Optimizing Final Exponentiation for Pairing-Friendly Elliptic Curves with Odd Embedding Degrees Divisible by 3

Walid Haddaji[1,3*], Loubna Ghammam[2], Nadia El Mrabet[3], Leila Ben Abdelghani[4]

[1,3*]Science and technology for defense lab LR19DN01, Center for military research, military academy, Tunis, Tunisia.
[2]ITK Engineering GmbH, Im Speyerer Tal 6 Rülzheim76761Germany.
[3]Laboratory of Secure System and Architecture (SSA), Ecole des Mines de Saint Etienne, 880 Rte de Mimet, Campus Georges Charpak Provence, 13120, Gardanne, France.
[4]Laboratory of Analysis, Probability and Fractals, Faculty of Sciences, Environment Avenue, Omrane, 5000, Monastir, Tunisia.

*Corresponding author(s). E-mail(s): haddajiwalid95@gmail.com;
Contributing authors: loubna.ghammam@itk-engineering.de;
nadia.elmrabet@emse.fr; leila.benabdelghani@fsm.rnu.tn;

Abstract

In pairing-based cryptography, final exponentiation with a large fixed exponent is crucial for ensuring unique outputs in Tate and optimal Ate pairings. While optimizations for elliptic curves with even embedding degrees have been well-explored, progress for curves with odd embedding degrees, particularly those divisible by **3**, has been more limited. This paper presents new optimization techniques for computing the final exponentiation of the optimal Ate pairing on these curves. The first exploits the fact that some existing seeds have a form enabling cyclotomic cubing and extends this to generate new seeds with the same form. The second is to generate new seeds with sparse ternary representations, replacing squaring with cyclotomic cubing. The first technique improves efficiency by **1.7%** and **1.5%** compared to the square and multiply (**SM**) method for existing seeds at **192**-bit and **256**-bit security levels, respectively. For newly generated seeds, it achieves efficiency gains of **3.6%** at **128**-bit, **5%** at **192**-bit, and **8.5%**

at **256**-bit security levels. The second technique improves efficiency by **3.3%** at **128**-bit, **19.5%** at **192**-bit, and **4.3%** at **256**-bit security levels.

Keywords: Elliptic curves, pairings, final exponentiation, cyclotomic cubing, complexity

1 Introduction

Pairings on elliptic curves are crucial for various cryptographic applications, e.g., identity-based Encryption [1], short signatures[2], and tri-partite Diffie-Hellman [3]. Consequently, substantial efforts [4–6] have been made to develop several families of elliptic curves specifically optimized for pairing applications. Other works have focused on optimizing the Miller loop [7, 8] and the final exponentiation [9–12], as these operations account for nearly all of the computational complexity involved in pairings. Recently, Barbulescu and Duquesne presented, in [13], new parameters that resist an attack for a discrete logarithm problem (DLP) proposed by Kim et al.in [14]. Additionally, they demonstrated that at 128−security level Barreto-Lynn-Scott family of elliptic curves of embedding degree $k = 12$ ($BLS12$) and Kachisa-Schaefer-Scott family of elliptic curves with $k = 16$ ($KSS16$) can provide a more efficient pairing than the Barreto-Naehrig family (BN). Unexpectedly, Barbulescu et al. announced in [15] that other elliptic curve families with $k = 9, 15, 27$ can be competitive with $BLS12$, $KSS16$ and BN. There are two components to the pairing's final exponentiation: 'easy part' refers to the first, and 'hard part' to the second. The easy part's computation is thought to be somewhat straightforward, whereas the hard part demands a great deal of work. The hard part is computed in a cyclotomic subgroup where the efficiency of certain cyclotomic operations in the target field can be exploited. Specifically, the primary operation in the hard part calculation is exponentiation in a cyclotomic subgroup using a prefixed integer called the 'seed'. This operation employs the square-and-multiply routine (**SM**), which squares for each binary digit of the seed and multiplies when the digit is active (non-zero). If cyclotomic squaring [11] is available, it should be used instead of the typical squaring to improve efficiency. Thus, cyclotomic squaring plays a crucial role in enhancing the efficiency of the hard part of the final exponentiation in pairings on elliptic curves with even embedding degrees, such as $BLS12$, $KSS16$, and $BLS24$. However, this operation is not available for the curves with odd embedding degrees, such as $BLS9$, $BLS15$, and $BLS27$. Thus, the exponentiation by a given seed in the cyclotomic subgroup of the target fields of these curves is done using the square and multiply routine depending on the typical squaring and multiplication. Fortunately, Granger and Scott noted that the techniques developed in [11] for efficient cyclotomic squaring could also be adapted to create an alternative approach for elliptic curves with an odd embedding degree divisible by 3. We refer to this substitute as 'cyclotomic cubing'. Later, Nanjo et al. [16] provided an explicit formula for calculating cyclotomic cubing, stating that it is 30% more efficient than conventional cubing. Furthermore, we found that cyclotomic cubing outperforms the standard method of squaring followed by multiplication, particularly in the field

$\mathbb{F}_{p^{27}}$. By analyzing the structure of the hard part of the final exponentiation on BLS curves and the specific forms of certain seeds, we were inspired to apply cyclotomic cubing in a partial manner. This partial application allows us to retain the efficiency provided by the NAF representation [17], while further enhancing by the use of cyclotomic cubing. According to Nanjo et al. [16], the performance improvement from using cyclotomic cubing on the $BLS15$ curve is not enough to justify replacing the square-and-multiply (**SM**) method with a cubing and multiplication (**CM**) approach. Another approach worth considering is the application of **CM** to new seeds with sparse ternary representations. This motivated us to explore **CM** as an alternative to **SM** to leverage the efficiency of cyclotomic cubing in the final exponentiation of the optimal Ate pairing on $BLS15$ and $BLS27$ curves.

Our contribution

In this paper, we propose the following methods:

1. **Two Consecutive Active Bits (TCAB):** This method utilizes a special form of two consecutive active bits in certain seeds to perform cyclotomic cubing through simple factorization. It exploits the advantages of cyclotomic cubing while maintaining the efficiency of the NAF representation. New seeds tailored for this method will be generated, and results are compared with those obtained by applying **SM** to the same seeds, as well as with results from existing seeds.
2. **Exponentiation using Sparse Ternary Representation:** We generate new seeds with sparse ternary representation and apply the **CM** method, where squaring is replaced by cyclotomic cubing. The efficiency of **CM** is evaluated by comparing the results with those obtained from existing seeds using the **SM** method.

Although these methods apply to any elliptic curve with an odd embedding degree divisible by 3, we focus on the $BLS15$ curve at 128 and 192-bit security levels, and the $BLS27$ curve at 192 and 256-bit levels. We particularly emphasize $BLS27$, as it is well-suited for computing the Miller loop and pairing products.

Paper overview

This paper is structured as follows: Section 2 provides background on arithmetic in finite fields with an odd extension degree divisible by 3, along with an introduction to cyclotomic arithmetic, elliptic curves, and pairings. Section 3 introduces our main method, **TCAB**, which computes partial cyclotomic cubing by exploiting specific bits within the seed. This section also analyzes the seed forms suitable for **TCAB** and compares its computational complexity with the **SM** method, focusing on the $BLS15$ and $BLS27$ curves. Section 4 investigates the use of ternary seed representations for final exponentiation on the $BLS15$ and $BLS27$ curves, compares the efficiency of ternary and NAF representations, and explores the generation of new sparse ternary seeds. Finally, we summarise our findings and outline potential future research directions.

Notations

Let $k \in \mathbb{N}^*$. In the rest of this paper, we use the following notations:

- $p \geq 3$ is a prime number,

- E is an elliptic curve defined over $\mathbb{F}_p$,
- $\mathbf{M_k}$ stands for the cost of multiplication in $\mathbb{F}_{p^k}$,
- $\mathbf{S_k}$ denotes the cost of squaring in $\mathbb{F}_{p^k}$,
- $\mathbf{F_k}$ a Frobenius operation $\mathbb{F}_{p^k}$,
- $\mathbf{I_k}$ denotes the costs of inversion in $\mathbb{F}_{p^k}$,
- $\mathbf{C_{c_k}}$ is the cost of cyclotomic cubing in $\mathbb{F}_{p^k}$,
- $\mathbf{I_{c_k}}$ represents the of cyclotomic inversion in $\mathbb{F}_{p^k}$,
- Let $u \in \mathbb{Z}$, $\mathbf{E_u}$ denotes the cost of the exponentiation by u,
- **SL** denotes the security level.

2 Background

Consider the finite field $\mathbb{F}_{p^k}$, with $3|k$. This section offers the essential fundamentals of arithmetic over $\mathbb{F}_{p^k}$ with k being odd and divisible by 3, elliptic curves, and pairings on curves of embedding degree k. For further information, the reader is referred to [15, 16, 18–20].

2.1 Costs of arithmetic operations over $\mathbb{F}_{p^k}$

In the remainder of this paper, we will neglect additions and multiplications by constants in $\mathbb{F}_{p^k}$ when we compute the complexity. We also assume that $\mathbf{M_1} = \mathbf{S_1}$. We suppose that $3|k$ and represent $\mathbb{F}_{p^k}$ as follows:

$$\mathbb{F}_{p^k} = \mathbb{F}_{p^{\frac{k}{3}}}[x]/(x^3 - \xi),$$

where ξ is a cubic non-residue in $\mathbb{F}_{p^{\frac{k}{3}}}$. Using Karatsuba's method [21], the complexity of the **multiplication** in $\mathbb{F}_{p^k}$ is $6\mathbf{M}_{\frac{\mathbf{k}}{\mathbf{3}}}$. To compute the **squaring** in fields with extension degree 3, it is recommended to utilize the Chung-Hasan method [22] which costs $2\mathbf{M}_{\frac{\mathbf{k}}{\mathbf{3}}} + 3\mathbf{S}_{\frac{\mathbf{k}}{\mathbf{3}}}$. Let $i, \in \mathbb{N}^*$ such that $i < \varphi(k)$, where φ is the Euler's totient function. The **Frobenius operation** is an endomorphism defined over $\mathbb{F}_{p^k}$ as follows:

$$\begin{aligned} \pi_i : \mathbb{F}_{p^k} &\longrightarrow \mathbb{F}_{p^k} \\ \alpha &\longmapsto \alpha^{p^i}. \end{aligned}$$

For more details the reader is referred to reference [18]. The costs of Frobenius operations are given in Table 1. According to [23], the complexity of the **inversion** in $\mathbb{F}_{p^k}$ is $9\mathbf{M}_{\frac{\mathbf{k}}{\mathbf{3}}} + 3\mathbf{S}_{\frac{\mathbf{k}}{\mathbf{3}}} + \mathbf{I}_{\frac{\mathbf{k}}{\mathbf{3}}}$.
Before presenting cyclotomic cubing and inversion in $\mathbb{F}_{p^k}$, we first define the cyclotomic subgroup of this field.

