Efficient Software Implementation of GMT6-672 and GMT8-542 Pairing-Friendly Curves for a 128-Bit Security Level

Zihao Song[1], Junichi Sakamoto[1,2], Shigeo Mitsunari[3], Naoki Yoshida[1], Riku Anzai[1], and Tsutomu Matsumoto[1(✉)]

[1] Yokohama National University, 79-7 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan
tsutomu@ynu.ac.jp

[2] National Institute of Advanced Industrial Science and Technology, 2-3-26 Aomi, Koto-ku, Tokyo 135-0064, Japan

[3] Cybozu Labs, Inc, Tokyo Nihombashi Tower 27F, 2-7-1 Nihombashi, Chuo-ku, Tokyo 103-6028, Japan

Abstract. A Bilinear pairing on an elliptic curve defined over a finite field provides an attractive prospect for designing cryptographic schemes with various functionalities. An elliptic curve over which a computationally efficient bilinear pairing can be defined is called a "pairing-friendly curve". Finding families of pairing-friendly curves with sufficient anticipated bit security has attracted significant research attention. For example, the Barreto-Neahrig (BN) and Barreto-Lynn-Scott (BLS) curves, are existing curves of this type. However, there is a need for alternatives to back up these already evaluated curves. In 2020 Guillevic, Masson, and Thomé (GMT) proposed pairing-friendly curves with embedding degrees 5 to 8 range. GMTk denotes curves with an embedding degree k. A composite k is preferred from the efficiency viewpoint. However, to the best of the GMT6 and GMT8 curves have been reported in the literature. In this paper, novel field-towering methods using two types of extension method and constructions are developed. These methods are applied to efficiently implement and analyze the bilinear pairings based on the GMT6 curve over a 672-bit prime field and the GMT8 curve over a 542-bit prime field. The pairing-computation times of our developed software evaluated using an Intel Core i7-8700 (@4.3 GHz Turbo Boost on) is computer are 0.987 ms and 1.12 ms for GMT6-672 and GMT8-542, respectively indicating the practicality of these curves.

Keywords: Software implementations · Bilinear pairings · Type-I AOPF

1 Introduction

A Bilinear pairing (hereafter simply "pairing") over an elliptic curve is valuable for implementing advanced cryptography, such as aggregate signatures [1],

J. Zhou et al. (Eds.): ACNS 2022 Workshops, LNCS 13285, pp. 461–478, 2022.
https://doi.org/10.1007/978-3-031-16815-4_25

homomorphic encryption [2], etc. One of the recent innovative protocols based on pairing is the zero-knowledge succinct noninteractive argument of knowledge (zk-SNARKs) [3]. Pairing is a nondegenerate bilinear map obtained from the direct product of two additive groups $\mathbb{G}_1$ and $\mathbb{G}_2$, resulting in a multiplicative group $\mathbb{G}_3$. The groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are generally subgroups obtained from elliptic curve groups $E(\mathbb{F}_p)$ and $E(\mathbb{F}_{p^k})$, where E, p, and k denote an elliptic curve, field characteristic, and embedding degree, respectively. Pairings constructed over elliptic curves require different properties and security levels depending on the particular application. Therefore, the investigation of new curves of efficient pairing computation (called "pairing-friendly" curves) constitutes a significant research area. The Barreto-Naehrig (BN) curves [5], Barreto-Lynn-Scott curves (BLS) [6], and Kachisa-Schaefer-Scott (KSS) curves are the most well-known families of pairing-friendly curves, which have been widely studied as efficient candidates for 128-bit level security pairings. Besides, there is an attack reported in [7] improves the number field sieve algorithm in discrete-logarithm problems in extension fields and affects the security level of many pairing-friendly curves. Hence, the parameters of pairing-friendly curves are forced to be replaced with their parameters for 128-bit security levels with enough margin. This parameter replacement has been studied only for a short period since the year 2016, after it the performance and security assessment for the well-known curves appear vague. In 2020, Guillevic, Masson, and Thomé (GMT) [8] proposed new curves generated by a modified Cocks-Pinch method. These curves satisfy the 128-bit level security against the attack mentioned in [7]. We refer to the paper [8] as "the GMT paper". Moreover, we denote the curves with $k = 6, 8$ proposed in [8] as the GMT6 and GMT8 curves, respectively. The GMT paper presented algorithms for fast pairing calculation. A simple model estimates the computational timings of pairing computation over the GMT6 and GMT8, where both results are 1.5 ms using an Intel Core i7-8700@3.2 GHz computer. Although the results reported in the GMT paper are promising, to the best of our knowledge, there is no study on rigorous software implementation for these curves. For this purpose, this paper aims to provide the first and efficient software implementation of the GMT curves with a detailed cost analysis.

Our Contributions. The following three main contributions are present in this paper. First, two types of efficient field towering methods for the GMT6 and GMT8 curves with the type-I all-one polynomial field (AOPF) [11] and the optimal extension field (OEF) [12] are proposed. These fields are used as the first subextension fields for fast paring calculation. Furthermore, the number of arithmetic operations of the proposed extension field towering is investigated and a new GMT8 curve parameter optimized for our extension fields is provided. Second, an unique detailed cost at the algorithm level is provided for implementing Miller's algorithm [4] with twists [13], and the required cost is reevaluated using an accurate expression. Moreover, the polynomials suggested by the GMT paper are reviewed for calculating the fast final exponentiation calculation and revised for efficiently calculating the orders of both curves. Finally, the experimental results obtained from the implemented software regarding the pairings

over the GMT6 and GMT8 curves based on the proposed constructions are presented. The software implementations, which are based on the crypto library [14] and the GNU MP library (GMP) version 6.2.1, give rise to pairing computation of the GMT6 and GMT8 curves in 0.987 ms and 1.12 ms, respectively, using an i7-8700 (@4.3 GHz Turbo Boost enabled) computer without using the lazy reduction technique.

Related Research Works. Lavice et al. [9] proposed a small-area pairing-computation architecture using the FPGA for the updated 128-bit level pairing-friendly curves. They also proposed an attractive formula for calculating the squaring in the quadratic cyclotomic subgroup. We adopt this suggested squaring method employed in the quadratic cyclotomic subgroup and use it in our proposed tower of extension fields to reduce the calculation cost.

Notation. In this paper, a multiplication, squaring, and inversion cost in $\mathbb{F}_{p^k}$ is denoted as m_k, s_k, and i_k, respectively. The symbol a_k denotes an addition cost in $\mathbb{F}_{p^k}$, where it is assumed that subtraction, left-shift, and right-shift costs in $\mathbb{F}_{p^k}$ are identical to a_k. $\mathbf{m}$ is used with m_1, and s_1 summarizes the total cost of $m_1 + s_1$ in $\mathbb{F}_p$. To distinguish parameters with different characteristics with the same embedding degree, each curve parameter is given a different designation using a bit length of characteristic p as the suffix, such as the GMT8-544 and GMT8-542.

