Revisiting Pairing-Friendly Curves with Embedding Degrees 10 and 14

Yu Dai[1], Debiao He[2], Cong Peng[2]✉, Zhijian Yang[1], and Chang-an Zhao[3]✉

[1] School of Mathematics and statistics, Wuhan University, Wuhan, China.
eccdaiy39@gmail.com, zjyang.math@whu.edu.cn
[2] School of Cyber Science and Engineering, Wuhan University, Wuhan, China.
hedebiao@whu.edu.cn, cpeng@whu.edu.cn
[3] School of Mathematics, Sun Yat-sen University, Guangzhou, China.
zhaochan3@mail.sysu.edu.cn

Abstract. Since 2015, there has been a significant decrease in the asymptotic complexity of computing discrete logarithms in finite fields. As a result, the key sizes of many mainstream pairing-friendly curves have to be updated to maintain the desired security level. In PKC'20, Guillevic conducted a comprehensive assessment of the security of a series of pairing-friendly curves with embedding degrees ranging from 9 to 17. In this paper, we focus on five pairing-friendly curves with embedding degrees 10 and 14 at the 128-bit security level, with BW14-351 emerging as the most competitive candidate. First, we extend the optimized formula for the optimal pairing on BW13-310, a 128-bit secure curve with a prime p in 310 bits and embedding degree 13, to our target curves. This generalization allows us to compute the optimal pairing in approximately $\log r/(2\varphi(k))$ Miller iterations, where r and k are the order of pairing groups and the embedding degree respectively. Second, we develop optimized algorithms for cofactor multiplication for $\mathbb{G}_1$ and $\mathbb{G}_2$, as well as subgroup membership testing for $\mathbb{G}_2$ on these curves. Finally, we provide detailed performance comparisons between BW14-351 and other popular curves on a 64-bit platform in terms of pairing computation, hashing to $\mathbb{G}_1$ and $\mathbb{G}_2$, group exponentiations, and subgroup membership testings. Our results demonstrate that BW14-351 is a strong candidate for building pairing-based cryptographic protocols.

Keywords: pairing-friendly curves · BW14-351 · the 128-bit security level

1 Introduction

The past two decades have witnessed the application of elliptic curve pairings in public-key cryptosystems, such as Direct Anonymous Attestation (DAA) [13, 51], Succinct Non-interactive Arguments of Knowledge (SNARKs) [3, 21, 22, 30], and Verifiable Delay Function(VDF) [20]. A cryptographic pairing is a non-degenerate bilinear map defined as $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where the three pairing groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ have the same large prime order r. Specifically, $\mathbb{G}_1$ and

$\mathbb{G}_2$ are two independent subgroups of an elliptic curve E over a finite field $\mathbb{F}_{p^k}$, while $\mathbb{G}_T$ is a subgroup of the multiplicative group $\mathbb{F}_{p^k}^*$. The value of k is the smallest positive integer such that $E[r] \subseteq E(\mathbb{F}_{p^k})$.

The security of pairing-based cryptographic protocols relies on the hardness of the discrete logarithm problem (DLP) in the three pairing groups. The best-known attack algorithm for solving the DLP on an elliptic curve (ECDLP) in the two input pairing groups $\mathbb{G}_1$ and $\mathbb{G}_2$ is the Pollard rho algorithm [43], which requires around $\sqrt{r}$ group operations. Thus, the size of the prime r is at least 256 bits for reaching the 128-bit security level. As for the DLP on a finite field (FFDLP) $\mathbb{F}_{p^k}$ in $\mathbb{G}_T$ whose characteristic p is not small, the best-known algorithm is the number field sieve (NFS) [44]. According to the standardization reported by ENISA [1] in 2013, a 3072-bit finite field is 128-bit secure. Since then, a series of variants of NFS have been proposed [9, 36, 38], resulting in a drastic decrease for the security level of mainstream pairing-friendly curves. In particular, Kim and Barbulescu [38] proposed the special extended tower number field sieve (SexTNFS), which is applied to a composite extension field whose characteristic p can be parameterized by a tiny-coefficients polynomial of moderate degree. This variant is almost tailored to mainstream pairing-friendly curves, such as the Barreto-Naehrig(BN) [11] and Barreto-Lynn-Scott(BLS) [11] families. For example, the estimates in [8, 32] suggest that the updated security level of the previous 128-bit secure BN curve has dropped down to $100 \sim 103$ bits.

In PKC'20, Guillevic [31] analyzed the consequence of the improvement of NFS in detail and recommended a list of pairing-friendly curves with embedding degrees from 10 to 16. In particular, Guillevic pointed out that the size of the prime p on both BN and BLS12 curves has to be increased to 446 bits to match the updated 128-bit security level, and the BLS12-446 curve is the most efficient choice for pairing computation at this security level across different pairing-friendly curves. However, owing to the large size of the characteristic p, both BLS12-446 and BN446 suffer a performance penalty concerning operations associated with $\mathbb{G}_1$. Therefore, two new curves derived from [24, Construction 6.6] have emerged for fast group exponentiation in $\mathbb{G}_1$: BW13-310 and BW19-286 [15]. Recently, Dai, Zhang and Zhao [18] proposed a new formula for computing pairing on BW13-310. More specifically, the number of iterations in Miller's algorithm on the curve is only around $\log r/(2\varphi(k))$. However, due to the lack of twists, the trick of denominator elimination is no longer applicable. In other words, even though the length of the Miller loop on BW13-310 is extremely short, the computational cost for each Miller doubling/addition step is expensive. In addition, since the group $\mathbb{G}_2$ on BW13-310 is defined over the full extension field $\mathbb{F}_{p^{13}}$, the operations associated with $\mathbb{G}_2$ are costly, such as hashing to $\mathbb{G}_2$ and group exponentiation in $\mathbb{G}_2$. It motivates us to search for new pairing-friendly curves such that the Miller loop can be performed in $\log r/(2\varphi(k))$ iterations, and the trick of denominator elimination applies as well.

1.1 Our Contributions

In this work, we revisit the cyclotomic pairing-friendly curves presented in [24] with embedding degrees 10 and 14. A comprehensive research is presented that aims to facilitate the implementation of pairing-based cryptographic protocols using these curves. Our contributions are summarized as follows.

- We generalize the optimized formula for the optimal pairing on BW13-310 to our target curves. Specifically, the automorphism action can be extracted from the Miller function evaluation, reducing the number of Miller iterations to approximately $\log r/(2\varphi(k))$. In addition, we refine the best-known algorithm for the final exponentiation to save several field multiplications.
- We develop new algorithms for key building blocks involved in implementing pairing-based protocols on our target curves, including cofactor multiplication for $\mathbb{G}_1$ and $\mathbb{G}_2$, and subgroup membership testing for $\mathbb{G}_2$.
- Utilizing the RELIC toolkit [2], we present high-speed software implementations of pairing computation, hashing to $\mathbb{G}_1$ and $\mathbb{G}_2$, group exponentiations, and subgroup membership testings over two target curves named BW10-511 and BW14-351 on a 64-bit platform. Our results show that compared to popular curves at the updated 128-bit security level, including BLS12-446, BN446, and BW13-310, BW14-351 is competitive for building pairing-based cryptographic protocols. In more detail,
  - the performance of pairing computation on BW14-351 is even slightly faster than that on BN446 and BW13-310, while about 16.2% slower than that on BLS12-446;
  - in terms of group exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_T$, BW14-351 is about 49.4% and 20.4% faster than BLS12-446, 118.5% and 100% faster than BN446, while 35.1% and 3.4% slower than BW13-310;
  - compared to BW13-310, BW14-351 incurs a lighter penalty for hashing to $\mathbb{G}_2$ and group exponentiation in $\mathbb{G}_2$, while is still slower than BN446 and BLS12-446.

Code: Our code is available at https://github.com/eccdaiy39/BW10-14.

2 Preliminaries

In this section, we recall some basic properties of ordinary elliptic curves, pairings and endomorphisms.

2.1 Ordinary elliptic curves over finite fields

Let $\mathbb{F}_p$ be a prime field with characteristic $p > 3$. Let E be an elliptic curve over $\mathbb{F}_p$ of the form $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_p$ such that $4a^3 + 27b^2 \neq 0$. The j-invariant of E is defined as $j(E) = 1728\frac{4a^3}{4a^3+27b^2}$. We denote by $E(\mathbb{F}_p)$ the group of $\mathbb{F}_p$-rational points of E, together with the identity element $\mathcal{O}_E$. Then the order of $E(\mathbb{F}_p)$ is given by $\#E(\mathbb{F}_p) = p + 1 - t$, where t is the trace of the

Frobenius endomorphism $\pi : (x, y) \to (x^p, y^p)$. If $t \neq 0$, then the curve E is said to be *ordinary*, and *supersingular* otherwise. Let r be a large prime divisor of $\#E(\mathbb{F}_p)$, and let $E[r]$ denote the r-torsion subgroup of E. The embedding degree k of E with respect to r and p is the smallest positive integer such that $E[r] \subseteq E(\mathbb{F}_{p^k})$. If $k > 1$, then k is the smallest integer such that $r \mid p^k - 1$.

An endomorphism α of E over $\bar{\mathbb{F}}_p$ is a non-constant rational map from E to itself over $\bar{\mathbb{F}}_p$, where $\bar{\mathbb{F}}_p$ is the algebraic closure of $\mathbb{F}_p$. The set of all endomorphisms of E over $\bar{\mathbb{F}}_p$, together with the zero map defined by $0(P) = \mathcal{O}_E$, forms a ring denoted as $\mathrm{End}(E)$. We denote by K the imaginary quadratic field $K = \mathbb{Q}(\sqrt{-D})$, where D is the square-free part of $4p - t^2$. Let O_K be the largest subring of K. Since E is ordinary, $\mathrm{End}(E)$ is an order in O_K, i.e., $\mathbb{Z}[\pi] \subseteq \mathrm{End}(E) \subseteq O_K$. For any $\alpha \in \mathrm{End}(E)$, the characteristic equation of α can be represented as $x^2 + mx + n = 0$ for two integers m and n, where n is called the norm of α, i.e., $\mathrm{Nrd}(\alpha) = n$. In particular, the characteristic equation of π is given as $\pi^2 - t\pi + p = 0$. For each endomorphism α, there is a unique endomorphism $\hat{\alpha}$ such that $\alpha \circ \hat{\alpha} = \mathrm{Nrd}(\alpha)$, which is called the dual of α.

Let $\mathrm{Aut}(E)$ be the automorphism group of E, and let $d = \gcd(k, \#\mathrm{Aut}(E))$. If $d > 1$, then there exists a unique degree-d twist E' such that $r \mid \#E'(\mathbb{F}_{p^{k/d}})$ with an untwisting isomorphism ϕ: $E' \to E$. In elliptic curve cryptography, ordinary elliptic curves with j-invariant 0 or 1728 are particularly interesting as they are equipped with an efficiently computable endomorphism. More precisely,

- if $j(E) = 0$, then we have $a = 0$ and $p \equiv 1 \bmod 3$ [50, Proposition 4.33]. There exists an endomorphism $E \to E$ given as $\tau : (x, y) \to (\omega \cdot x, y)$, where ω is a primitive cube root of unity in $\mathbb{F}_p^*$. The characteristic equation of τ is $\tau^2 + \tau + 1 = 0$ and the dual of τ is $\hat{\tau} : (x, y) \to (\omega^2 \cdot x, y)$;
- if $j(E) = 1728$, then we have $b = 0$ and $p \equiv 1 \bmod 4$ [50, Theorem 4.23]. There exists an endomorphism $E \to E$ given as $\tau : (x, y) \to (-x, i \cdot y)$, where i is a primitive fourth root of unity in $\mathbb{F}_p^*$. The characteristic equation of τ is $\tau^2 + 1 = 0$ and the dual of τ is $\hat{\tau} : (x, y) \to (-x, -i \cdot y)$.

For the two types of curves, there exist the following two endomorphisms on E':

$$\eta = \phi^{-1} \circ \tau \circ \phi, \psi = \phi^{-1} \circ \pi \circ \phi,$$

where η and ψ have the same characteristic equations as τ and π, respectively.

2.2 Optimal pairing

Given a random point $Q \in E(\mathbb{F}_{p^k})$ and an integer m, a Miller function $f_{m,Q}$ is a normalized rational function in $\mathbb{F}_{p^k}(E)$ with divisor

$$div(f_{m,Q}) = m(Q) - ([m]Q) - (m - 1)(\mathcal{O}_E). \tag{1}$$

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be respectively 1- and p-eigenspaces of π acting on $E[r]$, i.e., $\mathbb{G}_1 = E(\mathbb{F}_p)[r]$ and $\mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi - [p])$. Let $\mathbb{G}_T$ be the group of r-th roots of unity in $\mathbb{F}_{p^k}^*$. Let $\lambda = \sum_{i=0}^{L} c_i p^i$ be a multiple of the prime r with $c_i \in \mathbb{Z}$ for

each i. Then, the general expression of the optimal pairing [49, Theorem 7] on E is given as:

$$\begin{aligned} e : & \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{G}_T, \\ & (Q,P) \to \left(\prod_{i=0}^{L} f_{c_i,Q}^{p^i}(P) \cdot \prod_{i=0}^{L-1} \frac{\ell_{[s_{i+1}]Q,[c_i p^i]Q}(P)}{\nu_{[s_i]Q}(P)} \right)^{\frac{(p^k-1)}{r}}, \end{aligned} \tag{2}$$

where $s_i = \sum_{j=i}^{L} c_j p^j$, $\ell_{[i]R,[j]R}$ is the straight line passing through $[i]R$ and $[j]R$, and $\nu_{[i+j]R}$ is the vertical line passing through $[i+j]R$. The Miller function $f_{c_i,Q}$ evaluated at the point P for each i can be obtained by executing Miller's algorithm [42], which is described in **Alg**. 1. Vercauteren [49, Theorem 7] proved that there exists a short vector $(c_0, c_1, \cdots, c_L)$ satisfying that $\max |c_i| \approx r^{1/\varphi(k)}$. Thus, the optimal pairing can be computed in approximately $\log r/\varphi(k)$ Miller iterations. Moreover, if the embedding degree k is even, the vertical line evaluations can be ignored because these values lie in the subfield $\mathbb{F}_{p^{k/2}}$ and can be "killed" by the exponentiation by $(p^k - 1)/r$.

