Journal of Cryptographic Engineering (2024) 14:43–55
https://doi.org/10.1007/s13389-022-00304-y

REGULAR PAPER

The elliptic net algorithm revisited

Shiping Cai[1] · Zhi Hu[2] · Zheng-An Yao[1] · Chang-An Zhao[1,3]

Received: 27 April 2022 / Accepted: 15 October 2022 / Published online: 4 November 2022

Abstract
Efficient implementation of pairings is a fundamental ingredient in pairing-based cryptographic protocols. The Elliptic Net algorithm is an alternative method to Miller's algorithm for computing the (Optimal) Ate pairing in polynomial time. In this paper, we utilize several tricks to speed up the Elliptic Net algorithm. Firstly, we eliminate the inversion in the improved Elliptic Net algorithm, which allows for further improvements under certain circumstances. Second, we apply lazy reduction to the Elliptic Net algorithm for a better performance. Finally, we propose a new derivation of the formulas for computing the (Optimal) Ate pairing on the twisted curves. In addition, we provide implementations of all versions of the Elliptic Net algorithm on personal computers based on RELIC toolkit. Our implementations indicate that on this research line the Elliptic Net algorithm is about 80% faster than the previous fastest ones on the twisted 381-bit BLS12 curve and 71.5% faster on the twisted 676-bit KSS18 curve on 64-bit platforms, respectively.

Keywords Pairings · Elliptic net algorithm · Twists of elliptic curves · Denominator elimination · High security level

1 Introduction

Pairing-based cryptography, as a member of elliptic curve cryptography (ECC), has utilized pairings to several notable applications beyond traditional and advanced cryptographic tools containing identity-based encryption [1,2], aggregate signature [3,4], functional encryption [5] and zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) [6,7]. In addition, pairings can be used for public-key compression [8] and verifiable delay function construction [9] in isogeny-based cryptography. Most of these applications are built on the Tate pairing and its variants such as the Eta pairing [10], the Ate pairing [11,12], the R-ate pairing [13] and the Optimal Ate pairing [14].

Efficient algorithms for pairing computations play an essential role in implementations of relevant protocols. There are two polynomial time algorithms for computing the Tate pairing and its variants. One is Miller's algorithm [15] which was proposed in 1986. The other is Elliptic Net algorithm (ENA) which was proposed in 2007 by Stange [16]. Miller's algorithm has been heavily optimized since 2000 and is now in a relatively mature phase. Many tricks have been employed to Miller's algorithm for efficiency, such as twist maps and lazy reduction. Twist maps of elliptic curves allow us to transfer the operations of an extension field to its own proper subfield, which considerably reduces the number of multiplications. For a more in-depth description of twist maps, we refer to [11,17]. Lazy reduction was first introduced in quadratic extension field arithmetic for Miller's algorithm by Michael Scott [18] and further developed in [19]. It saves several modular reductions. Hence, it also brings significant improvements to Miller's algorithm. Stange [16] first defined elliptic nets and gave a relationship between elliptic nets and the Tate pairing [1]. Elliptic nets are generalized from elliptic divisibility sequences [20]. These sequences arise from any choice of an elliptic curve and rational points on such a curve. For more information about elliptic divisibility sequences, see [21]. The method called Double-and-Add for updating each value of the block in the ENA was proposed by Shipsey [22]. One can compute pairings using elliptic nets of rank 2. The explicit formulas for computing some variants of the Tate pairing using the ENA were given in [23,24]. In 2015, an improved version of the ENA (IENA) was proposed by

✉ Chang-An Zhao
zhaochan3@mail.sysu.edu.cn

[1] School of Mathematics, Sun Yat-sen University, Guangzhou 510275, Guangdong, People's Republic of China

[2] School of Mathematics and Statistics, Central South University, Changsha 410083, Hunan, People's Republic of China

[3] Guangdong Key Laboratory of Information Security, Guangzhou 510275, Guangdong, People's Republic of China

Chen *et al.* [25]. They reduced the dimension of the block for pairing computations at the price of one inversion at the DoubleAdd step. Hence, the IENA can perform well if the parameter of the Miller loop has low Hamming weight. It should be noted that most popular pairing-friendly curves meet this condition. Due to the properties of the structure of elliptic nets, it is possible to design some parallel strategies for the ENA to compute pairings. Moreover, we can update all values of a block in each iteration of the ENA simultaneously if we change the dimension of the first vector of a block from eight to ten. More detailed information can be found in [26]. It is known that the (I)ENA as an alternative method to pairing computations is still more costly than Miller's algorithm. Till now, there is a relative lack of research on the implementation of the (I)ENA.

Our Contributions. In this work, we aim to strengthen the (I)ENA as an easy-to-implement and efficient algorithm and to reduce the gap between the (I)ENA and Miller's algorithm. Previous works on the (I)ENA preferred to implement this algorithm in Magma. Therefore, our work first implements the (I)ENA based on RELIC toolkit [27] in a combination of C and assembly language. Note that the base field arithmetic is implemented in the assembly language in the RELIC library. Our specific contributions are as follows.

- We eliminate the inversion completely in the IENA. For the IENA, an inversion is always involved at the DoubleAdd step in the Double-and-Add algorithm. In this paper, we find that an inversion can be eliminated at the price of several multiplications in the IENA. The implementation indicates that the IENA works well when utilizing this trick.
- We use the (I)ENA to compute the Optimal Ate pairing entirely on the twisted curve inspired by the previous work of Costello *et al.* [28], who explored the pairing computation entirely on the twisted curve via the process of Miller's algorithm. In Miller's algorithm, the pairing computation contains addition and doubling steps of the Miller loop. Since the procedure of the (I)ENA relies on the updating of the block and does not involve the evaluation of the line and vertical line at either addition or doubling step of the Miller loop, the use of the (I)ENA to compute the pairing on the twisted curve still requires a general proof. In this work, we present a new proof based on the theory of divisor to verify the relationship between the (Optimal) Ate pairing on an elliptic curve and its twisted curve. This proof relies only on the definition of the Tate pairing and the theory of divisors. Furthermore, we derive the explicit formulas of the line function of the Optimal Ate pairing on the twisted curve. In our implementations, we boost the performance of the (I)ENA on a 381-bit BLS12 curve at the 128-bit security level and a 676-bit KSS18 curve at the 192-bit security level by using twist maps, respectively [29].
- We adopt lazy reduction [30] which only performs one reduction for the sum of several multiplications to the (I)ENA. In these algorithms, we observe that there are many terms of the form $A \cdot B - C \cdot D$, where A, B, C, D belong to a finite field. This inspires us to apply lazy reduction for the (I)ENA. In our implementation, lazy reduction reduces by around 27% the number of modular reductions.

We conclude that pairings can be efficiently computed with the ENA. Our optimized implementation of the ENA with an execution time of $2.16\,ms$ runs up to 4.9 times faster than the previous one on the 381-bit BLS12 curve on x64 platforms. Notice that Miller's algorithm performs well in our implementation which takes about $1.57\,ms$ on such a curve. Even though the ENA is still slower than Miller's algorithm, the ratio between the cost of ENA and Miller's algorithm is reduced from over 9 times to less than 2 times after the development of this work.

The rest of this paper is organized as follows. Section 2 gives an overview of pairings, twists of elliptic curves and the (I)ENA. In Sect. 3, we replace an inversion by several multiplications in the IENA. Section 4 analyzes the (Optimal) Ate pairing entirely on the twisted curve that is computed by the (I)ENA. In Sect. 5, we apply lazy reduction to the (I)ENA. The implementation and efficiency analysis are discussed in Sect. 6. Section 7 concludes the paper.

2 Preliminaries

In this section, we will give the definition of the Tate pairing and the (Optimal) Ate pairing. A brief description of twists of elliptic curves and the (I)ENA will also be provided.

2.1 Pairings

Let $\mathbb{F}_q$ be a finite field with the characteristic not equal to 2 or 3. Let $E : y^2 = x^3 + Ax + B$ be a short Weierstrass curve over $\mathbb{F}_q$, where $A, B \in \mathbb{F}_q$ and $4A^3 + 27B^2 \neq 0$. We denote the q-power Frobenius endomorphism on E by π_q. The order of $E(\mathbb{F}_q)$ is given by $\#E(\mathbb{F}_q) = q + 1 - t$, where t is the Frobenius trace of π_q. Choose a large prime r with $r \mid \#E(\mathbb{F}_q)$. Let $k \in \mathbb{Z}$ be the embedding degree with respect to r, i.e., the minimal positive integer satisfying $r \mid q^k - 1$.