2 Preliminaries

The GMT curves with embedding degrees $k = 6, 8$ and ate pairing over the GMT curves are reviewed in this section.

2.1 Guillevic-Masson-Thomé (GMT) Curves with Embedding Degrees 6 and 8

Guillevic, Masson, and Thomé [8] proposed pairing-friendly elliptic curves based on the Cocks-Pinch algorithm with embedding degrees $k = 5, 6, 7, 8$. The curves with even embedding degrees $k = 6$ and 8 (GMT6 and GMT8) are capable of calculating pairing efficiently. The parameters of the GMT curves (field characteristic $p(u)$, order $r(u)$, and Frobenius trace $t(u)$ with coefficient h_t, h_y) are given by the following polynomials, where the integer parameters $u, h_y, h_t \in \mathbb{Z}$ are selected as p and r are prime numbers. The complex multiplication (CM) discriminant of the GMT6 curve is $D = 3$ with elliptic curve $E : y^2 = x^3 + b$ where $x, y \in \mathbb{F}_{p^6}$ with non-zero coefficient $b \in \mathbb{F}_p$. The ρ-value$= log(p)/log(r)$ of GMT6 is 2.63. For GMT8 curve, the CM discriminant is $D = 4$ with the elliptic curve $E : y^2 = x^3 + ax$ where $x, y \in \mathbb{F}_{p^8}$ with the nonzero coefficient $a \in \mathbb{F}_p$. For $k = 8$, the obtained GMT8-542 curve has a slightly better ρ-value$= 2.12$ than the GMT6 curve.

Algorithm 1. Ate pairing over the GMT6 and GMT8 curves using a 2-NAF loop parameter expression

Require: T, $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$
Ensure: $f_{T,Q}(P) \in \mathbb{F}^*_{p^k}$
1 $f \leftarrow 1, R \leftarrow Q$
2: **for** $i = \lfloor \log_2(T) \rfloor - 1$ down to 0 **do**
3: $\lambda \leftarrow l_{R,R}(P)$, $R \leftarrow 2R$ ▷ //DBLLine
4: $f \leftarrow f^2 \cdot \lambda$ ▷ //UPDATE1
5: **if** $T[i] = 1$ **then**
6: $\lambda \leftarrow l_{R,Q}(P)$, $R \leftarrow R + Q$ ▷ //ADDLine
7: $f \leftarrow f \cdot \lambda$ ▷ //UPDATE2
8: **if** $T[i] = -1$ **then**
9: $\lambda \leftarrow l_{R,-Q}(P)$, $R \leftarrow R - Q$ ▷ //ADDLine
10: $f \leftarrow f \cdot \lambda$ ▷ //UPDATE2
11: $f \leftarrow f^{(p^k-1)/r}$
12: **return** f

The number of rational points on the elliptic curve E over the finite field $\mathbb{F}_p$ is expressed as $\#E(\mathbb{F}_p) = p+1-t$ according to the Hasse's theorem. The elliptic curve E also forms an additive group in the extension field $E(\mathbb{F}_{p^k})$, where k is the embedding degree of the curve. The order of $E(\mathbb{F}_{p^k})$ is $\#E(\mathbb{F}_{p^k}) = p^k+1-t_k$, where $t_k = \alpha^k + \beta^k$ and α and β are complex conjugate numbers. The r-torsion subgroup of E, which is defined as $E[r] := \{P | P \in E, [r]P = \mathcal{O}\}$ has two unique subgroups of order r. These subgroups are useful for efficient pairing computation. Let the π_p be Frobenius endomorphism and the first subgroup $\mathbb{G}_1 = E[r] \cap \ker(\pi_p - [1]) \subset E(\mathbb{F}_p)[r]$, which is defined over $\mathbb{F}_p$. The second subgroup $\mathbb{G}_2 = E[r] \cap \ker(\pi_p - [p]) \subset E(\mathbb{F}_{p^k})[r]$, which is defined over $\mathbb{F}_{p^k}$. The subgroup order r satisfies the condition $r|(p^k - 1)$, $r|\#E(\mathbb{F}_p)$, $r^2|\#E(\mathbb{F}_{p^k})$ which are important for pairing computation optimization.

2.2 Ate Pairing over the GMT6 and GMT8 Curves

Let $\mathbb{G}_3$ be a multiplicative subgroup defined as

$$\mathbb{G}_3 = \mathbb{F}_{p^k}[r] \tag{1}$$

where k is the embedding degree of the pairing-friendly curve. For three Abelian groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3$, an ate pairing a_T can be defined as follows:

$$a_T : \mathbb{G}_2 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3, \tag{2}$$

$$(Q, P) \mapsto (f_{T,Q}(P))^{(p^k-1)/r} \tag{3}$$

where $T = u - 1$ and $f_{T,Q}$ is a rational function with a divisor $\mathrm{div}(f_{T,Q}) = T(Q) - ([T]Q) - (T-1)(\mathcal{O})$. Ate pairing for the GMT6 and GMT8 curves is calculated by using Algorithm 1.

In Algorithm 1, the calculation steps 1 to 10 are identified as Miller's algorithm, where steps 2 to 10 are particularly called Miller's loop. Steps 3, 6 and 9 describe the calculations of the rational functions $l_{R,R}, l_{R,Q}$ together with the elliptic curve doubling (ECD) and elliptic curve addition (ECA) calculations. We call the calculations of $l_{R,R}$ together with the ECD as DBLLine. Similarly, we call the calculations of and $l_{R,Q}$ together with the ECA as ADDLine. UPDATE1 and UPDATE2 are "sparse" multiplications in $\mathbb{F}_{p^k}$ with less computational load than the standard multiplication in $\mathbb{F}_{p^k}$.

The input $T = u - 1$ is expressed in a nonadjacent form (NAF), which represents integers in certain conditions as three value types of 1, 0, -1, rather than a binary form for efficient pairing calculating. Miller's algorithm is also known as an algorithm capable of using a 2-NAF since the inversion operation $l_{R,-Q}(P)$ can be easily calculated in this case.

Step 11 is known as the final exponentiation, the details of the final exponentiation calculation for the GMT6 and GMT8 curves are described in Sect. 5.2.