Algorithm 1: Miller's Algorithm

Input: $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, $m = \sum_{i=0}^{L} m_i 2^i$ with $m_i \in \{-1, 0, 1\}$
Output: $f_{m,Q}(P)$
1: $T \leftarrow Q$, $f \leftarrow 1$
2: **for** $i = L - 1$ **down to** 0 **do**
3: $\quad f \leftarrow f^2 \cdot \frac{\ell_{T,T}(P)}{\nu_{[2]T}(P)}$, $T \longleftarrow [2]T$
4: $\quad$ **if** $m_i = 1$ **then**
5: $\quad\quad f \leftarrow f \cdot \frac{\ell_{T,Q}(P)}{\nu_{T+Q}(P)}$, $T \leftarrow T + Q$
6: $\quad$ **elif** $m_i = -1$ **then**
7: $\quad\quad f \leftarrow f \cdot \frac{\ell_{T,-Q}(P)}{\nu_{T-Q}(P)}$, $T \leftarrow T - Q$
8: $\quad$ **end if**
9: **end for**
10: **return** f

3 Elliptic Curves with Embedding Degrees 10 and 14

The construction of pairing-friendly curves necessitates special methods to ensure a small embedding degree k, which is crucial for efficient pairing computation. In addition, we also expect pairing-friendly curves have j-invariant 0 or 1728 such that they are equipped with efficiently computable endomorphisms and efficient formulas for point operation. Using the Brezing-Weng method [12], Freeman, Scott and Teske [24, **Sect**. 6] constructed a list of such curves with embedding degrees 10 and 14, which are summarized in **Tab**. 1. The formulas

for optimal pairing on these curves are given in **Tab**. 2. It is straightforward to see that the number of iterations in Miller's algorithm on these curves is approximately $\log r/\varphi(k)$.

Remark 1. Using Eq. (2), the formula for the optimal pairing for the Cyclo(6.5)-10 family is expressed as $\left(f^p_{z^2,Q}(P)\right)^{(p^{10}-1)/r}$. Since a non-degenerate power of a pairing remains a pairing, we can replace $f^p_{z^2,Q}(P)$ by $f_{z^2,Q}(P)$ to save one Frobenius map.

Table 1. Important parameters for pairing-friendly curves with embedding degrees 10 and 14 from [24, **Sect**. 6].

family	k	$j(E)$	p	r	t
Cyclo(6.3)	10	1728	$\frac{1}{4}(z^{14}-2z^{12}+z^{10}+z^4+2z^2+1)$	$\Phi_{20}(z)$	z^2+1
Cyclo(6.5)	10	1728	$\frac{1}{4}(z^{12}-z^{10}+z^8-5z^6+5z^4-4z^2+4)$	$\Phi_{20}(z)$	$-z^6+z^4-z^2+2$
Cyclo(6.6)	10	0	$\frac{1}{3}(z^3-1)^2(z^{10}-z^5+1)+z^3$	$\Phi_{30}(z)$	z^3+1
Cyclo(6.3)	14	1728	$\frac{1}{4}(z^{18}-2z^{16}+z^{14}+z^4+2z^2+1)$	$\Phi_{28}(z)$	z^2+1
Cyclo(6.6)	14	0	$\frac{1}{3}(z-1)^2(z^{14}-z^7+1)+z^{15}$	$\Phi_{42}(z)$	z^8-z+1

3.1 New formulas for optimal pairings on target curves

Recently, Dai, Zhang and Zhao [18] proposed a faster formula for pairing computation on the BW13-310 curve such that the length of the Miller loop can be reduced to around $\log r/(2\varphi(k))$. In this subsection, we show how to generalize this technique to our target curves. On this basis, we further propose an improved algorithm to reduce the performance penalty introduced by this new technique.

By the fact that the endomorphism ring of ordinary elliptic curves is commutative, we find that $\tau(Q) \in \mathbb{G}_2$ for any $Q \in \mathbb{G}_2$ as

$$\pi \circ \tau(Q) = \tau \circ \pi(Q) = \tau([p]Q) = [p]\tau(Q) \text{ and } [r]\tau(Q) = \tau([r]Q) = \mathcal{O}_E.$$

Furthermore, since the group order of $\mathbb{G}_2$ is prime, the endomorphism τ acts on $\mathbb{G}_2$ as a scalar, which is denoted as λ_2. In detail, we can fix the parameter of τ such that

$$\lambda_2 = \begin{cases} -z^{k/2}, & \text{in the Cyclo(6.3)-10, 14 and Cyclo(6.5)-10 families;} \\ z^k, & \text{in the Cyclo(6.6)-10 family;} \\ -z^k-1, & \text{in the Cyclo(6.6)-14 family.} \end{cases} \tag{3}$$

Table 2. Original formulas for optimal pairings on pairing-friendly curves with embedding degrees 10 and 14.

family-k	short vector	optimal pairing
Cyclo(6.3)-10	$[z^2, -1, 0, 0]$	$\left(f_{z^2,Q}(P)\right)^{(p^{10}-1)/r}$
Cyclo(6.5)-10	$[-1, z^2, 0, 0]$	$\left(f_{z^2,Q}(P)\right)^{(p^{10}-1)/r}$
Cyclo(6.6)-10	$[z, 0, -1, z^2]$	$\left(f_{z,Q}(P) \cdot f_{z^2,Q}^{p^3}(P) \cdot \ell_{\pi^7(Q),\pi^3([z^2]Q)}(P)\right)^{(p^{10}-1)/r}$
Cyclo(6.3)-14	$[z^2, -1, 0, 0, 0, 0]$	$\left(f_{z^2,Q}(P)\right)^{(p^{14}-1)/r}$
Cyclo(6.6)-14	$[z^2, z, 1, 0, 0, 0]$	$\left(f_{z^2,Q}(P) \cdot f_{z,Q}^{p}(P) \cdot \ell_{\pi^2(Q),\pi([z]Q)}(P)\right)^{(p^{14}-1)/r}$

By combining the two endomorphisms π and τ, we fortunately find that $\pi^m \circ \tau(Q) = [z]Q$ for any $Q \in \mathbb{G}_2$, where

$$m = \begin{cases} (k+2)/4, & \text{in the Cyclo(6.3)-10 and Cyclo(6.3)-14 families;} \\ 7, & \text{in the Cyclo(6.5)-10 and Cyclo(6.6)-10 families;} \\ 1, & \text{in the Cyclo(6.6)-14 family.} \end{cases}$$

This observation enables us to rewrite the formulas for optimal pairings on our target curves such that the number of Miller iterations can be reduced to around $\log r/(2\varphi(k))$, which is summarized in Lemma 1 below.

Lemma 1. *Let notation be as above. Then* $f_{z^2,Q} = f_{z,Q}^{z} \cdot f_{z,Q}^{p^m} \circ \hat{\tau}$, *where* $\hat{\tau}$ *is the dual of* τ.

Proof. It can obtained from [35, Lemma 3.5] that

$$f_{z^2,Q} = f_{z,Q}^{z} \cdot f_{z,[z]Q}. \tag{4}$$

Since $\pi^m \circ \tau(Q) = [z]Q$, it follows from [53, Theorem 1] and [17, Theorem 1] that

$$f_{z,[z]Q} = f_{z,\pi^m\circ\tau(Q)} = f_{z,\tau(Q)}^{p^m} = f_{z,Q}^{p^m} \circ \hat{\tau}^{p^m} = f_{z,Q}^{p^m} \circ \hat{\tau}. \tag{5}$$

Inserting Eq. (5) into Eq. (4), we have

$$f_{z^2,Q} = f_{z,Q}^{z} \cdot f_{z,Q}^{p^m} \circ \hat{\tau},$$

which completes the proof. □

Based on Lemma 1, we can derive new formulas for optimal pairings on our target curves by executing the following two steps:

-Step 1. We first replace $f_{z^2,Q}(P)$ by $f_{z,Q}^{z}(P) \cdot f_{z,Q}^{p^m}(\hat{\tau}(P))$ in the original formulas for optimal pairings. In particular, we can also replace the point $[z]Q$ by $\pi^m \circ \tau(Q)$ at the final line in the Cyclo(6.6)-10 and Cyclo(6.6)-14 families.

Table 3. Optimized formulas for optimal pairings on pairing-friendly curves with embedding degrees 10 and 14.

family-k	optimal pairing
Cyclo(6.3)-10	$\left(f_{z,Q}^{z\cdot p^7}(P)\cdot f_{z,Q}(\hat{\tau}(P))\right)^{(p^{10}-1)/r}$
Cyclo(6.5)-10	$\left(f_{z,Q}^{z\cdot p^3}(P)\cdot f_{z,Q}(\hat{\tau}(P))\right)^{(p^{10}-1)/r}$
Cyclo(6.6)-10	$\left(f_{z,Q}^{1+z\cdot p^3}(P)\cdot f_{z,Q}(\hat{\tau}(P))\cdot (y_P-y_Q)^{p^7}\right)^{(p^{10}-1)/r}$
Cyclo(6.3)-14	$\left(f_{z,Q}^{z\cdot p^{10}}(P)\cdot f_{z,Q}(\hat{\tau}(P))\right)^{(p^{14}-1)/r}$
Cyclo(6.6)-14	$\left(f_{z,Q}^{1+z\cdot p^{13}}(P)\cdot f_{z,Q}(\hat{\tau}(P))\cdot (y_P-y_Q)^{p}\right)^{(p^{14}-1)/r}$

Table 4. Parameters of the pairing-friendly curves with embedding degrees 10 and 14 at the updated 128-bit security level.

curve	family-k	seed z	r bits	p bits	p^k bits	DL cost in $\mathbb{F}_{p^k}$
BW10-480	Cyclo(6.5)-10	$2^5+2^{14}+2^{15}+2^{18}+2^{36}+2^{40}$	321	480	4791	128
BW10-511	Cyclo(6.6)-10	$2^7+2^{13}+2^{26}-2^{32}$	256	511	5101	150
BW10-512	Cyclo(6.3)-10	$1+2^3+2^{17}+2^{32}+2^{35}+2^{36}$	294	512	5111	129
BW14-351	Cyclo(6.6)-14	$2^6-2^{12}-2^{14}-2^{22}$	265	351	4908	149
BW14-382	Cyclo(6.3)-14	$1+2^{10}+2^{13}-2^{16}+2^{19}+2^{21}$	256	382	5338	129

-Step 2. Utilizing the property that a non-degenerate power of a pairing remains a pairing, we then can raise the output of the Miller loop to a p^{k-m}-power such that the exponent of the second Miller function is equal to 1. The new formulas for the optimal pairing for our selected curves are summarized in **Tab.** 3. Clearly, the most costly part of the Miller loop is to compute $f_{z,Q}^{z\cdot p^{k-m}}(P)\cdot f_{z,Q}(\hat{\tau}(P))$, enabling the execution of Miller's algorithm in around $\log|z|$ iterations within the same loop, albeit with a slightly higher computational cost per iteration. However, compared to the original formulas, the new ones require an additional exponentiation by z. Fortunately, the exponentiation can be integrated with the computation of $f_{z,Q}(\hat{\tau}(P))$ to share several squarings. Specifically, we first calculate $f_{z,Q}(P)$ and store all line function evaluations required for computing $f_{z,Q}(\hat{\tau}(P))$ at the first loop. Subsequently, given the initial value $f_{z,Q}^{p^{k-m}}(P)$, we then compute $f_{z,Q}^{z\cdot p^{k-m}}(P)\cdot f_{z,Q}(\hat{\tau}(P))$ at the second loop. The optimized procedure for computing this value is presented in **Alg**. 2. Thanks to the final exponentiation, the value of g^{-1} can be replaced by $\bar{g}$ in Line 10 of **Alg**. 2, where $\bar{g}$ represents the conjugate of g.

3.2 Choice of parameters at the 128-bit security level

The choice of parameters of pairing-friendly curves should be careful for achieving high performance implementation at the desired security level. In this paper,

Algorithm 2: Computing $f_{z,Q}^{z\cdot p^{k-m}}(P) \cdot f_{z,Q}(\hat{\tau}(P))$

Input: $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, $z = \sum_{i=0}^{L} z_i \cdot 2^i$ with $z_i \in \{-1, 0, 1\}$
Output: $f_{z,Q}^{z\cdot p^{k-m}}(P) \cdot f_{z,Q}(\hat{\tau}(P))$
1: $T \leftarrow Q$, $f \leftarrow 1$, tab$\leftarrow [\,]$, $j \leftarrow 0$
2: **for** $i = L-1$ **down to** 0 **do**
3: $\quad f \leftarrow f^2 \cdot \ell_{T,T}(P)$, tab$[j] \leftarrow \ell_{T,T}(\hat{\tau}(P))$, $T \longleftarrow 2T$, $j \leftarrow j+1$ `// SDBL`
4: $\quad$ **if** $z_i = 1$ **then**
5: $\quad\quad f \leftarrow f \cdot \ell_{T,Q}(P)$,tab$[j] \leftarrow \ell_{T,Q}(\hat{\tau}(P))$, $T \leftarrow T+Q$, $j \leftarrow j+1$ `// SADD`
6: $\quad$ **elif** $z_i = -1$ **then**
7: $\quad\quad f \leftarrow f \cdot \ell_{T,-Q}(P)$, tab$[j] \leftarrow \ell_{T,-Q}(\hat{\tau}(P))$, $T \leftarrow T-Q$, $j \leftarrow j+1$ `// SADD`
8: $\quad$ **end if**
9: **end for**
10: $g \leftarrow f^{p^{k-m}}$, $h \leftarrow g$, $j \leftarrow 0$
11: **for** $i = L-1$ **down to** 0 **do**
12: $\quad h \leftarrow h^2 \cdot$tab$[j]$, $j \leftarrow j+1$
13: $\quad$ **if** $z_i = 1$ **then**
14: $\quad\quad h \leftarrow h \cdot g \cdot$tab$[j]$, $j \leftarrow j+1$
15: $\quad$ **elif** $z_i = -1$ **then**
16: $\quad\quad h \leftarrow h \cdot \bar{g} \cdot$tab$[j]$, $j \leftarrow j+1$
17: $\quad$ **end if**
18: **end for**
19: **return** h

we focus on the performance of pairing computation at the 128-bit security level. To this end, the size of full extension field $\mathbb{F}_{p^k}$ should be large enough to resist attacks from the variants of NFS. The concrete security can be estimated using the source code provided by Guillevic and Singh [33], which is available at
https://gitlab.inria.fr/tnfs-alpha/alpha/tree/master/sage.
On this basis, to maximize the efficiency of pairing computation, we also expect

(a) the selected prime p satisfies that $p \equiv 1 \bmod k$;
(b) the sum of bit length and Hamming weight (in non-adjacent form) of the selected seed z is as small as possible.