Definition 1.
The cyclotomic subgroup of $\mathbb{F}_{p^k}$ is given in [24] by:

$$G_{\Phi_k(p)} = \{\alpha \in \mathbb{F}_{p^k}^*; \alpha^{\Phi_k(p)} = 1\}, \tag{1}$$

where ϕ_k is the $k-th$ cyclotomic polynomial. The order of $G_{\Phi_k(p)}$ is $\Phi_k(p)$.

Example 1.
$G_{\Phi_{27}(p)}$ is the cylotomic subgroup of $\mathbb{F}^*_{q^{27}}$ of order $\Phi_{27}(p) = p^{18} + p^9 + 1$.

Let $\alpha \in \mathbb{F}_{p^k}$. If $\alpha \in G_{\Phi_k(p)}$, the **inversion** of α can be inferred from the cyclotomic subgroup $\alpha \in G_{\Phi_k(p)}$'s membership relation. In this case, it is called cyclotomic inversion, which is frequently more efficient than the usual one. In the following example, we demonstrate how to compute the cyclotomic inversion in $\mathbb{F}_{p^{15}}$ and $\mathbb{F}_{p^{27}}$, along with the associated computational cost.

Example 2.

- *Let $\alpha \in G_{\Phi_{15}(p)}$, then $\alpha^{\Phi_{15}(p)} = 1$, which gives $\alpha^{-1} = \alpha^{p^{10}}\alpha^{p^5}$.*
- *Let $\alpha \in G_{\Phi_{27}(p)}$, then $\alpha^{\Phi_{27}(p)} = 1$, which gives $\alpha^{-1} = \alpha^{p^{18}}\alpha^{p^9}$.*

Based on the method of Fouotsa et al. [18], we have $I_{c_{15}} = 3 \times \mathbf{M_5} + 3 \times \mathbf{S_5}$ and $I_{c_{27}} = 3 \times \mathbf{M_9} + 3 \times \mathbf{S_9}$. Using the estimates of Aranha et al. [20], the cyclotomic inversion costs $78\mathbf{M_1}$ in $\mathbb{F}_{p^{15}}$ and $189\mathbf{M_1}$ in $\mathbb{F}_{p^{27}}$.

Similarly to cyclotomic inversion, the cubing of an element in $G_{\Phi_k(p)}$ can be computed taking advantage of the structure of this group. In this case, cubing is called cyclotomic cubing. The method of calculating this operation is detailed in [16]. As noted in [16], the cost of cyclotomic cubing is $4\mathbf{S}_{\frac{\mathbf{k}}{\mathbf{3}}} + 5\mathbf{M}_{\frac{\mathbf{k}}{\mathbf{3}}}$. This costs $117\mathbf{M_1}$ in $\mathbb{F}_{p^{15}}$ and $288\mathbf{M_1}$ in $\mathbb{F}_{p^{27}}$. Since we will apply our improvements to $BLS15$ and $BLS27$, we present, in Table 1, the costs of operations in $\mathbb{F}_{p^{15}}$ and $\mathbb{F}_{p^{27}}$ based on the estimates of Aranha et al. [20].

Fields	Operations	Costs
$\mathbb{F}_{p^{15}}$	Multiplication $\mathbf{M_{15}}$	$78\mathbf{M_1}$
	Squaring $\mathbf{S_{15}}$	$65\mathbf{M_1}$
	Inversion $\mathbf{I_{15}}$	$229\mathbf{M_1}$
	Fronenius $\mathbf{F_{15}}$	$14\mathbf{M_1}$
	Cyclotomic inversion $\mathbf{I_{c_{15}}}$	$78\mathbf{M_1}$
	Cyclotomic cubing $\mathbf{C_{c_{15}}}$	$117\mathbf{M_1}$
$\mathbb{F}_{p^{27}}$	Multiplication $\mathbf{M_{27}}$	$216\mathbf{M_1}$
	Squaring $\mathbf{S_{27}}$	$153\mathbf{M_1}$
	Inversion $\mathbf{I_{27}}$	$536\mathbf{M_1}$
	Fronenius $\mathbf{F_k}$	$26\mathbf{M_1}$
	Cyclotomic inversion $\mathbf{I_{c_{27}}}$	$189\mathbf{M_1}$
	Cyclotomic cubing $\mathbf{C_{c_{27}}}$	$288\mathbf{M_1}$

Table 1: The arithmetic operations costs in the fields $\mathbb{F}_{p^{15}}$ and $\mathbb{F}_{p^{27}}$ [20].

2.2 Pairings

Let E be an elliptic curve defined over $\mathbb{F}_p$. Let r be a large prime factor of $\#E(\mathbb{F}_p)$ and k be the smallest positive integer such that $r|(p^k - 1)$. Let $P \in E(\mathbb{F}_p)[r]$ of order r and $f_{r,P}$ be the rational function with the following divisor(See [19] for details about

divisors):

$$Div(f_{r,P}) = r(P) - r(P_\infty).$$

Let us consider the point $Q \in E(\mathbb{F}_{p^k})[r]$ of order r and let μ_r be the group of $r-th$ roots of unity of $\mathbb{F}^*_{p^k}$. Then, the reduced Tate pairing is a bilinear and non-degenerate map defined as follows:

$$\begin{aligned} e_t \colon E(\mathbb{F}_p)[r] \times E(\mathbb{F}_{p^k})[r] &\longrightarrow \mu_r \\ (P,Q) &\longmapsto f_{r,P}(Q)^{\frac{p^k-1}{r}}. \end{aligned}$$

The Ate pairing is a variant of Tate pairing defined as follows:

$$\begin{aligned} e_A \colon \mathbb{G}_2 \times \mathbb{G}_1 &\longrightarrow \mathbb{G}_3 \\ (Q,P) &\longmapsto f_{t-1,Q}(P)^{\frac{p^k-1}{r}}, \end{aligned}$$

where $\mathbb{G}_1 = E(\overline{\mathbb{F}_p})[r] \cap Ker(\pi_p - 1) = E(\mathbb{F}_p)[r]$, $\mathbb{G}_2 = E(\overline{\mathbb{F}_p})[r] \cap Ker(\pi_p - p)$, $\mathbb{G}_3 = \mu_r$, and $\pi_p : E(\overline{\mathbb{F}_p}) \to E(\overline{\mathbb{F}_p})$, $\quad \pi_p(x,y) = (x^p, y^p)$, $\quad$ be the Frobenius endomorphism on the curve E, and t is its trace. According to [7], a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_3$, with $\#\mathbb{G}_1 = \#\mathbb{G}_2 = \#\mathbb{G}_3 = r$, is said to be an optimal pairing if it can be computed in $\frac{log_2(r)}{\varphi(k)} + \epsilon(k)$ basic Miller iterations, with $\epsilon(k) \leq log_2(k)$. The computation of pairing is composed of two stages. The first consists of calculating the function $f_{.,.}(.)$ using the Miller algorithm [25]. The second is to raise the output of Miller's algorithm to the power of $\frac{p^k-1}{r}$. We refer to this stage as the final exponentiation. Its efficient calculation has become a significant undertaking. Recall that the exponent $\frac{p^k-1}{r}$ is split into the two parts as follows:

$$\frac{p^k-1}{r} = \frac{p^k-1}{\phi_k(p)} \frac{\phi_k(p)}{r},$$

where ϕ_k is the $k-th$ cyclotomic polynomial. Let A be the Miller's algorithm's output and $B = A^{\frac{p^k-1}{\phi_k(p)}}$. Then the pairing's output is $B^{\frac{\phi_k(p)}{r}}$. Since there are few multiplications, inversions, and Frobenius operations in $\mathbb{F}_{p^k}$, the computation of the first part B, is typically inexpensive and is called the 'easy part'. However, the computation of the second part, $B^{\frac{\phi_k(p)}{r}}$, is seen to be more challenging and is referred to as the 'hard part'. Several methods have been proposed for performing this calculation [9, 18, 26–28]. In particular, Zhang et al. [29] used a recursion relation to expand $\frac{\phi_k(p)}{r}$ in the base p and compute the hard part of the final exponentiation for $k = 27$. Hayashida et al.[30] generalized Zhang et al.'s method to arbitrary embedding degrees using the homogeneous cyclotomic polynomial constructed from cyclotomic polynomial.
In the next section, we will focus on calculating the optimal Ate pairing on elliptic curves with embedding degrees 15 and 27.

2.3 Optimal Ate pairing on $BLS15$ and $BLS27$

Generally, BLS curves are defined over $\mathbb{F}_p$ by the equation:

$$E : y^2 = x^3 + b,$$

but our focus here is specifically on the $BLS15$ and $BLS27$ curves.

The case of $BLS15$

$BLS15$ is a family of parametrized elliptic curves with embedding degree 15 defined in [31] by the following parameters:

$$\begin{cases} p = \frac{u^{12}-2u^{11}+u^{10}+u^7-2u^6+u^5+u^2+u+1}{3}, \\ r = u^8 - u^7 + u^5 - u^4 + u^3 - u + 1, \\ t = u + 1, \end{cases}$$

where u is an integer, known as seed, and it is chosen so that both p and r are integers and simultaneously primes. For each seed u, we denote the curve in question by E_u. As indicated in [18], the optimal Ate pairing on the curve E_u is given by:

$$\begin{aligned} e_o \colon \mathbb{G}_2 \times \mathbb{G}_1 &\longrightarrow \mathbb{G}_3 \\ (Q, P) &\longmapsto f_{u,Q}(P)^{\frac{p^{15}-1}{r}}, \end{aligned}$$

where $\mathbb{G}_1 = E_u(\overline{\mathbb{F}_p})[r] \cap Ker(\pi_p - 1) \subset E_u(\mathbb{F}_p)$, $\mathbb{G}_2 = E_u(\overline{\mathbb{F}_p})[r] \cap Ker(\pi_p - p) \subset E(\mathbb{F}_{p^{15}})$, $\mathbb{G}_3 = \mu_r \subset \mathbb{F}^*_{p^{15}}$. The value $f_{u,Q}(P)$ is calculated using the Miller algorithm [25]. Let A be the output of the Miller algorithm. The final exponentiation of the optimal Ate pairing consists of calculating $A^{\frac{p^{15}-1}{r}}$. According to [30], $\frac{p^{15}-1}{r}$ is split as follows:

$$\frac{p^{15}-1}{r} = (p^5 - 1)(p^2 + p + 1)\frac{\Phi_{15}(p)}{r}.$$

Therefore, $A^{\frac{p^{15}-1}{r}} = (A^{(p^5-1)(p^2+p+1)})^{\frac{\Phi_{15}(p)}{r}}$. The easy part of the final exponentiation for $BLS15$ lies in the computation of $B = A^{(p^5-1)(p^2+p+1)}$, while the hard part involves calculating $B^{\frac{\Phi_{15}(p)}{r}}$. Hayashida et al. indicate, in [30], that the exponent of the hard part can be parameterized as $3.\frac{\Phi_{15}(p)}{r}$. This exponent is given, in [30], as follows:

$$3.\frac{\Phi_{15}(p)}{r} = (u-1)^2(u^2 + u + 1) + \sum_{i=0}^{7} \lambda_i(u)p^i(u) + 3,$$

where $\lambda_7 = 1$, $\lambda_6 = u\lambda_7 - 1$, $\lambda_5 = u\lambda_6$, $\lambda_4 = u\lambda_5 + 1$, $\lambda_3 = u\lambda_4 - 1$, $\lambda_2 = u\lambda_3 + 1$, $\lambda_1 = u\lambda_2$, and $\lambda_0 = u\lambda_1 - 1$. To express the complexity of the final exponentiation on $BLS15$, we distinguish two cases. The first is when we apply cyclotomic cubing, exploiting the term 3 in the hard part decomposition. In this case, we express this

complexity as follows:

$$I_{15} + 18 \times \mathbf{M_{15}} + \mathbf{C_{c_{15}}} + 10 \times \mathbf{F_{15}} + I_{c_{15}} + 2 \times \mathbf{E_{u-1}} + 9 \times \mathbf{E_u}. \quad (2)$$

