Efficient Ternary Computation of Optimal Ate Pairing on BLS27 Curves

Walid Haddaji
Science and Technology for Defense Lab (LR19DN01)
Center for Military Research, Military Academy
Tunis, Tunisia
haddajiwalid95@gmail.com

Abstract

The computation of optimal Ate pairings on elliptic curves with embedding degree $k = 27$ (BLS27) is highly relevant for achieving the 256-bit security level, especially in the context of recent advances in the Number Field Sieve (NFS) and its variants (exTNFS, SexTNFS). Traditional binary approaches fail to fully exploit the degree-3 extension tower of $\mathbb{F}_{p^{27}}$. In this work, we propose an efficient ternary version of the Miller loop, restricting the seed representation to sparse ternary digits $\{0, 1\}$ to streamline point operations and eliminate costly inversions. Furthermore, we generate two new parameter seeds tailored for exTNFS and SexTNFS security levels. These seeds feature sparse ternary representations that simultaneously guarantee the efficiency of the Miller loop and allow the full exploitation of cyclotomic cubing in $\mathbb{F}_{p^{27}}$ during the hard part of the final exponentiation. Compared to the state of the art binary approach by Fouotsa et al. (2020), our exTNFS seed yields a 22% improvement in the overall optimal Ate pairing computation cost. Concurrently, our proposed SexTNFS seed ensures a higher level of security against the most advanced NFS variants.

Keywords: Optimal Ate pairing, BLS27 curves, Ternary Miller algorithm, Cyclotomic cubing, Final exponentiation, exTNFS, SexTNFS.

1 Introduction

Pairing-based cryptography has enabled the realization of numerous advanced cryptographic protocols, such as Identity-Based Encryption (IBE) [4], short signatures [5], and attribute-based encryption [11]. The efficiency of these protocols relies heavily on the fast computation of bilinear pairings, particularly the optimal Ate pairing, on pairing-friendly elliptic curves.
For a long time, curves with even embedding degrees (e.g., BN12, BLS12) were favored due to the availability of quadratic twists and efficient binary Miller algorithms. However, recent breakthroughs in the resolution of the Discrete Logarithm Problem (DLP) in finite fields using the Number Field Sieve (NFS) and its variants, such as the Extended Tower NFS (exTNFS) and Special exTNFS (SexTNFS), have necessitated a re-evaluation of parameter sizes. According to Barbulescu and Duquesne [2], at the 256-bit security level, the extension field $\mathbb{F}_{p^k}$ must be as large as 2^{18480} for exTNFS and 2^{27410} for SexTNFS. For $k = 27$, this implies base field primes p of roughly 684 and 1015 bits, respectively. BLS27 curves, with an embedding degree $k = 27$, offer a practical resistance against these NFS variants. Indeed, the larger embedding degree allows the extension field $\mathbb{F}_{p^k}$ to reach the massive sizes required for 256-bit security without requiring an impractically large base field prime p. Because $27 = 3^3$, the extension field $\mathbb{F}_{p^{27}}$ is naturally constructed as a tower of degree-3 extensions. In this tower, computing the cubing in the cyclotomic subgroup of $\mathbb{F}_{p^{27}}$ is exceptionally cheap, allowing us to replace the traditional square-and-multiply routine. Furthermore, the curve's j-invariant of 0 allows for a cubic twist,

meaning the point operations for the first argument of the optimal Ate pairing can be performed entirely in the smaller subfield.
Several works have investigated pairings on curves with odd embedding degrees and cubic twists. Zhang and Lin [17] provided an early analysis of optimum pairing products at high security levels, including $k = 27$. Fouotsa, El Mrabet, and Pecha [9] provided a detailed arithmetic and cost estimation of operations in the tower extensions for $k = 9, 15, 27$, improving the theoretical cost of the Miller loop and final exponentiation by applying the lattice-based method of Fuentes-Castañeda et al. [10]. However, their approach relied on the traditional binary Miller algorithm and binary exponentiation in the final exponentiation. More recently, Hayashida et al. [12] proposed a new framework for efficient final exponentiation via cyclotomic structure, directly factorizing the hard part using homogeneous cyclotomic polynomials. For odd embedding degrees divisible by 3, where cyclotomic squaring is not available, Nanjo et al. [14] derived an explicit formula for cyclotomic cubing. Ben Abdelghani and Haddaji [3] further optimized these arithmetic operations, providing precise costs for Frobenius maps and demonstrating the applicability of cyclotomic cubing to reduce final exponentiation costs for BLS27 by generating specific seeds. In this work, we build upon these advancements to present a fully ternary framework for the entire optimal Ate pairing computation. Despite the structural advantages of BLS27, the use of a binary Miller loop fails to fully exploit the base-3 structure of the extension tower. In this paper, we adapt the entire pairing computation to exploit this base-3 representation. Our contributions are as follows. We provide the mathematical proofs about the ternary version of Miller loop.

1. **Efficient ternary Miller loop:** we introduce a ternary Miller algorithm that operates strictly on seeds with ternary digits in $\{0, 1\}$. This removes the need for the digit 2, avoiding precomputation of doublings and extra field multiplications, while exploiting cheap cubing in $\mathbb{F}_{p^{27}}$.

2. **Optimal seeds for NFS Security:** we generate and utilize specific seeds:

   $$x_1 = 3^{23} + 3^5 + 3^3 + 1$$

   for exTNFS (256-bit) and

   $$x_2 = 3^{32} + 3^{31} + 3^{27} + 3^{22} + 3^{10} + 1$$

   for SexTNFS (256-bit), both fitting our $\{0, 1\}$ constraint.

3. **Cyclotomic Cubing in Final Exponentiation:** we adapt the hard part of the final exponentiation to use ternary exponentiation via cyclotomic cubing.

4. **Complexity analysis and comparison:** we give a detailed complexity analysis for the ternary Miller loop and provide a detailed cost estimation and compare our theoretical costs with the recent work by Fouotsa et al. [9]. Notably, we mathematically verify the exact congruence of p modulo 27 for both our seeds and theirs, correcting previous Frobenius cost assumptions and demonstrating substantial efficiency gains.

The remainder of this paper is organized as follows. Section 2 recalls the fundamental definitions of the BLS27 family, the optimal Ate pairing, and the arithmetic properties of the underlying finite field extensions. Section 3 presents our novel ternary Miller algorithm, providing rigorous mathematical proofs for its correctness, alongside our ternary exponentiation approach for the hard part of the final exponentiation. Section 4 details the generation of our specific ternary seeds and provides a comprehensive complexity analysis of both the Miller loop and the final exponentiation. Section 5 contrasts our theoretical costs against the state of the art binary approach by Fouotsa et al., highlighting the significant efficiency gains and security advantages achieved by our method. Finally, Section 6 concludes the paper.

2 Background

We recall the arithmetic properties in $\mathbb{F}_{p^{27}}$, the fundamental definitions of the BLS27 family, and the optimal Ate pairing.