In more detail, the condition (a) ensures that the full extension field $\mathbb{F}_{p^k}$ can be represented as $\mathbb{F}_p[v]/(v^k - \alpha)$ for some $\alpha \in \mathbb{F}_p^*$ [40, Theorem 3.75], which actually can be constructed as a tower of quadratic and $k/2$-th extensions:

$$\mathbb{F}_p \xrightarrow{\xi^{k/2}-\alpha} \mathbb{F}_{p^{k/2}} \xrightarrow{v^2-\xi} \mathbb{F}_{p^k}.$$

This construction induces fast multiplication and squaring arithmetic operations in $\mathbb{F}_{p^k}$; the condition (b) aims to minimize the number of Miller iterations in **Alg**. 2. In fact, the computation of the final exponentiation also benefits from condition (b) since this step consists of a large number of exponentiations by z (see **Sect**. 4.3). **Tab**. 4 summarizes our chosen seeds z under the above conditions, together with the corresponding sizes of the curve parameters. Notably,

while Guillevic [31, **Tab**. 6] selected a seed for the Cyclo(6.6)-14 family, this seed fails to meet the condition (a).

Curve name: For convenience, all the candidate curves listed in **Tab**. 4 are collectively called BW curves since they are essentially generated using the Brezing-Weng method. Moreover, we distinguish each curve by its embedding degree and the size of the characteristic p. For instance, the BW14-351 curve is referred to as the curve constructed from the Cyclo(6.6)-14 family defined over a 351-bit prime field.

4 Pairing Computation

In this section, we first describe explicit formulas for Miller doubling and addition steps. In particular, the technique of lazy reduction [6,45] has been fully exploited to reduce the number of modular reductions required in Miller's algorithm. Then, we show how to perform the final exponentiation efficiently. Finally, we present detailed operation counts for pairing computation on different curves.

Notations. The cyclotomic group $\mathbb{G}_{\Phi_k(p)}$ is defined by $\mathbb{G}_{\Phi_k(p)} = \{a \in \mathbb{F}_{p^k} \mid a^{\Phi_k(p)} = 1\}$. The notation $\times$ is used to denote field multiplication without reduction. We use the following notation to denote the cost of operations:(i) $\mathbf{a}$, $\mathbf{m}$, $\mathbf{m}_u$, $\mathbf{s}$, $\mathbf{s}_u$, $\mathbf{i}$ and $\mathbf{r}$ denote the cost of addition, multiplication, multiplication without reduction, squaring, squaring without reduction, inversion and modular reduction in $\mathbb{F}_p$, respectively; (ii) $\tilde{\mathbf{a}}$, $\tilde{\mathbf{m}}$, $\tilde{\mathbf{m}}_u$, $\tilde{\mathbf{m}}_\xi$, $\tilde{\mathbf{s}}$, $\tilde{\mathbf{s}}_u$, $\tilde{\mathbf{i}}$ and $\tilde{\mathbf{r}}$ represent the cost of addition, multiplication, multiplication without reduction, multiplication by ξ, squaring, squaring without reduction, inversion and modular reduction in $\mathbb{F}_{p^{k/2}}$, respectively; (iii) $\mathbf{M}$, $\mathbf{S}$, $\mathbf{f}$ and $\mathbf{I}$ represent the cost of multiplication, squaring, Frobenius map and inversion in $\mathbb{F}_{p^k}$, respectively; (iv) $\mathbf{S}_c$ and $\mathbf{e}$ represent the cost of squaring and exponentiation by z in the cyclotomic group $\mathbb{G}_{\Phi_k(p)}$, respectively.

4.1 Miller loop on curves of form $y^2 = x^3 + b$

Let $E : y^2 = x^3 + b$ be a curve over $\mathbb{F}_p$ with embedding degree 10 or 14. Then E admits a quadratic twist $E' : y^2 = x^3 + b/\xi^3$ over $\mathbb{F}_{p^{k/2}}$. The associated untwisting isomorphism from E' to E is given by

$$\phi : (x, y) \to (x\xi, y\xi v).$$

To avoid field inversions when performing point operations, points can be represented in projective coordinates. For this curve shape, it is convenient to use Jacobian coordinates, that is, an affine point (x, y) corresponds to a triplet (X, Y, Z) with $x = X/Z^2$ and $y = Y/Z^3$.

Shared doubling step (SDBL) Let $T = (X, Y, Z) \in E'(\mathbb{F}_{p^{k/2}})[r]$ be in Jacobian coordinates. The formula for computing the doubling point $[2]T =$

(X_3, Y_3, Z_3) is derived from [7, **Sect**. 4.3], where

$$X_3 = X(\frac{9}{4}X^3 - 2Y^2), Y_3 = \frac{9}{4}X^3(2Y^2 - \frac{3}{2}X^3) - Y^4, Z_3 = YZ.$$

By the form of the untwisting map ϕ, the image point $\phi(T) \in \mathbb{G}_2$ can be represented as $(X\xi, Y\xi v, Z)$. Thanks to the technique of denominator elimination, the line function $l_{\phi(T),\phi(T)}$ evaluated at $P = (x_P, y_P)$ and $\hat{\tau}(P) = (\tilde{x}_P, y_P)$ can be simplified as

$$\begin{aligned} l_{\phi(T),\phi(T)}(P) &= 2Z_3Z^2y_P + \big((3X^3 - 2Y^2) \cdot \xi - 3X^2Z^2x_P\big)v, \\ l_{\phi(T),\phi(T)}(\hat{\tau}(P)) &= 2Z_3Z^2y_P + \big((3X^3 - 2Y^2) \cdot \xi - 3X^2Z^2\tilde{x}_P\big)v. \end{aligned}$$

It is evident that the two line evaluations share a large amount of variables. In addition, the technique of lazy reduction can be employed when computing Y_3. Thus, we can obtain the above two line evaluations using the following sequence of operations:

$$\begin{aligned} &A = 3X^2, B = A \cdot X, C = \frac{B}{2}, D = C + \frac{C}{2}, E = Y^2, F = 2E, U_0 = D \times (F - C), \\ &U_1 = E \times E, \quad Y_3 = (U_0 - U_1) \bmod p, X_3 = X \cdot (D - F), Z_3 = Y \cdot Z, G = Z^2, \\ &I = G \cdot Z_3 \cdot (2y_P), J = A \cdot G, K = (B - F) \cdot \xi, L = J \cdot x_P, M = J \cdot \tilde{x}_P, \\ &l_{\phi(T),\phi(T)}(P) = I + (K - L)v, l_{\phi(T),\phi(T)}(\hat{\tau}(P)) = I + (K - M)v. \end{aligned}$$

The total operation count for point doubling and two line evaluations is $5\tilde{\mathbf{m}} + \tilde{\mathbf{m}}_u + \tilde{\mathbf{s}}_u + \tilde{\mathbf{m}}_\xi + 3\tilde{\mathbf{s}} + \frac{3k}{2}\mathbf{m} + \tilde{\mathbf{r}} + 13\tilde{\mathbf{a}} + \mathbf{a}$, assuming that the computation of division by 2 and $U_0 - U_1$ requires one and two additions, respectively.

Shared addition step (SADD) Let $T = (X, Y, Z), Q = (X_2, Y_2, Z_2) \in E'(\mathbb{F}_{p^{k/2}})[r]$ be in Jacobian coordinates with $Z \neq 0$, $Z_2 = 1$ and $T \neq Q$. Then one can compute the point $T + Q = (X_3, Y_3, Z_3)$ using the mixed addition formula [7, **Sect**. 4.3], which is given as

$$\theta = Y_2Z^3 - Y, \beta = X_2Z^2 - X, X_3 = \theta^2 - 2X\beta^2 - \beta^3, Y_3 = \theta(X\beta^2 - X_3) - Y\beta^3, Z_3 = Z\beta.$$

Subsequently, the line function $l_{\phi(T),\phi(Q)}$ evaluated at P and $\hat{\tau}(P)$ can be expressed as

$$\begin{aligned} l_{\phi(T),\phi(Q)}(P) &= Z_3y_p + \big((\theta X_2 - Y_2Z_3) \cdot \xi - \theta x_P\big)v, \\ l_{\phi(T),\phi(Q)}(\hat{\tau}(P)) &= Z_3y_p + \big((\theta X_2 - Y_2Z_3) \cdot \xi - \theta \tilde{x}_P\big)v. \end{aligned}$$

Again, by taking advantage of the technique of lazy reduction, we perform the following sequence of operations to compute the above point addition and two

line evaluations, which costs $6\tilde{\mathbf{m}} + 4\tilde{\mathbf{m}}_u + \tilde{\mathbf{m}}_\xi + 3\tilde{\mathbf{s}} + \frac{3k}{2}\mathbf{m} + 2\tilde{\mathbf{r}} + 12\tilde{\mathbf{a}}$:

$$
\begin{aligned}
&A = Z^2, \theta = Y_2 \cdot A \cdot Z - Y, \beta = X_2 \cdot A - X, B = \beta^2, C = \beta \cdot B, D = X \cdot B,\\
&X_3 = \theta^2 - 2D - C, U_0 = \theta \times (D - X_3), U_1 = Y \times C, Y_3 = (U_0 - U_1) \bmod p,\\
&Z_3 = Z \cdot \beta, E = Z_3 \cdot y_P, F = \theta \cdot x_P, G = \theta \cdot \tilde{x}_P, U_2 = \theta \times X_2, U_3 = Y_2 \times Z_3,\\
&H = (U_2 - U_3) \bmod p, I = H \cdot \xi, l_{\phi(T),\phi(Q)}(P) = E + (I - F)v,\\
&l_{\phi(T),\phi(Q)}(\hat{\tau}(P)) = E + (I - G)v.
\end{aligned}
$$

4.2 Miller loop on curves of form $y^2 = x^3 + ax$

Let $E : y^2 = x^3 + ax$ be a curve over $\mathbb{F}_p$ with embedding degree 10 or 14. Then E admits a quadratic twist $E' : y^2 = x^3 + a'$ over $\mathbb{F}_{p^{k/2}}$, where $a' = a \cdot \xi^2$. As a consequence, the associated untwisting isomorphism from E' to E can be expressed as

$$\phi : (x, y) \to (x, y) \to (x/\xi, y/(\xi v)).$$

For this curve shape, we represent an affine point (x, y) in the weight-$(1, 2)$ coordinates (X, Y, Z) satisfying that $x = X/Z$ and $y = Y/Z^2$ [16, **Sect**. 4].

Shared doubling step (SDBL) Let $T = (X, Y, Z) \in E'(\mathbb{F}_{p^{k/2}})[r]$ be in weight-$(1, 2)$ coordinates. For this curve shape, the point doubling formula for computing $[2]T = (X_3, Y_3, Z_3)$ is derived from [16, **Sect**. 4], which is expressed as

$$X_3 = (X^2 - a'Z^2)^2, Y_3 = 2Y(X^2 - a'Z^2)\big(2(X^2 + a'Z^2)^2 - X_3\big), Z_3 = 4Y^2.$$

In this case, it is more convenient to perform line evaluations on the twisted curve. In other words, we compute the line function $l_{T,T}$ evaluated at $\phi^{-1}(P) = (x_P\xi, y_P\xi v)$ and $\phi^{-1} \circ \hat{\tau}(P) = (-x_P\xi, \tilde{y}_P\xi v)$. The explicit formulas are given by

$$
\begin{aligned}
l_{T,T}(\phi^{-1}(P)) &= (X^3 - a'XZ^2) - (3X^2Z + a'Z^3)x_P\xi + 2YZy_P\xi v,\\
l_{T,T}(\phi^{-1} \circ \hat{\tau}(P)) &= (X^3 - a'XZ^2) + (3X^2Z + a'Z^3)x_P\xi + 2YZ\tilde{y}_P\xi v.
\end{aligned}
$$

Accordingly, the point doubling and two line evaluations can be accomplished by performing the following sequences of operations at a cost of $5\tilde{\mathbf{m}} + 5\tilde{\mathbf{s}} + \frac{3k}{2}\mathbf{m} + 2\tilde{\mathbf{m}}_\xi + \tilde{\mathbf{m}}_{a'} + 9\tilde{\mathbf{a}}$ ($\tilde{\mathbf{m}}_{a'}$ denotes the cost of multiplication by a'):

$$
\begin{aligned}
&A = X^2, B = 2Y, C = a' \cdot Z^2, D = A - C, E = A + C, X_3 = D^2, Z_3 = B^2, F = B \cdot Z,\\
&Y_3 = B \cdot D \cdot (2E^2 - X_3), G = F \cdot \xi, I = X \cdot D, H = (2A + E) \cdot Z \cdot x_P, J = y_P \cdot G,\\
&\tilde{J} = \tilde{y}_P \cdot G, K = H \cdot \xi, l_{T,T}(\phi^{-1}(P)) = I - K + Jv, l_{T,T}(\phi^{-1} \circ \hat{\tau}(P)) = I + K + \tilde{J}v.
\end{aligned}
$$

Shared addition step (SADD) Let $T = (X, Y, Z), Q = (X_2, Y_2, Z_2) \in E'(\mathbb{F}_{p^{k/2}})[r]$ be in weight-(1, 2) coordinates with $Z \neq 0$, $Z_2 = 1$ and $T \neq Q$. We

adopt the mixed addition formula [16, **Sect**. 4] to compute the point $T + Q = (X_3, Y_3, Z_3)$, which is given by

$$U = X - X_2Z, S = U^2Z, X_3 = (Y - Y_2Z^2)^2 - (X + X_2Z)S,$$
$$Y_3 = \big((Y - Y_2Z^2)(XS - X_3) - YSU\big)UZ, Z_3 = (UZ)^2.$$

Subsequently, the line function $l_{T,Q}$ evaluated at $\phi^{-1}(P)$ and $\phi^{-1} \circ \hat{\tau}(P)$ are given by

$$l_{T,Q)}(\phi^{-1}(P)) = \big((Y - Y_2Z^2)X_2 - UZY_2\big) - (Y - Y_2Z^2)\xi x_P + y_P UZ\xi v,$$
$$l_{T,Q}(\phi^{-1} \circ \hat{\tau}(P)) = \big((Y - Y_2Z^2)X_2 - UZY_2\big) + (Y - Y_2Z^2)\xi x_P + \tilde{y}_P UZ\xi v.$$

The following sequence of operations can be used for computing the above mixed point addition and two line evaluations at a cost of $6\tilde{\mathbf{m}} + 6\tilde{\mathbf{m}}_u + 2\tilde{\mathbf{m}}_\xi + 3\tilde{\mathbf{s}} + \frac{3k}{2}\mathbf{m} + 3\tilde{\mathbf{r}} + 10\tilde{\mathbf{a}}$:

$A = Z^2, B = X_2 \cdot Z, C = Y_2 \cdot A, D = X - B, E = Y - C, F = Z \cdot D, G = F \cdot D,$
$X_3 = \big(E \times E - (X + B) \times G\big) \bmod p, H = X \cdot G - X_3, I = E \cdot F, J = G^2,$
$Y_3 = (I \times H - Y \times J) \bmod p, Z_3 = F^2, K = (E \times X_2 - F \times Y_2) \bmod p, L = E \cdot x_P \cdot \xi,$
$M = F \cdot \xi, N = M \cdot y_P, \tilde{N} = M \cdot \tilde{y}_P, l_{T,Q}(\phi^{-1}(P)) = (K - L) + Nv,$
$l_{T,Q}(\phi^{-1} \circ \hat{\tau}(P)) = (K + L) + \tilde{N}v.$

4.3 The final exponentiation

The final exponentiation is the other time-consuming stage of pairing computation. This step aims to raise the output of the Miller loop to the power of $(p^k - 1)/r$. For our target curves, the large exponent can be split into two parts:

$$(p^{k-1})/r = \underbrace{(p+1)(p^{k/2} - 1)}_{\text{easy part}} \cdot \underbrace{\Phi_k(p)/r}_{\text{hard part}} \cdot$$

The exponentiation to the power of the easy part yields an element $f \in \mathbb{G}_{\Phi_k(p)}$, which costs only $\mathbf{I} + 3\mathbf{M} + 2\mathbf{f}$. The major bottleneck of the final exponentiation arises from the exponentiation to the power of the hard part. Observing that a non-degenerate power of a pairing remains a pairing, Fuentes-Castañeda *et al*. [25] proved that it suffices to raise f to the power of a multiple h of $\Phi_k(p)/r$, where h can be written in the base p as

$$h = h_0 + h_1 \cdot p + \cdots + h_{\varphi(k)-1} \cdot p^{\varphi(k)-1}.$$

As a consequence, the LLL algorithm is applied to obtain small coefficients h_i. In essence, this method aims to minimize the number of iterations required for the final exponentiation. Nevertheless, it may still be challenging to devise an optimized routine of the $\varphi(k)$ small exponentiations f^{h_i}. For example, when

applying this method to the BW14-351 curve, the six coefficients h_i are given as follows:

$$\begin{aligned}
h_0 &= z^{13} + z^{12} + z^{11} - z^6 + 3z^5 + z^3,\\
h_1 &= -z^{13} - z^{12} - 2z^{11} - z^{10} - z^9 + z^6 - 2z^5 + z^4 - 3z^3,\\
h_2 &= (1+z^3)(z^{10} + z^9 + z^8) - z^6 + 2z^5 - z^4 - z^3 + 2z^2 - z,\\
h_3 &= -z^{13} - z^{12} - z^{11} + z^6 - 2z^5 + z^4 + z^2 + z + 1,\\
h_4 &= z^{13} + z^{12} + z^{11} - z^8 - z^7 - 2z^6 + 2z^5 - z^4 - 3,\\
h_5 &= z^{14} - z^{11} + 4z^6 - 2z^5 + z^4.
\end{aligned}$$

Thus, the cost of computing f^{h_i} consists of 14 exponentiations by z and a large amount of full extension field multiplications.

Based on the fact that $f^{\Phi_k(p)} = 1$, we can further substitute the exponent h with $\lambda = h + \delta\Phi_k(p)$ for some integer δ. In particular, since $\Phi_k(p) = \sum_{i=0}^{\varphi(k)} (-1)^i p^i$ in our case, the new exponent λ can be written in the base of p as

$$\lambda = \lambda_0 + \lambda_1 \cdot p + \cdots + \lambda_{\varphi(k)} \cdot p^{\varphi(k)},$$

where $\lambda_i = h_i + (-1)^i\delta$ for $i \in \{0, 1, \cdots, \varphi(k) - 1\}$ and $\lambda_{\varphi(k)} = \delta$. Therefore, the careful selection of the parameter δ may facilitate faster final exponentiation. We now revisit the final exponentiation on the BW14-351 curve. By setting $\lambda_6 = -(z^{13} + z^{12} + z^{11} + 3z^5) + (z^6 + z^5 + z^4)$, we have

$$\begin{aligned}
\lambda_0 &= h_0 + \lambda_6 = z^5 + z^4 + z^3, & \lambda_1 &= h_1 - \lambda_6 = -z^{11} - z^{10} - z^9 - 3z^3,\\
\lambda_2 &= h_2 + \lambda_6 = z^{10} + z^9 + z^8 - z^3 + 2z^2 - z, & \lambda_3 &= h_3 - \lambda_6 = z^2 + z + 1,\\
\lambda_4 &= h_4 + \lambda_6 = -z^8 - z^7 - z^6 - 3, & \lambda_5 &= h_5 - \lambda_6 = z^{14} + z^{13} + z^{12} + 3z^6.
\end{aligned}$$

It is straightforward to see that the six coefficients λ_i satisfy the following relations:

$$\begin{aligned}
&\lambda_3 = z^2 + z + 1, \quad \lambda_0 = z^3\lambda_3, && \lambda_4 = -(z^3\lambda_0 + 3), \lambda_2 = -(z^2\lambda_4 + z\lambda_3),\\
&\lambda_1 = z^3\lambda_4, && \lambda_6 = z^2\lambda_1 + z\lambda_0, \quad \lambda_5 = -z^3\lambda_1.
\end{aligned}$$

In conclusion, the hard part exponentiation on the BW14-351 curve benefits from the easy relation between λ_i. In **Tab**. 5, we list our selected coefficients $\lambda_0, \lambda_1, \cdots, \lambda_{\varphi(k)}$ and the corresponding sequence of operations on the five candidate curves.

4.4 Computational cost

The construction of tower fields and the curve equations for the five candidate pairing-friendly curves are presented in **Tab**. 6. We now discuss the operation counts of pairing computation on these curves. To this aim, we first count the number of finite field arithmetic operations. For the Frobenius map and the

Table 5. The exponentiation of the hard part on the five candidate pairing-friendly curves. The values of f^3 and f^4 can be obtained during the computation of f^z. Therefore, we assume that the computation of f^3 requires one multiplication, while the computation of f^4 is free. The notation $\bar{f}_i$ is denoted as the conjugate of f_i.

BW10-480	$\lambda_0 = z^8-4z^2, \lambda_1 = z^{10}-z^8-4z^4+4z^2, \lambda_2 = z^6-z^4-4, \lambda_3 = -z^6+4, \lambda_4 = 0$
	Input:$f \in \mathbb{G}_{\Phi_{14}(p)}$, **Output**:$h \in \mathbb{G}_T$, **Cost**:$10\mathbf{e}+6\mathbf{M}+3\mathbf{f}$
	$f_1 \leftarrow f^{z^4}, f_2 \leftarrow f_1^{z^2} \cdot \bar{f}^4, f_3 \leftarrow f_2^{z^2}, f_4 \leftarrow f_3^{z^2}, f_5 \leftarrow f_2 \cdot \bar{f}_1, f_6 \leftarrow f_4 \cdot \bar{f}_3,$ $h \leftarrow f_3 \cdot f_6^p \cdot {f_5^{p^2}} \cdot {\bar{f}_2}^p$
BW10-511	$\lambda_0 = -z^{13}+2z^{10}-z^7-3, \lambda_1 = -z^{10}+2z^7-z^4, \lambda_2 = -z^7+2z^4-z,$ $\lambda_3 = (z^{14}-2z^{11}+z^8+3z)-(z^9-2z^6+z^3), \lambda_4 = (z^{11}-2z^8+z^5)-(z^6-2z^3+1)$
	Input:$f \in \mathbb{G}_{\Phi_{10}(p)}$, **Output**:$h \in \mathbb{G}_T$, **Cost**:$14\mathbf{e}+9\mathbf{M}+\mathbf{S}_c+4\mathbf{f}$
	$f_1 \leftarrow f^{z^6-2z^3+1}, f_2 \leftarrow f_1^z, f_3 \leftarrow f_2^{z^2}, f_4 \leftarrow f_3^z, f_5 \leftarrow f_4^z, f_6 \leftarrow f_5^{z^2} \cdot f^3,$ $f_7 \leftarrow f_6^z \cdot \bar{f}_3, f_8 \leftarrow f_5 \cdot \bar{f}_1, h \leftarrow \bar{f}_6 \cdot {\bar{f}_4}^p \cdot {\bar{f}_2}^{p^2} \cdot {f_7}^{p^3} \cdot {f_8}^{p^4}$
BW10-512	$\lambda_0 = z^6-2z^4+z^2, \lambda_1 = z^4-2z^2+1, \lambda_2 = -z^{12}+2z^{10}-z^8-4,$ $\lambda_3 = -z^{10}+2z^8-z^6, \lambda_4 = -z^8+2z^6-z^4$
	Input:$f \in \mathbb{G}_{\Phi_{10}(p)}$, **Output**:$h \in \mathbb{G}_T$, **Cost**:$12\mathbf{e}+7\mathbf{M}+\mathbf{S}_c+4\mathbf{f}$
	$f_1 \leftarrow f^{z^4-2z^2+1}, f_2 \leftarrow f_1^{z^2}, f_3 \leftarrow f_2^{z^2}, f_4 \leftarrow f_3^{z^2}, f_5 \leftarrow f_4^{z^2} \cdot f^4,$ $h \leftarrow f_2 \cdot f_1^p \cdot {\bar{f}_5}^{p^2} \cdot {\bar{f}_4}^{p^3} \cdot {\bar{f}_3}^{p^4}$
BW14-351	**Input**:$f \in \mathbb{G}_{\Phi_{14}(p)}$, **Output**:$h \in \mathbb{G}_T$, **Cost**:$14\mathbf{e}+12\mathbf{M}+6\mathbf{f}$
	$f_1 \leftarrow f^{z^2+z+1}, f_2 \leftarrow f_1^z, f_3 \leftarrow f_2^{z^2}, f_4 \leftarrow f_3^z, f_5 \leftarrow f_4^{z^2} \cdot f^3 f_5 \leftarrow \bar{f}_5, f_6 \leftarrow f_5^{z^2},$ $f_7 \leftarrow f_2 \cdot f_6, f_8 \leftarrow f_6^z, f_9 \leftarrow f_8^{z^2}, f_{10} \leftarrow f_4 \cdot f_9, f_{11} \leftarrow {\bar{f}_9}^z,$ $h \leftarrow f_3 \cdot {\bar{f}_8}^p \cdot {\bar{f}_7}^{p^2} \cdot {f_1}^{p^3} \cdot {f_5}^{p^4} \cdot {f_{11}}^{p^5} \cdot {f_{10}}^{p^6}$
BW14-382	$\lambda_0 = z^{10}-2z^8+z^6, \lambda_1 = z^8-2z^6+z^4, \lambda_2 = z^6-2z^4+z^2, \lambda_3 = z^4-2z^2+1,$ $\lambda_4 = -z^{16}+2z^{14}-z^{12}-4, \lambda_5 = -z^{14}+2z^{12}-z^{10}, \lambda_6 = -z^{12}+2z^{10}-z^8$
	Input:$f \in \mathbb{G}_{\Phi_{14}(p)}$, **Output**:$h \in \mathbb{G}_T$, **Cost**:$16\mathbf{e}+7\mathbf{M}+\mathbf{S}_c+4\mathbf{f}$
	$f_1 \leftarrow f^{z^4-2z^2+1}, f_2 \leftarrow f_1^{z^2}, f_3 \leftarrow f_2^{z^2}, f_4 \leftarrow f_3^{z^2}, f_5 \leftarrow f_4 \cdot f_3^p \cdot f_2^{p^2},$ $f_6 \leftarrow (f_5^{z^6} \cdot f^4)^{p^4}, h \leftarrow f_1^{p^3} \cdot f_5 \cdot \bar{f}_6$

inversion operation, we adopt the formulas described in [32, **Sect**. 5]. For multiplication and squaring arithmetic, we combine the lazy reduction technique [6,45] and the Karatsuba algorithm [37]. In particular, cyclotomic squaring arithmetic can be accelerated using the formula described in [29, **Sect**.2.1]. The exact operation counts for finite field arithmetic across different pairing-friendly curves are presented in **Tab**. 7.

Recall from **Sect**. 3.1 that the optimized formulas for the Miller function on our target curves are expressed as

$$\begin{cases} f_{z,Q}^{z \cdot p^{k-m}}(P) \cdot f_{z,Q}(\hat{\tau}(P)) \cdot f_{z,Q}(P) \cdot (y_P - y_Q)^{p^m}, & \text{if } j(E) = 1728; \\ f_{z,Q}^{z \cdot p^{k-m}}(P) \cdot f_{z,Q}(\hat{\tau}(P)), & \text{if } j(E) = 0. \end{cases}$$

Table 6. Parameters of full extension fields and curve equations for the five candidate pairing-friendly curves.

curve	full extension field	original curve E	twisted curve E'
BW10-480	$\mathbb{F}_p \xrightarrow{\xi^5+11} \mathbb{F}_{p^5} \xrightarrow{v^2-\xi} \mathbb{F}_{p^{10}}$	$y^2 = x^3 + x$	$y^2 = x^3 + \xi^2 x$
BW10-511	$\mathbb{F}_p \xrightarrow{\xi^5+4} \mathbb{F}_{p^5} \xrightarrow{v^2-\xi} \mathbb{F}_{p^{10}}$	$y^2 = x^3 - 2$	$y^2 = x^3 - 2/\xi^3$
BW10-512	$\mathbb{F}_p \xrightarrow{\xi^5+17} \mathbb{F}_{p^5} \xrightarrow{v^2-\xi} \mathbb{F}_{p^{10}}$	$y^2 = x^3 + x$	$y^2 = x^3 + \xi^2 x$
BW14-351	$\mathbb{F}_p \xrightarrow{\xi^7-2} \mathbb{F}_{p^7} \xrightarrow{v^2-\xi} \mathbb{F}_{p^{14}}$	$y^2 = x^3 + 3$	$y^2 = x^3 + 3/\xi^3$
BW14-382	$\mathbb{F}_p \xrightarrow{\xi^7-17} \mathbb{F}_{p^7} \xrightarrow{v^2-\xi} \mathbb{F}_{p^{14}}$	$y^2 = x^3 + x$	$y^2 = x^3 + \xi^2 x$

The computation of $f_{z,Q}^{z\cdot p^{k-m}}(P)\cdot f_{z,Q}(\hat{\tau}(P))$ can be performed using **Alg**. 2, and it requires additional $2\mathbf{M}+\mathbf{f}+\tilde{\mathbf{a}}$ to complete the final step of the Miller iteration on curves with j-invariant 1728. In conclusion, the total operation count of Miller Loop (ML) is

$$\begin{aligned}\mathrm{ML} = &\underbrace{2\mathbf{M}+\mathbf{f}+\tilde{\mathbf{a}}}_{\text{if } j(E)=1728} + \underbrace{(\mathtt{nbits}(z)-1)\cdot\mathtt{SDBL} + (\mathtt{hw}(z)-1)\cdot\mathtt{SADD}}_{\text{Lines 1-9 in Alg.2}} + \\ &\underbrace{\big((\mathtt{nbits}(z)-1)+2\mathtt{hw}(z)-2\big)\cdot\mathbf{M} + \big(\mathtt{nbits}(z)-1\big)\cdot\mathbf{S}+\mathbf{f}}_{\text{Lines 10-16 in Alg.2}},\end{aligned} \tag{6}$$

where $\mathtt{nbits}(z)$ and $\mathtt{hw}(z)$ represent the bit length and the Hamming weight in 2-non-adjacent form of the seed z, respectively. We use n_1, n_2, n_3 and n_4 to denote the number of $\mathbf{e}$, $\mathbf{M}$, $\mathbf{S}_c$ and $\mathbf{f}$ required for the exponentiation to the power of the hard part, respectively. Then the total operation count of the final exponentiation (FE) is

$$\begin{aligned}\mathrm{FE} =& \underbrace{\mathbf{I}+3\mathbf{M}+2\mathbf{f}}_{\text{easy part}} + \underbrace{n_1\big((\mathtt{nbits}(z)-1)\mathbf{S}_c+(\mathtt{hw}(z)-1)\mathbf{M}\big)+n_2\mathbf{M}+n_3\mathbf{S}_c+n_4\mathbf{f}}_{\text{hard part}} \\ =&\mathbf{I}+\big(n_1(\mathtt{hw}(z)-1)+n_2+3\big)\mathbf{M}+\big(n_1(\mathtt{nbits}(z)-1)+n_3\big)\mathbf{S}_c+(n_4+2)\mathbf{f}.\end{aligned} \tag{7}$$

In the following, we take BW14-351 as an example to analyze the detailed operation count of pairing computation.