We use this expression to support the complexity of our subsequent methods, which will leverage cyclotomic cubes. The second case considers the computation without cyclotomic cubing, based on squaring. Here, the complexity is:

$$I_{15} + 19 \times \mathbf{M_{15}} + \mathbf{S_{15}} + 10 \times \mathbf{F_{15}} + I_{c_{15}} + 2 \times \mathbf{E_{u-1}} + 9 \times \mathbf{E_u}. \quad (3)$$

The case of $BLS27$

$BLS27$ is a family of parametrized elliptic curves with embedding degree 27 given in [29] by the following parameters:

$$\begin{cases} r(u) = \frac{u^{18}+u^9+1}{3}, \\ p(u) = (u-1)^2 r(u) + u, \\ t(u) = u + 1, \end{cases}$$

where u is a seed chosen in the same manner as $BLS15$. For each seed u, we denote the curve in question by E_u. According to [18], the optimal Ate pairing on the curve E_u is given by:

$$\begin{aligned} e_o \colon \mathbb{G}_2 \times \mathbb{G}_1 &\longrightarrow \mathbb{G}_3 \\ (Q, P) &\longmapsto f_{u,Q}(P)^{\frac{p^{27}-1}{r}}, \end{aligned}$$

where $\mathbb{G}_1 = E_u(\overline{\mathbb{F}_p})[r] \cap Ker(\pi_p - 1) \subset E_u(\mathbb{F}_p)$, $\mathbb{G}_2 = E_u(\overline{\mathbb{F}_p})[r] \cap Ker(\pi_p - p) \subset E(\mathbb{F}_{p^{27}})$, $\mathbb{G}_3 = \mu_r \subset \mathbb{F}^*_{p^{27}}$. The value $f_{u,Q}(P)$ is computed using the Miller algorithm [25]. Let A be the output of the Miller algorithm. The final exponentiation of the optimal Ate pairing consists of calculating $A^{\frac{p^{27}-1}{r}}$. The exponent $\frac{p^{27}-1}{r}$ is given in [30] by:

$$\frac{p^{27}-1}{r} = (p^9 - 1)\frac{\Phi_{27}(p)}{r}.$$

Thus, $A^{\frac{p^{27}-1}{r}} = (A^{(p^9-1)})^{\frac{\Phi_{27}(p)}{r}}$. The easy part of the final exponentiation for $BLS27$ represents the computation of $B = A^{(p^9-1)}$, whereas the hard part involves calculating $B^{\frac{\Phi_{27}(p)}{r}}$. As given in [30], $\frac{\Phi_{27}(p)}{r}$ is decomposed as follows:

$$(x-1)^2(x^2 + px + p^2)(x^6 + p^3x^3 + p^6)(x^9 + p^9 + 1) + 3.$$

Using this decomposition, and similarly to $BLS15$, the complexity of the final exponentiation of the optimal Ate pairing on $BLS27$ is:

$$I_{27} + 8 \times \mathbf{M_{27}} + \mathbf{C_{c_{27}}} + 6 \times \mathbf{F_{27}} + 2 \times \mathbf{E_{u-1}} + 17 \times \mathbf{E_u} \quad (4)$$

if cyclotomic cubing is considered. Otherwise, it is given by:

$$I_{27} + 9 \times \mathbf{M_{27}} + \mathbf{S_{27}} + 6 \times \mathbf{F_{27}} + 2 \times \mathbf{E_{u-1}} + 17 \times \mathbf{E_u}. \tag{5}$$

3 Two Consecutive Active Bits(TCAB)

We observed that exponentiation of an element in the cyclotomic subgroup of $\mathbb{F}_{p^k}$ by an integer u, referred to as the 'seed', is the dominant operation in the hard part of pairing final exponentiation for the BLS family. Specifically, curves with odd embedding degrees divisible by 3 are hindered by the lack of cyclotomic squaring. Despite this limitation, existing works employ the **SM** method for this exponentiation. Fortunately, cyclotomic cubing is available and allows us to leverage its efficiency. In this context, we have noticed that we can find two consecutive active bits in the NAF representation of u with a particular form ((6)) that allows a partial use of the cyclotomic cubing. This section will explore scenarios where **TCAB** improves exponentiation by u in the cyclotomic subgroup. We will examine existing 'seeds' and generate new ones for optimal cases. In both cases, the application of **TCAB** maintains the efficiency of the **SM** method while leveraging the cyclotomic cubing's superiority over the simultaneous use of squaring and multiplication.

3.1 Description

Recall that for k odd integer and divisible by 3, the field $\mathbb{F}_{p^k}$ is represented as follows:

$$\mathbb{F}_{p^k} = \mathbb{F}_{p^{\frac{k}{3}}}[x]/(x^3 - \xi)$$

where ξ is a cubic non-residue in $\mathbb{F}_{p^{\frac{k}{3}}}$. Let E be the BLS curve with embedding k and $u \in \mathbb{Z}$ be the seed from which all the parameters of E are constructed. The NAF representation of u is given as follows:

$$signed(u) = (b_{l-1} \cdots b_1 b_0)_2,$$

where $l \in \mathbb{N}^*$ is the length of u and $b_0, b_1, \cdots, b_{l-2} \in \{-1, 0, 1\}, b_{l-1} \neq 0$. We denote by h the Hamming weight of u. The bits of u that are not equal to zero are called 'active bits'. Let $b_{i_1}, b_{i_2}, \cdots, b_{i_h}$ be the active bits in the NAF representation of u, where $0 \leq i_1 < i_2 < \cdots < i_{h-1} < i_h \leq l-1$. In the remainder of this section, we fix $1 < j \leq h$ and $\alpha \in G_{\Phi_k(p)} \subset \mathbb{F}_{p^k}$. We write α^u in the following form:

$$\alpha^u = \alpha^{b_{i_1}2^{i_1} + \cdots + b_{i_{j-2}}2^{i_{j-2}}} (\alpha^{\mathbf{2^{i_{j-1}}}})^{(\mathbf{b_{i_{j-1}}} + \mathbf{b_{i_j}} \mathbf{2^{i_j - i_{j-1}}})} \alpha^{b_{i_{j+2}}2^{i_{j+2}} + \cdots + b_{i_h}2^{i_h}}$$

We want to establish a condition at the level of the $j-th$ bit that allows the execution of some cyclotomic cubings. The desired condition is specified as follows:

$$b_{i_{j-1}} + b_{i_j} 2^{i_j - i_{j-1}} = \pm 3^{c_j}, \tag{6}$$

where $c_j \in \mathbb{N}^*$. We use the following theorem to determine the possible pairs taken by $(c_j, i_j - i_{j-1})$.

Theorem 1. *[32, 33]*
The only solutions of the equation

$$x^a - y^b = 1$$

in integers $a, b \geq 2$ and nonzero integers x, y are given by $a = 2, b = 3$ and $x = \pm 3$, $y = 2$.

If the property (6) is verified, we examine all possible values for $b_{i_{j-1}}$ and b_{i_j}, and apply the Theorem 1 to conclude that the only pairs taken by $(c_j, i_j - i_{j-1})$ are $\{(1,1),(1,2),(2,3)\}$. Furthermore, α^u takes the following form:

$$\alpha^u = \alpha^{b_{i_1} 2^{i_1} + \cdots + b_{i_{j-2}} 2^{i_{j-2}}} (\alpha^{\mathbf{2^{i_j - 1}}})^{\pm \mathbf{3^{c_j}}} \alpha^{b_{i_{j+2}} 2^{i_{j+2}} + \cdots + b_{i_h} 2^{i_h}},$$

This form allows for a combination of the usual calculation using the **SM** method and the computation of c_j cyclotomic cubing. In this case, **TCAB** method is said to apply to the seed u at position j.

3.2 Complexity

Assuming that condition (6) holds, we introduce some notations to analyze the complexity of the **TCAB** method. For $g \in \{2, \cdots, h\}$, we define g^- by:

$$g^- = \begin{cases} 1 \text{ if } \exists i \in \{i_1, \cdots, i_h\} \setminus \{i_{g-1}, i_g\}; b_i = -1 \\ 0 \text{ otherwise.} \end{cases}$$

Specifically, we use g^- to indicate whether there is a negative bit outside the positions g and $g-1$. This helps us to determine whether a cyclotomic inversion is needed. Let s_j be given by:

$$s_j = \begin{cases} 0 \text{ if } b_{i_{j-1}} + b_{i_j} 2^{i_j - i_{j-1}} = 3^{c_j} \\ 1 \text{ if } b_{i_{j-1}} + b_{i_j} 2^{i_j - i_{j-1}} = -3^{c_j}. \end{cases}$$

To compute α^u, the **TCAB** method needs the following complexity:

$$[\mathbf{1}_{\{j \neq h\}}(j) i_h + \mathbf{1}_{\{h\}}(j) i_{h-1}]\mathbf{S_k} + (h-2)\mathbf{M_k} + \max\{j^-, s_j\}\mathbf{I_{c_k}} + c_j \mathbf{C_{c_k}},$$

where $\mathbf{1}_A$ represents the indicator function over a set of integers A. For the same goal, the **SM** method requires:

$$i_h \mathbf{S_k} + (h-1)\mathbf{M_k} + \max\{(h-2)^-, h^-\}\mathbf{I_{c_k}}.$$

The gain of **TCAB** compared to **SM**, is:

$$[i_h - \mathbf{1}_{\{j \neq h\}}(j) i_h - \mathbf{1}_{\{h\}}(j) i_{h-1}]\mathbf{S_k} + 1\mathbf{M_k} + [\max\{(h-2)^-, h^-\} - \max\{j^-, s_j\}]\mathbf{I_{c_k}} - c_j \mathbf{C_{c_k}}.$$

Remark 1. *(Multi-positions application)*
*We suppose that it is possible to apply the method **TCAB** to a given seed u n times,*

with $n \in \mathbb{N}^*$. Let $J = \{j_1, j_2, \cdots, j_n\} \subset \{2, 3, \cdots, h\}$ be the set of positions where we can apply this method. Then the total complexity of this application is given by:

$$[\mathbf{1}_{\{j_1 \neq h\}}(j_1)i_h + \mathbf{1}_{\{h\}}(j_1)i_{h-1}]\mathbf{S_k} + (h-1-n)\mathbf{M_k} + [\max\{{j_1}^-, s_{j_1}\}]\mathbf{I_{c_k}} + \sum_{r=1}^{n} c_{j_r}\mathbf{C_{c_k}},$$

3.3 Application

In this section, we study the **TCAB** method on the curves $BLS15$ and $BLS27$ based on the placement of positions within the seed u where **TCAB** is applicable: at the beginning, in the middle, and at the end. We will use existing seeds for analysis. Then, we interpret the optimal scenario. Finally, we propose new seeds for **TCAB** use, offering comparisons to highlight the efficiency compared to the **SM** method. Throughout this section, we will use Table 1 to estimate the costs.