Choose $P \in E(\mathbb{F}_q)[r]$ and $Q \in E(\mathbb{F}_{q^k})[r]$. We denote by μ_r the group of the rth roots of unity in $\mathbb{F}_{q^k}$. For an integer i and a point S on E, let $f_{i,S}$ be a rational function such that

$$\mathrm{Div}(f_{i,S}) = i(S) - (iS) - (i-1)(\infty).$$

We also call the function $f_{i,S}$ as the Miller function. Then the reduced Tate pairing [31] is defined as

$$\begin{aligned} &Tate : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k}) \to \mu_r \\ &(P, Q) \mapsto Tate(P, Q) = f_{r,P}(Q)^{q^k-1/r}. \end{aligned}$$

Furthermore, if we choose P and Q in specific subgroups of $E[r]$, the pairing computation can be sped up. Define

$$\begin{aligned} &\mathbb{G}_1 \triangleq E[r] \bigcap Ker(\pi_q - [1]), \\ &\mathbb{G}_2 \triangleq E[r] \bigcap Ker(\pi_q - [q]). \end{aligned}$$

Choose $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, respectively. Let $T = t - 1$. We can define a pairing as follows.

$$\begin{aligned} &Ate_E : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r \\ &(Q, P) \mapsto Ate_E(Q, P) = f_{T,Q}(P)^{q^k-1/r}, \end{aligned}$$

which is called the Ate pairing [11].

The Ate pairing is the case of the Eta pairing [10] on ordinary elliptic curves. The length of the Miller loop of the Ate pairing is shorter than that of the Tate pairing [12,13]. The Optimal Ate pairing allows us to obtain the shortest loop length [14,32,33]. Let $l_{S,T}$ be the line through two points S and T. Let v_T be the vertical line passing through point T. The definition of the Optimal Ate pairing is given in the following theorem.

Theorem 1 *[14, Theorem 4] Let $\lambda = \alpha r = \sum_{i=0}^{\varphi(k)} c_i q^i$ with $r \nmid \alpha$, where $\varphi(k)$ is the Euler function of k, then we can define a bilinear map*

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

where $s_i = \sum_{j=i}^{\varphi(k)} c_j q^j$.
If $\alpha k q^{k-1} \neq \left(\left(q^k - 1\right)/r\right) \sum_{i=0}^{\varphi(k)-1} i c_i q^{i-1} \pmod r$, then Opt_E is non-degenerate. We call Opt_E as the Optimal Ate pairing.

The implementation of the Tate pairing and its variants contains the Miller loop step and the final exponentiation step. At the Miller loop step, we first compute the value of the Miller function. We then raise this value to the power of $(q^k - 1)/r$ at the final exponentiation step. For the computation of the Optimal Ate pairing, one should consider computing the value of line functions at the Miller loop step, which depends on the family of pairing-friendly curves. In this work, we mainly consider the implementation of the Optimal Ate pairing on the BLS12 and KSS18 curves. More specific information will be discussed in Sect. 6.

2.2 Twists of elliptic curves

In this subsection, we first recall the definition of twists of elliptic curves.

Definition 1 Let E be an elliptic curve over $\mathbb{F}_q$. An elliptic curve $E'/\mathbb{F}_{q^{k/d}}$ is a twist of degree d of E if there exists an isomorphism $\Psi_d : E' \to E$ defined over $\mathbb{F}_{q^k}$ and d is minimal.

The potential degree d of twists is 2, 3, 4 or 6 [1,17]. For the BLS12 and KSS18 curves, the parameter $A = 0$ and the degree $d = 6$. In this case, the equation of the curve E is $y^2 = x^3 + B$, and the curve E' is the sextic twist of E. The M-type and D-type twists are given below [34].

$$\begin{aligned} &M - type : \ E' : y^2 = x^3 + B\xi \\ &\quad \Psi_6 : E' \to E \\ &\quad (x, y) \mapsto \left(\xi^{1/3}x, \xi^{1/2}y\right), \\ &D - type : \ E' : y^2 = x^3 + B/\xi \\ &\quad \Psi_6 : E' \to E \\ &\quad (x, y) \mapsto \left(\xi^{-1/3}x, \xi^{-1/2}y\right). \end{aligned} \tag{1}$$

Furthermore, we have the following theorem for the Tate pairing:

Theorem 2 *[35, Chapter IX, Theorem 9] Let $E_1/\mathbb{F}_q$ be an elliptic curve. Let r_0 be a prime such that $r_0 \mid \#E_1(\mathbb{F}_q)$. Suppose that the embedding degree with respect to q and r_0 is k. Let $\phi : E_1 \to E_2$ be an isogeny, where E_2 is an elliptic curve over $\mathbb{F}_{q^k}$. Choose $P \in E_1(\mathbb{F}_q)[r_0]$ and $Q \in E_2(\mathbb{F}_{q^k})$. We have $e(\phi(P), Q) = e(P, \hat{\phi}(Q))$, where $\hat{\phi}$ is the dual of ϕ.*

Note that Ψ_d is an isogeny of degree 1. If we denote the dual of Ψ_d by $\hat{\Psi}_d$, then $\hat{\Psi}_d \circ \Psi_d = [1]$. Recall the definition of E in Sect. 2.1. Choose $P \in E(\mathbb{F}_q)[r]$ and $Q' \in E'(\mathbb{F}_{q^{k/d}})$. We can compute pairings $(Opt)Ate_{E'}(\hat{\Psi}_d(P), Q')$ on the twisted curve E' whose Miller loop length is the same as the loop length of $(Opt)Ate_E(P, \Psi_d(Q'))$ on the original curve E.
Furthermore, define

$$\Phi_d = \Psi_d^{-1} \circ \pi_q \circ \Psi_d.$$

One can verify that $\Phi_d : E' \to E'$ is a homomorphism defined over $\mathbb{F}_{q^k}$ [36,37], which can help us get some useful conclusions in Sect. 4.

2.3 The elliptic net algorithm

An elliptic net is a map W from a finitely generated free Abelian group G to an integral domain R, satisfying a certain recurrence relation as follows.

$$\begin{aligned} &W(\alpha+\beta+\delta)W(\alpha-\beta)W(\gamma+\delta)W(\gamma) \\ &\quad + W(\beta+\gamma+\delta)W(\beta-\gamma)W(\alpha+\delta)W(\alpha) \\ &\quad + W(\gamma+\alpha+\delta)W(\gamma-\alpha)W(\beta+\delta)W(\beta) \\ &= 0, \end{aligned} \tag{2}$$

where $\alpha,\ \beta,\ \gamma,\ \delta \in G$.

For each $n \in \mathbb{Z}^+$, division polynomials $\psi_n \in \mathbb{Z}[A, B, x, y]$ are defined as follows [17].

$$\begin{aligned} &\psi_0 = 0,\ \psi_1 = 1,\ \psi_2 = 2y, \\ &\psi_3 = 3x^4 + 6Ax^2 + 12Bx - A^2, \\ &\psi_4 = 4y(x^6 + 5Ax^4 + 20Bx^3 - 5A^2x^2 - 4ABx \\ &\qquad - 8B^2 - A^3), \\ &\psi_{2n+1}\psi_1 = \psi_{n+2}\psi_n^3 - \psi_{n-1}\psi_{n+1}^3 \ (n \geq 2), \\ &\psi_{2n}\psi_2 = \psi_n(\psi_{n+2}\psi_{n-1}^2 - \psi_{n-2}\psi_{n+1}^2) \ (n \geq 3). \end{aligned}$$

Division polynomials are examples of elliptic nets of rank 1, i.e., $W(i) = \psi_i,\ \forall i \in \mathbb{Z}$. They can be used to compute scalar multiplication. Elliptic nets of rank 2 are applied for pairing computations. The relationship between the Tate pairing and an elliptic net is given below.

Theorem 3 *[16, Theorem 6] Choose $P \in E(\mathbb{F}_q)[r]$ and $Q \in E(\mathbb{F}_{q^k})[r]$. Let ∞ be the point in infinity. We denote the elliptic net associated with E, P, Q by $W_{P,Q}$; then, we have*

$$f_{r,P}(D_Q) = \frac{W_{P,Q}(r+1,1)W_{P,Q}(1,0)}{W_{P,Q}(r+1,0)W_{P,Q}(1,1)},$$

where $D_Q = (Q) - (\infty)$.