Example 1. The selected seed z of BW14-351 has $\mathtt{nbits}(z) = 23$ and $\mathtt{hw}(z) = 4$. Then, it follows from Eq. (6) that the cost of the Miller Loop is:

$$\begin{aligned}\mathrm{ML} =& 22(\mathbf{M}+\mathbf{S}+5\tilde{\mathbf{m}}+\tilde{\mathbf{m}}_u+\tilde{\mathbf{s}}_u+\tilde{\mathbf{m}}_\xi+3\tilde{\mathbf{s}}+21\mathbf{m}+\tilde{\mathbf{r}}+13\tilde{\mathbf{a}}+\mathbf{a})+ \\ & 3(\mathbf{M}+6\tilde{\mathbf{m}}+4\tilde{\mathbf{m}}_u+\tilde{\mathbf{m}}_\xi+3\tilde{\mathbf{s}}+21\mathbf{m}+2\tilde{\mathbf{r}}+12\tilde{\mathbf{a}})+(28\mathbf{M}+22\mathbf{S}+\mathbf{f}) \\ =&53\mathbf{M}+44\mathbf{S}+128\tilde{\mathbf{m}}+34\tilde{\mathbf{m}}_u+75\tilde{\mathbf{s}}+22\tilde{\mathbf{s}}_u+25\tilde{\mathbf{m}}_\xi+525\mathbf{m}+28\tilde{\mathbf{r}}+\mathbf{f}+322\tilde{\mathbf{a}}+22\mathbf{a} \\ =&216\tilde{\mathbf{m}}+193\tilde{\mathbf{m}}_u+219\tilde{\mathbf{m}}_\xi+75\tilde{\mathbf{s}}+22\tilde{\mathbf{s}}_u+134\tilde{\mathbf{r}}+\mathbf{f}+525\mathbf{m}+966\tilde{\mathbf{a}}+22\mathbf{a} \\ =&537\mathbf{m}+11271\mathbf{m}_u+582\mathbf{s}_u+83543\mathbf{a}+2975\mathbf{r} \\ =&11808\mathbf{m}_u+582\mathbf{s}_u+83543\mathbf{a}+3512\mathbf{r}.\end{aligned}$$

Table 7. Costs of arithmetic operations in a tower extension field $\mathbb{F}_{p^k}$ on the five candidate curves.

curve	$\tilde{\mathbf{m}} = \tilde{\mathbf{m}}_u + \tilde{\mathbf{r}}$	$\tilde{\mathbf{s}} = \tilde{\mathbf{s}}_u + \tilde{\mathbf{r}}$	$\tilde{\mathbf{i}}$	$\tilde{\mathbf{m}}_\xi$, $\tilde{\mathbf{m}}_{a'}$
BW10-480	$15\mathbf{m}_u+122\mathbf{a}+5\mathbf{r}$	$7\mathbf{m}_u+6\mathbf{s}_u+84\mathbf{a}+5\mathbf{r}$	$\approx \mathbf{i}+2\tilde{\mathbf{m}}+22\mathbf{m}$	$5\mathbf{a}$, $10\mathbf{a}$
BW10-511	$15\mathbf{m}_u+98\mathbf{a}+5\mathbf{r}$	$7\mathbf{m}_u+6\mathbf{s}_u+60\mathbf{a}+5\mathbf{r}$	$\approx \mathbf{i}+2\tilde{\mathbf{m}}+22\mathbf{m}$	$2\mathbf{a}$, -
BW10-512	$15\mathbf{m}_u+122\mathbf{a}+5\mathbf{r}$	$7\mathbf{m}_u+8\mathbf{s}_u+84\mathbf{a}+5\mathbf{r}$	$\approx \mathbf{i}+2\tilde{\mathbf{m}}+22\mathbf{m}$	$5\mathbf{a}$, $10\mathbf{a}$
BW14-351	$24\mathbf{m}_u+162\mathbf{a}+7\mathbf{r}$	$15\mathbf{m}_u+6\mathbf{s}_u+106\mathbf{a}+7\mathbf{r}$	$\approx \mathbf{i}+3\tilde{\mathbf{m}}+38\mathbf{m}$	$\mathbf{a}$, -
BW14-382	$24\mathbf{m}_u+210\mathbf{a}+7\mathbf{r}$	$15\mathbf{m}_u+6\mathbf{s}_u+154\mathbf{a}+7\mathbf{r}$	$\approx \mathbf{i}+3\tilde{\mathbf{m}}+38\mathbf{m}$	$\mathbf{5a}$, $10\mathbf{a}$
$\mathbf{S}_c$	$\mathbf{S}$	$\mathbf{M}$	$\mathbf{I}$	$\mathbf{f}$
$\tilde{\mathbf{m}}+\tilde{\mathbf{s}}+2\tilde{\mathbf{a}}$	$2\tilde{\mathbf{m}}+5\tilde{\mathbf{a}}+2\tilde{\mathbf{m}}_\xi$	$3\tilde{\mathbf{m}}_u+8\tilde{\mathbf{a}}+2\tilde{\mathbf{m}}_\xi+2\tilde{\mathbf{r}}$	$\tilde{\mathbf{i}}+2\tilde{\mathbf{m}}+\tilde{\mathbf{m}}_\xi+2\tilde{\mathbf{s}}+\tilde{\mathbf{a}}$	$(k-2)\mathbf{m}$

Furthermore, it can be obtained from **Tab**. 5 that the parameters n_1, n_2, n_3 and n_4 are equal to 14, 12, 0 and 6, respectively. By Eq. (7), the cost of the final exponentiation is:

$$\begin{aligned}\mathrm{FE} &= (\mathbf{I}+3\mathbf{M}+2\mathbf{f})+(14\mathbf{e}+12\mathbf{M}+6\mathbf{f}) = \mathbf{I}+57\mathbf{M}+308\mathbf{S}_c+8\mathbf{f}\\ &= \tilde{\mathbf{i}}+310\tilde{\mathbf{m}}+171\tilde{\mathbf{m}}_u+115\tilde{\mathbf{m}}_\xi+310\tilde{\mathbf{s}}+114\tilde{\mathbf{r}}+96\mathbf{m}+1073\tilde{\mathbf{a}}\\ &= \mathbf{i}+16266\mathbf{m}_u+134\mathbf{m}+1860\mathbf{s}_u+5159\mathbf{r}+118894\mathbf{a}\\ &= \mathbf{i}+16400\mathbf{m}_u+1860\mathbf{s}_u+118894\mathbf{a}+5293\mathbf{r}.\end{aligned}$$

In total, the cost of pairing computation on BW14-351 is

$$\mathrm{ML}+\mathrm{FE} = \mathbf{i}+28208\mathbf{m}_u+2442\mathbf{s}_u+202437\mathbf{a}+8805\mathbf{r}.$$

Table 8. Operation Counts of pairing computation on the five candidate pairing-friendly curves.

curve	ML	FE	ML+FE
BW10-480	$12861\mathbf{m}_u+1290\mathbf{s}_u$ $+115357\mathbf{a}+4761\mathbf{r}$	$\mathbf{i}+11591\mathbf{m}_u+2412\mathbf{s}_u$ $+111610\mathbf{a}+4682\mathbf{r}$	$\mathbf{i}+24452\mathbf{m}_u+3702\mathbf{s}_u$ $+226967\mathbf{a}+9443\mathbf{r}$
BW10-511	$10027\mathbf{m}_u+822\mathbf{s}_u$ $+71412\mathbf{a}+3508\mathbf{r}$	$\mathbf{i}+12452\mathbf{m}_u+2706\mathbf{s}_u$ $+94203\mathbf{a}+5130\mathbf{r}$	$\mathbf{i}+22479\mathbf{m}_u+3528\mathbf{s}_u$ $+165615\mathbf{a}+8638\mathbf{r}$
BW10-512	$11761\mathbf{m}_u+1170\mathbf{s}_u$ $+105417\mathbf{a}+4341\mathbf{r}$	$\mathbf{i}+12820\mathbf{m}_u+2610\mathbf{s}_u$ $+123314\mathbf{a}+5130\mathbf{r}$	$\mathbf{i}+24581\mathbf{m}_u+3780\mathbf{s}_u$ $+228731\mathbf{a}+9471\mathbf{r}$
BW14-351	$11808\mathbf{m}_u+582\mathbf{s}_u$ $+83543\mathbf{a}+3512\mathbf{r}$	$\mathbf{i}+16400\mathbf{m}_u+1860\mathbf{s}_u$ $+118894\mathbf{a}+5293\mathbf{r}$	$\mathbf{i}+28208\mathbf{m}_u+2442\mathbf{s}_u$ $+202437a\mathbf{a}+8805\mathbf{r}$
BW14-382	$12594\mathbf{m}_u+720\mathbf{s}_u$ $+115874\mathbf{a}+3874\mathbf{r}$	$\mathbf{i}+19883\mathbf{m}_u+2034\mathbf{s}_u$ $+191396\mathbf{a}+6137\mathbf{r}$	$\mathbf{i}+32477\mathbf{m}_u+2754\mathbf{s}_u$ $+307270\mathbf{a}+10011\mathbf{r}$

In **Tab**. 8, we summarize the costs of pairing computation on the five candidate curves. It should be noted that the selected primes p for BW10-480, BW10-511, and BW10-512 can be represented by 8 computer words in a 64-bit processor, while for BW14-351 and BW14-382 only require 6 computer words. As illustrated in [4, **Sect**. 8], it is reasonable to estimate that $\mathbf{m}_8 \approx (136/78)\mathbf{m}_6 \approx 1.74\mathbf{m}_6$ and $\mathbf{a}_8 \approx (8/6)\mathbf{a}_6 \approx 1.33\mathbf{a}_6$, where $\mathbf{m}_i$ and $\mathbf{a}_i$ denote the costs of multiplication and addition in $\mathbb{F}_p$, with p a i computer word size prime in a 64-bit processor. Based on the estimate and **Tab**. 8, we predict that BW14-351 is the most efficient choice among the five candidate curves for pairing computation.

5 Subgroup Membership Testing

In pairing-based cryptographic protocols, subgroup membership testing plays a critical role in defending against small subgroup attacks. [10, 41]. Recent research [17, 47] has demonstrated that efficiently computable endomorphisms are powerful tools for accelerating these testings in various pairing groups. In this section, we describe the application of state-of-the-art technique [17] to our specific pairing-friendly curves. Furthermore, we also introduce a faster method for $\mathbb{G}_2$ membership testing.

5.1 $\mathbb{G}_1$ membership testing

Given a candidate point P, the process of verifying whether $P \in \mathbb{G}_1$ can be divided into two phases. Concretely, one can first check whether $P \in E(\mathbb{F}_p)$, followed by verifying that the order of P is exactly r. It is clear that the computational cost largely comes from the second phase. Let the endomorphism τ on $\mathbb{G}_1$ act as scalar multiplication by λ_1, and let $\mathcal{L}_\tau$ be a two dimensional lattice as

$$\mathcal{L}_\tau = \{(a_0, a_1) \in \mathbb{Z}^2 | a_0 + a_1 \cdot \lambda_1 \equiv 0 \bmod r\}.$$

By [49, Theorem 2], the norm of the shortest vector in $\mathcal{L}_\tau$ is about $\log r/2$. We let (a_0, a_1) be a vector in $\mathcal{L}_\tau$ with $\gcd(h_1, h_1') = 1$, where $h_1 = \#E(\mathbb{F}_p)/r$ and

$$h_1' = \begin{cases} (a_0^2 - a_0 \cdot a_1 + a_1^2)/r, & \text{if } j(E) = 0; \\ (a_0^2 + a_1^2)/r, & \text{if } j(E) = 1728. \end{cases} \tag{8}$$

Dai et al. [17] prove that the short vector (a_0, a_1) can be used to accelerate $\mathbb{G}_1$ membership testing, i.e.,

$$P \in \mathbb{G}_1 \Leftrightarrow P \in E(\mathbb{F}_p) \text{ and } [a_0]P + [a_1]\tau(P) = \mathcal{O}_E.$$

In general, the constraint $\gcd(h_1, h_1') = 1$ is mild and thus one can find a valid short vector close to the shortest one on many pairing-friendly curves. It means that the process of $\mathbb{G}_1$ membership testing requires about $\log r/2$ iterations.