3.3.1 The case of $BLS15$

Assume that $\mathbb{F}_{p^{15}}$ is represented by:

$$\mathbb{F}_{p^{15}} = \mathbb{F}_{p^5}[x]/(x^3 - \xi_5),$$

where ξ_5 is a cubic non-residue in $\mathbb{F}_{p^5}$. Let $\alpha \in G_{\Phi_{15}(p)} \subset \mathbb{F}_{p^{15}}$ and $u \in \mathbb{Z}$. We apply the **TCAB** method for $BLS15$ at the middle and end of some existing seeds, as no seeds in the literature were found where **TCAB** is applicable at the beginning.

TCAB in the middle of the seed:

Let $u_{m_{15}}$ be the seed proposed in [18], given by:

$$u_{m_{15}} = 1 + \mathbf{2^8} + \mathbf{2^9} + 2^{41} + 2^{48},$$

and matches the 192−bit security level. We write $\alpha^{u_{m_{15}}}$

$$\alpha^{u_{m_{15}}} = \alpha(\alpha^{\mathbf{2^8}})^{\mathbf{3}}\alpha^{2^{41}+2^{48}}.$$

The complexity of the exponentiation by $u_{m_{15}}$ in $G_{\phi_{15}(p)}$ using **TCAB** and **SM** is given in Table 2. We give, in Table 3, the cost of exponentiation in $G_{\phi_{15}(p)}$ by the

Method	**Complexity**
TCAB	$3\mathbf{M_{15}} + 48\mathbf{S_{15}} + \mathbf{C_{c_{15}}}$
SM	$4\mathbf{M_{15}} + 48\mathbf{S_{15}}$

Table 2: The complexity of exponentiation in $G_{\Phi_{15}(p)}$ by $u_{m_{15}}$ using **SM** and **TCAB**.

application of **TCAB** and **SM** to $u_{m_{15}}$ and provide the possible gain. From Table

Method	Cost	Gain
TCAB	$3471\mathbf{M_1}$	$-39\mathbf{M_1}$
SM	$3432\mathbf{M_1}$	

Table 3: Comparison of costs of one exponentiation by $u_{m_{15}}$ in $G_{\Phi_{15}(p)}$ using **SM** and **TCAB**.

3, we remark that replacing **TCAB** by **SM** is not advantageous in the middle of the seed.

TCAB at the end of the seed:

Let $u_{e_{15}}$ be the seed proposed in [20], given by:

$$u_{e_{15}} = 2^6 + 2^{59} + 2^{62} + \mathbf{2^{73}} + \mathbf{2^{74}},$$

and matches the 128−bit security level. We put $\alpha^{u_{e_{15}}}$ in the following form:

$$\alpha^{u_{e_{15}}} = \alpha^{2^6+2^{59}+2^{62}}(\alpha^{\mathbf{2^{73}}})^{\mathbf{3}}.$$

For this seed, the complexity of **TCAB** is $3\mathbf{M_{15}} + 73\mathbf{S_{15}} + \mathbf{C_{c_{15}}}$, while **SM** costs $4\mathbf{M_{15}} + 74\mathbf{S_{15}}$. Since $u_{e_{15}}$ is even and its NAF representation does not contain any negative digit, the exponentiation in $G_{\phi_{15}(p)}$ by $u_{e_{15}} - 1$ requires more $\mathbf{M_{15}}$ and $I_{c_{15}}$ than that by $u_{e_{15}}$. The cost of **TCAB** is $5096\mathbf{M_1}$, whereas the cost of **SM** is $5122\mathbf{M_1}$. Consequently, the gain provided by the application of **TCAB** instead of **SM** is $26\mathbf{M_1}$. For this reason, we extend the comparison to the final exponentiation. Using the expressions (2) and (3), we compute this complexity and give it in Table 4.

Method	Complexity
TCAB	$\mathbf{I_{15}} + 53\mathbf{M_{15}} + 803\mathbf{S_{15}} + 12\mathbf{C_{c_{15}}} + 3\mathbf{I_{c_{15}}} + 10\mathbf{F_{15}}$
SM	$\mathbf{I_{15}} + 65\mathbf{M_{15}} + 815\mathbf{S_{15}} + 3\mathbf{I_{c_{15}}} + 10\mathbf{F_{15}}$

Table 4: The complexity of the final exponentiation applying **SM** and **TCAB** to $u_{e_{15}}$.

Applying **TCAB** to the seed $u_{e_{15}}$, using the Table 1 and the expression (2), the final exponentiation of the optimal Ate pairing on $BLS15$ costs:

$$229\mathbf{M_1}+53\times(78\mathbf{M_1})+803\times(65\mathbf{M_1})+12\times(117\mathbf{M_1})+3\times(78\mathbf{M_1})+10\times(14\mathbf{M_1}) = 58336\mathbf{M_1}.$$

By applying **SM** to the same seed, the final exponentiation complexity is:

$$229\mathbf{M_1} + 65 \times (78\mathbf{M_1}) + 815 \times (65\mathbf{M_1}) + 3 \times (78\mathbf{M_1}) + 10 \times (14\mathbf{M_1}) = 58648\mathbf{M_1}.$$

We give these costs in Table 5 and provide the gain of using **TCAB** instead of **SM**.

Method	**Cost**	**Gain**
TCAB	$58336\mathbf{M_1}$	$312\mathbf{M_1}$
SM	$58648\mathbf{M_1}$	

Table 5: Comparison of costs of the final exponentiation on BLS applying **SM** and **TCAB** to $u_{e_{15}}$.

3.3.2 The case of $BLS27$

Let $\mathbb{F}_{p^{27}}$ be represented as follows:

$$\mathbb{F}_{p^{27}} = \mathbb{F}_{p^9}[x]/(x^3 - \xi),$$

where ξ is a cubic non-residue in $\mathbb{F}_{p^9}$. Let $\alpha \in G_{\Phi_{27}(p)} \subset \mathbb{F}_{p^{27}}$ and $u \in \mathbb{Z}$. We will apply the **TCAB** method in three situations: at the beginning, middle, and end of the given seed u.

Application of TCAB at the beginning of the seed:
Let $u_{b_{27}}$ be the seed proposed in [15], given by:

$$u_{b_{27}} = \mathbf{2^3} + \mathbf{2^4} + 2^{11} + 2^{15},$$

and matches the 128−bit security level. Then, we have:

$$\alpha^{u_{b_{27}}} = (\alpha^{\mathbf{2^3}})^{\mathbf{3}} \alpha^{2^{11}+2^{15}}.$$

The complexity of the exponentiation by $u_{b_{27}}$ in $G_{\phi_{27}(p)}$ using **TCAB** and **SM** is given in Table 6. We give, in Table 7, the cost of exponentiation in $G_{\phi_{27}(p)}$ by the application of **TCAB** and **SM** to $u_{b_{27}}$ and provide the possible gain. From Table 7, we remark that replacing **TCAB** by **SM** is not advantageous in the middle of the seed. In the current situation, **TCAB** shows no advantage over **SM**.

Application of TCAB in the middle of the seed:
Let $u_{m_{27}}$ be the seed proposed in [15], given by:

$$u_{m_{27}} = \mathbf{1} + \mathbf{2} + 2^4 + \mathbf{2^6} + \mathbf{2^7} + \mathbf{2^9} + \mathbf{2^{10}} + 2^{12} + 2^{29},$$

Method	Complexity
TCAB	$2\mathbf{M_{27}} + 15\mathbf{S_{27}} + \mathbf{C_{c_{27}}}$
SM	$3\mathbf{M_{27}} + 15\mathbf{S_{27}}$

Table 6: The complexity of exponentiation in $G_{\Phi_{27}(p)}$ by $u_{b_{27}}$ using **SM** and **TCAB**.

Method	Cost	Gain
TCAB	$3015\mathbf{M_1}$	$-72\mathbf{M_1}$
SM	$2943\mathbf{M_1}$	

Table 7: The costs of one exponentiation by $u_{b_{27}}$ in $G_{\Phi_{27}(p)}$ using **SM** and **TCAB**.

and matches the 256−bit security level. It should be noted that **TCAB** is applied three times to the middle of this seed. We can express $\alpha^{u_{m_{27}}}$ as:

$$\alpha^{u_{m_{27}}} = \alpha^{\mathbf{3}}\alpha^{2^4}(\alpha^{2^6})^3(\alpha^{2^9})^3\alpha^{2^{12}+2^{29}},$$

Table 8 contains the complexity of the exponentiation by $u_{m_{27}}$ in $G_{\phi_{27}(p)}$ using **TCAB** and **SM**. In Table 9, we give the cost of exponentiation in $G_{\phi_{27}(p)}$ applying **TCAB**

Method	Complexity
TCAB	$5\mathbf{M_{27}} + 29\mathbf{S_{27}} + 3\mathbf{C_{c_{27}}}$
SM	$8\mathbf{M_{27}} + 29\mathbf{S_{27}}$

Table 8: The complexity of exponentiation in $G_{\Phi_{27}(p)}$ by $u_{m_{27}}$ using **SM** and **TCAB**.

and **SM** to $u_{m_{27}}$ and provide the possible gain. In the current situation, **TCAB** offers no advantage over **SM**.

Application of TCAB at the end of the seed:

Since we expect a positive gain in this situation and the utility of BLS27 at 192- and 256-bit security levels, we present two existing seeds for these levels. However, no valid seed has been found for the 128-bit security level.