One can compute the Tate pairing in polynomial time by using the ENA. For simplicity, we abbreviate $W_{P,Q}(n,s)$ to $W(n,s)$. Assume that $W(1,0) = W(0,1) = 1$. In [16], Stange defined a block that consists of a first vector of eight consecutive terms centered on the term $W(i,0)$ and a second vector of three consecutive terms centered on $W(i,1)$, where $i \in \mathbb{Z}$. For the first vector, all of $W(n,0)$ terms can be updated by two formulas in Eqs. (3)-(4). By updating the first vector in the iteration, we can compute the scalar multiplication. For the second vector, we update the $W(n,1)$ terms by Eqs. (5)-(8). The updating process of a block V centered on i is shown in Figure 1.

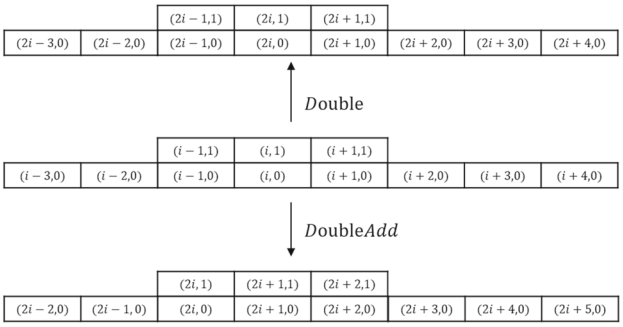

Fig. 1 Updating process of a block centered on i

$$W(2i-1,0) = W(i+1,0)W(i-1,0)^3 - W(i-2,0)W(i,0)^3, \tag{3}$$

$$W(2i,0) = (W(i,0)W(i+2,0)W(i-1,0)^2 - W(i,0)W(i-2,0)W(i+1,0)^2)/W(2,0). \tag{4}$$

$$W(2i-1,1) = (W(i+1,1)W(i-1,1)W(i-1,0)^2 - W(i,0)W(i-2,0)W(i,1)^2)/W(1,1), \tag{5}$$

$$W(2i,1) = (W(i-1,1)W(i+1,1)W(i,0)^2 - W(i-1,0)W(i+1,0)W(i,1)^2), \tag{6}$$

$$W(2i+1,1) = (W(i-1,1)W(i+1,1)W(i+1,0)^2 - W(i,0)W(i+2,0)W(i,1)^2)/W(-1,1), \tag{7}$$

$$W(2i+2,1) = (W(i+1,0)W(i+3,0)W(i,1)^2 - W(i-1,1)W(i+1,1)W(i+2,0)^2)/W(2,-1). \tag{8}$$

In some certain conditions, the value $W(2,0)$ can be fixed to 1 by the equivalence of elliptic nets [38].

Algorithm 1 The improved Elliptic Net algorithm [25]

Require: Initial terms $a = W(2,0), b = W(3,0), c = W(4,0), d = W(2,1), e = W(-1,1), f = W(2,-1), g = W(1,1)$ of an elliptic net which satisfies $W(1,0) = W(0,1) = 1$ and $n = (d_l d_{l-1} \ldots d_0)_2 \in \mathbb{Z}$ with $d_l = 1$ and $d_i \in \{0,1\}$ for $0 \leq i \leq l-2$
Ensure: $W(n,0), W(n,1)$
1: $V \leftarrow [[-a,-1,0,1,a,b,c],[1,g,d]]$
2: **for** $i = l-1$ downto 0 **do**
3: **if** $d_i == 0$ **then**
4: $V \leftarrow Double(V)$
5: **else**
6: $V \leftarrow DoubleAdd(V)$
7: **end if**
8: **end for**
9: **return** $V[0,3], V[1,1]$

Generally, updating a block centered on i to a block centered on $2i$ is called the Double step, and updating a block centered on i to a block centered on $2i+1$ is called the DoubleAdd step, which is represented by $Double(V)$ and $DoubleAdd(V)$, respectively. In [25], the authors improved the ENA by reducing the dimension of the first vector in an elliptic net from eight to seven. But we need to update the last term of the first vector at the DoubleAdd step by the following formula:

$$W(2i+4,0) = (W(2i+3,0)W(2i+1,0)W(2,0)^2 - W(3,0)W(1,0)W(2i+2,0)^2)/W(2i,0). \tag{9}$$

We show the IENA in Algorithm 1. The algorithm to compute the process of Lines 2-8 in Algorithm 1 is called the Double-and-Add algorithm. In fact, the overall process of the ENA is similar to that of the IENA. The main difference between the ENA and IENA is the saved term in the first vector, which affects the initial values of the block and the Double-and-Add algorithm.

3 Elimination of the inversion

In the IENA, an inversion is always involved at the DoubleAdd step. In this section, we will exploit the equivalence of the elliptic nets and perform local processing on the IENA to avoid the inversion in exchange for few multiplications.

When a block centered on i is updated to a block centered on $2i+1$, we need to compute the inverse of $W(2i,0)$ for updating the value of $W(2i+4,0)$ from Equation (9). To eliminate this inversion, we multiply $W(\lambda,0)_{2i-3\le\lambda\le 2i+4}$ by $W(2i,0)$ simultaneously at the DoubleAdd step. We have the following theorem to support this approach.

Theorem 4 *Given a block V centered on i, i.e.,*

$W(\lambda,0)_{i-3\le\lambda\le i+3}$ *and* $W(\lambda,1)_{i-1\le\lambda\le i+1}\in\mathbb{F}_{q^k}$.

1. *If $W(\lambda,0)_{i-3\le\lambda\le i+3}$ are multiplied by $\alpha\in\mathbb{F}_{q^k}^*$, i.e.,*

$$\hat{W}(\lambda,0)_{i-3\le\lambda\le i+3} = \alpha\cdot W(\lambda,0)_{i-3\le\lambda\le i+3},$$

then at the DoubleAdd step, we will obtain the block centered on $2i+1$, which is updated as follows.

$$\hat{W}(\lambda,0)_{2i-2\le\lambda\le 2i+4} = \alpha^4\cdot W(\lambda,0)_{2i-2\le\lambda\le 2i+4},$$
$$\hat{W}(\lambda,1)_{2i\le\lambda\le 2i+2} = \alpha^2\cdot W(\lambda,1)_{2i\le\lambda\le 2i+2}.$$

Furthermore, if $\alpha\neq 0$ is in a proper subfield of $\mathbb{F}_{q^k}$, then

$$\left(\frac{\hat{W}_{P,Q}(s,1)}{\hat{W}_{P,Q}(s,0)}\right)^{\frac{q^k-1}{r}} = \left(\frac{W_{P,Q}(s,1)}{W_{P,Q}(s,0)}\right)^{\frac{q^k-1}{r}}, \tag{10}$$

where s is an integer.

2. *If $W(\lambda,0)_{i-3\le\lambda\le i+3}$ and $W(\lambda,1)_{i-1\le\lambda\le i+1}$ are multiplied by $\alpha\in\mathbb{F}_{q^k}^*$, then at the DoubleAdd step, all the values of the block will be multiplied by α^4, and Equation (10) still holds.*

Proof Recall the recursive formula for $W(2i-1,0)$ and $W(2i,0)$ in Equations (3)-(4). We multiply $W(\lambda,0)_{i-3\le\lambda\le i+3}$ by α; then, we have

$$\begin{aligned}\hat{W}(2i-1,0) &= \alpha^4(W(i+1,0)W(i-1,0)^3 - W(i-2,0)W(i,0)^3)\\ &= \alpha^4\cdot W(2i-1,0).\\ \hat{W}(2i,0) &= \alpha^4\cdot W(2i,0).\end{aligned} \tag{11}$$

Therefore,

$$\hat{W}(\lambda,0)_{2i-2\le\lambda\le 2i+3} = \alpha^4\cdot W(\lambda,0)_{2i-2\le\lambda\le 2i+3}.$$

For the term $W(2i+4,0)$,

$$\begin{aligned}\hat{W}(2i+4,0) &= \alpha^8(W(2i+3,0)W(2i+1,0)W(2,0)^2)\\ &\quad - \alpha^8(W(3,0)W(1,0)W(2i+2,0)^2)/\alpha^4 W(2i,0)\\ &= \alpha^4 W(2i+4,0).\end{aligned}$$

This finishes the proof for the first assertion.

Now we consider the second vector in the block. Note that there are only two values of the first vector involved for computing each $W(\lambda,1)_{2i\le\lambda\le 2i+2}$. The new updated $\hat{W}(\lambda,1)_{2i\le\lambda\le 2i}$ will be multiplied by α^2.