2.1 BLS27 Curves

The Barreto-Lynn-Scott (BLS) family with embedding degree $k = 27$, denoted BLS27, is parameterized by an integer x as [1]:

$$\begin{aligned} p(x) &= \frac{1}{3}(x-1)^2(x^{18}+x^9+1)+x \\ r(x) &= \frac{1}{3}(x^{18}+x^9+1) \\ t(x) &= x+1 \end{aligned}$$

where p is the field characteristic, r is the prime subgroup order, and t is the trace of Frobenius. For a suitable integer x such that both $p(x)$ and $r(x)$ are prime, the elliptic curve E is defined over the base field $\mathbb{F}_p$ by the Weierstrass equation $E : Y^2 = X^3 + b$, where $b \in \mathbb{F}_p^*$. Because the coefficient of X is zero, this curve has a j-invariant of 0, which is a necessary condition for admitting a degree-3 (cubic) twist.
Let $\overline{\mathbb{F}_p}$ denote the algebraic closure of $\mathbb{F}_p$. The Frobenius endomorphism $\pi_p : E(\overline{\mathbb{F}_p}) \to E(\overline{\mathbb{F}_p})$ is defined by $\pi_p(X, Y) = (X^p, Y^p)$. The map π_p has exactly two eigenvalues, 1 and p, which allows us to define the two distinct r-torsion subgroups on the curve:

$$\begin{aligned} G_1 &= E(\overline{\mathbb{F}_p})[r] \cap \mathrm{Ker}(\pi_p - [1]) = E(\mathbb{F}_p)[r] \\ G_2 &= E(\overline{\mathbb{F}_p})[r] \cap \mathrm{Ker}(\pi_p - [p]) \subset E(\mathbb{F}_{p^{27}})[r] \end{aligned}$$

Because $k = 27 = 3^3$, the extension field $\mathbb{F}_{p^{27}}$ can be towered as a degree-3 extension of $\mathbb{F}_{p^9}$. To exploit this structure for efficient arithmetic, we utilize a cubic twist E' of E [7]. Let $\omega \in \mathbb{F}_{p^{27}}$ be an element such that $\omega^3 = \nu \in \mathbb{F}_{p^9}$ but $\omega \notin \mathbb{F}_{p^9}$. The twisted curve E' is defined over $\mathbb{F}_{p^9}$ by the equation $E' : Y'^2 = X'^3 + b\omega^6$.

The curves E and E' are isomorphic over the full extension field $\mathbb{F}_{p^{27}}$ via the degree-3 twist isomorphism $\psi : E' \to E$ given by:

$$\psi(X', Y') = \left(\frac{X'}{\omega^2}, \frac{Y'}{\omega^3}\right) = \left(\frac{X'\omega}{\nu}, \frac{Y'}{\nu}\right)$$

The fundamental property of this twist is that while the isomorphism ψ is defined over $\mathbb{F}_{p^{27}}$, it maps the subgroup $G_2 \subset E(\mathbb{F}_{p^{27}})[r]$ to a subgroup G_2' that is entirely contained within $E'(\mathbb{F}_{p^9})[r]$. This allows all elliptic curve point operations involving the second argument Q to be performed over the significantly smaller subfield $\mathbb{F}_{p^9}$.

2.2 Arithmetic in $\mathbb{F}_{p^{27}}$ and Cyclotomic Subgroup

Following Fouotsa et al. [9] and Ben Abdelghani et al. [3], we construct the tower:

$$\begin{aligned} \mathbb{F}_{p^3} &= \mathbb{F}_p[u]/(u^3 - 7). \\ \mathbb{F}_{p^9} &= \mathbb{F}_{p^3}[v]/(v^3 - u). \\ \mathbb{F}_{p^{27}} &= \mathbb{F}_{p^9}[w]/(w^3 - v). \end{aligned}$$

The costs of operations in these fields are established as follows:

- Multiplication in $\mathbb{F}_{p^{27}}$: $M_{27} = 216M_1$.

- Squaring in $\mathbb{F}_{p^{27}}$: $S_{27} = 153M_1$.
- The inversion in $\mathbb{F}_{p^{27}}$: $I_{27} = 536M_1$.
- Standard Cubing in $\mathbb{F}_{p^{27}}$: $C_{27} = 3S_9 + 8M_9 = 363M_1$.
- Cyclotomic Cubing in $G_{\phi_3(p^9)}$: $C_{\text{cyc27}} = 252M_1$.
- Cyclotomic Inversion in $G_{\phi_3(p^9)}$: $I_{\text{cyc27}} = 189M_1$.
- Frobenius Operations: According to Ben Abdelghani et al. [3], the cost of the p^i-Frobenius map in $\mathbb{F}_{p^{27}}$ depends heavily on the congruence of p modulo 27. Applying Corollary 1 from [3], we distinguish the following cases:
  - Case 1 ($27 \mid p-1$): the p^9-Frobenius map costs $18M_1$, p^3 and p^6 cost $24M_1$, while the $p, p^2, p^4, p^5, p^7, p^8$-Frobenius maps cost $26M_1$.
  - Case 2 ($9 \mid p-1$ but $27 \nmid p-1$): the p^9-Frobenius map costs $18M_1$, p^3 and p^6 cost $24M_1$, while the $p, p^2, p^4, p^5, p^7, p^8$-Frobenius maps cost $116M_1$.
  - Case 3 ($3 \mid p-1$ but $9 \nmid p-1$): the p^9-Frobenius map costs $18M_1$, p^3 and p^6 cost $114M_1$, while the $p, p^2, p^4, p^5, p^7, p^8$-Frobenius maps cost $686M_1$.

To ensure a rigorous comparison, we must mathematically verify the exact condition each seed satisfies, rather than assuming a generic case. For the BLS27 family, the base field prime is defined by $p(x) = \frac{1}{3}(x-1)^2(x^{18}+x^9+1)+x$. To determine $p(x) \pmod{27}$ for seeds where $x \equiv 1 \pmod 3$, we can use the binomial theorem. If $x \equiv 1 \pmod 3$, then $x = 1+3k$ for some integer k. By the binomial expansion:

$$x^9 = (1+3k)^9 = 1 + 9(3k) + \binom{9}{2}(3k)^2 + \sum_{i=3}^{9}\binom{9}{i}(3k)^i \equiv 1 + 27k + 36(9k^2) \equiv 1 \pmod{27}.$$

Consequently, $x^{18} \equiv 1 \pmod{27}$ as well. This implies that the cyclotomic polynomial factor $\Phi_{27}(x) = x^{18}+x^9+1 \equiv 1+1+1 = 3 \pmod{27}$. Substituting this back into the parameter equation yields:

$$p(x) \equiv \frac{1}{3}(x-1)^2 \cdot 3 + x = (x-1)^2 + x = x^2 - 2x + 1 + x = x^2 - x + 1 \pmod{27}. \quad (1)$$

For the seed $x_F = 2^{51} + 2^{42} + 2^{28} + 2^9 + 1$ used by Fouotsa et al. [9], we find $x_F \equiv 25 \pmod{27}$, yielding $p(x_F) \equiv 25^2 - 25 + 1 \equiv 7 \pmod{27}$. This implies $3 \mid p-1$ but $9 \nmid p-1$, placing their implementation in the expensive case (Case 3). In their work, they assumed Case 1, leading to a severe underestimation of their Frobenius costs.

2.3 Optimal Ate Pairing

Building upon the subgroups $G_1 \subset E(\mathbb{F}_p)[r]$ and $G_2 \subset E(\mathbb{F}_{p^{27}})[r]$ defined previously, we formalize the pairing computation. For the BLS27 family, the trace of Frobenius satisfies $t(x) - 1 = x$. According to the framework of optimal pairings established by Vercauteren [16], the shortest non-trivial vector in the lattice of pairing functions corresponds exactly to the parameter x.

Consequently, the optimal Ate pairing on BLS27 is defined as the non-degenerate bilinear map:

$$e : G_2 \times G_1 \to \mu_r \subset \mathbb{F}_{p^{27}}^*$$
$$e(Q,P) = f_{x,Q}(P)^{\frac{p^{27}-1}{r}}$$

where $f_{x,Q}$ is the Miller function with divisor $\text{div}(f_{x,Q}) = x(Q) - ([x]Q) - (x-1)(\mathcal{O})$, evaluated at the base field point $P \in G_1$. The computation of this map naturally divides into two distinct

phases: the evaluation of the Miller function $f_{x,Q}(P)$, and the final exponentiation to the power $\frac{p^{27}-1}{r}$.