2.3 Ate Pairings over GMT Curves with Twists

The **GMT6** curve with the CM discriminant $D = 3$ and input $Q \in \mathbb{G}_2$ used for ate paring over the elliptic curve $E : y^2 = x^3 + b$ known as having an isomorphism ψ. The isomorphism ψ projects the subgroup $\mathbb{G}_2 \subset E(\mathbb{F}_{p^6})$ to a same order subgroup $\mathbb{G'}_2 \subset E'(\mathbb{F}_p)$ where the sextic twist $E' : y^2 = x^3 + b/z, z \in \mathbb{F}_p$. Since two subgroups have the same information, the required cost heavy arithmetics in $\mathbb{F}_{p^6}$ can be replaced by the simple calculations in $\mathbb{F}_p$. The isomorphism ψ from the twisted curve to the original curve can be defined as follows:

$$\psi : E' \to E, \tag{4}$$

$$Q'(x, y) \mapsto Q(xz^{-1/3}, yz^{-1/2}) \tag{5}$$

With assuming that both $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G'}_2 \subset E'(\mathbb{F}_p)$, the twisted ate pairing for the GMT6 curve can be computed as follows:

$$a_T : \mathbb{G'}_2 \times \mathbb{G}_1 \to \mathbb{G}_3, \tag{6}$$

$$(Q', P) \mapsto (f_{T,\psi(Q')}(P))^{(p^6-1)/r} \tag{7}$$

In this case, the ADDLine and DBLLine can be computed in $\mathbb{F}_p$.

The **GMT8** curve $E : y^2 = x^3 + ax$ with a CM discriminant $D = 4$ has a different type of twist called "quartic twist". The map φ from the twisted elliptic curve E' to the original curve E is defined as follows:

$$\varphi : E' \to E, \tag{8}$$

$$Q'(x, y) \mapsto Q(xz^{-1/2}, yz^{-3/4}) \tag{9}$$

where $z \in \mathbb{F}_{p^2}$ is a quadratic non-residue in $\mathbb{F}_p$, $x^4 - z \in \mathbb{F}_{p^2}[x]$ is irreducible, the twisted curve $E' : y^2 = x^3 + ax/z$, and $Q' \in \mathbb{G'}_2 \subset E'(\mathbb{F}_{p^2})$. Similar to

the GMT6 curve, ate pairing with the quartic twist for the GMT8 curve can be computed as follows:

$$a_T : \mathbb{G}'_2 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3, \tag{10}$$

$$(Q', P) \mapsto (f_{T,\varphi(Q')}(P))^{(p^8-1)/r} \tag{11}$$

In this case, the GMT8 curves with embedding degree 8 can compute the ADDLine and DBLLine functions of ate pairing in $\mathbb{F}_{p^2}$ arithmetics. Even though the twist maps reduce the number of arithmetic operations in the pairing, the total cost of Miller's algorithm depends on other elements, such as the Miller's loop parameter T and the coordinate system. Furthermore, optimizing the final exponentiation calculation and not only Miller's algorithm (for example, factorizing the polynomial $(p^k - 1)/r$ $(k = 6, 8)$ and performing fast squaring in the extension fields), is also a key component for fast pairing computations.

3 Review of Extension Field Classes

For fast pairing computation, the efficiency of the multiplication over the extension fields heavily decides it's efficiency. To construct an extension field, first the primitive root c of $f(x)$ is preferred to choose from the twist curve parameter z [13]. Second, the primitive root of $f(x)$ is preferred to be as simple as possible (for example $c = 2$). These constraints make impose a difficulty to find efficient irreducible polynomials for pairing. A tower of extension fields that have nested structures is proposed based on [10]. In this section, the existing classes of practical extension fields are initially reviewed and then the candidates for the tower of fields available for the GMT curves are indicated.

3.1 Optimal Extension Fields

Bailey and Paar [12] introduced the following formal definition for constructing extension fields consisting of a polynomial basis:

Definition 1 (Optimal extension fields, OEFs). *OEFs are the extension fields satisfying the following three properties.*

1. ***Characteristic:*** *A pseudo-Mersenne prime number p of the form $p = 2^l \pm c$, where $l, c \in \mathbb{Z}$.*
2. ***Modular Polynomial:*** *An irreducible binomial $x^m - s$, where $s \in \mathbb{F}_p$ and m is the extension degree.*
3. ***Basis:*** *A set $\{1, \omega, \omega^2, ..., \omega^{m-1}\}$, where ω is a primitive root of the modular polynomial.*

Although the characteristic p is a pseudo-Mersenne prime number in the OEF definition, it is known that an OEF is actually capable of general prime numbers. An OEF has several fast multiplication algorithms for different degrees m, such as Karatsuba method [19], the Karatsuba-like method [20], and Toom-Cook method [21]. Specifically $m = 2$ and $s = -1$ constitute the most important

variant, where the squaring computation in $\mathbb{F}_{p^2}$ requires only two multiplications in $\mathbb{F}_p$, using the Karatsuba method. We call this technique "Karatsuba complex method," which is a famous and standard technique for accelerating pairing calculation.

3.2 All-One Polynomial Extension Fields

Unlike an extension field such as the polynomial based OEF described above, a special extension with a Gaussian normal basis was introduced by Nogami et al. [11]. This field is called all-one polynomial field (AOPF). Later Nekado et al. extended its definition and classified several types of AOPFs, such as type-I X [18] and type-II X [15,17]. The definition of Type-I AOPF is given as follows:

Definition 2 (Type-I All-one polynominal Fields). *Type-I AOPFs are the extension fields satisfying the following three properties.*

1. ***Characteristic:*** *A pseudo-Mersenne prime number p of the form $p = 2^l \pm c$, where $l, c \in \mathbb{Z}$.*
2. ***Modular Polynomial:*** *An all-one irreducible polynomial $(x^{m+1}-1)/(x-1)$, where $s \in \mathbb{F}_p$ and $m+1$ is a prime number.*
3. ***Basis:*** *A pseudo basis $\{\omega, \omega^2, , \omega^3 ..., \omega^m\}$ is equivalent to the normal basis $\{\omega, \omega^p, \omega^{p^2}, ..., \omega^{p^{m-1}}\}$ where ω is a primitive root of the modular polynomial.*