Next, we will verify Equation (10). For any integer s,

$$\hat{W}_{P,Q}(s,0) = \alpha^{2l}\cdot W_{P,Q}(s,0),$$
$$\hat{W}_{P,Q}(s,1) = \alpha^{l}\cdot W_{P,Q}(s,1),$$

where $l\in\mathbb{Z}$. If the constant α is chosen to be in a proper subfield of $\mathbb{F}_{q^k}$, then $\alpha^{\frac{q^k-1}{r}}$ is equal to 1. This verifies Equation (10).

If $W(\lambda,0)_{i-3\le\lambda\le i+3}$ and $W(\lambda,1)_{i-1\le\lambda\le i+1}$ are multiplied by α simultaneously, then from Case 1, we have

$$\hat{W}(\lambda,0)_{2i-2\le\lambda\le 2i+4} = \alpha^4 W(\lambda,0)_{2i-2\le\lambda\le 2i+4}.$$

As for the second vector of the block, since each $W(\lambda,1)_{i-1\le\lambda\le i+1}$ is multiplied by α, $W(\lambda,1)_{2i\le\lambda\le 2i+2}$ will be

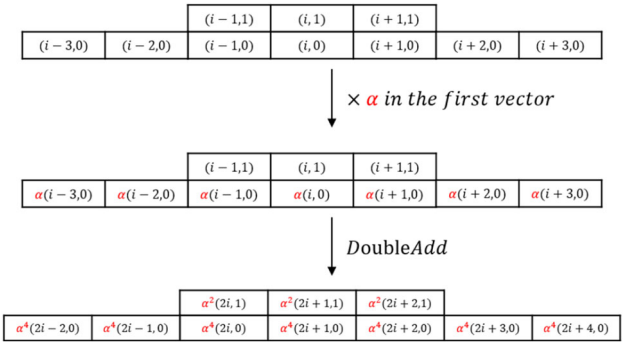

Fig. 2 Updating a block centered on i at the DoubleAdd step

multiplied by α^4 according to Equations (6)-(8). It is clearly seen that Equation (10) still holds, because

$$\frac{\hat{W}_{P,Q}(s,1)}{\hat{W}_{P,Q}(s,0)} = \frac{\alpha^l W_{P,Q}(s,1)}{\alpha^l W_{P,Q}(s,0)} = \frac{W_{P,Q}(s,1)}{W_{P,Q}(s,0)},$$

for some integer l.

So far the proof has been finished. □

Remark 1 The process of Case 1 in Theorem 4 is shown in Figure 2. Theorem 4 considers the situation at the DoubleAdd step. Indeed, we have a similar result at the Double step. It can be proved in the same manner. Theorem 4 can also be extended for any pairing-friendly curves. However, if we need to guarantee that Equation (10) holds for any $\alpha \in \mathbb{F}_{q^k}^*$, we should multiply every term in the block by α.

For some popular pairing-friendly curves, we will have a friendly situation. Taking the BLS12 curve, we choose in this work as an example; there is a proposition which is available for the (I)ENA. The related parameters of the BLS12 curve can be seen in Sect. 6.1 and the towering scheme of the extension field is shown as follows.

- $\mathbb{F}_{q^2} = \mathbb{F}_q[u]/\langle u^2 - \beta\rangle$, where $\beta = -1$;
- $\mathbb{F}_{q^6} = \mathbb{F}_{q^2}[v]/\langle v^3 - \xi\rangle$, where $\xi = u + 1$;
- $\mathbb{F}_{q^{12}} = \mathbb{F}_{q^6}[\omega]/\langle \omega^2 - v\rangle$.

Recall the definition of Ψ_6 in Sect. 2.2. The corresponding M-type twist Ψ_6 is

$$\begin{aligned} &\Psi_6 : E' \to E \\ &(x,y) \mapsto \left(\xi^{1/3}x, \xi^{1/2}y\right). \end{aligned} \tag{12}$$

From the towering scheme of $\mathbb{F}_{q^{12}}$, we know $\omega^2 = v$ and $v^2 = \xi$. Hence, we have $\xi^{1/3} = v$, $\xi^{1/2} = v\omega$.

Proposition 5 *Choose $P \in E(\mathbb{F}_q)$ and $Q' = (x_{Q'}, y_{Q'}) \in E'(\mathbb{F}_{q^2})$. Define $Q \triangleq \Psi_6(Q')$.*

Write $W_{Q,P}(s,0) = a_0 + a_1\boldsymbol{\omega}$, where $s \in \mathbb{Z}$, $a_0, a_1 \in \mathbb{F}_{q^6}$. Then we have

$$W_{Q,P}(s,0) = \begin{cases} a_0, & s \text{ is odd} \\ a_1\boldsymbol{\omega}, & s \text{ is even} \end{cases}.$$

Proof We abbreviate $W_{Q,P}(s,0)$ to $W_Q(s,0)$ for convenience. Note that $W(s,0) = \psi_s \in \mathbb{Z}[x, y, A, B]$, where ψ_s is a division polynomial. Therefore, we just prove the proposition in two situations according to [39, Section 3.2].

1. Assume that s is odd. Then ψ_s is a polynomial in $\mathbb{Z}[x, y^2, A, B]$. For the short Weierstrass elliptic curve $y^2 = x^3 + Ax + B$, we can replace y^2 by a polynomial in x. Hence, if we evaluate ψ_s at the point Q, then the x-coordinate of Q is $x_{Q'}\boldsymbol{v} \in \mathbb{F}_{q^6}$. Hence, $W_Q(s,0)$ will be always in a proper subfield of $\mathbb{F}_{q^{12}}$, i.e., $W_Q(s,0) = a_0$.
2. If s is even, then ψ_s is a polynomial in $2y\mathbb{Z}[x, y^2, A, B]$. Evaluating ψ_s at Q, we find that the y-coordinate of Q is $y_{Q'}\boldsymbol{v\omega} \in \mathbb{F}_{q^{12}}$. According to Case 1, the evaluation of a polynomial in $\mathbb{Z}[x, y^2, A, B]$ at Q will be in $\mathbb{F}_{q^6}$. Combined with the y-coordinate of Q, we have a_0 is equal to 0. Therefore, $W_Q(s,0) = a_1\boldsymbol{\omega}$.

So far the proposition has been proved. □

In order to eliminate the inverse of $W(2i,0)$ in Equation (9), we multiply $W(\lambda,0)_{2i-2\le\lambda\le 2i+4}$ by $W(2i,0)$ at the DoubleAdd step. It follows from Proposition 5 that the term $W(2i,0)$ can be written as $a_1\boldsymbol{\omega}$ on the BLS12 curve, where $a_1 \in \mathbb{F}_{q^6}$. Recall the towering scheme of $\mathbb{F}_{q^{12}}$. We have $\omega^2 = v$ and $v \in \mathbb{F}_{q^6}$. Then both $W(2i,0)^2 = a_1^2\boldsymbol{\omega^2} = a_1^2\boldsymbol{v}$ and $W(2i,0)^4 = a_1^4\boldsymbol{v}^2$ will always be in $\mathbb{F}_{q^6}$. Hence, $W(2i,0)^2$ and $W(2i,0)^4$ are equal to 1 if we raise them to the power of $\frac{q^k-1}{r}$. As long as we do not meet the inversion in the last iteration, Equation (10) will always be true. Then we can use the new block to compute pairings by using the IENA. Fortunately, the last iteration of the Miller loop on the BLS12 and KSS18 curves always invoke the Double step. This means that we can use 5 multiplications instead of 1 inversion.

4 The elliptic net algorithm on the twisted curve

The application of the twists maps has brought significant improvements in Miller's algorithm. When we use the (I)ENA to compute the (Optimal) Ate pairing, the algorithm will also have a good improvement if all the related parameters are on the twisted curve. In 2010, Costello *et al.* [28] proposed the Ate pairing entirely on the twisted curve. The authors considered the line and vertical line on twisted curves

at addition and doubling steps of the Miller loop and proved the correctness of this case via the procedure of Miller's algorithm. It seems to be natural that the (I)ENA should also be able to compute pairings entirely on the twisted curve. However, in the (I)ENA, we only need to update the values of the block and use these values to compute the value of the Tate pairing and its variants in the end. The proof of Costello *et al.* [28] is not suitable for the case of the (I)ENA. In the following, we will present a new proof that verifies the relationship between the (Optimal) Ate pairing on the elliptic curve and its corresponding twisted curve based on the divisor theory.