Method	Cost	Gain
TCAB	$6381\mathbf{M_1}$	$-216\mathbf{M_1}$
SM	$6165\mathbf{M_1}$	

Table 9: The costs of one exponentiation by $u_{m_{27}}$ in $G_{\Phi_{27}(p)}$ using **SM** and **TCAB**.

- **At** $192-$**bit security level:**
  Let $u_{e_{27.192}}$ be the seed proposed in [15], given by:

$$u_{e_{27.192}} = -2^5 + 2^8 + 2^{12} + 2^{16} + \mathbf{2^{21}} + \mathbf{2^{22}}.$$

  Thus, $\alpha^{u_{e_{27.192}}}$ can be expressed as:

$$\alpha^{u_{e_{27.192}}} = \alpha^{-2^5+2^8+2^{12}+2^{16}}(\alpha^{\mathbf{2^{21}}})^{\mathbf{3}}.$$

  The complexity for **TCAB** is $4\mathbf{M_{27}} + 21\mathbf{S_{27}} + \mathbf{C_{c_{27}}} + \mathbf{I_{c_{27}}}$, while for **SM** it is $5\mathbf{M_{27}} + 22\mathbf{S_{27}} + \mathbf{I_{c_{27}}}$. As $u_{e_{27.192}}$ is even and its NAF representation contains a negative digit, the exponentiation in $G_{\phi_{27}(p)}$ by $u_{e_{27.192}} - 1$ requires more $\mathbf{M_{27}}$ than that by $u_{e_{27.192}}$. The costs of exponentiation by $u_{e_{27.192}}$ in $G_{\Phi_{27}(p)}$ using **TCAB** and **SM** are $4554\mathbf{M_1}$ and $4635\mathbf{M_1}$, respectively. The gain from **TCAB** over **SM** is $81\mathbf{M_1}$. Since this gain is positive, we can proceed to the final exponentiation for comparison. Using the expressions (4) and (5), this complexity of is given in Table 10. Applying **TCAB** to the seed $u_{e_{27.192}}$, using the Table 1 and the expression (4),

Method	Complexity
TCAB	$\mathbf{I_{27}} + 86\mathbf{M_{27}} + 399\mathbf{S_{27}} + 20\mathbf{C_{c_{27}}} + 19\mathbf{I_{c_{27}}} + 6\mathbf{F_{27}}$
SM	$\mathbf{I_{27}} + 106\mathbf{M_{27}} + 419\mathbf{S_{27}} + 19\mathbf{I_{c_{27}}} + 6\mathbf{F_{27}}$

Table 10: The complexity of the final exponentiation applying **SM** and **TCAB** to $u_{e_{27.192}}$.

the final exponentiation of the optimal Ate pairing on $BLS27$ costs:

$536\mathbf{M_1}+86\times(216\mathbf{M_1})+399\times(153\mathbf{M_1})+20\times(288\mathbf{M_1})+19\times(189\mathbf{M_1})+6\times(26\mathbf{M_1}) = 89666\mathbf{M_1}$.

By applying **SM** to the same seed, the final exponentiation complexity is:

$536\mathbf{M_1} + 106 \times (216\mathbf{M_1}) + 419 \times (153\mathbf{M_1}) + 19 \times (189\mathbf{M_1}) + 6 \times (26\mathbf{M_1}) = 91286\mathbf{M_1}$.

Those costs are given in Table 11 with the gain of using **TCAB** instead of **SM**.

Method	Cost	Gain
TCAB	$89666\mathbf{M_1}$	$1620\mathbf{M_1}$
SM	$91286\mathbf{M_1}$	

Table 11: Comparison of costs of the final exponentiation on $BLS27$ applying **SM** and **TCAB** to $u_{e_{27.192}}$.

- **At** $256-$**bit security level:**
  Let $u_{e_{27.256}}$ be the seed proposed in [29], given by:

$$u_{e_{27.256}} = -2^3 + 2^8 + 2^{25} + 2^{27} + 2^{28},$$

and matches the $256-$bit security level. Let $\alpha \in G_{\Phi_{27}(p)} \subset \mathbb{F}_{p^{27}}$. Then, we have:

$$\alpha^{u_{e_{27.256}}} = \alpha^{-2^3+2^8+2^{25}}(\alpha^{\mathbf{2^{27}}})^{\mathbf{3}},$$

The complexity of exponentiation by $u_{e_{27.256}}$ in $G_{\Phi_{27}(p)}$ using **TCAB** is $3\mathbf{M_{27}} + 27\mathbf{S_{27}} + \mathbf{C_{c_{27}}} + \mathbf{I_{c_{27}}}$, while it is $4\mathbf{M_{27}} + 28\mathbf{S_{27}} + \mathbf{I_{c_{27}}}$ using **SM**. The exponentiation in $G_{\phi_{27}(p)}$ by $u_{e_{27.256}} - 1$ requires more $\mathbf{M_{27}}$ than that by $u_{e_{27.256}}$ because $u_{e_{27.256}}$ is even and its NAF representation contains a negative digit. The costs of **TCAB** and **SM** are $5256\mathbf{M_1}$ and $5337\mathbf{M_1}$, respectively. We use the expressions (4) and (5) to give the complexity of the final exponentiation in Table 12. Applying **TCAB** to

Method	Complexity
TCAB	$\mathbf{I_{27}} + 67\mathbf{M_{27}} + 513\mathbf{S_{27}} + 20\mathbf{C_{c_{27}}} + 19\mathbf{I_{c_{27}}} + 6\mathbf{F_{27}}$
SM	$\mathbf{I_{27}} + 87\mathbf{M_{27}} + 533\mathbf{S_{27}} + 19\mathbf{I_{c_{27}}} + 6\mathbf{F_{27}}$

Table 12: The complexity of the final exponentiation applying **SM** and **TCAB** to $u_{e_{27.256}}$.

the seed $u_{e_{27.192}}$, using the Table 1 and the expression (4), the final exponentiation of the optimal Ate pairing on $BLS27$ costs:

$$536\mathbf{M_1}+67\times(216\mathbf{M_1})+513\times(153\mathbf{M_1})+20\times(288\mathbf{M_1})+19\times(189\mathbf{M_1})+6\times(26\mathbf{M_1}) = 103004\mathbf{M_1}.$$

By applying **SM** to the same seed, the final exponentiation complexity is:

$$536\mathbf{M_1} + 87\times(216\mathbf{M_1}) + 533\times(153\mathbf{M_1}) + 19\times(189\mathbf{M_1}) + 6\times(26\mathbf{M_1}) = 104624\mathbf{M_1}.$$

Those costs are given in Table 13 with the gain of using **TCAB** instead of **SM**.

Method	Cost	Gain
TCAB	$103004\mathbf{M_1}$	$1620\mathbf{M_1}$
SM	$104624\mathbf{M_1}$	

Table 13: Comparison of costs of the final exponentiation on $BLS27$ applying **SM** and **TCAB** to $u_{e_{27.256}}$.

3.4 Optimal Gain Analysis

In this section, we evaluate the efficiency of **TCAB** based on its application positions. Although our analysis focuses on the curves $BLS27$ and $BLS15$, the insights are also relevant for other curves with an odd embedding degree divisible by 3. Assuming also that $\mathbf{M_1} = \mathbf{S_1}$, based on all previous assumptions, using the same notation as in **TCAB** description, and assuming $k \in \{15, 27\}$, we present the following possible cases.

1. We suppose that we can apply **TCAB** to a given seed u over the positions set $J = \{j_1, j_2, \cdots, j_n\}$, such that $h \notin J$. Here, **SM** and **TCAB** need the same squaring number. The gain of **TCAB** compared to **SM** is $n\mathbf{M_k} - (\sum_{r=1}^{n} c_{j_r})\mathbf{C_{c_k}}$. For this gain to be positive, we require that $\sum_{r=1}^{n} c_{j_r} < \frac{3}{4}n$ for $BLS27$, and $\sum_{r=1}^{n} c_{j_r} < \frac{2}{3}n$ for $BLS15$. These conditions are not feasible because $\sum_{r=1}^{n} c_{j_r} \geq n > 1$. In this case, **we can not benefit from applying TCAB**,
2. Let us suppose that we can perform the first case and apply **TCAB** at the end of the seed u. In this case, **SM** need more squaring than **TCAB** The gain of **TCAB** is
$$(n+1)\mathbf{M_k} + (i_h - i_{h-1})\mathbf{S_k} - (c_h + \sum_{r=1}^{n} c_{j_r})\mathbf{C_{c_k}}.$$
This gain **can exceed** $27M_1$ for $BLS27$, but **it cannot be positive** for $BLS15$.
3. Suppose that we can apply **TCAB** only at the end of the seed u. In this case, the possible positive gains are $81\mathbf{M_1}$ and $99\mathbf{M_1}$ for $BLS27$, and $26\mathbf{M_1}$ and $39\mathbf{M_1}$ for $BLS15$.

We conclude that optimal gain is achieved by applying the **TCAB** method at the end of the seed.

3.5 Optimal new seeds

We focus on the optimal case mentioned above to generate new seeds valid for **TCAB** on the curves BLS27 and BLS15. For this goal, we follow Barbulescu and Duquesne's recommendations [13] for discrete logarithm computation. We also considered the size of the seeds produced in [15, 18] for the curves BLS27 and BLS15.

3.5.1 New seed for TCAB at the 128−bit security level

Case of $k = 15$

Based on Barbulescu and Duquesne's recommendations [13], we generated the seed $u = 2+2^{12}+2^{26}+2^{28}+2^{29}$. This results in a prime p with 355 bits and a prime r with 238 bits. The complexity of exponentiation in $G_{\Phi_{15}(p)}$ by the seed u provided above using **TCAB** is $3\mathbf{M_{15}}+28\mathbf{S_{15}}+\mathbf{C_{c_{15}}}$. Using the estimates mentioned in Table 1, this exponentiation costs $2171\mathbf{M_1}$. Note that the exponentiation in $G_{\Phi_{15}(p)}$ by $u-1$ has the same cost. Using expression (2) (where complexity is based on cyclotomic cubing), we apply **TCAB** to the seed u, showing that the final exponentiation of the optimal ate pairing on the curve $BLS15$ has the following cost:

$$\begin{aligned} 229\mathbf{M_1} &+ 18\times(78\mathbf{M_1}) + 117\mathbf{M_1} \\ &+ 10\times(14\mathbf{M_1}) + 78\mathbf{M_1} \\ &+ 2\times(2171\mathbf{M_1}) + 9\times(2171\mathbf{M_1}) \\ &= 25849\mathbf{M_1} \end{aligned}$$