Although the characteristic p is a pseudo-Mersenne prime number in the Definition 2, it is known that an AOPF is actually capable of general prime numbers. An efficient way to calculate the multiplication in an AOPF is to use the cyclic vector multiplication algorithm (CVMA), which is more efficient than the multiplication in an OEF. According to Nekado et al. [18], the squaring in the quadratic type-I AOPF: $\mathbb{F}_{p^2} = \mathbb{F}_p[\omega]/(\omega^2 + \omega + 1)$ only requires two multiplications in $\mathbb{F}_p$ as follows:

$$\alpha = (a_0, a_1),\ \alpha^2 = \beta = (b_0, b_1), \tag{12}$$

$$b_0 = \{-a_1(a_0 - a_1) + a_0\},\ b_1 = \{-a_0(a_0 - a_1) - a_1\} \tag{13}$$

where $\alpha, \beta \in \mathbb{F}_{p^2}$ and $a_0, a_1, b_0, b_1 \in \mathbb{F}_p$. Unlike the OEF, the type-I AOPF has much constraints. For example, $m+1$ must be a prime number, which restricts the degree of AOPF extension to an even number only. Furthermore, since the degree of type-II AOPF, the squaring in $\mathbb{F}_{p^2}$ requires three multiplications in $\mathbb{F}_p$, which is less efficient compared with the type-I AOPF or the adapted $z = s = -1$ OEF in Karatsuba complex method. In addition, if the probability of a general prime number to construct a degree-2 type-I AOPF is at most 50%. Still, the 2 **m** cost squaring, the quadratic extension field of both OEF with $s = -1$ and type-I AOPF are still good candidates for a fast pairing calculation.

4 Proposal of Efficient GMT6 and GMT8 Curve Parameters and Their Field-Towering Schemes

As described in Sect. 3, building an efficient tower of an extension field with twist capability has a few constraints regarding the selection of the polynomial primitive root. Although the field towering system reduces the degree of irreducible polynomials to be explored, finding the curve parameters with an efficient computation cost is still complicated. In this section, the parameter selection for both GMT6 and GMT8 curves is described, and new curve parameters and tower construction methods for efficient pairing computation over GMT curves are proposed.

4.1 GMT6 Curve Parameters and Towers

The GMT paper [8] suggests the use of parameters for the GMT6 curve, as shown in Table 1. These parameters are denoted as GMT6-672. The GMT paper suggests the direct sextic extension using the irreducible polynomial $x^6 - s, s = 2 \in \mathbb{F}_p$ for the pairing computation over GMT6-672. Since 2 is a quadratic non-residue (QNR) and a cubic non-residue (CNR) in $\mathbb{F}_p$, the twist parameter z can be equivalent to $z = 2$. In this work, a field towering scheme τ_1 based on the extension proposed in the GMT paper was derived, as shown in Table 3. However, we found that the suggested τ_1 cost 2 extra addition in $\mathbb{F}_{p^2}$ squaring compare to $z = s = -1$; therefore, the arithmetic costs in τ_1 is not the best for pairing calculation.

We propose a new variant of field towering scheme τ_2 for efficient pairing computation over the GMT6 curve using both the sextic twist and Karatsuba complex techniques. -1 is not CNR in $\mathbb{F}_p$. Therefore, we had to find an alternative, QNR and CNR elements in $\mathbb{F}_p$ for the twist parameter z without changing the entire tower construction. Using numerical experiments, we found that the element $-4 \in \mathbb{F}_p$ satisfies the requirements. The cost estimations presented in Table 3 show that the extension field construction τ_2 exhibits less $\mathbb{F}_p$ addition costs in $\mathbb{F}_{p^6}$ than τ_1.

4.2 GMT8 Curve Parameters and Towers

The GMT8-544 curve proposed in [8] with the extension Field $\mathbb{F}_{p^8} = \mathbb{F}_p[x]/(x^8 - 5)$ which only capable with OEF and Type-II AOPF. We present an alternative characteristic p for the GMT8 curve with both OEF and Type-I AOPF construction available which can achieve flexible and efficient implementation. To find such a characteristic, we focus on finding a prime number available with either the Karatsuba complex or type-I AOPF. According to the GMT paper [24], the 2-NAF weight of some parameters is required for efficient computation over the GMT8-544 curves as follows:

$$u : \text{2-naf weight} \leq 5, h_y : \text{ 2-naf weight} \leq 7, h_t : \text{ 2-naf weight} \leq 4 \tag{14}$$

Table 1. GMT6-672 parameters [8]

Param	2-NAF weights	Bit length	Value
u	2	128	`0xefffffffffffffe00000000000000000`
h_t	1	–	`-1`
h_y	4	–	`0xffbbffffffffffffc020`
p	–	672	`0x9401ff90f28bffb0c610fb10bf9e0fefd59211629a7991 563c5e468d43ec9cfe1549fd59c20ab5b9a7cda7f27a0067 b8303eeb4b31555cf4f24050ed155555cd7fa7a5f8aaaaaa ad47ede1a6aaaaaaaab69e6dcb`

In the GMT paper, the evaluated security level of the proposed GMT8-544 curve is 131-bits. We investigated for new parameters which satisfies approximately the 128-bit security level by focusing the characteristic search in the 525 - 544 bit range. Looking for only the characteristic satisfying the condition above for quadratic type-I AOPF and OEF construction could be obtained. The parameters found are denoted as GMT8-542; these are presented in Table 2. The subgroup security and twist subgroup security of our GMT8-542 are the same with original GMT8-544; $\mathbb{G}_1, \mathbb{G}_2$ subgroup-security are confirmed, the twist-subgroup is not secure.

Compared to the original GMT8-544 curve, h_y in the proposed curve has 2 more weights in the 2-NAF. A part from this disadvantage, the proposed GMT8-542 parameters are available only with the type-I AOPF, and 1/3 cost reduction is achieved for the squaring operation in the extension fields. Based on the extension proposed in the GMT paper, we derived a field-towering scheme τ_3 as shown in Table 3. In this case, the element 3 is a QNR in $\mathbb{F}_p$, and the square root of 3 in $\mathbb{F}_{p^2}$ is also a QNR element which makes the quartic twist available for this tower.

We propose a more efficient towering scheme τ_4, where the first subextension field $\mathbb{F}_{p^2}$ is constructed using the type-I AOPF method. A simple QNR element $(1, -1) \in \mathbb{F}_{p^2}$ for the second and third stage OEFs is selected. Since $(1, -1)$ is QNR, the quartic twist is also available in τ_4. Two towers of extension fields and their arithmetic costs for each curve are summarized in Table 3. The newly proposed towers τ_2 and τ_4 exhibit less number of arithmetic operations for the squaring and the cyclotomic subgroup squaring in $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^6}$, and $\mathbb{F}_{p^8}$.

The final exponentiation raising power of $(p^k - 1)/r$ is heavily dependent on the squaring cost in $\mathbb{F}_{p^k}$. We can use two strategies to accelerate the final exponentiation: compressed squaring introduced by Karabina [25] and cyclotomic subgroup squaring [9,26]. Both algorithms are efficient compared with the regular squaring in the extension fields; however, the compressed squaring requires inversion operation in $\mathbb{F}_p$, which could be the bottleneck of pairing computation.