Use of Twists and Denominator Elimination: As established in [7], the standard practice when computing the optimal Ate pairing is to map the point $Q \in G_2$ to the twisted curve via $Q' = \psi^{-1}(Q) \in G_2'$, so that the point operations (doublings and triplings) inside the Miller loop are performed entirely within the smaller subfield $\mathbb{F}_{p^9}$. When it is time to evaluate the rational line functions, the points are "untwisted" back to $E(\mathbb{F}_{p^{27}})$ via ψ.

Furthermore, by evaluating the line functions at the base field point $P \in E(\mathbb{F}_p)$ during the Miller loop, the denominators (vertical lines) evaluate to elements in $\mathbb{F}_p^*$. Because the final exponent is a multiple of $p-1$, these elements are mapped to 1 during the final exponentiation and can be safely ignored—a technique known as denominator elimination.

2.3.1 The Binary Miller Algorithm

The standard method for evaluating the Miller function $f_{m,Q}(P)$ relies on the binary representation of the integer m. Introduced by Miller [13], this algorithm processes the bits of m from the most significant to the least significant, using a double-and-add approach.

Let m be a positive integer with binary representation $m = \sum_{i=0}^{k-1} d_i 2^i$, where $d_i \in \{0, 1\}$. The algorithm relies on the algebraic relations between Miller functions of different orders. Let $h_{U,V}$ denote the rational line function with divisor $\mathrm{div}(h_{U,V}) = (U) + (V) - (U+V) - (\mathcal{O})$. The transitions from a state n to $2n$ (doubling) and from $2n$ to $2n+1$ (addition) are governed by the following function updates:

$$f_{2n,Q} = f_{n,Q}^2 \cdot h_{[n]Q,[n]Q},$$
$$f_{2n+1,Q} = f_{2n,Q} \cdot h_{[2n]Q,Q}.$$

In the context of the optimal Ate pairing on BLS27, the evaluation point is $P \in G_1 = E(\mathbb{F}_p)[r]$. The point Q formally belongs to $G_2 \subset E(\overline{\mathbb{F}_p})[r] \cap \mathrm{Ker}(\pi_p - [p]) \subset E(\mathbb{F}_{p^{27}})$.

To avoid computationally expensive arithmetic in the full extension field $\mathbb{F}_{p^{27}}$, we exploit the degree-3 twist isomorphism $\psi : E' \to E$. By utilizing the preimage of Q under the twist, denoted as $Q' = \psi^{-1}(Q) \in E'(\mathbb{F}_{p^9})$, the point operations (doublings and additions) can be performed efficiently on the twist E' over the smaller subfield $\mathbb{F}_{p^9}$. The function updates (squarings and multiplications by line functions) are performed in $\mathbb{F}_{p^{27}}$.

Evaluating the line functions $h_{U,V}(P)$ at the base field point $P \in E(\mathbb{F}_p)$ ensures that the vertical line denominators evaluate to elements in $\mathbb{F}_p^*$. Since $p-1$ divides $\frac{p^{27}-1}{r}$, these denominators are mapped to 1 during the final exponentiation and can thus be ignored during the Miller loop, a technique known as denominator elimination. The binary Miller loop is described in Algorithm 1

2.3.2 Final Exponentiation

The final exponentiation (FE) raises the output of the Miller loop $f \in \mathbb{F}_{p^{27}}^*$ to the power $\frac{p^{27}-1}{r}$. Following the standard framework described by Scott et al. [15] and applied by Fouotsa et al. [9] for $k = 27$, this exponent is split into a computationally inexpensive easy part and a more intensive hard part:

$$\frac{p^{27}-1}{r} = \underbrace{[p^9 - 1]}_{\text{Easy part}} \cdot \underbrace{\left[\frac{p^{18}+p^9+1}{r}\right]}_{\text{Hard part}}.$$

We denote the exponent of the hard part by $d = \frac{p^{18}+p^9+1}{r}$. The easy part maps the element to the cyclotomic subgroup $G_{\Phi_3(p^9)}$, which has order $p^{18} + p^9 + 1$. It is computed efficiently using a single p^9-Frobenius map, an inversion, and a multiplication in $\mathbb{F}_{p^{27}}$.

Algorithm 1 Binary Miller Algorithm for Optimal Ate Pairing on BLS27

Require: $Q' \in E'(\mathbb{F}_{p^9})$ (representing $Q = \psi(Q') \in G_2$), $P \in G_1 \subset E(\mathbb{F}_p)$, and $m = \sum_{i=0}^{k-1} d_i 2^i$ with $d_i \in \{0, 1\}$
Ensure: $f_{m,Q}(P) \in \mathbb{F}^*_{p^{27}}$

$T \leftarrow Q'$; $f \leftarrow 1$
for $i = k - 2$ down to 0 **do**
    // **Doubling Step**
    $f \leftarrow f^2$
    $f \leftarrow f \cdot h_{T,T}(P)$
    $T \leftarrow 2T$
    if $d_i = 1$ **then**
        // **Addition Step**
        $f \leftarrow f \cdot h_{T,Q'}(P)$
        $T \leftarrow T + Q'$
    end if
end for
return f

The hard part, raising the result to the power d, is the computational bottleneck. Fouotsa et al. [9] computed this using the lattice-based method of Fuentes-Castañeda et al. [10], expressing d as a polynomial in x and evaluating it via a vectorial addition chain.
Alternatively, Hayashida et al. [12] proposed a framework that directly factorizes the hard part using homogeneous cyclotomic polynomials. For the BLS27 family, where $r(x) = (x^{18}+x^9+1)/3$, they derived the following particular factorization for the hard part $d(x)$:

$$d(x) = (x-1)^2 \cdot \Psi_1(x,p)\Psi_3(x,p)\Psi_9(x,p) \cdot (x^9 + p^9 + 1) + 3,$$

where $\Psi_1(x,p) = 1$, $\Psi_3(x,p) = x^2 + px + p^2$, and $\Psi_9(x,p) = x^6 + p^3x^3 + p^6$ are the homogeneous cyclotomic polynomials. The authors evaluated this factorization using 17 exponentiations by x, 2 exponentiations by $x - 1$, and several Frobenius maps. The total complexity of the final exponentiation based on Hayashida's method is expressed as:

$$I_{27} + 9 \times M_{27} + S_{27} + F_{27} + 2 \times E_{x-1} + 17 \times E_x, \tag{2}$$

where $F_{27} = p + p^2 + p^3 + p^6 + 2p^9$ denotes the total cost of Frobenius maps, and E_x, E_{x-1} denote the exponentiations by x and $x - 1$ in the cyclotomic subgroup, respectively.

3 Pairing using ternary computation

In this section, we present a ternary version of Miller algorithm and detail the mathematical proofs that ensure its correctness, alongside our ternary exponentiation approach for the final exponentiation.

3.1 The Ternary Miller Algorithm

We propose a ternary version of Miller's algorithm that evaluates $f_{x,Q}(P)$ by processing the base-3 representation of x. To maximize efficiency, we restrict the ternary digits to $d_i \in \{0, 1\}$, avoiding the digit 2. This eliminates the need to precompute $2Q$ and evaluate the corresponding line function. The Miller ternary loop is given in Algorithm 2.
As established for the binary algorithm, we use Q' and T to denote the points as they are represented on the twist $E'(\mathbb{F}_{p^9})$. The rational line functions are evaluated at $P \in E(\mathbb{F}_p)$ using

the images of these points under the twist isomorphism ψ. This allows the point operations (triplings and additions) to be performed efficiently over the subfield $\mathbb{F}_{p^9}$, while the function updates (cubings and multiplications) are performed in $\mathbb{F}_{p^{27}}$. Evaluating the lines at $P \in E(\mathbb{F}_p)$ allows for denominator elimination techniques, where multiplying by the appropriate conjugate factor pushes the denominators into a proper subfield of $\mathbb{F}_{p^{27}}$, which is then eliminated during the final exponentiation.