Recall the definition of the elliptic curve E in Sect. 2.1. Let $E'/\mathbb{F}_{q^e}$ be the twist of E of degree d with $e = k/d$. Let π'_{q^e} be the q^e-power Frobenius map on E'. There exists an isomorphism $\Psi_d : E' \to E$ over $\mathbb{F}_{q^k}$. Then we can define two subgroups

$$\mathbb{G}'_1 \triangleq E'[r] \bigcap Ker(\pi'_{q^e} - [1]),$$
$$\mathbb{G}'_2 \triangleq E'[r] \bigcap Ker(\pi'_{q^e} - [q^e]).$$

Actually, the parameter of the Miller loop length T can be set as $(t-1) \bmod r$ [12]. Firstly, we give a new derivation of the theorem about the Ate pairing entirely on the twisted curve as follows.

Theorem 6 *For $P' \triangleq \Psi_d^{-1}(P) \in \mathbb{G}'_2$ and $Q' \in \mathbb{G}'_1$, we can define a pairing on $\mathbb{G}'_1 \times \mathbb{G}'_2$ if $r^2 \nmid T^k - 1$:*

$$\begin{aligned} &Ate_{E'} : \mathbb{G}'_1 \times \mathbb{G}'_2 \to \mu_r \\ &(Q', P') \mapsto Ate_{E'}(Q', P') \\ &= \left(f_{T,Q'}\left(P'\right)\right)^{q^k-1/r}. \end{aligned}$$

Proof We only need to prove $f_{T,\Psi_d(Q')} = f_{T,Q'} \circ \Psi_d^{-1}$, for all $Q' \in \mathbb{G}'_1$. The divisor of $f_{T,\Psi_d(Q')}$ is

$$\mathrm{Div}(f_{T,\Psi_d(Q')}) = T(\Psi_d(Q')) - ([T]\Psi_d(Q')) - (T-1)(\infty).$$

Since Ψ_d is an isomorphism, we get

$$\begin{aligned} (\Psi_d)^*\mathrm{Div}(f_{T,\Psi_d(Q')}) &= T(Q') - ([T]Q') - (T-1)(\infty), \\ &= (f_{T,Q'}). \end{aligned}$$

Furthermore, we have

$$(\Psi_d)^*\mathrm{Div}(f_{T,\Psi_d(Q')}) = \mathrm{Div}(f_{T,\Psi_d(Q')} \circ \Psi_d).$$

Thus, we can deduce that $f_{T,\Psi_d(Q')} \circ \Psi_d = f_{T,Q'}$. Composing the formula with Ψ_d^{-1} on both sides, we get

$$f_{T,\Psi_d(Q')}(P) = \left(f_{T,Q'} \circ \Psi_d^{-1}\right)(P).$$

This completes the proof of the theorem. □

Remark 2 In Miller's algorithm, when we compute the Ate pairing on the original curve with twists, the field arithmetic is in the field where Q' is located. Since the cost of the transformations involved in each iteration is very small, the final value we require can be easily obtained by twist maps. In detail, we only need to multiply the value by a fixed value α in $\mathbb{F}_{q^k}$. This multiplication is sparse in general. But for the (I)ENA, if we adopt the same idea to use twist maps, the value will be multiplied by a different value of α in each iteration, which means that the transformation is not a friendly process. Therefore, we consider computing the Ate pairing on the twisted curve for the (I)ENA.

4.1 The optimal ate pairing on the twisted curve

For the situation of the Optimal Ate pairing entirely on the twisted curve, we can present the following theorem.

Theorem 7 *Let $\lambda = mr$ with $r \nmid m$ and $\lambda = \sum_{i=0}^{\varphi(k)} c_i q^i$. Define*

$$\Phi_{d,i} = \Psi_d^{-1} \circ [c_i q^i] \circ \Psi_d,$$

where $\Phi_{d,s_i} = \Psi_d^{-1} \circ [s_i] \circ \Psi_d$ and $s_i = \sum_{j=i}^{\varphi(k)} c_j q^j$. There exists a pairing on $\mathbb{G}'_1 \times \mathbb{G}'_2$:

$$\begin{aligned} &Opt_{E'} : \mathbb{G}'_2 \times \mathbb{G}'_1 \to \mu_r \\ &(Q', P') \mapsto Opt_{E'}(Q', P') \\ &= \left(\prod_{i=0}^{\varphi(k)} f_{c_i,Q'}^{q^i}(P') \cdot \prod_{i=0}^{\varphi(k)-1} \frac{l_{\Phi_{d,s_{i+1}},\Phi_{d,i}(Q')}}{v_{\Phi_{d,s_i}(Q')}}(P')\right)^{\frac{q^k-1}{r}}. \end{aligned}$$

Proof It follows from Theorem 6 that

$$\mathrm{Div}(\prod_{i=0}^{\varphi(k)} f_{c_i,Q'}^{q^i} \circ \Psi_d^{-1}) = \mathrm{Div}(\prod_{i=0}^{\varphi(k)} f_{c_i,\Psi_d(Q')}).$$

Let $Q_i \triangleq [s_{i+1}] \circ \Psi_d(Q')$. Consider the relation between $l_{\Phi_{d,s_{i+1}},\Phi_{d,i}(Q')}$ and $l_{Q_i,[c_i q^i]\Psi_d(Q')}$. Then we have the following divisor of the line function:

$$\begin{aligned} \mathrm{Div}(l_{Q_i,[c_i q^i]\Psi_d(Q')}) &= (Q_i) + ([c_i q^i]\Psi_d(Q')) \\ &\quad + (-Q_{i+1}) - 3(\infty). \end{aligned}$$

Since Ψ_d is an isomorphism,

$$\begin{aligned} (\Psi_d)^*\mathrm{Div}(l_{Q_i,[c_i q^i]\Psi_d(Q')}) &= (\Psi_d^{-1}(Q_i)) + (-Q_{i+1}) - 3(\infty) \\ &\quad + (\Psi_d^{-1} \circ [c_i q^i] \circ \Psi_d(Q')) \\ &= (l_{\Phi_{d,s_{i+1}},\Phi_{d,i}(Q')}). \end{aligned}$$

Therefore,

$$l_{Q_i,[c_i q^i]\Psi_d(Q')}(P) = l_{\Phi_{d,s_{i+1}},\Phi_{d,i}(Q')} \circ \Psi_d^{-1}(P).$$

Similarly,

$$v_{Q_i}(P) = v_{\Phi_{d,i}}(Q') \circ \Psi_d^{-1}(P).$$

This completes the proof of the theorem. □

Remark 3 We have $\pi_q \circ \Psi_d(Q') = [q]\Psi_d(Q')$, for each $\Psi_d(Q') \in \mathbb{G}_2$. Since π_q is an endomorphism and Ψ_d is an isomorphism over $\mathbb{F}_{q^k}$, we have

$$\pi_q \circ \Psi_d(Q') = \Psi_d \circ [q](Q').$$

Therefore,

$$\Psi_d^{-1} \circ \pi_q \circ \Psi_d(Q') = [q](Q'), i.e., \Phi_{d,1}(Q') = [q](Q').$$

It is known that a point in $\mathbb{G}_1'$ can be mapped to a point in $E[r]$. This also means that the points on the line function on the twisted curve can be obtained via Remark 3.

Note that the ratio between the inversion and multiplication costs over $\mathbb{F}_{q^k}$ decreases as the embedding degree k becomes larger. It follows that the cost of 1 inversion may be close to that of 5 multiplications. However, when we compute the (Optimal) Ate pairing entirely on the twisted curve, each term of the first vector in the block centered on i will be in $\mathbb{F}_{q^e}$. Hence, it is necessary for this situation to eliminate the inversion at the DoubleAdd step.

5 The elliptic net algorithm with lazy reduction

Lazy reduction can also be employed to speed up the (I)ENA. It can save the number of modular reductions during the calculation. The main idea of lazy reduction is to put the required modular reductions for the sum of several multiplications like $\sum a_i b_i$ over $\mathbb{F}_q$ to the end. Therefore, these multiplications only need 1 modular reduction over $\mathbb{F}_q$. In this paper, we use Montgomery reduction [40], so the cost of a modular reduction is equal to the cost of one multiplication without reduction.

When we use the (I)ENA to compute the (Optimal) Ate pairing, lots of multiplications of the form $A \cdot B \pm C \cdot D$ are contained, which needs 2 modular reductions normally. But lazy reduction allows us to use one modular reduction only. It should be noted that we are not concerned about violating the upper bound of Montgomery reduction for this situation since one only uses lazy reduction once each time with $A, B, C, D \in \mathbb{F}_q$. We mainly improve the term $W(3, 0)$ and $W(4, 0)$ at the initialization step. The number of modular reductions of three situations is given in Table 1.