5.2 $\mathbb{G}_T$ membership testing

In the case of $\mathbb{G}_T$ membership testing, the Frobenius endomorphism is critical in finding valid short vectors. To illustrate it, we first use $\mathcal{L}_\pi$ to denote the following $\varphi(k)$ dimensional lattice:

$$\mathcal{L}_\pi = \{(a_0, \cdots, a_{\varphi(k)-1}) \in \mathbb{Z}^{\varphi(k)} | a_0 + a_1 \cdot p + \cdots + a_{\varphi(k)-1} \cdot p^{\varphi(k)-1} \equiv 0 \bmod r\}.$$

The norm of the shortest vector in $\mathcal{L}_\pi$ is about $\log r/\varphi(k)$. For a given short vector $\mathbf{c} = (c_0, c_1, \cdots, c_{\varphi(k)-1}) \in \mathcal{L}_\pi$, we define that

$$h_T = \Phi_k(p)/r \text{ and } h'_T = \sum\nolimits_{i=0}^{\varphi(k)-1} c_i \cdot p^i.$$

Dai et al. found that if the short vector $\mathbf{c}$ satisfies $\gcd(h_T, h'_T) = 1$, then

$$\alpha \in \mathbb{G}_T \Leftrightarrow \alpha^{\Phi_k(p)} = 1 \text{ and } \prod_{i=0}^{\varphi(k)-1} \alpha^{c_i \cdot p^i} = 1.$$

Likewise, the condition $\gcd(h_T, h'_T) = 1$ is mild, and thus the process of $\mathbb{G}_T$ membership testing requires about $\log r/\varphi(k)$ iterations.

Modified short vector: The previous idea for optimizing the final exponentiation still applies to $\mathbb{G}_T$ membership testing such that several full extension field multiplications can be saved. Specifically, once the candidate element α proved to be a member of $\mathbb{G}_{\Phi_k(p)}$, one can replace the original valid vector $\mathbf{c}$ by $\mathbf{c}' = (c_0 + \delta, c_1 - \delta, \cdots, c_{\varphi(k)-1} - \delta, \delta)$ for some integer δ for our target curves as

$$\prod_{i=0}^{\varphi(k)-1} \alpha^{c_i \cdot p^i} = 1 \Leftrightarrow \alpha^{\delta \cdot \Phi_k(p)} \cdot \prod_{i=0}^{\varphi(k)-1} \alpha^{c_i \cdot p^i} = 1.$$

In particular, if the first i tuples of $\mathbf{c}'$ are 0, we then can obtain a new vector as $(c_{i+1} + (-1)^{i+1}\delta, \cdots, c_{\varphi(k)-1} - \delta, \delta, 0, \cdots, 0)$. For instance, using the Magma code provided in [17, **Sect**. 5], a valid vector for $\mathbb{G}_T$ membership testing on BW14-351 is given by $\mathbf{c} = (1, -1, 1, z^2 - 1, -z^2 + z + 1, -z)$. Taking $\delta = -1$, we have

$$(c_0 - 1, c_1 + 1, \cdots, c_6 - 1, 1) = (0, 0, 0, z^2, -z^2 + z, -z + 1, -1).$$

By left-shifting the above vector, a modified short vector $(z^2, -z^2 + z, -z + 1, -1, 0, 0, 0)$ is obtained. Consequently, it is equivalent to checking that

$$\alpha \cdot \alpha^{(p+p^3+p^5)\cdot p} = \alpha^{p+p^3+p^5}, \alpha^{p^3} = \alpha^{z^2} \cdot \alpha^{(z-z^2)\cdot p} \cdot \alpha^{(1-z)\cdot p^2}.$$

5.3 $\mathbb{G}_2$ membership testing

Recall from **Sect**. 2.1 that ψ and η represent two efficiently computable endomorphisms on E' with $j(E') = 0$ or 1728. For a given short vector $\mathbf{c} = (c_0, c_1, \cdots, c_{\varphi(k)-1}) \in \mathcal{L}_\pi$, we define that

$$h_2 = \#E'(\mathbb{F}_{p^{k/2}})/r \text{ and } h'_2 = \sum\nolimits_{i=0}^{\varphi(k)-1} c_i \cdot p^i.$$

Dai et al. method for $\mathbb{G}_2$ membership testing is summarized as follows: If the short vector $\mathbf{c}$ satisfies that $\gcd(h_2, h_2') = 1$, then

$$Q \in \mathbb{G}_2 \Leftrightarrow Q \in E'(\mathbb{F}_{p^{k/2}}) \text{ and } \sum_{i=0}^{\varphi(k)-1} [c_i]\psi^i(Q) = \mathcal{O}_{E'}.$$

Again, the above computation requires about $\log r/\varphi(k)$ iterations. In the following, we develop a faster method for $\mathbb{G}_2$ membership testing, which is tailored to our target curves. To this aim, we first determine the characteristic equation of the endomorphism $\Psi = \psi \circ \eta$.

Lemma 2. *Let E be an ordinary curve over $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = p + 1 - t$, admitting a twist E'. If $j(E') = 0$ or 1728, then the characteristic equation of Ψ is given as follows:*

(1) $j(E') = 0: \quad \Psi^2 + \frac{t \pm 3f}{2}\Psi + p = 0$ *with* $t^2 - 4p = -3f^2$;
(2) $j(E') = 1728: \quad \Psi^2 \pm f\Psi + p = 0$ *with* $t^2 - 4p = -f^2$.

Proof. We only give the proof of the case $j(E') = 0$ (The proof of the remaining case is similar). As mentioned in **Sect**. 2.1, the characteristic equation of Ψ can be expressed as

$$\Psi^2 + m\Psi + n = 0 \tag{9}$$

for some integers m and n. Since $\mathrm{Nrd}(\psi) = p$ and $\mathrm{Nrd}(\eta) = 1$, we have

$$n = \mathrm{Nrd}(\Psi) = \mathrm{Nrd}(\psi) \cdot \mathrm{Nrd}(\eta) = p.$$

Furthermore, the characteristic equation of ψ and η are given as follows:

$$\psi^2 - t\psi + p = 0, \quad \eta^2 + \eta + 1 = 0.$$

It is easy to deduce that

$$\psi = \frac{t \pm \sqrt{-3} \cdot f}{2} \text{ and } \eta = \frac{-1 \pm \sqrt{-3}}{2}.$$

By the fact that $\Psi = \psi \circ \eta$, we have

$$\Psi = \frac{t \pm \sqrt{-3} \cdot f}{2} \cdot \frac{-1 \pm \sqrt{-3}}{2} = \frac{-(t \pm 3f) \pm \sqrt{-3} \cdot (t - f)}{4}. \tag{10}$$

On the other hand, it can be obtained from Eq. (9) that

$$\Psi = \frac{-m \pm \sqrt{m^2 - 4n}}{2}. \tag{11}$$

By comparing Eqs.(10) and (11), we conclude that $m = (t \pm 3f)/2$, which completes the proof. □

Recall that the endomorphism η acts on $\mathbb{G}_2$ as scalar multiplication by λ_2 that is defined in Eq. (3). By combining the actions of ψ and η on $\mathbb{G}_2$ together, we have $\Psi(Q) = [\ell]Q$ for any $Q \in \mathbb{G}_2$, where $\ell = p \cdot \lambda_2 \bmod r$. Since the order of Ψ restricting on the $\mathbb{F}_{p^{k/2}}$ rational endomorphism ring is equal to $2k$ or $3k$ on our target curves, we have $r \mid \Phi_{2k}(\ell)$ or $r \mid \Phi_{3k}(\ell)$. The degree of each of the two cyclotomic polynomials is equal to $2\varphi(k)$. For this reason, we can construct the following $2\varphi(k)$ dimensional lattice:

$$\mathcal{L}_\Psi = \{(a_0, \cdots, a_{2\varphi(k)-1}) \in \mathbb{Z}^{2\varphi(k)} | a_0 + a_1 \cdot \ell + \cdots + a_{2\varphi(k)-1} \cdot \ell^{2\varphi(k)-1} \equiv 0 \bmod r\}.$$

Given a short vector $\mathbf{c} = (c_0, c_1, \cdots, c_{2\varphi(k)-1}) \in \mathcal{L}_\Psi$, we define that

$$g(\Psi) = \Psi^2 - t_\Psi \Psi + p \text{ and } h(\Psi) = \sum\nolimits_{i=0}^{2\varphi(k)-1} c_i \Psi^i,$$

where t_Ψ is the trace of Ψ that is given in Lemma 2. By taking full advantage of the endomorphism Ψ, a new method for $\mathbb{G}_2$ membership testing is proposed, which is tailored to our target curves.

Theorem 1. *Let E be an ordinary curve over $\mathbb{F}_p$ with j-invariant 0 or 1728. Let r be a large prime such that $r \mid \#E(\mathbb{F}_p)$. Suppose E admit a twist E' of degree 2 such that $r \mid \#E'(\mathbb{F}_{p^{k/2}})$. Let $\boldsymbol{c} = (c_0, c_1, \cdots, c_{2\varphi(k)-1}) \in \mathcal{L}_\Psi$,and let $\mathbf{Res}(h(\Psi), g(\Psi))$ be the resultant of $h(\Psi)$ and $g(\Psi)$. Assume that the short vector $\boldsymbol{c}$ satisfies that*

$$\gcd(\mathbf{Res}\big(h(\Psi), g(\Psi)\big), h_2 \cdot r) = r. \tag{12}$$

For any non-identity point Q of $E'(\mathbb{F}_{p^{k/2}})$, the point $Q \in \mathbb{G}_2 = E'(\mathbb{F}_{p^{k/2}})[r]$ if and only if

$$\sum\nolimits_{i=0}^{2\varphi(k)-1} [c_i]\Psi^i(Q) = \mathcal{O}_{E'}. \tag{13}$$

Proof. If $Q \in \mathbb{G}_2$, then we have $\Psi(Q) = [\ell]Q$. As a result, we can easily check that

$$\sum\nolimits_{i=0}^{2\varphi(k)-1} [c_i]\Psi^i(Q) = \sum\nolimits_{i=0}^{2\varphi(k)-1} [c_i \ell^i]Q = \mathcal{O}_{E'}.$$

Conversely, we let b_0 and b_1 be two integers satisfying that $b_0 + b_1\Psi = h(\Psi) \bmod g(\Psi)$. By the property of resultant, we have

$$\mathbf{Res}(h(\Psi), g(\Psi)) = \mathbf{Res}(b_0 + b_1\Psi, g(\Psi)) = b_0^2 + b_0 b_1 t_\Psi + b_1^2 p.$$

Furthermore, by the fact that $h(\Psi)(Q) = g(\Psi)(Q) = \mathcal{O}_{E'}$, we have

$$[b_0^2 + b_0 b_1 t_\Psi + b_1^2 p]Q = (b_0 + b_1\hat{\Psi})(b_0 + b_1\Psi)(Q) = \mathcal{O}_{E'}.$$

Therefore, the order of Q divides $\gcd(\mathbf{Res}(h(\Psi), g(\Psi)), h_2 \cdot r)$. Since the selected vector $\mathbf{c}$ is restricted by Eq. (12), we conclude that $Q \in E'(\mathbb{F}_{p^{k/2}})[r] = \mathbb{G}_2$, which completes the proof. □

Likewise, the new approach requires about $\log r/(2\varphi(k))$ bit operations, which is about $2\times$ as fast as the previous leading work [17]. In **Tab**. 9, we list the short vectors that can be used for $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ membership membership testings on the five candidate pairing-friendly curves. It is straightforward to see that the computational cost of $\mathbb{G}_2$ membership testing on the five candidate curves comes largely from a scalar multiplication by z.

Table 9. Short vectors for subgroup membership testings on five candidate pairing-friendly curves.

curve	$\mathbb{G}_1$ (a_0, a_1)	$\mathbb{G}_2$	$\mathbb{G}_T$
BW10-480	$(z^3 - z, -1 - a_0 \cdot z)$	$(1, 0, 0, -z, 0, 0, 0, 0)$	$(z^2, 0, 0, 0, 1)$
BW10-511	$(a_1 \cdot z - 1, z^3 + z^2 - 1)$	$(1, 0, -z - 1, -1, 0, 0, 1, 1)$	$(1, -z^2, 0, z, 0)$
BW10-512	$(z^3 - z, -a_0 \cdot z - 1)$	$(0, 1, 0, z - 1, 0, 1, -z + 1, -1)$	$(1, z^2 - 1, 0, z^2 - 1)$
BW14-351	$(z^5 + z^4 - z^2 - z, (1 - z) \cdot a_0 - 1)$	$(1, 1, 0, -1, -1, 0, 1, 0, -1, -1, 0, z + 1)$	$(z^2, z - z^2, 1 - z, -1)$
BW14-382	$(z^5 - z^3 + z, -1 + a_0 \cdot z)$	$(0, 1, z, -1, 0, 1, 0, -1, 1, 1, 0, z - 1)$	$(z^2, -1, z^2, -1)$

6 Cofactor Multiplication

Hashing a string into $\mathbb{G}_1$ or $\mathbb{G}_2$ is an important building block in pairing-based cryptographic protocols. This operation consists of two phases: first mapping a string into a curve point, followed by a cofactor multiplication so that the resulting point falls into the target subgroup. In this section, we present efficient algorithms for cofactor multiplication for $\mathbb{G}_1$ and $\mathbb{G}_2$ on our chosen target curves.

6.1 Cofactor multiplication for $\mathbb{G}_1$

Given a random point $P \in E(\mathbb{F}_p)$, cofactor multiplication for $\mathbb{G}_1$ is to map the point P into $\mathbb{G}_1$. The naive way is to perform the scalar multiplication $[h_1]P$, where the cofactor $h_1 = \#E(\mathbb{F}_p)/r$. EI Housni, Guillevic and Piellard [23] observed that the cofactor h_1 can be replaced by a smaller cofactor $\tilde{h}_1$ on a large class of cyclotomic pairing-friendly curves, where $\tilde{h}_1$ is determined by the group structure of $E(\mathbb{F}_p)$:

$$E(\mathbb{F}_p) \cong \mathbb{Z}_{m_1} \oplus \mathbb{Z}_{\tilde{h}_1 \cdot r} \text{ with } m_1 \mid \tilde{h}_1 \text{ and } m_1 \cdot \tilde{h}_1 = h_1.$$

In particular, if the curve E has j-invariant 0 or 1728, then m_1 is the largest integer such that $m_1^2 \mid \#E(\mathbb{F}_p)$ and $m_1 \mid (p-1)$. Thus, it is not difficult to determine the value of m_1 on the five candidate curves. In the optimal case, we have $m_1 \approx \tilde{h}_1$ and thus the new method would be twice as fast as the naive one, e.g. for the BW10-480 curve.