Algorithm 2 Ternary Miller Algorithm for BLS27 (Digits in $\{0, 1\}$)

Require: $Q' \in E'(\mathbb{F}_{p^9})$ (representing $Q = \psi(Q') \in G_2$), $P \in G_1 \subset E(\mathbb{F}_p)$, $x = \sum_{i=0}^{k-1} d_i 3^i$ with $d_i \in \{0, 1\}$
Ensure: $f_{x,Q}(P) \in \mathbb{F}^*_{p^{27}}$
1: $T \leftarrow Q'$; $f \leftarrow 1$
2: **for** $i = k - 2$ down to 0 **do**
3: // **Tripling Step**
4: $f \leftarrow f^3$
5: $f \leftarrow f \cdot h_{T,T}(P) \cdot h_{2T,T}(P)$
6: $T \leftarrow 3T$
7: **if** $d_i = 1$ **then**
8: // **Addition Step**
9: $f \leftarrow f \cdot h_{T,Q'}(P)$
10: $T \leftarrow T + Q'$
11: **end if**
12: **end for**
13: **return** f

To rigorously prove the correctness of our algorithm, we rely on the properties of the Miller function divisor in $E(\overline{\mathbb{F}_p})$. Let $Q = \psi(Q') \in E(\overline{\mathbb{F}_p})$ be the abstract mathematical point on E corresponding to the algorithmic input. The Miller function $f_{m,Q}$ has divisor $\mathrm{div}(f_{m,Q}) = m(Q) - ([m]Q) - (m-1)(\mathcal{O})$. Let $h_{U,V}$ denote the abstract rational line function on E with divisor $(U) + (V) - (U+V) - (\mathcal{O})$. When $U = V$, $h_{U,U}$ represents the tangent line at U.

Theorem 1. *For any integer m, the transition from $f_{m,Q}$ to $f_{3m,Q}$ is given by the relation:*

$$f_{3m,Q} = f_{m,Q}^3 \cdot h_{[m]Q,[m]Q} \cdot h_{[2m]Q,[m]Q}$$

Proof. We verify the equality by comparing the divisors of both sides. The divisor of a product is the sum of the divisors.

- $\mathrm{div}(f_{m,Q}^3) = 3m(Q) - 3([m]Q) - 3(m-1)(\mathcal{O})$
- $\mathrm{div}(h_{[m]Q,[m]Q}) = 2([m]Q) - ([2m]Q) - (\mathcal{O})$
- $\mathrm{div}(h_{[2m]Q,[m]Q}) = ([2m]Q) + ([m]Q) - ([3m]Q) - (\mathcal{O})$

Summing these three divisors together yields:

$$\begin{aligned}
&3m(Q) + (-3([m]Q) + 2([m]Q) + ([m]Q)) + (-([2m]Q) + ([2m]Q)) - ([3m]Q) \\
&+ (-3(m-1) - 1 - 1)\,(\mathcal{O}) \\
&= 3m(Q) + 0([m]Q) + 0([2m]Q) - ([3m]Q) - (3m - 3 + 2)(\mathcal{O}) \\
&= 3m(Q) - ([3m]Q) - (3m-1)(\mathcal{O})
\end{aligned}$$

This exactly matches the definition of $\mathrm{div}(f_{3m,Q})$. Thus, the function update is mathematically exact. □

Theorem 2. *For any integer m, the transition from $f_{3m,Q}$ to $f_{3m+1,Q}$ is given by:*

$$f_{3m+1,Q} = f_{3m,Q} \cdot h_{[3m]Q,Q}$$

Proof. Again, summing the divisors of the right-hand side:

$$\begin{aligned}&(3m(Q) - ([3m]Q) - (3m-1)(\mathcal{O})) + (([3m]Q) + (Q) - ([3m+1]Q) - (\mathcal{O}))\\ &= (3m+1)(Q) - ([3m+1]Q) - (3m)(\mathcal{O}) = \operatorname{div}(f_{3m+1,Q})\end{aligned}$$

□

Theorem 3 (Loop Invariant and Algorithm Correctness)**.** *Let $x = \sum_{i=0}^{k-1} d_i 3^i$ be the ternary representation of the loop parameter with $d_i \in \{0,1\}$, and let $m_i = \sum_{j=i}^{k-1} d_j 3^{j-i}$ be the integer formed by the most significant trits down to index i. Let $T_i \in E(\overline{\mathbb{F}_p})$ be the mathematical point corresponding to the algorithmic state variable T at the end of iteration i (i.e., $T_i = \psi(T)$). At the end of iteration i, the ternary Miller algorithm maintains the following invariant:*

$$T_i = [m_i]Q \quad \textit{and} \quad f = f_{m_i,Q}(P)$$

Consequently, upon termination at $i = 0$, the algorithm outputs $f_{x,Q}(P)$.

Proof. We prove the correctness of the algorithm by mathematical induction on the loop index i.

Base Case ($i = k-1$): Before the main loop begins, the algorithm initializes $T \leftarrow Q'$ and $f \leftarrow 1$. Mathematically, this means $T_{k-1} = \psi(Q') = Q$. Since $x > 0$, its most significant trit must be $d_{k-1} = 1$, so $m_{k-1} = 1$. By definition, the Miller function $f_{1,Q}$ has the divisor:

$$\operatorname{div}(f_{1,Q}) = 1(Q) - ([1]Q) - (1-1)(\mathcal{O}) = (Q) - (Q) - 0(\mathcal{O}) = 0$$

A rational function with a zero divisor is a non-zero constant, which we normalize to 1. Thus, $f_{1,Q}(P) = 1$. The initialization therefore yields $T_{k-1} = [1]Q = [m_{k-1}]Q$ and $f = 1 = f_{m_{k-1},Q}(P)$, so the invariant holds at the start of the loop.

Inductive Hypothesis: Assume that at the end of iteration $i+1$, the invariant holds. That is, the variables are in the state $T_{i+1} = [m_{i+1}]Q$ and $f = f_{m_{i+1},Q}(P)$.

Inductive Step ($i+1 \to i$): During iteration i, the algorithm processes the trit d_i in two phases:

1. *Tripling Step:* The algorithm updates $f \leftarrow f^3 \cdot h_{T,T}(P) \cdot h_{2T,T}(P)$ and $T \leftarrow 3T$. Since T represents T_{i+1} mathematically, this update evaluates the lines through T_{i+1} and transitions the point to $T_i \leftarrow 3T_{i+1}$. By Theorem 2.1 (Tripling Step Correctness), the function transitions to represent $3m_{i+1}$. Therefore, immediately after the tripling step, the intermediate variables satisfy $T_i = [3m_{i+1}]Q$ and $f = f_{3m_{i+1},Q}(P)$.

2. *Addition Step:* If $d_i = 1$, the algorithm updates $f \leftarrow f \cdot h_{T,Q'}(P)$ and $T \leftarrow T + Q'$. This evaluates the line through T_i and Q, transitioning the point to $T_i \leftarrow T_i + Q$. By Theorem 2.2 (Addition Step Correctness), the function transitions to represent $3m_{i+1}+1$. If $d_i = 0$, no addition occurs, and the state remains $3m_{i+1}$.

By the definition of base-3 positional notation, the integer formed by the prefix of trits down to index i is derived by multiplying the previous prefix by 3 and adding the current digit:

$$m_i = 3m_{i+1} + d_i$$

Because the algorithm explicitly computes $3m_{i+1} + d_i$ through the tripling and conditional addition steps, at the end of iteration i, the variables exactly match the invariant: $T_i = [m_i]Q$ and $f = f_{m_i,Q}(P)$.