Table 1 Number of Modular Reductions at the Initialization Step

Algorithm	$A, B \neq 0$	$B = 0$	$A = 0$
ENA [16]	10	8	6
This work	7	6	5

Table 2 Number of Modular Reductions at the Double-and-Add Step

Algorithm	$Double(V)$	$DoubleAdd(V)$
ENA [16]	42	42
IENA [25]	37	40
This work	27	30

The explicit updating formulas at the Double-and-Add step are mentioned in Sect. 2.3. The $Double(V)$ and $DoubleAdd(V)$ functions are combined with lazy reduction, and we adopt the new Double-and-Add algorithm in [25], requiring 10 terms in total. We present the optimized Double-and-Add algorithm based on the IENA in Appendix A. Assume that our terms in the block are all in the finite field $\mathbb{F}_q$. At Lines 7-17 we compute the $Double(V)$ function. We update 7 terms in the first vector and 3 terms in the second vector that are both centered on $2i$ in a block. In the ENA, we need 42 modular reductions in each iteration. The number of modular reductions decreases to 37 in the IENA. With the help of lazy reduction, the updating process for each term can save one modular reduction, so 10 terms will save 10 modular reductions in total. The $DoubleAdd(V)$ function is computed at Lines 9-33. These steps contain 40 modular reductions in the IENA originally, and the number of modular reductions is reduced to 30 in each iteration with lazy reduction. Table 2 shows the number of modular reductions among the ENA, IENA, and our optimized algorithm at the Double-and-Add step, respectively.

6 Implementation and analysis

In this section, we implement the optimization of the (I)ENA for the pairing computation. Note that we mainly consider the improvement of the computation at the Miller loop. Our implementations are performed on an Intel Core i7-8550U CPU processor operating at 1.80 GHz with hyperthreading turned off and TurboBoost disabled. We compile the benchmarks on GCC 7.4.0 with the -O2 flag set and test them on a 64-bit Linux platform, running Ubuntu 18.04 LTS. Our code is based on version 0.5.0 of the RELIC toolkit [27], and we adopt the finite field arithmetic implemented in the RELIC library which needs the GMP library for both curves. Hence,

our benchmarks use the assembly language to improve the performance of our implementations. Our code is available at https://github.com/wennycai/ENA.

We will use different methods to test the efficiency of computing the Optimal Ate pairing on the pairing-friendly curves at the 128-bit security level and 192-bit security level, respectively. Notice that the $DoubleAdd(V)$ function is not friendly in the IENA. In general, we choose the Miller loop parameter which has a low Hamming weight so that we can use $Double(V)$ function more frequently in the whole iterations to accelerate the IENA. The pairing-friendly curves we choose are the 381-bit BLS12 and 676-bit KSS18 curves. We also implement Miller's algorithm for benchmarks. The version of Miller's algorithm we use in our work is the fastest one implemented by Aranha *et al.* in the RELIC library. For fair comparison, we use the same algorithms for the finite field arithmetic and the final exponentiation in the RELIC library when we implement the ENA, IENA and Miller's algorithm. The recent work of [41] proposed some more efficient field arithmetic algorithms to speed up the computation of the Optimal Ate pairing on the 381-BLS12 curve which obtained a 1.37× speedup on an x64 Intel processor. Our implementations can also benefit from their improvement. The authors plan to open-source the optimized implementation of pairing computations on the 381-bit BLS12 curve that was integrated to the state-of-the-art pairing library RELIC 0.5.0. We specify some symbols here to show the amount of operations in this section:

- M_k: Multiplication over $\mathbb{F}_{q^k}$, S_k: Squaring over $\mathbb{F}_{q^k}$,
- M: Multiplication over $\mathbb{F}_q$, S: Squaring over $\mathbb{F}_q$,
- I_k: Inversion over $\mathbb{F}_{q^k}$, A: Addition over $\mathbb{F}_q$.

6.1 381-bit BLS12 curve

The concrete parameters for the 381-bit BLS12 curve with embedding degree $k = 12$ are given as follows.

- $z = -2^{63} - 2^{62} - 2^{60} - 2^{57} - 2^{48} - 2^{16}$;
- $r = z^4 - z^2 + 1$;
- $q = (z-1)^2(z^4 - z^2 + 1)/3 + z$;
- $E : y^2 = x^3 + 4$ over $\mathbb{F}_q$;
- $\mathbb{F}_{q^2} = \mathbb{F}_q[u]/\langle u^2 - \beta\rangle$, where $\beta = -1$;
- $\mathbb{F}_{q^6} = \mathbb{F}_{q^2}[v]/\langle v^3 - \xi\rangle$, where $\xi = u + 1$;
- $\mathbb{F}_{q^{12}} = \mathbb{F}_{q^6}[\omega]/\langle \omega^2 - v\rangle$;
- Twisted curve $E' : y^2 = x^3 + 4\xi$ over $\mathbb{F}_{q^2}$.

Recall that $P \in E(\mathbb{F}_q)$ and $Q' \in E'(\mathbb{F}_{q^e})$. Note that we do not need to compute the evaluation of the line function on the BLS12 curve. Hence, we only compute the following formula:

$$\left(f_{z,\Psi_6(Q')}(P)\right)^{\frac{q^{12}-1}{r}} \quad or \quad \left(f_{z,Q'}(\Psi_6^{-1}(P))\right)^{\frac{q^{12}-1}{r}}.$$

The amount of operations for $f_{z,\Psi_6(Q')}$ and $f_{z,Q'}$ in one iteration is $7S_{12} + \frac{67}{2}M_{12}$ and $6S_2 + 62M_2 + S_{12} + \frac{3}{2}M_{12}$ at the Double step in the ENA, respectively. In our implementation, we use the ratios $1I_{12} \approx 3M_{12}, 1I_2 \approx 13M_2, 1M_2 \approx 3M$ and $1M_{12} \approx 54M$ [19]. In the IENA, we need $6S_{12}+31M_{12}+I_{12}$ without twists at the DoubleAdd step. If we compute pairings on the twisted curve, the operations can be reduced to $1S_{12} + 1M_{12} + 5S_2 + 39M_2 + 1I_2$. Without considering the influence of delay error, it is necessary to eliminate the inversion if we compute the Optimal Ate pairing on the twisted curve. Because the cost of 1 inversion is greater than that of 5 multiplications in $\mathbb{F}_{q^2}$. Moreover, we choose to compute $f_{-z,Q'}$ and use the relationship

$$(f_{z,Q'})^{(q^{12}-1)/r} = (\frac{1}{f_{-z,Q'}})^{(q^{12}-1)/r}$$

to revise the value, since z is a negative number. In order to make the IENA work well, we expand $-z$ in the non-adjacent form (NAF) to reduce the proportion of nonzero digits. Although the ENA is much slower than Miller's algorithm, it still counts in milliseconds. We cycle the benchmarks 10, 000 times and take the average value to ensure the stability and accuracy of our results. The comparison about the efficiency of different methods is provided in Table 3.

Table 3 shows that this work speeds up the ENA indeed. Twist maps have a good performance for the (I)ENA. The application of twist maps improves the efficiency of the ENA and IENA by 80.8% and 79.8%, respectively. In addition, lazy reduction further accelerates the (I)ENA and brings around 9% efficiency improvement on the twisted BLS12 curve. It should be noted that on the BLS12 curves, the cost of 1 inversion over $\mathbb{F}_{q^{12}}$ is close to that of 5 multiplications over $\mathbb{F}_{q^{12}}$. Hence, we only need to avoid the inversion on the twisted BLS12 curve, which makes the IENA be up to 3.66% faster than before.