In Table 3, the s_k^{cyclo} is represented by the square in the cyclotomic subgroup of extension field $\mathbb{F}_{p^k}$. s_k^{cyclo} represents the cost of the cyclotomic subgroups

Table 2. GMT8-542 parameters

Param	2-NAF weights	Bit length	Value
u	4	64	`0xffc0000004020002`
h_t	1	–	`-1`
h_y	6	–	`0x7452`

p	–	542	`0x347111bfc75e57d130de7be68437c8d75455d209459d42`
			`1455023bee14df9fe75aa4734686ca3d08c1fa594100d794`
			`21d56c53899ee0f066fad9eb45b0985dbdbba2dcc1`

squaring in $\mathbb{F}_{p^k}$. The cyclotomic subgroup squaring equation for τ_1 and τ_2 was adopted from [26, Sect. 3.2]. For τ_3, we adopt the cyclotomic subgroup squaring equation from [26, Sect. 3] was adopted, whereas for τ_4, the equation from a recent work [9] was selected to prevent the multiplication with $(\omega^2+\omega)^{-1}$ in the $\mathbb{F}_{p^4}$ multiplication.

5 Implementation of Ate Pairing over the GMT6 and GMT8 Curves

In this section, the details of the proposed pairing implementation are presented. Among the proposed towers of the extension fields, τ_2 and τ_4 are the best constructions for the GMT6 and GMT8 curves, respectively. This section provides a detailed calculation of pairing cost based on these towers.

5.1 Implementation of Miller's Algorithm

In previous studies, many sophisticated techniques were proposed to improve the performance of Miller's algorithm. For example, the optimal coordinate system depends on the type of the underlying elliptic curves. Base on the GMT paper [8, Table 5], the homogeneous projective coordinate system (weight[1:1])) for the GMT6-672 curve was adopted. This system was proposed by Costello et al. in [23] and later modified in [22, Section 5].

For the proposed GMT8-542 curve, the Miller's algorithm with the projective coordinate system (weight[1:2]) was adopted. This is also suggested by Costello et al. in [22, Sect. 4]. As a Miller's loop parameter, the GMT6-672 has 129-bit $T = u - 1$ with a 2-NAF weight of 2, whereas the GMT8-542 curve has a 65-bit $T = u - 1$ with a 2-NAF weight $= 4$. The cost of the implemented functions in Miller's loop based on the τ_2 and τ_4 field-towering schemes is summarized in Table 4.

In Table 4, the column "Call" indicates the number of function calls per Miller's algorithm execution. Since DBLLine does not require UPDATE1 in the first loop of Miller's algorithm, UPDATE1 has one less call than DBLLine. Two ADDLine functions are denoted "ADDLine" and "ADDLine′" in Table 4. Due to

$3i_1$ and $1m_1$ precomputation, ADDLine in Miller's loop can be replaced with "ADDLine′". Although the pairings and applications employ "ADDLine′", the only functional with constant P, Q is the subgroup generator of $\mathbb{G}_1$ and $\mathbb{G}'_2$. Thus, pairings without restricting any of the application functionalities were implemented.

5.2 Implementation of Final Exponentiation

In the second part of pairing calculation, the result of Miller's algorithm is raised to the power of $(p^k - 1)/r$. This is also known as final exponentiation $(p^k - 1)/r$ can be separated into two parts; the easy part and the hard part. The complexity of the final exponentiation largely depends on the curve parameters, especially the polynomials of characteristic $p(u)$, order $r(u)$, and Frobenius trace $t(u)$.

Table 3. Arithmetic calculation costs in the tower of the extension fields

Curve and tower	Extension fields	Operation	m_1	s_1	a_1	i_1	Note
GMT6, τ_1 $E(\mathbb{F}_p) : y^2 = x^3 - 1$ $E'(\mathbb{F}_p) : y^2 = x^3 - v^{-6}$	$\mathbb{F}_{p^2} : \mathbb{F}_p[i]/(i^2 - 2)$	m_2	3	0	6	0	
		s_2	2	0	5	0	
	$\mathbb{F}_{p^6} : \mathbb{F}_{p^2}[v]/(v^3 - i)$, where $i^2 = 2$	m_6	18	0	76	0	
		s_6	12	0	47	0	
		s_6^{cyclo}	6	0	37	0	[26] Sect. 3.2
		f_6	4	0	0	0	
		i_6	35	1	102	1	
GMT6, τ_2 $E(\mathbb{F}_p) : y^2 = x^3 - 1$ $E'(\mathbb{F}_p) : y^2 = x^3 - v^6$	$\mathbb{F}_{p^2} : \mathbb{F}_p[i]/(i^2 + 1)$	m_2	3	0	5	0	
		s_2	2	0	3	0	
	$\mathbb{F}_{p^6} : \mathbb{F}_{p^2}[v]/(v^3 - 2i)$, where $i^2 = -1$	m_6	18	0	64	0	
		s_6	12	0	41	0	
		s_6^{cyclo}	6	0	29	0	[26] Sect. 3.2
		f_6	4	0	0	0	
		i_6	36	1	80	1	
GMT8, τ_3 $E : y^2 = x^3 + x$ $E'(\mathbb{F}_{p^2}) : y^2 = x^3 + ix$	$\mathbb{F}_{p^2} : \mathbb{F}_p[i]/(i^2 - 3)$	m_2	3	0	7	0	
		s_2	2	0	5	0	
	$\mathbb{F}_{p^4} : \mathbb{F}_{p^2}[v]/(v^2 - i)$	m_4	9	0	33	0	
		s_4	6	0	25	0	
	$\mathbb{F}_{p^8} : \mathbb{F}_{p^4}[g]/(g^2 - v)$, where $i^2 = 3$	m_8	27	0	121	0	
		s_8	18	0	93	0	
		s_8^{cyclo}	12	0	69	0	[26] Sect. 3.1
		f_8	6	0	0	0	
		i_8	46	1	169	1	
GMT8, τ_4 $E : y^2 = x^3 + x$ $E'(\mathbb{F}_{p^2}) : y^2 = x^3 + v^2x$	$\mathbb{F}_{p^2} : \mathbb{F}_p[\omega]/(\omega^2 + \omega + 1)$	m_2	3	0	4	0	
		s_2	2	0	3	0	Type-I AOPF
	$\mathbb{F}_{p^4} : \mathbb{F}_{p^2}[v]/(v^2 - (\omega^2 + \omega))$	m_4	9	0	26	0	
		s_4	6	0	21	0	
	$\mathbb{F}_{p^8} : \mathbb{F}_{p^4}[g]/(g^2 - v)$, where $(1, -1) \in \mathbb{F}_{p^2}$ $\omega + \omega^p = 1$	m_8	27	0	102	0	
		s_8	18	0	83	0	
		s_8^{cyclo}	12	0	66	0	[9] Sect 3.3
		f_8	9	0	12	0	
		i_8	49	0	132	1	