Termination: The loop terminates after processing the least significant trit ($i = 0$). According to the invariant, at the end of $i = 0$, the function f holds $f_{m_0,Q}(P)$. By definition, $m_0 = \sum_{j=0}^{k-1} d_j 3^j = x$. Thus, the algorithm correctly outputs $f = f_{m_0,Q}(P) = f_{x,Q}(P)$. □

3.2 Ternary Hard Part of Final Exponentiation

The most computationally expensive operation in the hard part of the final exponentiation is the exponentiation by the seed x within the cyclotomic subgroup. Previous works, such as Fouotsa et al. [9], evaluated this exponentiation using the binary representation of x, performing a cyclotomic squaring at each step, followed by a cyclotomic multiplication depending on whether the current bit was 1. However, in the extension field $\mathbb{F}_{p^{27}}$, fast cyclotomic squaring is not available (only classical squaring is supported), as the embedding degree $k = 27$ does not allow the standard squaring-friendly cyclotomic structure. Fortunately, as demonstrated by Nanjo et al. [14] and Ben Abdelghani et al. [3], an efficient cyclotomic cubing formula is supported in this context. To fully exploit the efficiency of this operation, we specifically searched for seeds that are sparse in their ternary representation. This ternary sparsity is a critical design choice, as it simultaneously guarantees the efficiency of both the Miller loop and the final exponentiation. We successfully generated two sparse ternary seeds that achieve the 256-bit security level against exTNFS and SexTNFS attacks [2]. We generate two ternary seeds satisfying the following criteria:

- Both parameters $p(x)$ and $r(x)$ must be prime.
- The seed must guarantee a 256-bit security level against NFS variants.
- The ternary representation of the seed must be highly sparse to minimize the number of addition steps in the Miller loop.
- The ternary digits must be restricted to $\{0, 1\}$ to avoid the precomputation of $2Q$ and eliminate extra cyclotomic inversions.

With respect to the above conditions, we were able to generated seeds are as follows:

- exTNFS Seed: $x_1 = 3^{23} + 3^5 + 3^3 + 1$. This seed has a ternary loop length of $l_1 = 23$ and a Hamming weight of $hw = 4$.
- SexTNFS Seed: $x_2 = 3^{32} + 3^{31} + 3^{27} + 3^{22} + 3^{10} + 1$. This seed has a ternary loop length of $l_2 = 33$ and a Hamming weight of $hw = 6$.

To accommodate varying application requirements, we generated two distinct seeds targeting exTNFS and SexTNFS resistance, respectively. Both seeds maintain the 256-bit security level but offer a deliberate trade-off between performance and maximum cryptographic strength. The exTNFS seed is optimized for scenarios where computational efficiency is the primary objective. Conversely, for applications requiring the highest level of security against the more powerful SexTNFS attacks, the SexTNFS seed should be selected, acknowledging a marginal reduction in computational speed. Because of the presence of the cyclotomic cubing, we express the final exponentiation complexity as follows:

$$I_{27} + 8 \times M_{27} + C_{\text{cyc}27} + F_{27} + 2 \times E_{x-1} + 17 \times E_x. \tag{3}$$

4 Complexity Analysis

In this section, we discuss some optimization techniques and give a complexity analysis for optimal Ate pairing on the curve BLS27.

4.1 Optimizations and Complexity in the Ternary Miller Loop

In this subsection, we investigate optimization techniques to improve the efficiency of the ternary Miller loop and provide a thorough analysis of its computational complexity.

4.1.1 Preliminaries and Notation

We consider $E : y^2 = x^3 + b$ over $\mathbb{F}_p$ and $E' : y'^2 = x'^3 + b'$ over $\mathbb{F}_{p^9}$, where $b' = b \cdot \omega^6 = b \cdot \nu^2$. The twist isomorphism is $\psi(x, y) = \big(x/\omega^2, y/\omega^3\big) = ((x \cdot \omega)/\nu, y/\nu)$. In Jacobian coordinates, a point T on E' is $T = (X_1 : Y_1 : Z_1)$, with affine coordinates $x_1 = X_1/Z_1^2$ and $y_1 = Y_1/Z_1^3$. The evaluation point is $P = (x_P, y_P) \in E(\mathbb{F}_p)$. To evaluate lines on E', we map P to $P' = \psi^{-1}(P) = (x_S, y_S) \in E'(\mathbb{F}_{p^{27}})$, where $x_S = x_P \cdot \omega^2, \quad y_S = y_P \cdot \omega^3 = y_P \cdot \nu$. Consequently, $x_S^2 = (x_P \cdot \omega^2)^2 = x_P^2 \cdot \omega^4 = x_P^2 \cdot \nu \cdot \omega$. Because $x_P, y_P \in \mathbb{F}_p$ and $\nu \in \mathbb{F}_{p^9}$, the scalars $(x_P^2 \cdot \nu)$, x_P, and $(y_P \cdot \nu)$ are constants precomputed once in $\mathbb{F}_{p^9}$.

4.1.2 Denominator elimination

Lemma 1. *For a cubic twist, let the vertical line denominator through a point $U = (X_U : Y_U : Z_U)$ evaluated at P' be $v = x_S - (X_U/Z_U^2)$. Let its conjugate factor be $\bar{v} = x_S^2 + x_S \cdot (X_U/Z_U^2) + (X_U/Z_U^2)^2$. Then, $v \cdot \bar{v} \in \mathbb{F}_{p^9}^*$, and thus vanishes during the final exponentiation.*

Proof. We compute the product explicitly:

$$\begin{aligned} v \cdot \bar{v} &= \left(x_S - \left(\frac{X_U}{Z_U^2}\right)\right) \cdot \left(x_S^2 + x_S \cdot \left(\frac{X_U}{Z_U^2}\right) + \left(\frac{X_U}{Z_U^2}\right)^2\right) \\ &= x_S^3 - \left(\frac{X_U^3}{Z_U^6}\right) \\ &= (x_P \cdot \omega^2)^3 - (X_U^3 \cdot Z_U^{-6}) \\ &= x_P^3 \cdot \omega^6 - (X_U^3 \cdot Z_U^{-6}) \\ &= x_P^3 \cdot \nu^2 - (X_U^3 \cdot Z_U^{-6}) \end{aligned}$$

Using the Jacobian curve equation $Y_U^2 = X_U^3 + b \cdot Z_U^6 \implies X_U^3 = Y_U^2 - (b \cdot Z_U^6)$, we substitute:

$$v \cdot \bar{v} = x_P^3 \cdot \nu^2 - \big((Y_U^2 - (b \cdot Z_U^6)) \cdot Z_U^{-6}\big) = \underbrace{x_P^3 \cdot \nu^2 - (Y_U^2 \cdot Z_U^{-6}) + b}_{\in \mathbb{F}_{p^9}^*}$$

Since $v \cdot \bar{v} \in \mathbb{F}_{p^9}^*$, multiplying the numerator by $\bar{v}$ successfully pushes the denominator into a subfield where it is mapped to 1. The evaluated Miller update thus becomes the full polynomial $H(x_S, y_S) = (L_{2,0} \cdot x_S^2) + (L_{1,0} \cdot x_S) + (L_{0,1} \cdot y_S) + L_{0,0}$ [7]. □

4.1.3 Optimizations in the tripling step

The tripling step consists of executing the dedicated tripling formula, which simultaneously provides the new point coordinates $3T$ and the intermediate slope values H and R. Using these existing intermediates, we evaluate the line functions H_1 and H_2 and update the Miller function as $f \leftarrow f^3 \cdot H_1 \cdot H_2$. Finally, the point variable is updated by assigning the computed coordinates, $T \leftarrow 3T$.