6.2 676-bit KSS18 curve

Now we give the parameters of the 676-bit KSS18 curve with embedding degree $k = 18$ below:

- $z = -2^{85} - 2^{31} - 2^{26} + 2^6$;
- $r = (z^6 + 37z^3 + 343)/343$;
- $q = (z^8 + 5z^7 + 7z^6 + 37z^5 + 188z^4 + 259z^3 + 343z^2 + 1763z + 2401)/21$;
- $E : y^2 = x^3 + 2$ over $\mathbb{F}_q$;
- $\mathbb{F}_{q^3} = \mathbb{F}_q[u]/\langle u^3 - \beta\rangle$, where $\beta = -2$;
- $\mathbb{F}_{q^6} = \mathbb{F}_{q^2}[v]/\langle v^2 - \xi\rangle$, where $\xi = u$;
- $\mathbb{F}_{q^{18}} = \mathbb{F}_{q^6}[\omega]/\langle \omega^3 - v\rangle$;

Table 3 Efficiency Comparison for pairing computations on a 381-bit BLS12 Curve

Method	Clock cycle ($\times 10^3$)	Time (ms)
ENA [16]	25,524	12.81
ENA with lazy reduction	24,599	12.35
IENA [25]	23,508	11.80
IENA with lazy reduction	22,586	11.34
IENA (Eliminate Inversion)	23,554	11.82
IENA (Eliminate Inversion) with lazy reduction	22,722	11.41
ENA (Twist)	4890	2.45
ENA (Twist) with lazy reduction	4463	2.24
IENA (Twist)	4749	2.38
IENA (Twist) with lazy reduction	4325	2.17
IENA (Twist & Eliminate Inv)	4575	2.30
IENA (Twist & Eliminate Inv) with lazy reduction	4315	2.16
Miller's algorithm	3123	1.57

Table 4 Efficiency Comparison for pairing computations on a 676-bit KSS18 Curve

Method	Clock cycle ($\times 10^3$)	Time (ms)
ENA [16]	136,542	68.54
ENA with lazy reduction	132,700	66.61
IENA [25]	122,629	61.56
IENA with lazy reduction	119,991	60.23
IENA (Eliminate Inversion)	122,681	61.59
IENA (Eliminate Inversion) with lazy reduction	120,686	60.58
ENA (Twist)	40,949	20.56
ENA (Twist) with lazy reduction	39,440	19.80
IENA (Twist)	40,676	20.42
IENA (Twist) with lazy reduction	39,276	19.72
IENA (Twist & Eliminate Inv)	40,291	20.23
IENA (Twist & Eliminate Inv) with lazy reduction	38,904	19.53
Miller's algorithm	17,149	8.61

- Twisted curve $E' : y^2 = x^3 + 2/\xi$ over $\mathbb{F}_{q^2}$.

We need to calculate

$$\left(f_{z,\Psi_6(Q')} \cdot f^q_{3,\Psi_6(Q')} \cdot l_{[z]\Psi_6(Q'),[3q]\Psi_6(Q')}(P)\right)^{\frac{q^{18}-1}{r}}$$

or

$$\left(f_{z,Q'} \cdot f^q_{3,Q'} \cdot l_{\Psi_6^{-1}([z]\Psi_6(Q')),\Psi_6^{-1}([3q]\Psi_6(Q'))}(\Psi_6^{-1}(P))\right)^{\frac{q^{18}-1}{r}}$$

for computing the Optimal Ate pairing on this curve. Notice that the Optimal Ate pairing on the 676-bit KSS18 curve can achieve the 192-bit security level. Therefore, we just cycle the benchmark 1, 000 times and take the average value as the final result. Table 4 shows the timings of different methods for computing the Optimal Ate pairing.

Similar to the results on the BLS12 curve, the computation of the Optimal Ate pairing using the (I)ENA on twisted KSS18 curve can be significantly sped up compared with the computation on the KSS18 curve, which is 70% and 66.83% faster, respectively. As for the performance of lazy reduction, we find that the application of this technique saves more operations on the KSS18 curve than that on the BLS12 curve. This is mainly due to the fact that the embedding degree of the KSS18 curve is larger than that of the BLS12 curve. Moreover, there are more iterations of the Miller loop on the KSS18 curve than on the BLS12 curve. Since the runtime of the ENA on KSS18 curve is $68.54ms$, the effect of $2ms$ saved by lazy reduction is not obvious here. Considering the IENA on the twisted KSS18 curve, the elimination of the inversion can save about 385, 000 clock cycles compared to the original one, and the lazy reduction can save about 1.4 million clock cycles.

According to our results, the efficiency of computing the Optimal Ate pairing on the twisted curve is much higher than that on the original curve for the (I)ENA. In addition, we can further improve the efficiency of the algorithm by eliminating the inversion in the IENA. However, the gap between the optimized IENA and Miller's algorithm on the KSS18 curve is larger than that on the BLS12 curve. This is related to the extension field arithmetic. For example, we require $6M + 3M_3 + 6S_3 + 1S_{18}$ and one sparse multiplication over $\mathbb{F}_{q^{18}}$ at the Double step in Miller's algorithm [19]. But we require $1S_{18} + 9M_{18} + 5S_3 + 22M_3$ at the Double step in this work.

7 Conclusions

In this work, we improved the Elliptic Net algorithm. For the original Elliptic Net algorithm, we analyzed its efficiency and presented implementations on the computation of the Optimal Ate pairing on a 381-bit BLS12 curve and a 676-bit KSS18 curve with several tricks, respectively. In particular, lazy reduction was able to reduce by around 27% of the required modular reductions. Moreover, the application of twist maps helped us reduce the number of multiplications and the improvement was significant. Besides, the improved Elliptic Net algorithm was also further developed by eliminating the inversion in exchange for few multiplications. On the 381-bit BLS12 curve, this work improved the performance of the Optimal Ate pairing by 80% compared with the original version on a 64-bit Linux platform. The implementation on the 676-bit KSS18 curve had shown that this work was 71.5% faster than the previous ones. Although the Elliptic Net algorithm was still slower than Miller's algorithm, it can compute pairings efficiently on personal computers.

Acknowledgements The work of Chang-An Zhao is partially supported by the Major Program of Guangdong Basic and Applied Research under Grant No. 2019B030302008 and NSFC under Grant No. 61972428. The work of Zhi Hu is supported by the National Natural Science Foundation of China (Grant No. 61972420, No. 61602526) and Hunan Provincial Natural Science Foundation of China (2020JJ3050, 2019JJ50827).

Appendix A. Algorithm

Algorithm 2 Double-and-Add Algorithm with Lazy Reduction (Eliminate Inversion)

Require: Block V centered on i in which the first vector has 7 terms and the second vector has 3 terms. $\alpha = W(2,0)^{-1}$, $\beta = W(-1,1)^{-1}$, $\gamma_1 = W(2,-1)^{-1}$, $\delta = W(1,1)^{-1}$, $\omega_{13} = W(1,0)W(3,0)$, $\omega_2 = W(2,0)^2$, $flag \in \{0, 1\}$.

Ensure: Block centered on $2i$ If $flag = 0$, centered on $2i + 1$ If $flag = 1$.

$S_0 \leftarrow V[2,2]^2 \ mod\ p$, $P_0 \leftarrow (V[2,1] * V[2,3]) \ mod\ p$;
for $i = 1$ to 5 **do**
  $S[i] \leftarrow V[1,i+1]^2 \ mod\ p$;
  $P[i] \leftarrow (V[1,i] * V[1,i+2]) \ mod\ p$;
end for
if $flag = 0$ **then**
  for $j = 1$ to 3 **do**
    $t_0 \leftarrow S[j] * P[j+1]$, $t_1 \leftarrow S[j+1] * P[j]$, $V[1, 2j-1] \leftarrow (t_0 - t_1) \ mod\ p$;
    $t_0 \leftarrow S[j] * P[j+2]$, $t_1 \leftarrow S[j+2] * P[j]$, $V[1, 2j] \leftarrow (t_0 - t_1) \ mod\ p$;
    $V[1, 2j] \leftarrow (V[1, 2j] * \alpha) \ mod\ p$;
  end for
  $t_0 \leftarrow S[4] * P[5]$, $t_1 \leftarrow S[5] * P[4]$, $V[1, 7] \leftarrow (t_0 - t_1) \ mod\ p$;
  $k_0 \leftarrow S[2] * P_0$, $k_1 \leftarrow P[2] * S_0$, $V[2, 1] \leftarrow (k_0 - k_1) \ mod\ p$;
  $V[2, 1] \leftarrow (V[2, 1] * \delta) \ mod\ p$;
  $k_0 \leftarrow S[3] * P_0$, $k_1 \leftarrow P[3] * S_0$, $V[2, 2] \leftarrow (k_0 - k_1) \ mod\ p$;
  $k_0 \leftarrow S[4] * P_0$, $k_1 \leftarrow P[4] * S_0$, $V[2, 3] \leftarrow (k_0 - k_1) \ mod\ p$;
  $V[2, 3] \leftarrow (V[2, 3] * \beta) \ mod\ p$;
else
  for $j = 1$ to 3 **do**
    $t_0 \leftarrow S[j] * P[j+2]$, $t_1 \leftarrow S[j+2] * P[j]$, $V[1, 2j-1] \leftarrow (t_0 - t_1) \ mod\ p$;
    $V[1, 2j-1] \leftarrow (V[1, 2j-1] * \alpha) \ mod\ p$;
    $t_0 \leftarrow S[j+1] * P[j+2]$, $t_1 \leftarrow S[j+2] * P[j+1]$, $V[1, 2j] \leftarrow (t_0 - t_1) \ mod\ p$;
  end for
  $vt_1 \leftarrow (V[1, 4] * V[1, 6]) \ mod\ p$, $vt_2 \leftarrow (V[1, 5]^2) \ mod\ p$;
  $t_0 \leftarrow vt_1 * \omega_2$, $t_1 \leftarrow vt_2 * \omega_{13}$, $V[1, 7] \leftarrow (t_0 - t_1) \ mod\ p$;
  for $j = 1$ to 6 **do**
    $V[1, j] = (V[1, j] * V[1, 3]) \ mod\ p$;
  end for
  $k_0 \leftarrow S[3] * P_0$, $k_1 \leftarrow P[3] * S_0$, $V[2, 1] \leftarrow (k_0 - k_1) \ mod\ p$;
  $k_0 \leftarrow S[4] * P_0$, $k_1 \leftarrow P[4] * S_0$, $V[2, 2] \leftarrow (k_0 - k_1) \ mod\ p$;
  $V[2, 2] \leftarrow (V[2, 2] * \beta) \ mod\ p$;
  $k_0 \leftarrow S[5] * P_0$, $k_1 \leftarrow P[5] * S_0$, $V[2, 3] \leftarrow (k_0 - k_1) \ mod\ p$;
  $V[2, 3] \leftarrow (V[2, 3] * \gamma_1) \ mod\ p$.
end if
return V