The case of $k = 27$

We successfully generated the seed $u = 2+2^9+2^{12}+2^{15}$, resulting in a prime p of 303 bits and a prime r of 272 bits. Using this seed, the complexity of the exponentiation in $G_{\Phi_{27}(p)}$ using **TCAB** is $2\mathbf{M_{27}}+12\mathbf{S_{27}}+2\mathbf{C_{c_{27}}}$. For this seed, the exponentiation by u or $u-1$ costs $2844\mathbf{M_1}$. Applying **TCAB** to the current seed and using the complexity expression (4), the final exponentiation on $BLS27$ costs:

$$536\mathbf{M_1}+8\times(216\mathbf{M_1})+288\mathbf{M_1}+6\times(26\mathbf{M_1})+2\times(2844\mathbf{M_1})+17\times(2844\mathbf{M_1}) = 56744\mathbf{M_1}.$$

3.5.2 New seeds for TCAB at the 192−bit security level

Case of $k = 15$

Based on the recommendations in [13], we found the seed $u = 1+2^9+2^{16}+2^{68}+2^{71}$, which leads to a prime p of 853 bits and a prime r of 570 bits. The complexity of exponentiation in $G_{\Phi_{15}(p)}$ by this seed using **TCAB** is $3\mathbf{M_{15}}+68\mathbf{S_{15}}+2\mathbf{C_{c_{15}}}$. The cost of this exponentiation is $4888\mathbf{M_1}$. For the current seed, the exponentiation in $G_{\Phi_{15}(p)}$ by $u-1$ is $4810\mathbf{M_1}$. Applying **TCAB** to the seed u and using the complexity expression (2), the final exponentiation of optimal ate pairing on the curve $BLS15$ costs:

$$\begin{aligned} 229\mathbf{M_1} &+ 18\times(78\mathbf{M_1}) + 117\mathbf{M_1} \\ &+ 10\times(14\mathbf{M_1}) + 78\mathbf{M_1} + 2\times(4810\mathbf{M_1}) \\ &+ 9\times(4888\mathbf{M_1}) = 55580\mathbf{M_1}. \end{aligned}$$

Case of $k = 27$

Following the recommendations in [13], we found the seed $u = 1+2^{11}+2^{20}+2^{23}+2^{24}$, this gives a a prime p of 492 bits and a prime r of 443 bits. The current seed results

in an exponentiation in $G_{\Phi_{27}(p)}$ with a complexity of $3\mathbf{M_{27}} + 23\mathbf{S_{27}} + \mathbf{C_{c_{27}}}$. This costs $4455\mathbf{M_1}$, while the exponentiation by $u-1$ costs $4239\mathbf{M_1}$. Using the complexity expression (4) and applying **TCAB** to the current seed, the total cost of the final exponentiation of optimal ate pairing on the curve $BLS27$ is:

$$536\mathbf{M_1}+8\times(216\mathbf{M_1})+288\mathbf{M_1}+2\times(4239\mathbf{M_1})+17\times(4455\mathbf{M_1})+6\times(26\mathbf{M_1}) = 86921\mathbf{M_1}.$$

3.5.3 New seeds for TCAB at the 256−bit security level ($k = 27$)

We followed the same recommendations as in [13] to generate the seed $u = 2 + 2^{41} + 2^{45} + 2^{48}$ for **TCAB** use. This seed gives a prime p of 963 bits and a prime r of 866 bits. The complexity of exponentiation in $G_{\Phi_{27}(p)}$ by applying **TCAB** is $2\mathbf{M_{27}} + 45\mathbf{S_{27}} + 2\mathbf{C_{c_{27}}}$. This exponentiation costs $7893\mathbf{M_1}$, which is also the cost for exponentiation by $u-1$. Based on the complexity expression (4), the total cost of the final exponentiation of optimal ate pairing on the curve $BLS27$ by applying **TCAB** to the seed u is:

$$536\mathbf{M_1}+8\times(216\mathbf{M_1})+288\mathbf{M_1}+2\times(7893\mathbf{M_1})+17\times(7893\mathbf{M_1})+6\times(26\mathbf{M_1}) = 152675\mathbf{M_1}.$$

We group all newly generated seeds in Table 14, providing the following: the curve embedding degree, the size of the prime p, the security level, and the curve equation coefficient b.

Seeds	**k**	**Size(p)**	**Size(p^k)**	**SL**	**b**	**DL algorithm**
$2 + 2^{41} + 2^{45} + 2^{48}$	27	963	25975	256	3	exTNFS
$1 + 2^{11} + 2^{20} + 2^{23} + 2^{24}$	27	492	13265	192	2	SexTNFS
$1 + 2^{9} + 2^{16} + 2^{68} + 2^{71}$	15	853	12787	192	1	exTNFS
$2 + 2^{9} + 2^{12} + 2^{15}$	27	303	8160	128	16	SexTNFS
$2 + 2^{12} + 2^{26} + 2^{28} + 2^{29}$	15	355	5323	128	16	SexTNFS

Table 14: New valid seeds for **TCAB** use.

3.6 Comparison

This section presents two comparison steps. First, we conduct an **self comparison** where we compare the application of **TCAB** and **SM** to our new seeds, emphasizing the superiority of **TCAB**. Second, we perform an **external comparison** in which we compare the cost of the final exponentiation for optimal ate pairing on both $BLS15$ and $BLS27$, using our new seeds alongside existing ones.

3.6.1 Self comparison

In Table 15, we compare the complexity of the final exponentiation of the optimal Ate pairing on the $BLS15$ and $BLS27$ curves, applying **TCAB** and **SM** to our novel seeds. In this comparison, we used the expressions (2) and (4) to calculate the complexity of **TCAB**, and (3) and (5) to calculate the complexity of **SM**. Based on Tables 1

Seed	k	Method	Complexity					
			$\mathbf{I_k}$	$\mathbf{M_k}$	$\mathbf{S_k}$	$\mathbf{C_{c_k}}$	$\mathbf{I_{c_k}}$	$\mathbf{F_k}$
$2+2^{41}+2^{45}+2^{48}$	27	**SM**	1	66	913	0	0	6
		TCAB	1	46	855	39	0	6
$1+2^{11}+2^{20}+2^{23}+2^{24}$	27	**SM**	1	83	457	0	0	6
		TCAB	1	63	437	20	0	6
$1+2^{9}+2^{16}+2^{68}+2^{71}$	15	**SM**	1	61	782	0	1	10
		TCAB	1	49	748	23	1	10
$2+2^{9}+2^{12}+2^{15}$	27	**SM**	1	66	286	0	0	6
		TCAB	1	46	228	39	0	6
$2+2^{12}+2^{26}+2^{28}+2^{29}$	15	**SM**	1	63	320	0	1	10
		TCAB	1	51	308	12	1	10

Table 15: Comparison of the complexity of the final exponentiation on $BLS15$ and $BLS27$ by applying **TCAB** and **SM** to the new seeds.

and 15, we compare the final exponentiation cost on the $BLS15$ and $BLS27$ curves in Table 16 when applying **TCAB** and **SM**. Furthermore, we assess the gain of **TCAB** over **SM**. From Table 16, we confirm our previous analysis by noting that our new seeds are more beneficial when applying **TCAB** rather than **SM**.

3.6.2 External comparison

We select a seed for each security level to compare, in Table 17, with existing seeds in terms of the complexity of the final exponentiation cost on the curves $BLS15$ and $BLS27$. **TCAB** is applied to our seeds, while **SM** is used for the others. Furthermore, in Table 18, we compare the cost of the final exponentiation, highlighting the gains achieved with our seeds.

Seeds:	k	Methods	Cost in $\mathbb{F}_p$	Gain (TCAB/SM)
$2+2^{41}+2^{45}+2^{48}$	27	**SM**	$154637\mathbf{M_1}$	$1962\mathbf{M_1}$
		TCAB	$152675\mathbf{M_1}$	
$1+2^{11}+2^{20}+2^{23}+2^{24}$	27	**SM**	$88541\mathbf{M_1}$	$1620\mathbf{M_1}$
		TCAB	$86921\mathbf{M_1}$	
$1+2^{9}+2^{16}+2^{68}+2^{71}$	15	**SM**	$56035\mathbf{M_1}$	$455\mathbf{M_1}$
		TCAB	$55580\mathbf{M_1}$	
$2+2^{9}+2^{12}+2^{15}$	27	**SM**	$58706\mathbf{M_1}$	$1962\mathbf{M_1}$
		TCAB	$56744\mathbf{M_1}$	
$2+2^{12}+2^{26}+2^{28}+2^{29}$	15	**SM**	$26161\mathbf{M_1}$	$312\mathbf{M_1}$
		TCAB	$25849\mathbf{M_1}$	

Table 16: Comparison of the cost of the pairing's final exponentiation on $BLS15$ and $BLS27$ and by applying **TCAB** and **SM** to the new seeds.

Seeds	k	SL	Complexity
$2+2^{41}+2^{45}+2^{48}$ (**This work**)	27	256	$\mathbf{I_{27}}+46\mathbf{M_{27}}+855\mathbf{S_{27}}+39\mathbf{C_{c_{27}}}+6\mathbf{F_{27}}$
$1+2^{9}+2^{28}+2^{42}+2^{51}$ [18]			$\mathbf{I_{27}}+83\mathbf{M_{27}}+970\mathbf{S_{27}}+6\mathbf{F_{27}}$
$1+2^{11}+2^{20}+2^{23}+2^{24}$ (**This work**)	27	192	$\mathbf{I_{27}}+63\mathbf{M_{27}}+437\mathbf{S_{27}}+20\mathbf{C_{c_{27}}}+6\mathbf{F_{27}}$
$1+2^{4}+2^{14}+2^{17}+2^{25}$ [18]			$\mathbf{I_{27}}+83\mathbf{M_{27}}+476\mathbf{S_{27}}+6\mathbf{F_{27}}$
$2+2^{12}+2^{26}+2^{28}+2^{29}$ (**This work**)	15	128	$\mathbf{I_{15}}+51\mathbf{M_{15}}+308\mathbf{S_{15}}+12\mathbf{C_{c_{15}}}+\mathbf{I_{c_{15}}}+10\mathbf{F_{15}}$
$2^{2}+2^{5}+2^{19}+2^{31}$ [18]			$\mathbf{I_{15}}+52\mathbf{M_{15}}+342\mathbf{S_{15}}+\mathbf{I_{c_{15}}}+10\mathbf{F_{15}}$

Table 17: Comparison of our seeds with some existing seeds in terms of the paring's final exponentiation complexity on $BLS15$ and $BLS27$.