Point tripling and slope extraction we use the dedicated $a = 0$ Jacobian tripling formula [8] to compute $3T = (X_4 : Y_4 : Z_4)$ directly from $T = (X_1 : Y_1 : Z_1)$. This formula computes the

point $[3]T$ without explicitly storing $[2]T$, and generates the following intermediates:

$$\begin{aligned}
&A = X_1^2, \quad B = Y_1^2, \quad M = 3A,\\
&E = 12X_1B - M^2,\\
&T_v = 8B^2,\\
&W = ME - T_v \quad \text{(Mathematically } Y_3\text{)},\\
&V = 2T_v - ME \quad \text{(Mathematically } T_v - Y_3\text{)},\\
&Z_4 = Z_1 \cdot E,\\
&X_4 = 8B \cdot (T_v - ME) + X_1 \cdot E^2,\\
&Y_4 = Y_1 \cdot (4W \cdot V - E^3).
\end{aligned}$$

Theorem 4. *To evaluate the Miller update $h_{2T,T}(P) = l_{2T,T}(P) \cdot \bar{v}_{3T}(P)$, we need the standard addition slope intermediates $H = X_3Z_1^2 - X_1Z_3^2$ and $R = Y_3Z_1^3 - Y_1Z_3^3$ to find the chord line $l_{2T,T}$. These can be extracted directly from the tripling intermediates without computing $[2]T$ via the equalities:*

$$H = -E \cdot Z_1^2 \quad \textit{and} \quad R = -V \cdot Z_1^3.$$

Proof. In the doubling step ($[2]T$), the Jacobian Z-coordinate is $Z_3 = 2Y_1Z_1$. Therefore, $Z_3^2 = 4Y_1^2Z_1^2 = 4BZ_1^2$. The definition of H is $H = X_3Z_1^2 - X_1Z_3^2$. Substituting Z_3^2:

$$H = X_3Z_1^2 - X_1(4BZ_1^2) = (X_3 - 4X_1B)Z_1^2.$$

In the doubling formula, $X_3 = M^2 - 8X_1B$. Substituting this:

$$H = (M^2 - 8X_1B - 4X_1B)Z_1^2 = (M^2 - 12X_1B)Z_1^2.$$

The tripling formula explicitly computes $E = 12X_1B - M^2$. Therefore, $M^2 - 12X_1B = -E$, which yields:

$$H = -E \cdot Z_1^2.$$

For R, the definition is $R = Y_3Z_1^3 - Y_1Z_3^3$. We know $Z_3 = 2Y_1Z_1$, so $Z_3^3 = 8Y_1^3Z_1^3 = 8Y_1BZ_1^3$. Substituting Z_3^3 into R:

$$R = Y_3Z_1^3 - Y_1(8Y_1BZ_1^3) = (Y_3 - 8Y_1^2B)Z_1^3.$$

Since $B = Y_1^2$, we have $8Y_1^2B = 8B^2$. The tripling formula defines $T_v = 8B^2$. Thus:

$$R = (Y_3 - T_v)Z_1^3.$$

In the doubling formula, $Y_3 = M(12X_1B - M^2) - 8B^2 = ME - T_v$. Substituting Y_3, we get:

$$R = (ME - T_v - T_v)Z_1^3 = (ME - 2T_v)Z_1^3.$$

The tripling formula defines $V = 2T_v - ME$. Therefore, $ME - 2T_v = -V$, which yields:

$$R = -V \cdot Z_1^3.$$

□

The cost of the tripling formula is $8M_9 + 5S_9$. The slope extraction costs $S_9 + 3M_9$. Thus, the total cost of the tripling step is $11M_9 + 6S_9$.

Explicit formulas for H_1 and H_2 the product of the line and its conjugate collapses algebraically into polynomials [7]. For $H_1 = l_{T,T} \cdot \bar{v}_{2T}$, the polynomial is

$$H_1(x_S, y_S) = (L_{2,0}^{(1)} \cdot x_S^2) + (L_{1,0}^{(1)} \cdot x_S) + (L_{0,1}^{(1)} \cdot y_S) + L_{0,0}^{(1)},$$

where:

$$\begin{aligned} L_{2,0}^{(1)} &= 4 \cdot X_1 \cdot Y_1^2 \cdot Z_1, \\ L_{1,0}^{(1)} &= 3 \cdot X_1^2 \cdot (Y_1^2 - 3 \cdot b \cdot Z_1^6), \\ L_{0,1}^{(1)} &= -6 \cdot X_1^3 \cdot Y_1, \\ L_{0,0}^{(1)} &= (Y_1^2 - b \cdot Z_1^6) \cdot (Y_1^2 + 9 \cdot b \cdot Z_1^6). \end{aligned}$$

For $H_2 = l_{2T,T} \cdot \bar{v}_{3T}$, the polynomial is $H_2(x_S, y_S) = (L_{2,0}^{(2)} \cdot x_S^2) + (L_{1,0}^{(2)} \cdot x_S) + (L_{0,1}^{(2)} \cdot y_S) + L_{0,0}^{(2)}$, where:

$$\begin{aligned} L_{2,0}^{(2)} &= 2 \cdot Z_4^2, \\ L_{1,0}^{(2)} &= 2 \cdot X_4 \cdot Z_4, \\ L_{0,1}^{(2)} &= -2 \cdot H^2 \cdot R \cdot Z_4, \\ L_{0,0}^{(2)} &= 2 \cdot X_4^2 + 2 \cdot H^2 \cdot R \cdot Y_4. \end{aligned}$$

The coefficients of H_1 depend strictly on T, while the coefficients of H_2 depend on $[3]T$ and the extracted H, R. All of these are already available, so $L_{i,j}$ are computed with 0 extra multiplications.

Evaluation and function Update This phase is constituted by the following steps:

1. Computing f^3 with cost $C_{27} = 369M_1$.
2. Evaluate the collapsed polynomial template H_i (for $i \in \{1, 2\}$), given by:

   $$H_i = L_{0,0}^{(i)} + (L_{2,0}^{(i)} \cdot x_S^2) + (L_{1,0}^{(i)} \cdot x_S) + (L_{0,1}^{(i)} \cdot y_S).$$

   We substitute the constants ($x_S^2 = x_P^2 \nu w$, $x_S = x_P w^2$, $y_S = y_P \nu$), to get

   $$H_i = \underbrace{(L_{0,0}^{(i)} + L_{0,1}^{(i)} \cdot y_P \cdot \nu)}_{\text{coeff. of } w^0} + \underbrace{(L_{2,0}^{(i)} \cdot x_P^2 \cdot \nu)}_{\text{coeff. of } w^1} \cdot w + \underbrace{(L_{1,0}^{(i)} \cdot x_P)}_{\text{coeff. of } w^2} \cdot w^2.$$

   Although the raw affine line is sparse, multiplying by the conjugate factor $\bar{v}$ to eliminate the denominator makes the evaluated function H_i a full element in $\mathbb{F}_{p^{27}}$. To compute H_i, we must multiply the $\mathbb{F}_{p^9}$ coefficients by the precomputed constants. Note that $\nu = v \in \mathbb{F}_{p^9}$ is the base element of the tower, so multiplying by ν requires only a cyclic shift of the $\mathbb{F}_{p^3}$ coefficients (costing $0M_1$). Thus, multiplying by $x_P^2 \nu$, $x_P \in \mathbb{F}_p$, and $y_P \nu$ each costs only a multiplication by an $\mathbb{F}_p$ scalar, which is $9M_1$. Since there are 3 non-zero terms ($L_{2,0}, L_{1,0}, L_{0,1}$), evaluating each H_i costs $3 \times 9M_1 = 27M_1$. Cost for both lines is $2 \times 27M_1 = 54M_1$.
3. Computing $H_1 \cdot H_2$ with cost $6M_9$.
4. Updating the function by computing $f^3 \cdot (H_1 \cdot H_2)$. This costs $6M_9$.

Finally, the total tripling step cost is

$$C_{\text{trip}} = (11M_9 + 6S_9) + 369M_1 + (54M_1 + 6M_9 + 6M_9) = \mathbf{1413M_1}.$$

4.1.4 Optimizations in the addition Step

The addition step is executed conditionally only when the current trit $d_i = 1$. Following the exact sequence established for the tripling step, we first execute the mixed Jacobian addition (since Q' is a fixed affine point). This single operation simultaneously yields the new point coordinates and the slope intermediates. We then use these existing intermediates to evaluate the rational function H_{add}, update the Miller function as $f \leftarrow f \cdot H_{add}$, and finally finalize the point update by assigning the newly computed coordinates to T.