Using the following equation, the GMT curves feature a very unique and efficient construction [8]:

$$t' \equiv u^i + 1 \equiv p + 1 \pmod{r}, (i = 1) \tag{15}$$

$$j = \frac{p + 1 - t'}{r}, \tag{16}$$

GMT6-672 Final Exponentiation
As mentioned above, the final exponentiation can be separated into two parts such as follows:

$$\frac{p^6 - 1}{r} = (p^3 - 1)(p + 1) \times \frac{(p^2 - p + 1)}{r} \tag{17}$$

The easy part $(p^3 - 1)(p + 1)$ requires two Frobenius endomorphism calculations f_6 for p and p^3. The Frobenius endomorphism for the raised power of $p^k/2$ does not require any multiplication when k is even. Moreover, as shown in Table 3, the OEF nested tower of the extension field only requires 4 **m** for the Frobenius endomorphism f_k.

For the hard part, using the replacement technique given in (17) and (18) where $c = j$ (where $\Phi_6(x)$ is the 6-th cyclotomic polynomial), $\frac{(p^2-p+1)}{r}$ can be broken down to:

$$\frac{\Phi_6(t' - 1)}{r} + (p + t' - 2)c = 1 + (p + t' - 2)c \tag{18}$$

The hard part can be multiplied by a small integer, which does not change the bilinear pairing integrity. In this case, a multiplication by 3 is recommended, so that the polynomial $3c$ does not have any fraction terms, such as

$$3(1 + (p + t' - 2)c) = 3 + 3c(p + u - 1) \tag{19}$$

Table 4. Cost of miller's loop in τ_2 and τ_4

Curve: tower	Function	m_1	s_1	Total **m**	Call
GMT6-672: τ_2	DBLLine	4	7	11 **m**	128
	UPDATE1	25	0	25 **m**	127
	ADDLine	13	2	15 **m**	2
	ADDLine′	12	2	14 **m**	0
	UPDATE2	13	0	13 **m**	2
	Miller's loop	3739	900	4639 **m**	1
GMT8-542: τ_4	DBLLine	26	0	26 **m**	64
	UPDATE1	42	0	42 **m**	63
	ADDLine	44	0	44 **m**	4
	ADDLine′	41	0	41 **m**	0
	UPDATE2	24	0	24 **m**	4
	Miller's loop	4582	0	4582 **m**	1

However, bilinearity could not be achieved using (19) with the polynomial $3c$ provided in Sect. 5.2 (3) of the GMT paper. Thus, our version of $3c$ was recalculated and corrected as follows:

$$3c = ((1 + 3w + 9w^2)(u - 1) + (6w + 9w^2))(u - 1) + 9w + 9w^2, \tag{20}$$

where $w = h_y/2$. We propose an efficient calculation order for $3c$, which is shown in Table 5. However, we realize Nanjo et al. [16] already proposed the same equation in their paper's TABLE IX. The final exponentiation costs using the above $3c$ calculation based on τ_2 are summarized in Table 7. Compare with the original GMT672 final exponentiation hard part, our calculation order reduced $4m_6$ and s_6.

GMT8-542 Final Exponentiation

Similar to the GMT6-672 curve, the power of the GMT8-542 final exponentiation can also be divided into two parts as follows:

$$\frac{p^8 - 1}{r} = (p^4 - 1) \times \frac{(p^4 + 1)}{r} \tag{21}$$

In this case the easy part $(p^4 - 1)$ only requires 1 m_6 and 1 i_6. Using again the replacement technique given in (17) and (18) with parameter $u' = u - 1$. The hard part of GMT8-542 can be broken down as follows:

$$\frac{(p^4 + 1)}{r} = \frac{\Phi_8(t' - 1)}{r} + d(p + t' - 1)(p + (t' - 1)^2) = 1 + d(p + u)(p^2 + u^2) \tag{22}$$

Table 5. Calculation of the raised power of GMT6-672 hard part-$3c$

Computation	Term computed	Cost
Input: $M \in \mathbb{F}_{p^6}, w, u' \in \mathbb{F}_p$ Output: $M^{3c} \in \mathbb{F}_{p^6}$ Temp. var: t_0, t_1, t_2		
$t_0 \leftarrow M^w$	M^w	c_w
$t_1 \leftarrow {t_0}^2$	M^{2w}	s_6^{cyclo}
$t_0 \leftarrow t_0 t_1$	M^{3w}	m_6
$t_1 \leftarrow t_0 M$	M^{3w+1}	m_6
$t_1 \leftarrow {t_1}^w$	M^{3w^2+w}	c_w
$t_2 \leftarrow {t_1}^2$	M^{6w^2+2w}	s_6^{cyclo}
$t_1 \leftarrow t_2 t_1$	M^{9w^2+3w}	m_6
$t_2 \leftarrow t_1 t_0$	M^{9w^2+6w}	m_6
$t_1 \leftarrow t_1 M$	M^{9w^2+3w+1}	m_6
$t_1 \leftarrow {t_1}^{u'}$	$M^{(9w^2+3w+1)u'}$	$c_{u'}$
$t_1 \leftarrow t_1 t_2$	$M^{(9w^2+3w+1)u'+9w^2+6w}$	m_6
$t_1 \leftarrow {t_1}^{u'}$	$M^{((9w^2+3w+1)u'+9w^2+6w)u'}$	$c_{u'}$
$t_1 \leftarrow t_1 t_2$	$M^{((9w^2+3w+1)u'+9w^2+6w)u'+9w^2+6w}$	m_6
$t_0 \leftarrow t_1 t_0$	$M^{((9w^2+3w+1)u'+9w^2+6w)u'+9w^2+9w}$	m_6
return t_0		

According to the GMT paper, thr hard part of embedding degree 8 is multiplied by 4 as follows:

$$4 + 4d(p + u)(p^2 + u^2) \tag{23}$$

Similar to the GMT6-672, $4d$ was recalculated and corrected as follows:

$$4d = ((((4n^2 + 1)u - 4n)u + 4n + 1)u - 4)u + 4n^2, \tag{24}$$

where $n = h_y$. We propose an efficient calculation algorithm for $4d$, as shown in Table 6. The total calculation costs of the final exponentiation are summarized in Table 7. Compare with the original GMT672 final exponentiation hard part, our calculation order increased $2s_6$ reduced $5m_6$.