4 Exponentiation using the sparse ternary representation

In this section, we generate new seeds for the $BLS15$ and $BLS27$ curves, sparse in ternary representation. We then evaluate the efficiency of final exponentiation on these curves using the 'Cubing and Multiply' (**CM**) method (Algorithm 1) with these

Seeds	k	SL	Cost	Gain
$2+2^{41}+2^{45}+2^{48}$ (**This work**)	27	256	152675$\mathbf{M_1}$	14202$\mathbf{M_1}$
$1+2^{9}+2^{28}+2^{42}+2^{51}$ [18]			166877$\mathbf{M_1}$	
$1+2^{11}+2^{20}+2^{23}+2^{24}$ (**This work**)	27	192	86921$\mathbf{M_1}$	4527$\mathbf{M_1}$
$1+2^{4}+2^{14}+2^{17}+2^{25}$ [18]			91448$\mathbf{M_1}$	
$2+2^{12}+2^{26}+2^{28}+2^{29}$ (**This work**)	15	128	25771$\mathbf{M_1}$	962$\mathbf{M_1}$
$2^{2}+2^{5}+2^{19}+2^{31}$ [18]			26733$\mathbf{M_1}$	

Table 18: Comparison of our seeds with some existing seeds in terms of the paring's final exponentiation cost on $BLS15$ and $BLS27$.

seeds, compared to 'Square and Multiply' (**SM**) for exponentiation in the cyclotomic subgroup, utilizing cyclotomic cubing.

4.1 Cubing and multiplication (CM)

Let u be a positive integer. The ternary representation of u is given as follows:

$$tern(u) = (t_0 t_1 \cdots t_{n-2} t_{n-1})_3,$$

where $t_0, t_1, \cdots, t_{n-2}, t_{n-1} \in \{0,1,2\}$, $u = t_0 3^0 + t_1 3^1 + \cdots + t_{n-2} 3^{n-2} + t_{n-1} 3^{n-1}$ and n denotes the length of the ternary representation.

Example 3.
Let $u = 21456$. Then, its ternary representation is given by:

$$tern(u) = (0022012001)_3.$$

Let $k \in \mathbb{N}^*$, such that $3|k$. Let $\alpha \in G_{\Phi_k(p)} \subset \mathbb{F}_{p^k}$. To fully leverage the efficiency of cyclotomic cubing, we need a method that maximizes its use. Thus, we introduce the **CM** method, which processes the ternary seed representation, performing one cyclotomic cubing for each digit and one multiplication if the digit is non-zero. This method is described in Algorithm 1.

4.2 Ternary and NAF Representations: Efficiency Comparison

The immediate idea to benefit from the cyclotomic cubing efficiency is to apply **CM** to the existing seeds instead of **SM**. We will assess the feasibility of this idea as follows:

Algorithm 1 CM (Cubing and multiplication)

Input: Parameter $u = (t_0, t_1, \cdots, t_n)_3$, $\alpha \in G_{\Phi_k(p)} \subset \mathbb{F}_{p^k}$
Output: α^u.

1. $r = 1$.
2. $\beta = \alpha^2$.//**If the ternary representation of** α **contains** 2
3. **for** $j = n - 1$ **down to** 0 **do**
   3.1 $r \leftarrow r^3$.
   3.2 **if** $t_i = 1$ **then** $r \leftarrow r\alpha$.
   3.3 **if** $t_i = 2$ **then** $r \leftarrow r\beta$. //**If the ternary representation of a contains** 2
4. **return** r.

Let h_b be the Hamming weight of a given seed u in its NAF representation and h_t be its Hamming weight in its ternary representation. Then, if we use the method **CM** to compute the exponentiation by u in $\mathbb{F}_{p^k}$'s cyclotomic subgroup, the computation complexity is given by:

$$(h_t - 1)\mathbf{M_k} + \mathbf{S_k} + (log_3(u) - 1)\mathbf{C_{c_k}}.$$

If we utilize the method **SM** to perform the wanted exponentiation, the complexity is expressed as:

$$(h_b - 1)\mathbf{M_k} + (log_2(u) - 1)\mathbf{S_k}.$$

We should minimize h_t and h_b to reduce the above complexities. In other words, the seed u should be as sparse as possible. Since h_t and h_b are both used to count multiplications and are typically very small, for **CM** to be more efficient than **SM**, it is required that $log_3(u)\mathbf{C_{c_k}} < log_2(u)\mathbf{S_k}$. This implies $\frac{\mathbf{C_{c_k}}}{\mathbf{S_k}} < \frac{log(3)}{log(2)} \approx 1.58$. Nanjo et al. demonstrated in [16] that this condition does not hold for the BLS15 curve. Specifically, $\frac{\mathbf{C_{c_{15}}}}{\mathbf{S_{15}}} = 1.81 > \frac{log(3)}{log(2)} \approx 1.58$. Therefore, using **CM** instead of **SM** is not advantageous. We reached a similar conclusion with the $BLS27$ curve, as $\frac{\mathbf{C_{c_{27}}}}{\mathbf{S_{27}}} = 1.88 > \frac{log(3)}{log(2)}$. **Thus, applying CM to the ternary representations of existing binary seeds is not beneficial.**

4.3 Generating new sparse ternary seeds

Since **CM** did not outperform **SM** on existing $BLS15$ and $BLS27$ binary seeds, we now explore the potential of generating sparse ternary seeds to enhance the efficiency of the **CM** method. The new seeds should meet the security specifications for discrete logarithm computation outlined in [13]. Additionally, they should be comparable in complexity to the seeds proposed in [15, 18], which adhere to the same specifications. We denote each of our new seeds as u_0 and each seed from [18] as u. For **CM** applied to u_0 to be more efficient than **SM** applied to u, we require $\log_3(u_0)\mathbf{C_{c_{15}}} < log_2(u)\mathbf{S_{15}}$ for $BLS15$ and $\log_3(u_0)\mathbf{C_{c_{27}}} < log_2(u)\mathbf{S_{27}}$ for $BLS27$. This implies $\log_3(u_0) < log_2(u)\frac{\mathbf{S_{15}}}{\mathbf{C_{c_{15}}}}$ and $\log_3(u_0) < log_2(u)\frac{\mathbf{S_{27}}}{\mathbf{C_{c_{27}}}}$. In other words, $\log_3(u_0) < \frac{5}{9}log_2(u)$ for $BLS15$ and $\log_3(u_0) < \frac{17}{32}log_2(u)$ for $BLS27$. Once these inequalities are satisfied, we adjust the Hamming weights to generate the desired seeds.

For $BLS27$, we generated the seed $1+3+2\times 3^{20}+2\times 3^{26}$ at the 256−bit security level and the seed $1+2\times 3^{9}+3^{11}$ at the 192−bit security level. We also generated the seed $1+3^{2}+3^{5}+3^{10}+3^{16}$ for $BLS15$ at the 128−bit security level.

4.4 Comparison

We conduct this comparison in two steps. The first step assesses the security of the new seeds and compares the efficiency of their ternary and NAF representations for exponentiation in the cyclotomic subgroup. We refer to this step as self comparison. The second step contrasts the security and efficiency of our seeds with those proposed in [18], and we refer to it as an external comparison.

4.4.1 Self comparison

We summarize, in Table 19, the security properties of our new ternary seeds based on the security specifications presented in [13]. In Table 20, we compare the complexity

Seeds	**k**	**Size(p)**	**Size(p^k)**	**SL**	**b**	**DL algorithm**
$1+3+2\times 3^{20}+2\times 3^{26}$	27	843	22752	256	16	exTNFS
$1+2\times 3^{9}+3^{11}$	27	353	9529	192	16	exTNFS
$1+3^{2}+3^{5}+3^{10}+3^{16}$	15	303	4542	128	1	exTNFS

Table 19: Security proprieties of our new ternary seeds

of exponentiation in the cyclotomic subgroup using the **SM** method with the NAF representation versus the **CM** method with the ternary representation of these seeds. We give also the gains provided by applying **TCAB** to our seeds instead of **SM**. For our new seeds, it is clear from Table 20 that exponentiation in the cyclotomic subgroups of $BLS15$ and $BLS27$ is more efficient in ternary representation than in NAF representation.

Seeds	**k**	**Methods**	**Complexity**			**Gain(CM/SM)**
			$\mathbf{M_k}$	$\mathbf{S_k}$	$\mathbf{C_{c_k}}$	
$1+3+2\times 3^{20}+2\times 3^{26}$	27	**SM**	20	42	0	$2457\mathbf{M_1}$
		CM	3	1	26	
$1+2\times 3^{9}+3^{11}$	27	**SM**	8	17	0	$576\mathbf{M_1}$
		CM	2	1	11	
$1+3^{2}+3^{5}+3^{10}+3^{16}$	15	**SM**	15	25	0	$611\mathbf{M_1}$
		CM	4	0	16	

Table 20: The complexity of exponentiation in $G_{\phi_{15}(p)}$ and $G_{\phi_{27}(p)}$ using **CM** and **SM** applied to the new ternary seeds.

We now proceed to compare our new ternary seeds with existing seeds. In Table 21, we compare the security properties of our new ternary seeds with those proposed in [18]. Although our seeds meet security levels considering the sizes p and r in bits,

Seeds	**SL**	**Size(p)**	**Size(r)**	**Size($\mathbf{p^k}$)**	**DL Alg**
$1+3+2\times 3^{20}+2\times 3^{26}$	256	851	766	22976	exTNFS
$1+2^9+2^{28}+2^{42}+2^{51}$ [18]		1019	883	27499	SexTNFS
$1+2\times 3^9+3^{11}$	192	353	318	9529	exTNFS
$1+2^4+2^{14}+2^{17}+2^{25}$ [18]		511	410	13461	SexTNFS
$1+3^2+3^5+3^{10}+3^{16}$	128	303	203	4542	exTNFS
$2^2+2^5+2^{19}+2^{31}$ [18]		371	249	5557	SexTNFS

Table 21: Comparison of security proprieties of our ternary seeds with binary seeds in [18]. $k=27$ for **SL**= 256 or 192 and $k=15$ for **SL**= 128.

they are less secure than those proposed in [18].
In Table 22, we compare the complexity of exponentiation in the cyclotomic subgroups of $\mathbb{F}p^{15}$ and $\mathbb{F}p^{27}$ using our new seeds and those proposed in [18]. Additionally, we present the performance gains achieved by applying the CM method to ternary seeds, as compared to the SM method applied to binary seeds used in [18]. Since the gains are

Seed	**k**	**Complexity**			**Gain in $\mathbb{F}_p$**
		$\mathbf{M_k}$	$\mathbf{S_k}$	$\mathbf{C_{c_k}}$	
$1+3+2\times 3^{20}+2\times 3^{26}$	27	3	1	26	$378\mathbf{M_1}$
$1+2^9+2^{28}+2^{42}+2^{51}$ [18]		4	51	0	
$1+2\times 3^9+3^{11}$	27	2	1	11	$936\mathbf{M_1}$
$1+2^4+2^{14}+2^{17}+2^{25}$ [18]		4	25	0	
$1+3^2+3^5+3^{10}+3^{16}$	15	4	0	16	$65\mathbf{M_1}$
$2^2+2^5+2^{19}+2^{31}$ [18]		3	31	0	

Table 22: Comparison of the complexity of exponentiation by the ternary seeds and the seeds of [18] in $G_{\phi_{15}(p)}$ and $G_{\phi_{27}(p)}$.