Explicit point addition intermediates we use mixed Jacobian addition [6] since $Q' = (X_2 : Y_2 : 1)$ is a fixed affine point. Given $T = (X_1 : Y_1 : Z_1)$, the intermediate slope variables and the resulting point $T + Q' = (X_5 : Y_5 : Z_5)$ are computed as:

$$\begin{aligned}
U_1 &= X_1, \quad S_1 = Y_1, \\
U_2 &= X_2 \cdot Z_1^2, \\
S_2 &= Y_2 \cdot Z_1^3, \\
H &= U_2 - U_1 = X_2 \cdot Z_1^2 - X_1, \\
R &= S_2 - S_1 = Y_2 \cdot Z_1^3 - Y_1, \\
X_5 &= R^2 - H^3 - 2 \cdot U_1 \cdot H^2, \\
Y_5 &= R \cdot (U_1 \cdot H^2 - X_5) - S_1 \cdot H^3, \\
Z_5 &= Z_1 \cdot H.
\end{aligned}$$

This formula costs $8M_9 + 3S_9$.

Explicit Formula for H_{add} The rational function $h_{T,Q'} = l_{T,Q'}/v_{T+Q'}$ requires denominator elimination. As proven in Lemma 2.1, we multiply the numerator by the conjugate factor $\bar{v}$. This product collapses algebraically into a polynomial [7]. This polynomial, denoted H_{add}, is evaluated at the mapped point $P' = (x_S, y_S) = (x_P\omega^2, y_P\nu) \in E'(\mathbb{F}_{p^{27}})$ as follows

$$H_{add}(x_S, y_S) = (L_{2,0}^{(3)} \cdot x_S^2) + (L_{1,0}^{(3)} \cdot x_S) + (L_{0,1}^{(3)} \cdot y_S) + L_{0,0}^{(3)},$$

where the coefficients are explicitly:

$$\begin{aligned}
L_{2,0}^{(3)} &= 2 \cdot Z_5^2, \\
L_{1,0}^{(3)} &= 2 \cdot X_5 \cdot Z_5, \\
L_{0,1}^{(3)} &= -2 \cdot H^2 \cdot R \cdot Z_5, \\
L_{0,0}^{(3)} &= 2 \cdot X_5^2 + 2 \cdot H^2 \cdot R \cdot Y_5.
\end{aligned}$$

The coefficients $L_{i,j}^{(3)}$ are computed strictly from the mixed addition intermediates (H, R) and the result coordinates (X_5, Y_5, Z_5). Thus, they are formed with 0 extra multiplications.

Evaluation and function update

1. We substitute the precomputed constants ($x_S^2 = x_P^2\nu w$, $x_S = x_P w^2$, $y_S = y_P\nu$) into the polynomial. As established in the tripling step, multiplying the $\mathbb{F}_{p^9}$ coefficients by these constants requires only multiplications by $\mathbb{F}_p$ scalars. With 3 non-zero terms, this costs $3 \times 9M_1 = 27M_1$.

2. Finally we compute $f \cdot H_{add}$ with cost $6M_9$.

Thus, the total addition step cost is

$$C_{\text{add}} = (8M_9 + 3S_9) + 27M_1 + 6M_9 = 14M_9 + 3S_9 + 27M_1 = \mathbf{612M_1}.$$

We illustrate our optimization in a detailed ternary Miller loop version (Algorithm 3).

Algorithm 3 Ternary Miller Algorithm for BLS27 (Digits $\{0, 1\}$)

Require: $Q' \in E'(\mathbb{F}_{p^9})$ $(Q = \psi(Q') \in G_2)$, $P \in G_1 \subset E(\mathbb{F}_p)$, $x = \sum_{i=0}^{k-1} d_i 3^i$ with $d_i \in \{0, 1\}$
Ensure: $f_{x,Q}(P) \in \mathbb{F}^*_{p^{27}}$

$T \leftarrow Q'$; $f \leftarrow 1$
for $i = k - 2$ down to 0 **do**
  // **Tripling Step**
  Compute $3T = (X_4, Y_4, Z_4)$ and extract slope factors H, R
  $f \leftarrow f^3$
  Evaluate $H_1 \leftarrow h_{T,T}(P)\bar{v}_{2T}(P)$, $H_2 \leftarrow h_{2T,T}(P)\bar{v}_{3T}(P)$
  $f \leftarrow f \cdot H_1 \cdot H_2$; $T \leftarrow (X_4 : Y_4 : Z_4)$
  if $d_i = 1$ **then**
    // **Addition Step**
    Compute $T + Q' = (X_5, Y_5, Z_5)$, evaluate $H_{add} \leftarrow h_{T,Q'}(P)\bar{v}_{T+Q'}(P)$
    $f \leftarrow f \cdot H_{add}$; $T \leftarrow (X_5 : Y_5 : Z_5)$
  end if
end for
return f

Cost of the Ternary Miller Loop we now evaluate the total complexity of the ternary Miller loop for our two proposed seeds.
For the exTNFS seed $x_1 = 3^{23} + 3^5 + 3^3 + 1$, the ternary loop length is $l_1 = 24$ (yielding 23 tripling steps) and the Hamming weight is $hw = 4$ (yielding 3 addition steps). The total Miller loop cost is given by:

$$23 \cdot C_{\text{trip}} + 3 \cdot C_{\text{add}} = 23(1413) + 3(612) = 32499 + 1836 = 34335M_1.$$

For the SexTNFS seed $x_2 = 3^{32} + 3^{31} + 3^{27} + 3^{22} + 3^{10} + 1$, the loop length is $l_2 = 33$ (yielding 32 tripling steps) and the Hamming weight is $hw = 6$ (yielding 5 addition steps). The total Miller loop cost is:

$$32 \cdot C_{\text{trip}} + 5 \cdot C_{\text{add}} = 45216 + 3060 = 48276M_1.$$

4.2 Complexity of the Final Exponentiation

Prior to computing the complexity of the final exponentiation, we leverage Equation (1) to rigorously verify that our generated seeds satisfy the condition $27 \mid p - 1$. This places our implementation in the optimal Case 1 for Frobenius operations, a distinction that will significantly reduce the final exponentiation costs compared to prior approaches operating under less favorable congruence conditions.
Using the Equation (3), the cost of the final exponentiation is given by

$$I_{27} + 8 \times M_{27} + C_{\text{cyc27}} + F_{27} + 2 \times E_{x-1} + 17 \times E_x = 2652M_1 + 2 \times E_{x-1} + 17 \times E_x.$$

For the exTNFS seed x_1: $l_1 = 23, hw = 4$, and the costs of exponentiation by x_1 and $x_1 - 1$ are as follows.

- $E_{x_1} = 23(252) + 3(216) = 6444M_1$.

- $E_{x_1-1} = 23(252) + 2(216) = 6228M_1$.

The total final exponentiation cost for x_1 is

$$2652M_1 + 2 \times E_{x-1} + 17 \times E_x = (2652 + 2(6228) + 17(6444))M_1 = 124656M_1.$$

For the SexTNFS seed x_2: we exploit the form of x_2 to write it as follows.

$$x_2 = 3^{31} \times 2^2 + 3^{27} + 3^{22} + 3^{10} + 1.$$

Thus, we get:

- $E_{x_2} = (31(252) + 2(153) + 4(216))M_1 = 8982M_1$.
- $E_{x_2-1} = (31(252) + 2(153) + 3(216))M_1 = 8766M_1$.