6 Implementation Results

To confirm the efficiency of the proposed methods, all the towers shown in Table 3 were implemented for ate pairing cost and speed comparison. The software developed computes bilinear pairings based on the algorithms introduced in Sect. 2.3. In this section, the features of the software libraries used are initially introduced. Furthermore, the pairing implementation results with detailed calculation costs are presented.

Table 6. Calculation of the raised power of GMT8-542 hard part-$4d$

Computation	Term computed	Cost
Input: $M \in \mathbb{F}_{p^8}, u, n \in \mathbb{F}_p$ Output: $M^{4d} \in \mathbb{F}_{p^8}$ Temp. var: t_0, t_1, t_2, t_3		
$t_0 \leftarrow M^2$	M^2	s_8^{cyclo}
$t_0 \leftarrow {t_0}^2$	M^4	s_8^{cyclo}
$t_1 \leftarrow {t_0}^n$	M^{4n}	c_n
$t_2 \leftarrow {t_1}^n$	M^{4n^2}	c_n
$t_3 \leftarrow t_2 M$	M^{4n^2+1}	m_8
$t_3 \leftarrow {t_3}^u$	$M^{(4n^2+1)u}$	c_u
$t_3 \leftarrow t_3 {t_1}^{-1}$	$M^{(4n^2+1)u-4n}$	m_8
$t_3 \leftarrow {t_3}^u$	$M^{((4n^2+1)u-4n)u}$	c_u
$t_3 \leftarrow t_3 t_1$	$M^{((4n^2+1)u-4n)u+4n}$	m_8
$t_3 \leftarrow t_3 M$	$M^{((4n^2+1)u-4n)u+4n+1}$	m_8
$t_3 \leftarrow {t_3}^u$	$M^{(((4n^2+1)u-4n)u+4n+1)u}$	c_u
$t_3 \leftarrow t_3 {t_0}^{-1}$	$M^{(((4n^2+1)u-4n)u+4n+1)u-4}$	m_8
$t_3 \leftarrow {t_3}^u$	$M^{((((4n^2+1)u-4n)u+4n+1)u-4)u}$	c_u
$t_0 \leftarrow t_3 t_2$	$M^{((((4n^2+1)u-4n)u+4n+1)u-4)u+4n^2}$	m_8
return t_0		

6.1 Multi-precision Libraries and Implementation Features

As mentioned above, two libraries (mcl [14] and GMP) are used, which are combined in this work. mcl is a library for pairing-based cryptography, mainly supporting the optimal ate pairing over BN curves and BLS12-381 curves. This library is available on almost all x32 and x64 architecture available platforms. The implementation conducted in this work mainly uses the mpn function group of the GNU multiple precision (GMP) Library called by the C++ language, although some core operations such as multiplication, modulo, addition and bit shift are replaced by mcl functions. The multiplication in $\mathbb{F}_p$ is performed using the Montgomery multiplication techniques.

6.2 Pairing Benchmark Results

Miller's algorithm and final exponentiation costs are summarized in Table 8. It can be observed that the proposed towers τ_2 and τ_4 exhibit lower costs than τ_1 and τ_3 by applying all the techniques previously described. Specifically, compared with τ_1 and τ_2, they are addition almost 6% more efficient because of the addition cost reduced Karatsuba complex method. Although, τ_4 exhibits an approximate 2% higher costs due to the specially of type-I AOPF but it has lower addition in total. The implementation results are presented in Table 9. The program was compiled using the Clang++12 with the compile option `-Ofast -march=native`. The benchmarks were obtained using an i7-8700 (base clock 3.2 GHz, boost 4.3 GHz) computer.

Table 7. τ_2 and τ_4 final exponentiation costs.

Curve: tower	Part	m_6	s_6^{cyclo}	f_k	f_k^2	i_6	c_u	${c_{u-1}}^a$	${c_w}^a$	${c_n}^a$	Total **m**
GMT6-672 : τ_2	Easy	2	0	1	0	1	0	0	0	0	77 **m**
	Hard (without 3c)	4	1	1	0	0	1	0	0	0	886 **m**
	3c	8	2	0	0	0	0	2	2	0	2892 **m**
	Total	14	3	2	0	1	1	2	2	0	3855 **m**
GMT8-542 : τ_4	Easy	1	0	0	0	1	0	0	0	0	76 **m**
	Hard (without 4d)	3	2	1	1	0	3	0	0	0	2748 **m**
	4d	6	2	0	0	0	4	0	0	2	4320 **m**
	Total	10	4	1	1	1	7	0	0	2	7144 **m**

[a]GMT6-672:τ_2, the costs for the raised power of $u, u-1$ and w are $c_u = 804$ **m** , $c_{u-1} = 822$ **m** , $c_w = 546$ **m** respectively. For the GMT8-542:τ_4 case the costs for raised power of u and n are $c_u = 876$ **m**, $c_n = 315$ **m** respectively.

Table 8. Pairing total costs

Tower	Miller's algorithm cost	Final Exponentiation cost	Total Pairing cost
τ_1	4902 **m**	3854 **m** $+i_1$	8774 **m** $+i_1$
τ_2	4639 **m**	3855 **m** $+i_1$	8494 **m** $+i_1$
τ_3	4310 **m**	7135 **m** $+i_1$	11445 **m** $+i_1$
τ_4	4582 **m**	7144 **m** $+i_1$	11726 **m** $+i_1$

Table 9. Implementation results obtained using an i7-8700 CPU (3.2 GHz Turbo Boost off, 4.3 GHz on) computer compared with the GMT paper estimation results

Curve	Tower	MP library	Miller's algorithm [μs]	Final exponentiation [μs]	Pairing [μs]	Turbo Boost
GMT6-672	$\tau 1$	mcl	772	722	1494	Off
		mcl	539	503	1042	On
	$\tau 2$	mcl	721	693	1410	Off
		mcl	505	481	987	On
	–	RELIC (estimation)	800	700	1500	Off
GMT8-542	$\tau 3$	mcl	589	1050	1639	Off
		mcl	411	731	1142	On
	$\tau 4$	mcl	569	1050	1616	Off
		mcl	398	730	1120	On
GMT8-544	–	RELIC (estimation)	600	900	1500	Off

The proposed pairing computation over the GMT6-672 and GMT8-542 curves is achieved in 0.99 and 1.12 ms, respectively, with Turbo Boost enabled. The construction of tower τ_2 is 5.2% faster than τ_1. Moreover, the construction of tower τ_4 is 2% faster than that of τ_3 due to the addition reduction. It is also observed that τ_4 has this feature which does not require any squaring in $\mathbb{F}_p$, which is an interesting result. A comparison with the GMT paper estimation results is also provided. A comparison between our implementation result and the GMT paper estimation results is provided in Table 9.