References

1. Mrabet, N.E., Joye, M.: Guide to pairing-based cryptography. cryptography and network security series. CRC Press, Boca Raton (2017)
2. Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter identity-based encryption via asymmetric pairings. Des. Codes Cryptogr. **73**(3), 911–947 (2014)
3. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) Advances in cryptology – EUROCRYPT 2003, pp. 416–432. Springer, Berlin (2003)
4. Boneh, D., Drijvers, M., Neven, G.: Compact multi-signatures for smaller blockchains. In: Peyrin, T., Galbraith, S. (eds.) Advances in cryptology - ASIACRYPT 2018, pp. 435–464. Springer, Cham (2018)
5. Agrawal, S., Goyal, R., Tomida, J.: Multi-input quadratic functional encryption from pairings. In: Malkin, T., Peikert, C. (eds.) Advances in cryptology - CRYPTO 2021, pp. 208–238. Springer, Cham (2021)
6. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, N., Ward, N.: Marlin: preprocessing zksnarks with universal and updatable srs. In: Canteaut, A., Ishai, Y. (eds.) Advances in cryptology - EUROCRYPT 2020, pp. 738–768. Springer, Cham (2020)
7. 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 (2016)
8. Naehrig, M., Renes, J.: Dual isogenies and their application to public-key compression for isogeny-based cryptography. In: Galbraith, S.D., Moriai, S. (eds.) Advances in cryptology - ASIACRYPT 2019, pp. 243–272. Springer, Cham (2019)
9. 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, Cham (2019)
10. Barreto, P.S., Galbraith, S.D., Héigeartaigh, C.O., Scott, M.: Efficient pairing computation on supersingular abelian varieties. Des. Codes Cryptogr. **42**(3), 239–271 (2007)
11. Hess, F., Smart, N.P., Vercauteren, F.: The eta pairing revisited. IEEE Trans. Inf. Theor. **52**(10), 4595–4602 (2006)
12. Matsuda, S., Kanayama, N., Hess, F., Okamoto, E.: Optimised versions of the ate and twisted ate pairings. In: Galbraith, S.D. (ed.) Cryptography and coding, pp. 302–312. Springer, Berlin (2007)
13. Lee, E., Lee, H.S., Park, C.M.: Efficient and generalized pairing computation on abelian varieties. IEEE Trans. Inf. Theor. **55**(4), 1793–1803 (2009)
14. Vercauteren, F.: Optimal pairings. IEEE Trans. Inf. Theor. **56**(1), 455–461 (2009)
15. Miller, V.S.: The weil pairing, and its efficient calculation. J. Cryptol. **17**(4), 235–261 (2004)
16. Stange, K.E.: The tate pairing via elliptic nets. In: Takagi, T., Okamoto, E., Okamoto, T., Okamoto, T. (eds.) Pairing-based cryptography - pairing 2007, pp. 329–348. Springer, Berlin (2007)
17. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer, New York (2009)
18. Scott, M., Costigan, N., Abdulwahab, W.: Implementing cryptographic pairings on smartcards. In: Goubin, L., Matsui, M. (eds.) Cryptographic hardware and embedded systems - CHES 2006, pp. 134–147. Springer, Berlin (2006)
19. 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 (2011)
20. Ward, M.: Memoir on elliptic divisibility sequences. Am. J. Math. **70**(1), 31 (1948)
21. Einsiedler, M., Everest, G., Ward, T.: Primes in elliptic divisibility sequences. LMS J. Comput. Math. **4**, 1–13 (2001)
22. Shipsey, R.: Elliptic divisibility sequences. PhD thesis, Goldsmiths, University of London UK, (2001)
23. Tang, C., Ni, D., Xu, M., Guo, B., Qi, Y.: Implementing optimized pairings with elliptic nets. Sci. China Inf. Sci. **57**(5), 1–10 (2014)
24. Ogura, N., Kanayama, N., Uchiyama, S., Okamoto, E.: Cryptographic pairings based on elliptic nets. In: Iwata, T., Nishigaki, M. (eds.) Advances in information and computer security, pp. 65–78. Springer, Berlin (2011)
25. Chen, B.L., Zhao, C.A.: An improvement of the elliptic net algorithm. IEEE Trans. Computers **65**(9), 2903–2909 (2015)
26. Onuki, H., Teruya, T., Kanayama, N., Uchiyama, S.: Faster explicit formulae for computing pairings via elliptic nets and their parallel computation. In: Ogawa, K., Yoshioka, K. (eds.) Advances in information and computer security, pp. 319–334. Springer, Cham (2016)
27. Aranha, D.F., Gouvêa, C.P.L., Markmann, T., Wahby, R.S., Liao, K.: RELIC is an efficient library for cryptography. https://github.com/relic-toolkit/relic
28. 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 (2010)
29. Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. J. Cryptol. **32**(1), 1–39 (2018)
30. Lim, C.H., Hwang, H.S.: Fast implementation of elliptic curve arithmetic in $GF\ (p^n)$. In: Imai, H., Zheng, Y. (eds.) Public key cryptography. Springer, Berlin (2000)
31. Granger, R., Hess, F., Oyono, R., Thériault, N., Vercauteren, F.: Ate pairing on hyperelliptic curves. In: Naor, M. (ed.) Advances in cryptology - EUROCRYPT 2007, pp. 430–447. Springer, Berlin (2007)
32. Zhao, C.A., Zhang, F.G., Huang, J.W.: All pairings are in a group. IEICE Trans. **91–A**(10), 3084–3087 (2008)
33. Zhao, C.A., Zhang, F.G., Huang, J.W.: A note on the ate pairing. Int. J. Inf. Security Arch. **7**(6), 379–382 (2008)
34. Azarderakhsh, R., Fishbein, D., Grewal, G., Hu, S., Jao, D., Longa, P., Verma, R.: Fast software implementations of bilinear pairings. IEEE Trans. Dependable Secure Comput. **14**(6), 605–619 (2017)
35. Blake, I.F., Seroussi, G., Smart, N.P.: Advances in elliptic curve cryptography, vol. 317. Cambridge University Press, Cambridge (2005)
36. Galbraith, S.D., Scott, M.: Exponentiation in pairing-friendly groups using homomorphisms. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing-Based cryptography - pairing 2008, pp. 211–224. Springer, Berlin (2008)
37. Galbraith, S.D., Lin, X., Scott, M.: Endomorphisms for faster elliptic curve cryptography on a large class of curves. J. Cryptol. **24**(3), 446–469 (2011)
38. Chen, B., Hu, C., Zhao, C.-A.: Note on scalar multiplication using division polynomials. IET Inf. Secur. **11**(4), 195–198 (2017)

39. Washington, C.L.: Elliptic curves: number theory and cryptography. CRC press, Boca Raton (2008)
40. Montgomery, P.L.: Modular multiplication without trial division. Math. Comput. **44**(170), 519–521 (1985)
41. Longa, P.: Efficient Algorithms for Large Prime Characteristic Fields and Their Application to Bilinear Pairings and Supersingular Isogeny-Based Protocols. Cryptology ePrint Archive, Paper 2022/367. https://eprint.iacr.org/2022/367 (2022). https://eprint.iacr.org/2022/367