The total final exponentiation cost for x_1 is

$$2652M_1 + 2 \times E_{x-1} + 17 \times E_x = (2652 + 2(8766) + 17(8982))M_1 = 172878M_1.$$

We can summarize the total costs for the complete optimal Ate pairing computation as follows:

- exTNFS seed (x_1): the total cost is $34335 + 124656 \approx 158991M_1$.
- SexTNFS seed (x_2): the total cost is $48276 + 172878 = 221154M_1$.

5 Comparison

In this section, we compare the computational cost of the optimal Ate pairing on BLS27 using our ternary approach with the binary approach of Fouotsa et al. [9].

5.1 Complexity of binary approach (Fouotsa et al.)

In the binary approach by Fouotsa et al., the elliptic curve operations on the twist $E'(\mathbb{F}_{p^9})$ are performed using affine coordinates. In this setting, both the doubling (DBL) and addition (ADD) steps require a field inversion in $\mathbb{F}_{p^9}$. The cost of a single doubling step which includes the point doubling, tangent line evaluation, and the field update via squaring in $\mathbb{F}_{p^{27}}$ is given as:

$$C_{\text{dbl}} = 3M_9 + 2S_9 + I_9 + 9M_1 + M_{27} + S_{27} = 646M_1$$

The cost of a single addition step—which includes the mixed point addition, chord line evaluation, and the field update via multiplication in $\mathbb{F}_{p^{27}}$ is:

$$C_{\text{add}} = 3M_9 + 2S_9 + I_9 + 9M_1 + M_{27} = 493M_1$$

For the seed used by Fouotsa et al., $x_F = 2^{51} + 2^{42} + 2^{28} + 2^9 + 1$, the binary representation has a bit-length of $l = 52$ and a Hamming weight of $hw = 5$. Consequently, the binary Miller loop executes 51 doubling steps and 4 addition steps. The total complexity of the binary Miller loop is:

$$\text{Cost}_{\text{ML}} = (51 \cdot 646 + 4 \cdot 493)\, M_1 = 34918M_1.$$

The final exponentiation cost is expressed as:

$$\text{Cost}_{\text{FE}} = I_{27} + 9M_{27} + S_{27} + F_{27} + 2 \times E_{x_F-1} + 17 \times E_{x_F}.$$

Since $9 \nmid p - 1$ for x_F, the Frobenius maps cost $F_{27} = 2(18 + 114 + 686)M_1 = 1636M_1$. The exponentiation costs are

$$E_{x_F} = (51 \cdot 153 + 4 \cdot 216)\, M_1 = 8667M_1$$

and

$$E_{x_F-1} = (51 \cdot 153 + 3 \cdot 216)\, M_1 = 8451 M_1$$

. Substituting these values yields the following final exponentiation cost:

$$\text{Cost}_{\text{FE}} = (536 + 9(216) + 153 + 1636 + 2(8451) + 17(8667))\, M_1 = 168510 M_1.$$

Therefore, the total cost of the optimal Ate pairing using the binary representation is:

$$\text{Cost}_{\text{Total}} = 34918 M_1 + 168510 M_1 = 203428 M_1.$$

5.2 Comparison with Our Ternary Approach

Compared to the seed x_F of Fouotsa et al. [9], our exTNFS seed x_1 yields a gain of $44437 M_1$ (approximately 22% more efficient, reducing the total cost to $158991 M_1$).

Furthermore, our SexTNFS seed x_2 guarantees a higher level of security against SexTNFS attacks. Additionally, our seed satisfies the primality conditions for both parameters p and r, whereas we have verified that these parameters are not prime for the seed x_F. This demonstrates that our SexTNFS seed surpasses that of Fouotsa et al. in terms of both security and validity. This comparison is summarized in Table 1.

Table 1: Recap of the computational costs and security of the optimal Ate pairing on BLS27.

Method	**Total Cost**	**Gain vs. x_F**	**Security & Validity**
Fouotsa et al. (Binary)	$203428 M_1$	—	SexTNFS (p, r not prime)
Our Approach (exTNFS)	$158991 M_1$	$44437 M_1$ ($\approx 22\%$)	exTNFS (p, r prime)
Our Approach (SexTNFS)	$221154 M_1$	—	SexTNFS (p, r prime)

6 Conclusion

In this paper, we presented an efficient ternary framework for computing the optimal Ate pairing on BLS27 curves. By restricting the seed x to sparse ternary digits $\{0, 1\}$, we developed a faster Miller algorithm that uses cheap cubing in the $\mathbb{F}_{p^{27}}$ extension tower, and we rigorously proved its correctness. For the final exponentiation, we replaced standard binary methods with ternary exponentiation, using cyclotomic cubing and a new chaining trick to greatly reduce the computational cost.

Our two proposed seeds offer a clear trade-off between speed and security. The exTNFS seed makes the pairing computation 22% faster than the binary approach by Fouotsa et al., while ensuring that both p and r are prime—which the previous baseline seed failed to do. The SexTNFS seed is slower due to its larger size, but it provides the highest security against advanced SexTNFS attacks and also guarantees valid primes. Together, they show that our ternary method is both faster and more secure for BLS27 curves.

For future work, the pairing could be sped up further by improving basic arithmetic in $\mathbb{F}_{p^{27}}$, such as finding faster formulas for standard cubing and squaring. Additionally, finding new algebraic shortcuts to reduce the cost of the point tripling step in the Miller loop remains a promising direction.

References

[1] P. S. L. M. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In *Security in Communication Networks*, pages 257–267. Springer, 2002.

[2] R. Barbulescu and S. Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 32:1298–1336, 2019.

[3] L. Ben Abdelghani and W. Haddaji. Optimized final exponentiation for optimal ate pairings using cyclotomic cubing. *Cryptology ePrint Archive*, 2025.

[4] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[5] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, 2001.

[6] H. Cohen, A. Miyaji, and T. Ono. Efficient elliptic curve exponentiation using mixed coordinates. In *Advances in Cryptology – ASIACRYPT'98*, volume 1514 of *LNCS*, pages 51–65. Springer, 1998.

[7] C. Costello, T. Lange, and M. Naehrig. Faster pairing computations on curves with high-degree twists. In *Public Key Cryptography – PKC 2010*, pages 224–242. Springer, 2010.

[8] V. S. Dimitrov, L. Imbert, and P. K. Mishra. Efficient and secure elliptic curve point multiplication using double-base chains. In *ASIACRYPT 2008*, pages 59–78. Springer, 2008.

[9] E. Fouotsa, N. El Mrabet, and A. Pecha. Computing optimal ate pairings on elliptic curves with embedding degree 9, 15 and 27. *Journal of Groups, Complexity, Cryptology*, 12(1):3:1–3:25, 2020.

[10] L. Fuentes-Castañeda, E. Knapp, and F. Rodríguez-Henríquez. Faster hashing to G_2. In *Selected Areas in Cryptography*, pages 412–430. Springer, 2011.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pages 89–98. ACM, 2006.

[12] D. Hayashida, K. Hayasaka, and T. Teruya. Efficient final exponentiation via cyclotomic structure for pairings over families of elliptic curves. *Cryptology ePrint Archive*, 2020.

[13] V. S. Miller. The weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[14] Y. Nanjo, M. Shirase, T. Kusaka, and Y. Nogami. An explicit formula of cyclotomic cubing available for pairings on elliptic curves with embedding degrees of multiple of three. In *2020 35th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 288–292. IEEE, 2020.

[15] M. Scott, N. Benger, M. Charlemagne, L. J. Dominguez Perez, and E. J. Kachisa. On the final exponentiation for calculating pairings on ordinary elliptic curves. In *Pairing-Based Cryptography–Pairing 2009*, pages 78–88. Springer, 2009.

[16] F. Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.

[17] X. Zhang and D. Lin. Analysis of optimum pairing products at high security levels. In *Progress in Cryptology–INDOCRYPT 2012*, pages 412–430. Springer, 2012.