Our implementation results are Benchmarked in the same environment as the GMT paper estimation results. It is observed that the GMT6-672 curve with tower τ_2, our results are achieved faster by approximately 6% than the GMT paper estimation results. For the GMT8-542 curve with tower τ_4, our results are achieved by 0.116 ms slower than the GMT paper estimation results.

7 Conclusion and Future Work

The following results can be concluded:

1. After reviewing the GMT6 and GMT8 curve parameters and classes of the existed extension fields, two different types of towers for newly emerged pairing-friendly curves were proposed. Since the GMT6 curve original characteristic is considered sufficiently efficient, a unique and efficient tower construction consisting of nested OEF was proposed. This scheme is suitable to the minimal addition karatsuba complex method. For the GMT8 curve, the existed parameters cannot achieve the best performance. Thus, we reexplored the characteristic and proposed a new set of parameters with only 2 less bits suitable with the type-I AOPF.
2. To the best of the authors' knowledge, complete and efficient software implementations of pairings for the GMT6 and GMT8 curves have not been reported. The cost of the recommended Miller's algorithm with a twist on

the available rational functions of pairings was presented. For the final exponentiation, the polynomials were recalculated, and the costs for both curves were re-evaluated. The implementation results suggested that the GMT6 and GMT8 curves are excellent and efficient candidates for 128-bit security pairing applications.

Acknowledgments. A part of this work was supported by the Cabinet Office (CAO), Cross-ministerial Strategic Innovation Promotion Program (SIP), "Cyber Physical Security for IoT Society", JPNP18015 (Funding agency: NEDO). The authors thank Tadanori Teruya of CPSEC, AIST, for his assistance with the towering construction, and Yuki Nanjo of Okayama University for her comments on improving the manuscript.

References

1. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_26
2. Naehrig, M., Lauter, K., Vaikuntanathan, V.: Can homomorphic encryption be practical?. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop (CCSW 2011). Association for Computing Machinery, pp. 113–124 (2011)
3. Abdolmaleki, B., Baghery, K., Lipmaa, H., Zając, M.: A Subversion-Resistant SNARK. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10626, pp. 3–33. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_1
4. Boxall, J., El Mrabet, N., Laguillaumie, F., Le, D.-P.: A variant of Miller's formula and algorithm. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 417–434. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17455-1_26
5. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006). https://doi.org/10.1007/11693383_22
6. Barreto, P.S.L.M., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Cimato, S., Persiano, G., Galdi, C. (eds.) SCN 2002. LNCS, vol. 2576, pp. 257–267. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36413-7_19
7. Kim, T., Barbulescu, R.: Extended tower number field sieve: a new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 543–571. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_20
8. Guillevic, A., Masson, S., Thomé, E.: Cocks-Pinch curves of embedding degrees five to eight and optimal ate pairing computation. Des. Codes Crypt. **88**(6), 1047–1081 (2020). https://doi.org/10.1007/s10623-020-00727-w
9. Lavice, A., Mrabet, N.E., Berzati, A., Rigaud, J., Proy, J.: Hardware implementations of pairings at updated security levels. In: Grosso, V., Pöppelmann, T. (eds.) CARDIS 2021. LNCS, vol. 13173, pp. 189–209. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-97348-3_11
10. Benger, N., Scott, M.: Constructing tower extensions of finite fields for implementation of pairing-based cryptography. In: Hasan, M.A., Helleseth, T. (eds.) WAIFI 2010. LNCS, vol. 6087, pp. 180–195. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13797-6_13

11. Nogami, Y., Saito, A., Morikawa, Y.: Finite extension field with modulus of all-one polynomial and representation of its elements for fast arithmetic operations. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **86**(9), 2376–2387 (2003)
12. Bailey, D.V., Paar, C.: Optimal extension fields for fast arithmetic in public-key algorithms. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 472–485. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055748
13. Scott, M. (2009): A note on twists for pairing friendly curves. Personal. ftp://ftp.computing.dcu.ie/pub/resources/crypto/twists.pdf
14. Mitsunari, S.: A portable and fast pairing-based cryptographic library. https://github.com/herumi/mcl. Accessed 14 Jan 2021
15. Nanjo, Y., Kodera, Y., Matsumura, R., Shirase, M., Kusaka, T., Nogami, Y.: Evaluation of a pairing on elliptic curves of embedding degree 15 with type-II all-one polynomial extension field of degree 5. In: 2020 Symposium on Cryptography and Information Security (2020)
16. Nanjo, Y., Khandaker, M.M., Kusaka, T, Nogami, Y.: Consideration of efficient pairing applying two construction methods of extension fields. In: 2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW), pp. 445–451 (2018)
17. Kato, H., Nogami, Y., Yoshida, T., Morikawa, Y.: A multiplication algorithm in Fpm such that p¿ m with a special class of gauss period normal bases. EICE Trans. Fundam. Electron. Commun. Comput. Sci. **92**(1), 173–181 (2009)
18. Kato, H., Nogami, Y., Yoshida, T., Morikawa, Y.: Cyclic vector multiplication algorithm based on a special class of Gauss period normal basis. ETRI J. **29**(6), 769–778 (2007)
19. Karatsuba, A., Ofman, Y.: Multiplication of many-digital numbers by automatic computers. In: Proceedings of the USSR Academy of Sciences, vol. 145, pp. 595–596 (1963). Translation in the academic journal Physics-Doklady, pp. 293–294
20. Montgomery, P.L.: Five, six, and seven-term Karatsuba-like formulae. IEEE Trans. Comput. **54**(3), 362–369 (2005)
21. Toom, A. L.: The complexity of a scheme of functional elements realizing the multiplication of integers. In: Soviet Mathematics Dok-Lady, vol. 3, no. 4, pp. 714–716 (1963)
22. Costello, C., Lange, T., Naehrig, M.: Faster pairing computations on curves with high-degree twists. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 224–242. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_14
23. Costello, C., Hisil, H., Boyd, C., Gonzalez Nieto, J., Wong, K.K.-H.: Faster pairings on special weierstrass curves. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 89–101. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03298-1_7
24. Masson, S.: Cocks-Pinch variant. https://gitlab.inria.fr/smasson/cocks-pinch-variant. Accessed 14 Jan 2021
25. Karabina, K.: Squaring in cyclotomic subgroups. Math. Comput. **82**(281), 555–579 (2013)
26. Granger, R., Scott, M.: Faster squaring in the cyclotomic subgroup of sixth degree extensions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 209–223. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_13