Faster cofactor multiplication for $\mathbb{G}_1$: The algorithm of EI Housni-Guillevic-Piellard can be further optimized in the case that $m_1 \ll \tilde{h}_1$, such as for the

BW10-511, BW10-512, BW14-351 and BW14-382 curves. In fact, a random point $P \in E(\mathbb{F}_p)$ can be mapped into $\mathbb{G}_1$ as follows:

$$E(\mathbb{F}_p) \xrightarrow{m_1} E(\mathbb{F}_p)[n_1 \cdot r] \xrightarrow{a_0+a_1\tau} E(\mathbb{F}_p)[r] = \mathbb{G}_1.$$

In detail, the first step is to map the point P into the cyclic group $E(\mathbb{F}_p)[n_1 \cdot r]$ by performing a scalar multiplication by m_1, where $n_1 = \tilde{h}_1/m_1$; the next step is to clear the cofactor n_1 using the endomorphism $a_0 + a_1 \cdot \tau$, where a_0 and a_1 are integers satisfying $a_0 + a_1 \cdot s_1 \equiv 0 \bmod n_1$ and s_1 denotes the scalar of the endomorphism τ acting on $E(\mathbb{F}_p)[n_1 \cdot r]$. More specifically, the LLL algorithm can be exploited to look for two integers a_0 and a_1 such that $\max\{\log|a_0|, \log|a_1|\} \approx \log n_1/2$. In conclusion, cofactor multiplication for $\mathbb{G}_1$ can always be performed in around $\log m_1 + \log n_1/2 \approx \log h_1/2$ iterations, which does not depend on the group structure of $E(\mathbb{F}_p)$. In **Tab**.10, we summarize the parameters h_1, m_1 and $\tilde{h}_1$, and short vectors (a_0, a_1) across different pairing-friendly curves.

Table 10. Important parameters for cofactor multiplication for $\mathbb{G}_1$ on the five candidate pairing-friendly curves.

curve	h_1	m_1	$\tilde{h}_1$	n_1	(a_0, a_1)
BW10-480	$\frac{z^4}{4}$	$\frac{z^2}{2}$	$\frac{z^2}{2}$	1	–
BW10-511	$\frac{(z^2-z+1)(z^3-1)^2}{3}$	$\frac{(z^3-1)}{3}$	$(z^2-z+1)(z^3-1)$	$3(z^2-z+1)$	$(1, z)$
BW10-512	$\frac{(z^2-1)^2(z^2+1)}{4}$	$\frac{(z^2-1)}{2}$	$\frac{(z^2-1)(z^2+1)}{2}$	z^2+1	$(z, -1)$
BW14-351	$\frac{(z^2-z+1)(z^2+z+1)}{3}$	1	$\frac{(z^2-z+1)(z^2+z+1)}{3}$	$\frac{(z^2-z+1)(z^2+z+1)}{3}$	$(2z, z^2+z-1)$
BW14-382	$\frac{(z^2-1)^2(z^2+1)}{4}$	$\frac{(z^2-1)}{2}$	$\frac{(z^2-1)(z^2+1)}{2}$	z^2+1	$(z, 1)$

6.2 Cofactor multiplication for $\mathbb{G}_2$

Cofactor multiplication for $\mathbb{G}_2$ aims to map a random point Q of $E'(\mathbb{F}_{p^{k/2}})$ into $\mathbb{G}_2$. The naive way is to compute $[h_2]Q$ directly, where $h_2 = \#E'(\mathbb{F}_{p^{k/2}})/r$. Since the cofactor h_2 is much larger than the cofactor h_1 and $\mathbb{G}_2$ is defined over $\mathbb{F}_{p^{k/2}}$, the computational cost of the cofactor multiplication for $\mathbb{G}_2$ is more expensive than that for $\mathbb{G}_1$. To date, the fastest known algorithm [25] requires approximately $\log h_2/\varphi(k)$ iterations to clear the cofactor. Recently, Dai et al. [19] proposed a fast method for this operation on curves with the lack of twists. In this subsection, we show that this method can be generalized to our target curves such that the number of iterations can be further reduced to $\log h_2/(2\varphi(k))$.

Lemma 3. *Let* $G'_0 = \{Q \in E'(\mathbb{F}_{p^{k/2}}) | \Phi_k(\psi)(Q) = \mathcal{O}_{E'}\}$. *Then the order of* G'_0 *is precisely equal to* $\frac{\#E'(\mathbb{F}_{p^{k/2}}) \cdot \#E(\mathbb{F}_p)}{\#E(\mathbb{F}_{p^2})}$.

Proof. Let $G_0 = \{Q \in E(\mathbb{F}_{p^k}) | \Phi_k(\pi)(Q) = \mathcal{O}_E\}$. It is easy to see that $G_0 \cong G'_0$ and thus $\#G_0 = \#G'_0$. By [19, Proposition 2], we have

$$\#G_0 = \frac{\#E(\mathbb{F}_{p^k}) \cdot \#E(\mathbb{F}_p)}{\#E(\mathbb{F}_{p^{k/2}}) \cdot \#E(\mathbb{F}_{p^2})}. \tag{14}$$

On the other hand, it can be obtained from [34, Theorem 3] that

$$\#E(\mathbb{F}_{p^k}) = \#E(\mathbb{F}_{p^{k/2}}) \cdot \#E'(\mathbb{F}_{p^{k/2}}). \tag{15}$$

Inserting Eq.(14) into Eq.(15), it yields that

$$\#G'_0 = \#G_0 = \frac{\#E'(\mathbb{F}_{p^{k/2}}) \cdot \#E(\mathbb{F}_p)}{\#E(\mathbb{F}_{p^2})}, \tag{16}$$

which completes the proof of this lemma. □

Since $\mathbb{G}_2$ is a subgroup of G'_0, we define that $G'_0 \cong \mathbb{Z}_{m_2} \oplus \mathbb{Z}_{m_2 \cdot n_2 \cdot r}$ for some integers m_2 and n_2. As a consequence, the process of mapping a random point of $E'(\mathbb{F}_{p^{k/2}})$ into $\mathbb{G}_2$ can be divided into the following three steps:

$$E'(\mathbb{F}_{p^{k/2}}) \to G'_0 \to E'(\mathbb{F}_{p^{k/2}})[n_2 \cdot r] \to \mathbb{G}_2.$$

Since the integer $k/2$ is prime for our chosen curves, a random point $Q \in E'(\mathbb{F}_{p^{k/2}})$ can be mapped into the group G'_0 under the endomorphism $\psi + 1$. It is clear that the computational cost of operations largely comes from the last step. In the following, we show how to map a random point of $E'(\mathbb{F}_{p^{k/2}})[n_2 \cdot r]$ into $\mathbb{G}_2$. To illustrate it, we first introduce the two lemmas.

Lemma 4. *Let t' be the trace of the $p^{k/2}$ power Frobenius endomorphism of E'. Let $f, f' \in \mathbb{Z}$ be such that $t^2 - 4p = -Df^2$ and $t'^2 - 4p^{k/2} = -Df'^2$, where $-D$ is the square-free part of $t^2 - 4p$. Let H be a cyclic subgroup of G'_0 with order $n_2 \cdot r$. Then $\psi(P) = [a]Q$ for any $Q \in H$, where $a = \frac{t \pm f(t'-2)}{2f'} \bmod n_2 \cdot r$.*

Proof. The proof is given in [25, Lemma 2]. □

As illustrated in [25], Lemma 4 induces a fast approach for cofactor multiplication for $\mathbb{G}_2$ in $\log n_2/\varphi(k)$ iterations on a large class of pairing-friendly curves.

Lemma 5. *Let H be a cyclic subgroup of G'_0 with order $n_2 \cdot r$. Then $\eta(Q) = [b]Q$ for any $Q \in H$, where*

$$b = \begin{cases} \dfrac{-f \pm (2a - t)}{2f} \bmod n_2 \cdot r, & \text{if } j(E) = 0, \\ \dfrac{\pm(2a - t)}{f} \bmod n_2 \cdot r, & \text{if } j(E) = 1728. \end{cases}$$

Proof. The proof is derived from [19, Lemma 2]. □

In the following, we propose a more efficient approach for cofactor multiplication for $\mathbb{G}_2$ suitable for curves listed in **Tab**. 1. Our main idea is summarized in the theorem below.

Theorem 2. *Let E be an ordinary elliptic curve admitting a degree-2 twist E' over an extension field $\mathbb{F}_{p^{k/2}}$, where k is the even embedding degree. Let H be a cyclic subgroup of G_0'. If the curve E satisfies the following two conditions: $(i) j(E) \in \{0, 1728\}$; $(ii) 3 \nmid k$ and $4 \nmid k$, then there exists a polynomial*

$$h(x) = h_0 + h_1 x + \cdots + h_{s-1} x^{s-1} \in \mathbb{Z}[x]$$

such that $h(\Psi)(Q) \in \mathbb{G}_2$ for any $Q \in H$, where $s = 2\varphi(k)$ and $|h_i| < |n_2|^{1/s}$ for $i = 0, \cdots, s-1$.

Proof. Since $\Psi = \psi \circ \eta$, it can be deduced from Lemmas 4 and 5 that $\Psi(Q) = [\lambda_2]Q$, where $\lambda_2 = a \cdot b \bmod n_2 \cdot r$. Under the condition that $3 \nmid k$ and $4 \nmid k$, we can deduce that the order of Ψ acting on the group G_0' is $2k$ or $3k$, which means that

$$\begin{cases} \Phi_{3k}(\lambda_2) \equiv 0 \bmod n_2 \cdot r, & \text{if } j(E) = 0; \\ \Phi_{2k}(\lambda_2) \equiv 0 \bmod n_2 \cdot r, & \text{if } j(E) = 1728. \end{cases}$$

In both cases, the degree of the cyclotomic polynomial is $2\varphi(k)$. Analogous to [25, Theorem 1], there exists a polynomial

$$h(x) = h_0 + h_1 x + \cdots + h_{\varphi(k)-1} x^{2\varphi(k)-1} \in \mathbb{Z}[x]$$

such that $h(\lambda_2)$ is a multiple of n_2, where $|h_i| < |n|^{1/2\varphi(k)}$. Therefore, we have $h(\Psi)Q \in \mathbb{G}_2$ for any $Q \in H$, which completes the proof of this theorem. □

By Theorem 2, the number of iterations for $\mathbb{G}_2$ cofactor multiplication can be reduced to $\frac{\log n_2}{2\varphi(k)} \approx \frac{\log h_2}{2\varphi(k)}$ on the curves listed in **Tab**. 1, which is faster than the previous leading work [25]. In the following, we take the BW14-351 curve as an example to describe the main mechanics of the new algorithm.

Example 2 (Cofactor multiplication for $\mathbb{G}_2$ on BW14-351). We first can check that $\gcd(\#G_0', p^7 - 1) = 1$ on BW14-351, where $\#G_0'$ can be obtained from Lemma 3. It follows from [19, Proposition 1] that G_0' is cyclic. Applying the LLL algorithm, we can obtain a target vector $(h_0, h_1, \cdots, h_{11})$, where

$$h_i = \begin{cases} 0, & \text{if } 9 \le i \le 11; \\ 2, & \text{if } i = 8; \\ z^2 + z + 1, & \text{if } i = 6; \\ z h_{i+1}, & \text{if } 2 \le i \le 5; \\ z h_2 - 1, & \text{if } i = 1; \\ h_1 + h_4 - h_3 - h_6 + z + 2, & \text{if } i = 0; \\ -h_1 - h_4 + h_2 + h_5 + 1. & \text{if } i = 7. \end{cases}$$

Given a random point $Q \in E'(\mathbb{F}_{p^7})$, we fist obtain the point $P = (\psi + 1)(Q)$. Then, we have $h(\Psi)P = \sum_{i=0}^{8} \Psi^i(R_i) \in \mathbb{G}_2$, where R_i is given as follows:

$$
\begin{aligned}
R_8 &= [2]P, \\
R_6 &= [z^2 + z + 1]P, \\
R_i &= [z]R_{i+1},\ 2 \leq i \leq 5, \\
R_1 &= [z]R_2 - P, \\
R_7 &= -(R_1 + R_4) + (R_2 + R_5) - P, \\
R_0 &= (R_1 + R_4) - (R_3 + R_6) + [z]P + R_8.
\end{aligned}
$$

In total, cofactor multiplication for $\mathbb{G}_2$ on BW14-351 costs seven scalar multiplications by z, twenty one point additions, one ψ map, and eight Ψ maps.

7 Implementation Results

We first present Magma code to validate the correctness of our proposed algorithms and formulas. Furthermore, we also provide high-speed software implementation for several important pairing group operations on BW10-511 and BW14-351. These two target curves are the winners for pairing computation among our chosen five candidate curves with embedding degrees 10 and 14, respectively. Our implementation is based on the RELIC toolkit, which is a well-known cryptographic library for building pairing-based cryptographic protocols on popular curves at the updated 128 security level, such as BN446 and BLS12-446. In addition, we have observed that the implementation of pairing group operations on BW13-310 presented in [18] also relies on this library. Therefore, we have integrated our code into RELIC to enable fair performance comparisons between the two target curves and these popular curves. Besides our proposed algorithms, we exploit state-of-the-art techniques to implement the following operations.

- The indifferentiable hashing function $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1$ can be implemented by using the SwiftEC map [14], followed by a cofactor multiplication by h_1.
- Since the cofactor $h_2 > r$ for our chosen curves, the construction of the indifferentiable hashing function $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{G}_2$ only require the map $\mathbb{F}_{p^{k/2}} \to E'(\mathbb{F}_{p^{k/2}})$ to be well-distributed [39, **Sect**. 1.2]. Consequently, we can use either the Shallue–van de Woestijne map [48] or the SwiftEC map.
- We employ the GLV method [27] and GLS method [26] to perform group exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_T$, respectively.
- For group exponentiation in $\mathbb{G}_2$ on our target curves, we fortunately find that Dai et al. method [18, **Sect**. 5] can be exploited to achieve a $2\varphi(k)$-dimensional scalar decomposition.
- In terms of the computation of pairings products, we adopt the strategies proposed [28,46,52] such that the final exponentiation step and the squaring computations at the Lines 3 and 12 of **Alg**. 2 can be shared.

Table 11. Benchmarking results of pairing group operations across different pairing-friendly curves reported in 10^3 cycles averaged over 10^4 executions.