positive in Table 22, we will continue by comparing the costs of the final exponentiation of the optimal Ate pairing on $BLS15$ and $BLS27$ using our ternary seeds with those proposed in [18]. For the curve $BLS15$ and at 128−bit security level, we apply **CM** to the seed $u1+3^2+3^5+3^{10}+3^{16}$. We find $\mathbf{E_{u-1}}=2106\mathbf{M_1}$ and $\mathbf{E_u}=2184\mathbf{M_1}$. We use the complexity expression (3) to compute the cost of the final exponentiation

on $BLS15$ as follows:

$$\begin{aligned}&229\mathbf{M_1} + 18 \times (78\mathbf{M_1}) + 117\mathbf{M_1} + (78\mathbf{M_1}) + 10 \times (14\mathbf{M_1})\\&+ 2 \times (2106\mathbf{M_1}) + 9 \times (2184\mathbf{M_1})\\&= 25836\mathbf{M_1}.\end{aligned}$$

For $BLS27$ curve and at 192−bit security level, we apply **CM** to the seed $u = 1 + 2 \times 3^9 + 3^{11}$ to get $\mathbf{E_{u-1}} = 3537\mathbf{M_1}$ and $\mathbf{E_u} = 3753\mathbf{M_1}$. We use the complexity expression (4) to compute the cost of the final exponentiation on $BLS27$ as follows:

$$536\mathbf{M_1}+8\times(216\mathbf{M_1})+288\mathbf{M_1}+6\times(26\mathbf{M_1})+2\times(3537\mathbf{M_1})+17\times(3753\mathbf{M_1}) = 73583\mathbf{M_1}.$$

For $BLS27$ curve and at 256−bit security level, where the seed is $u = 1+3+2\times3^{20}+2\times3^{26}$, we apply **CM** to u to obtain $\mathbf{E_{u-1}} = 8073\mathbf{M_1}$ and $\mathbf{E_u} = 8289\mathbf{M_1}$. Thus, the cost of the final exponentiation on $BLS27$ for the current seed is computed as follows:

$$536\mathbf{M_1}+8\times(216\mathbf{M_1})+288\mathbf{M_1}+6\times(26\mathbf{M_1})+2\times(8073\mathbf{M_1})+17\times(8289\mathbf{M_1}) = 159767\mathbf{M_1}.$$

We give these costs in Table 23 and compare them with the costs provided by the seeds proposed in [18], highlighting the gains achieved by using our seeds to calculate the final exponentiation.

Seeds	**k**	**SL**	**Cost**	**Gain**
$1+3+2\times3^{20}+2\times3^{26}$ (**This work**)	27	256	$159767\mathbf{M_1}$	$7110\mathbf{M_1}$
$1+2^9+2^{28}+2^{42}+2^{51}$ [18]			$166877\mathbf{M_1}$	
$1+2\times3^9+3^{11}$ (**This work**)	27	192	$73583\mathbf{M_1}$	$17865\mathbf{M_1}$
$1+2^4+2^{14}+2^{17}+2^{25}$ [18]			$91448\mathbf{M_1}$	
$1+3^2+3^5+3^{10}+3^{16}$ (**This work**)	15	128	$25836\mathbf{M_1}$	$897\mathbf{M_1}$
$2^2+2^5+2^{19}+2^{31}$ [18]			$26733\mathbf{M_1}$	

Table 23: Comparison of the final exponentiation costs and the gain offered by our seeds

Based on the tables presented in this section, we conclude the following regarding the use of cyclotomic cubing and ternary representation for computing final exponentiation on elliptic curves curves with odd embedding degrees divisible by 3:

1. Applying the **CM** method to a seed where the **SM** method is already efficient offers no advantage,
2. We can generate a sparse ternary seed where the **CM** method outperforms the **SM** method on the NAF representation of the same seed,
3. Given a seed u_0 with a sparse binary representation where the **SM** method is efficient, we can generate a sparse ternary seed v_0 for which **CM** outperforms **SM** on u_0. However, the ternary seed has less security than the binary seed, according to the security constraints in [13].
4. If a sparse ternary seed v_0 is generated to match the same security of a sparse binary seed u_0, then **CM** applied to v_0 is less efficient than **SM** applied to u_0.

5 Conclusion

The choice of seed for generating pairing-friendly curve parameters is critical in optimizing the efficiency of the Miller algorithm and the final exponentiation. To enhance these tasks, selecting a seed with a sparse NAF representation is essential. For elliptic curves with odd embedding degrees divisible by 3, such as $BLS27$, cyclotomic cubing has been shown to outperform squaring followed by multiplication. Inspired by Nanjo et al. [16], we optimized the final exponentiation of the optimal Ate pairing on the BLS family, achieving a consistent slight improvement. Furthermore, we introduced the **TCAB** method, which leverages partial cyclotomic cubing based on the specific structure of certain seeds, while preserving the efficiency of the Miller algorithm through the sparse NAF representation of the seed. Cyclotomic cubing on these curves motivated us to generate new seeds with sparse ternary representations, accelerating exponentiation in the cyclotomic subgroups of $\mathbb{F}p^{15}$ and $\mathbb{F}p^{27}$, and improving final exponentiation on $BLS15$ and $BLS27$. However, two challenges emerged with ternary representations: first, generating seeds with competitive complexity to existing seeds with sparse binary representation risks reducing security; second, the ternary seeds do not always exhibit sparse NAF representations, limiting the efficiency of the Miller algorithm. Further improvements in cyclotomic cubing and the development of a ternary-based alternative to the double-and-add method [34] are needed to address these challenges.

References

[1] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Annual International Cryptology Conference, pp. 213–229 (2001). Springer

[2] Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 514–532 (2001). Springer

[3] Joux, A.: A one round protocol for tripartite diffie–hellman. In: International Algorithmic Number Theory Symposium, pp. 385–393 (2000). Springer

[4] Barreto, P.S., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Security in Communication Networks: Third International Conference, SCN 2002 Amalfi, Italy, September 11–13, 2002 Revised Papers 3, pp. 257–267 (2003). Springer

[5] Kachisa, E.J., Schaefer, E.F., Scott, M.: Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field. In: International Conference on Pairing-based Cryptography, pp. 126–135 (2008). Springer

[6] Barreto, P.S., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: International Workshop on Selected Areas in Cryptography, pp. 319–331 (2005). Springer

[7] Vercauteren, F.: Optimal pairings. IEEE transactions on information theory **56**(1), 455–461 (2009)

[8] Hess, F., Smart, N.P., Vercauteren, F.: The eta pairing revisited. IEEE transactions on information theory **52**(10), 4595–4602 (2006)

[9] Scott, M., Benger, N., Charlemagne, M., Dominguez Perez, L.J., Kachisa, E.J.: On the final exponentiation for calculating pairings on ordinary elliptic curves. In: Pairing-Based Cryptography–Pairing 2009: Third International Conference Palo Alto, CA, USA, August 12-14, 2009 Proceedings 3, pp. 78–88 (2009). Springer

[10] Stam, M., Lenstra, A.K.: Efficient subgroup exponentiation in quadratic and sixth degree extensions. In: Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13–15, 2002 Revised Papers 4, pp. 318–332 (2003). Springer

[11] Granger, R., Scott, M.: Faster squaring in the cyclotomic subgroup of sixth degree extensions. In: International Workshop on Public Key Cryptography, pp. 209–223 (2010). Springer

[12] Karabina, K.: Squaring in cyclotomic subgroups. Mathematics of Computation **82**(281), 555–579 (2013)

[13] Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. Journal of cryptology **32**, 1298–1336 (2019)

[14] Kim, T., Barbulescu, R.: Extended tower number field sieve: A new complexity for the medium prime case. In: Annual International Cryptology Conference, pp. 543–571 (2016). Springer

[15] Barbulescu, R., El Mrabet, N., Ghammam, L.: A taxonomy of pairings, their security, their complexity (2020)

[16] Nanjo, Y., Shirase, M., Kusaka, T., Nogami, Y.: 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), pp. 288–292 (2020). IEEE

[17] Hankerson, D., Menezes, A.: Elliptic curve cryptography. In: Encyclopedia of Cryptography, Security and Privacy, pp. 1–2. Springer, ??? (2021)

[18] Fouotsa, E., El Mrabet, N., Pecha, A.: Optimal ate pairing on elliptic curves with embedding degree $9, 15$ and 27. Journal of groups, complexity, cryptology **12** (2020)

[19] Khamseh, E.: The review on elliptic curves as cryptographic pairing groups. Mathematics and Computational Sciences **2**(2), 50–59 (2021)

[20] Aranha, D.F., Fotiadis, G., Guillevic, A.: A short-list of pairing-friendly curves resistant to the special tnfs algorithm at the 192-bit security level (2024)

[21] Cryptography, C.: Elliptic and hyperelliptic curve cryptography

[22] Chung, J., Hasan, M.A.: Asymmetric squaring formulae. In: 18th IEEE Symposium on Computer Arithmetic (ARITH'07), pp. 113–122 (2007). IEEE

[23] Duquesne, S., El Mrabet, N., Haloui, S., Rondepierre, F.: Choosing and generating parameters for pairing implementation on bn curves. Applicable Algebra in Engineering, Communication and Computing **29**(2), 113–147 (2018)

[24] Galbraith, S.D.: Mathematics of Public Key Cryptography. Cambridge University Press, ??? (2012)

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

[26] Fuentes-Castaneda, L., Knapp, E., Rodríguez-Henríquez, F.: Faster hashing to. In: International Workshop on Selected Areas in Cryptography, pp. 412–430 (2011). Springer

[27] Ghammam, L., Fouotsa, E.: Adequate elliptic curves for computing the product of n pairings. In: International Workshop on the Arithmetic of Finite Fields, pp. 36–53 (2016). Springer

[28] Ghammam, L., Fouotsa, E.: Improving the computation of the optimal ate pairing for a high security level. Journal of Applied Mathematics and Computing **59**(1), 21–36 (2019)

[29] Zhang, X., Lin, D.: Analysis of optimum pairing products at high security levels. In: Progress in Cryptology-INDOCRYPT 2012: 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings 13, pp. 412–430 (2012). Springer

[30] Hayashida, D., Hayasaka, K., Teruya, T.: Efficient final exponentiation via cyclotomic structure for pairings over families of elliptic curves. Cryptology ePrint Archive (2020)

[31] Duan, P., Cui, S., Chan, C.W.: Special polynomial families for generating more suitable elliptic curves for pairing-based cryptosystems. Cryptology ePrint Archive (2005)

[32] Ribenboim, P.: Catalan's conjecture. Séminaire de Philosophie et Mathématiques (6), 1–11 (1994)

[33] Schoof, R.: Catalan's Conjecture. Springer, ??? (2010)

[34] Coron, J.-S.: Resistance against differential power analysis for elliptic curve cryptosystems, 292–302 (1999). Springer