Operation\Curve	BLS12-446	BN446	BW13-310	BW10-511	BW14-351
hashing to $\mathbb{G}_1$	327	149	125	621	204
hashing to $\mathbb{G}_2$	1630	1361	16699	11981	7236
exp in $\mathbb{G}_1$	541	791	268	592	362
exp in $\mathbb{G}_2$	918	1394	7247	4621	3531
exp in $\mathbb{G}_T$	1322	2243	1062	1476	1098
test in $\mathbb{G}_1$	389	8	269	723	345
test in $\mathbb{G}_2$	333	487	1176	1262	923
test in $\mathbb{G}_T$	372	540	223	586	384
ML	1554	2480	1719	2819	1600
FE	1835	1589	2579	3872	2337
Single pairing	3389	4069	4298	6691	3937
2-pairings	4439	5717	5640	9016	5205
5-pairings	7614	10532	9621	15621	9008
8-pairings	10790	15349	13603	22191	12811

It should be noted that RELIC supports the GLV decomposition and the SwiftEC map once the associated curve parameters are given. Specifically, fast constant-time evaluation of the SwiftEC map in RELIC is based on the technique proposed in [5].

The implementations are compiled with GCC 11.4.0 and flags `-O3 -funroll-loops -march=native -mtune=native`. The benchmarks are executed on an Intel Core i9-12900K processor running at @3.2GHz with TurboBoost and hyper-threading features disabled. **Tab**. 11 reports detailed performance comparisons for each building block across different curves. The results reveal that BW14-351 outperforms BW10-511 for all pairing group operations. Moreover, BW14-351 exhibits competitive performance compared to mainstream pairing-friendly curves. Specifically, single pairing computation on BW14-351 is slightly faster than that on BN446 and BW13-310, while about 16.2% slower than that on BLS12-446. Regarding group exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_T$, BW14-351 is about 49.4% and 20.4% faster than BLS12-446, 118.5% and 100% faster than BN446, while 35.1% and 3.4% slower than BW13-310. Moreover, compared to BW13-310, BW14-351 benefits from a lighter performance penalty for hashing to $\mathbb{G}_2$ and group exponentiation in $\mathbb{G}_2$, although it remains slower than BN446 and BLS12-446.

These results show that each curve has its own strengths and no one can be said to be perfect. The selection of a curve should be based on a careful analysis of the protocol requirements and a thorough evaluation of the performance trade-offs. The BW14-351 curve may be an appropriate choice if a protocol pursues fast group exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_T$, while wishes to minimize the performance penalty for group exponentiations in $\mathbb{G}_2$. In addition, this curve provides

the 149-bit security level on the finite field side, making it advantageous for achieving long-term security.

8 Conclusion

In this paper, we provided a comprehensive research for a list of pairing-friendly curves with embedding degrees 10 and 14. We generalized Dai-Zhang-Zhao algorithm for pairing computation on BW13-310 to our target curves, so that the number of Miller iterations can be reduced to approximately $\log r/(2\varphi(k))$, while the denominator elimination trick still can be applied. We also proposed optimized algorithms for cofactor multiplication for $\mathbb{G}_1$ and $\mathbb{G}_2$, and subgroup membership testing for $\mathbb{G}_2$ on these curves. After checking the correctness of our proposed algorithms via Magma code, we presented high-speed software implementations on the BW10-511 and BW14-351 curves inside the RELIC library, and compared performance tradeoffs with other popular curves at the same security level, including BN446, BLS12-446 and BW13-310. Our results showed that the BW14-351 curve is competitive for building pairing-based cryptographic protocols at the updated 128-bit security level.

Acknowledgment

We would like to thank Dmitrii Koshelev for insightful discussion about hashing to $\mathbb{G}_2$ on pairing-friendly curves. The work was supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the National Natural Science Foundation of China (Nos. 62325209, 61972428), the Major Program(JD) of Hubei Province (No. 2023BAA027), Guangdong Major Project of Basic and Applied Basic Research (No. 2019B030302008), Guangdong Provincial Key Laboratory of Information Security Technology (No. 2023-B1212060026) and the Fundamental Research Funds for the Central Universities (Nos. 2042023KF0203, 2042024kf1013).

References

1. European union agency of network and information security (ENISA): Algorithms, key sizes and parameters report. https://www.enisa.europa.eu/publications/algorithms-key-sizes-and-parameters-report (2013)
2. Aranha, D.F., Gouvêa, C.P.L.: RELIC is an Efficient LIbrary for Cryptography, https://github.com/relic-toolkit/relic
3. Aranha, D.F., El Housni, Y., Guillevic, A.: A survey of elliptic curves for proof systems. Designs, Codes and Cryptography **91**(11), 3333–3378 (Nov 2023). https://doi.org/10.1007/s10623-022-01135-y
4. Aranha, D.F., Fuentes-Castañeda, L., Knapp, E., Menezes, A., Rodríguez-Henríquez, F.: Implementing pairings at the 192-bit security level. In: Abdalla, M., Lange, T. (eds.) Pairing-Based Cryptography – Pairing 2012. pp. 177–195. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36334-4_11

5. Aranha, D.F., Hvass, B.S., Spitters, B., Tibouchi, M.: Faster constant-time evaluation of the kronecker symbol with application to elliptic curve hashing. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. p. 3228–3238. CCS '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3576915.3616597
6. Aranha, D.F., Karabina, K., Longa, P., Gebotys, C.H., López, J.: Faster explicit formulas for computing pairings over ordinary curves. In: Paterson, K.G. (ed.) Advances in Cryptology – EUROCRYPT 2011. pp. 48–68. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_5
7. Azarderakhsh, R., Fishbein, D., Grewal, G., Hu, S., Jao, D., Longa, P., Verma, R.: Fast software implementations of bilinear pairings. IEEE Transactions on Dependable and Secure Computing **14**(6), 605–619 (2017). https://doi.org/10.1109/TDSC.2015.2507120
8. Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. Journal of Cryptology **32**(4), 1298–1336 (2019). https://doi.org/10.1007/s00145-018-9280-5
9. Barbulescu, R., Gaudry, P., Kleinjung, T.: The tower number field sieve. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015. pp. 31–55. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_2
10. Barreto, P.S.L.M., Costello, C., Misoczki, R., Naehrig, M., Pereira, G.C.C.F., Zanon, G.: Subgroup security in pairing-based cryptography. In: Lauter, K., Rodríguez-Henríquez, F. (eds.) Progress in Cryptology – LATINCRYPT 2015. pp. 245–265. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-22174-8_14
11. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) Selected Areas in Cryptography – SAC 2005. pp. 319–331. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11693383_22
12. Brezing, F., Weng, A.: Elliptic curves suitable for pairing based cryptography. Designs, Codes and Cryptography **37**(1), 133–141 (Oct 2005). https://doi.org/10.1007/s10623-004-3808-4
13. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security. pp. 132–145. Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1030083.1030103
14. Chavez-Saab, J., Rodríguez-Henríquez, F., Tibouchi, M.: SwiftEC: Shallue-van de woestijne indifferentiable function to elliptic curves: Faster indifferentiable hashing to elliptic curves. In: Advances in Cryptology – ASIACRYPT 2022. pp. 63–92. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-22963-3_3
15. Clarisse, R., Duquesne, S., Sanders, O.: Curves with fast computations in the first pairing group. In: Krenn, S., Shulman, H., Vaudenay, S. (eds.) Cryptology and Network Security – CNS2020. pp. 280–298. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-65411-5_14
16. Costello, C., Lange, T., Naehrig, M.: Faster pairing computations on curves with high-degree twists. In: Nguyen, P.Q., Pointcheval, D. (eds.) Public Key Cryptography – PKC 2010. pp. 224–242. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_14
17. Dai, Y., Lin, K., Zhao, C.A., Zhou, Z.: Fast subgroup membership testings for $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ on pairing-friendly curves. Designs, Codes and Cryptography **91**(10), 3141–3166 (2023). https://doi.org/10.1007/s10623-023-01223-7

18. Dai, Y., Zhang, F., Zhao, C.A.: Don't forget pairing-friendly curves with odd prime embedding degrees. IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(4), 393–419 (2023). https://doi.org/10.46586/tches.v2023.i4.393-419
19. Dai, Y., Zhang, F., Zhao, C.A.: Fast hashing to $\mathbb{G}_2$ on pairing-friendly curves with the lack of twists. Finite Fields and Their Applications **91**, 102–263 (2023). https://doi.org/https://doi.org/10.1016/j.ffa.2023.102263
20. De Feo, L., Masson, S., Petit, C., Sanso, A.: Verifiable delay functions from supersingular isogenies and pairings. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 248–277. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-34578-5_10
21. El Housni, Y., Guillevic, A.: Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition. In: Krenn, S., Shulman, H., Vaudenay, S. (eds.) Cryptology and Network Security – CANS 2020. pp. 259–279. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-65411-5_13
22. El Housni, Y., Guillevic, A.: Families of SNARK-friendly 2-chains of elliptic curves. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 367–396. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-07085-3_13
23. El Housni, Y., Guillevic, A., Piellard, T.: Co-factor clearing and subgroup membership testing on pairing-friendly curves. In: Batina, L., Daemen, J. (eds.) Progress in Cryptology – AFRICACRYPT 2022. pp. 518–536. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-17433-9_22
24. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. Journal of Cryptology **23**(2), 224–280 (2010). https://doi.org/10.1007/s00145-009-9048-z
25. Fuentes-Castañeda, L., Knapp, E., Rodríguez-Henríquez, F.: Faster hashing to $\mathbb{G}_2$. In: Miri, A., Vaudenay, S. (eds.) Selected Areas in Cryptography – SAC 2011. pp. 412–430. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28496-0_25
26. Galbraith, S.D., Lin, X., Scott, M.: Endomorphisms for faster elliptic curve cryptography on a large class of curves. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009. pp. 518–535. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
27. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: Kilian, J. (ed.) Advances in Cryptology — CRYPTO 2001. pp. 190–200. Springer Berlin Heidelberg, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_11
28. Granger, R., Smart, N.P.: On computing products of pairings. Cryptology ePrint Archive, Paper 2006/172 (2006), https://eprint.iacr.org/2006/172
29. Granger, R., Scott, M.: Faster squaring in the cyclotomic subgroup of sixth degree extensions. In: Nguyen, P.Q., Pointcheval, D. (eds.) Public Key Cryptography – PKC 2010. pp. 209–223. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_13
30. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology – EUROCRYPT 2016. pp. 305–326. Springer Berlin Heidelberg, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_11

31. Guillevic, A.: A short-list of pairing-friendly curves resistant to special TNFS at the 128-bit security level. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography – PKC 2020. pp. 535–564. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-45388-6_19
32. Guillevic, A., Masson, S., Thomé, E.: Cocks-Pinch curves of embedding degrees five to eight and optimal ate pairing computation. Designs, Codes and Cryptography **88**(6), 1047–1081 (2020). https://doi.org/10.1007/s10623-020-00727-w
33. Guillevic, A., Singh, S.: On the alpha value of polynomials in the tower number field sieve algorithm. Mathematical Cryptology **1**(1) (Feb 2021), open access at https://journals.flvc.org/mathcryptology/article/view/125142
34. Hess, F., Smart, N.P., Vercauteren, F.: The Eta pairing revisited. IEEE Transactions on Information Theory **52**(10), 4595–4602 (2006). https://doi.org/10.1109/TIT.2006.881709
35. Ionica, S., Robert, D.: Pairings. In: El Mrabet, N., Joye, M. (eds.) Guide to pairing-based cryptography. Chapman and Hall/CRC Press, BocaRaton (2016)
36. Joux, A., Pierrot, C.: The special number field sieve in $\mathbb{F}_{p^n}$. In: Cao, Z., Zhang, F. (eds.) Pairing-Based Cryptography – Pairing 2013. pp. 45–61. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-04873-4_3
37. Karatsuba, A.: Multiplication of multidigit numbers on automata. In: Soviet physics doklady. vol. 7, pp. 595–596 (1963)
38. Kim, T., Barbulescu, R.: Extended tower number field sieve: A new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016. pp. 543–571. Springer Berlin Heidelberg, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_20
39. Koshelev, D.: Some remarks on how to hash faster onto elliptic curves. Journal of Computer Virology and Hacking Techniques (Mar 2024). https://doi.org/10.1007/s11416-024-00514-4
40. Lidl, R., Niederreiter, H.: Introduction to finite fields and their applications. Cambridge university press (1994)
41. Lim, C.H., Lee, P.J.: A key recovery attack on discrete log-based schemes using a prime order subgroup. In: Kaliski, B.S. (ed.) Advances in Cryptology — CRYPTO 1997. pp. 249–263. Springer Berlin Heidelberg, Berlin, Heidelberg (1997). https://doi.org/10.1007/BFb0052240
42. Miller, V.S.: The Weil pairing, and its efficient calculation. Journal of Cryptology **17**(4), 235–261 (2004). https://doi.org/10.1007/s00145-004-0315-8
43. Pollard, J.M.: A Monte Carlo method for factorization. Bit Numerical Mathematics **15**(3), 331–334 (1975). https://doi.org/10.1007/BF01933667
44. Schirokauer, O.: Discrete logarithms and local units. Philosophical Transactions: Physical Sciences and Engineering **345**(1676), 409–423 (1993). https://doi.org/10.1098/rsta.1993.0139
45. Scott, M.: Implementing cryptographic pairings. In: Proceedings of the First International Conference on Pairing-Based Cryptography – Pairing 2007. p. 177–196. Springer-Verlag, Berlin, Heidelberg (2007)
46. Scott, M.: On the efficient implementation of pairing-based protocols. In: Chen, L. (ed.) Cryptography and Coding. pp. 296–308. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25516-8_18
47. Scott, M.: A note on group membership tests for $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ on BLS pairing-friendly curves. Cryptology ePrint Archive, Report 2021/1130 (2021), https://ia.cr/2021/1130

48. Shallue, A., van de Woestijne, C.E.: Construction of rational points on elliptic curves over finite fields. In: Hess, F., Pauli, S., Pohst, M. (eds.) Algorithmic Number Theory Symposium– ANTS 2006. pp. 510–524. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
49. Vercauteren, F.: Optimal pairings. IEEE Transactions on Information Theory **56**(1), 455–461 (2009). https://doi.org/10.1109/TIT.2009.2034881
50. Washington, L.C.: Elliptic curves: number theory and cryptography. Chapman and Hall/CRC (2008)
51. Yang, K., Chen, L., Zhang, Z., Newton, C.J., Yang, B., Xi, L.: Direct anonymous attestation with optimal TPM signing efficiency. IEEE Transactions on Information Forensics and Security **16**, 2260–2275 (2021). https://doi.org/10.1109/TIFS.2021.3051801
52. Zhang, X., Lin, D.: Analysis of optimum pairing products at high security levels. In: Galbraith, S., Nandi, M. (eds.) Progress in Cryptology - INDOCRYPT 2012. pp. 412–430. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
53. Zhao, C.A., Zhang, F., Huang, J.: A note on the ate pairing. International Journal of Information Security **7**(6), 379–382 (Nov 2008). https://doi.org/10.1007/s10207-008-0054-1