A taxonomy of pairings, their security, their complexity

Razvan Barbulescu[1], Nadia El Mrabet[2], and Loubna Ghammam[2]

[1] CNRS, Sorbonne université, Univ Paris Diderot, France
razvan.barbulescu@imj-prg.fr
[2] Mines Saint-Etienne, CEA-Tech, Centre CMP, Departement SAS, France
nadia.el-mrabet@emse.fr loubna.ghammam@itk-engineering.de

Abstract. A recent NFS attack against pairings made it necessary to increase the key sizes of the most popular families of pairings : BN, BLS12, KSS16, KSS18 and BLS24. The attack applies to other families of pairings but not to all. In this paper we compute the key sizes required for more than 150 families of pairings to verify if there are any other families which are better than BN. The security estimation is not straightforward because it is not a mathematical formula, but rather one has to instantiate the Kim-Barbulescu attack by proposing polynomials and parameters.
After estimating the practical security of an extensive list of families, we compute the complexity of the optimal Ate pairing at 128 and 192 bits of security. For some of the families the optimal Ate has never been studied before. We show that a number of families of embedding degree 9, 14 and 15 are very competitive with BN, $BLS12$ and $KSS16$ at 128 bits of security. We identify a set of candidates for 192 bits and 256 bits of security.

Keywords: Discrete Logarithm Problem; Number Field Sieve; Elliptic Curves; Pairings

1 Introduction

Pairings are a crucial ingredient in a series of public-key protocols. After Joux' [Jou00] tri-partite Diffie-Hellman key echange and the identity-base encryption scheme of Boneh and Franklin [BF01], it became clear that pairings can have applications which could not be obtained with any other mathematical primitives. Many more public-key protocols followed, including short signatures [BLS04], a wide variety of aggregate, instance and verifier-local revocation signatures [BGLS03,BBS04,JN09], broadcast encryption [BGW05], cloud computing [AFGH06], privacy enhancing environments [She10], deep package inspection over encrypted traffic [SLPR15,CDK⁺17] and many others. The NIST [MC11] pilots a project dedicated to pairings. Efficient implementations of pairings [BLM⁺09], [BGDM⁺10], [GAL⁺12], [UW14], [KNG⁺17] made them interesting for industrial development [Tea05,Cha08].

At a high level, a pairing is a non-degenerated and bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of an elliptic curve and $\mathbb{G}_3$ is a sub-group of a finite field.

The security of pairing-based cryptography relies on one side on the discrete logarithm problem (DLP) over $\mathbb{G}_1$ (and consecutively over $\mathbb{G}_2$) which are elliptic curves, we call this the curve side security and note that it is very well understood on the classical computers (pairing-based cryptography is not resisting to quantum computers, whose feasibility is not known to this day). On the other side, it relies on the discrete logarithm problem over $\mathbb{G}_3$ which is the multiplicative group of a finite field, this is the field side security.

The hardness of computing discrete logarithms in a finite field is difficult to evaluate. In a first time one used the approximation that its cost is the same as of that of factoring, which is done with a variant of the same algorithm : the number field sieve (NFS). Hence, the first key sizes proposed for pairings [Len01] were such that $\log_2 \#\mathbb{G}_3$ matches the required bitsize for an RSA module offering the same security level (the RSA hypothesis). In a second time, one computed the cost using a theoretical upper bound [MSS16],[SG18] (the asymptotic hypothesis). In a recent article, Barbulescu and Duquesne [BD18] made a precise real-life analysis with no theoretical assumption (this is practical estimation). Hence, they found the optimal parameters for each variant of NFS and obtained key sizes which can be used in a future standardization for 5 families of pairing friendly elliptic curves. Many more families exist and our article, together with very recent other works [GS19],[GMT19], extends these key evaluations to other families.

The use of approximations was not a problem before 2013. Indeed, the difficulty of the DLP in fields $\mathbb{F}_p$ with p prime is the same as that of factoring an RSA module of the same bit size as p. The NFS variants used to attack pairings were either analogies of the one used for $\mathbb{F}_p$, as the function field sieve for the pairings of small characteristic, or cumbersome adaptations of NFS to the case of $\mathbb{F}_{p^k}$ when p is non-small and $k > 1$. However, the small characteristic pairings are now forbidden [Eur13, page 32] because of a series of attacks culminating with a quasi-polynomial algorithm [BGJT14]. A series of new variants of NFS between 2013 and 2016 [JP13,BGGM15,BGK15] showed that the finite fields $\mathbb{F}_{p^k}$ can actually be easier than the prime case, from an asymptotic point of view. Kim and Barbulescu proposed a variant of NFS which either encompass the previous variants or it improves on them [KB16]. The Kim-Barbulescu attack depends highly on two specific pairing-friendly elliptic curves parameters: on one side on the parametrization of the characteristic and on the other side on the embedding degree. The precise estimation of Barbulescu and Duquesne [BD18] concluded that also from a practical point of view, certain pairings require a larger bitsize than prime fields for the same level of security. In this work we extend the list of pairing families from 5 in [BD18] to over 150 families.

The starting point of our work is the remark that the fastest pairings before the Kim-Barbulescu attack, as BN, KSS and BLS, are precisely those which are the most affected by the attack. Indeed, the complexity of the NFS variants is well-expressed using the L-notation:

$$L_N[c] = \exp((c/9)^{\frac{1}{3}}(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}})^{1+o(1)}.$$

The constant c takes various values depending on the variant of NFS, a list of these variants being made in Section 3.2. We have then four situations for the DLP in a field $\mathbb{F}_{q^k}$, represented in Figure 1:

- When k is prime and q doesn't have a polynomial form, at a constant bit size of q^k, c is 64 when k is small (TNFS or NFS-GJL) and 96 when k is large (NFS-Conj).
- When k is prime and q has a special form, at a constant bit size of q^k, c is 32 when k is small (STNFS or Joux-Pierrot) and 64 otherwise (Joux-Pierrot).
- When k is composite and q doesn't have a polynomial form, at a constant it size of q^k, c is 64 when k is small (NFS-GJL or TNFS) and 48 when k is large (exTNFS-Conj).
- When k is composite and q has a polynomial form, c is always 32 (STNFS or Joux-Pierrot if k is small and SexTNFS otherwise).

Hence, the most popular pairings (BN, KSS16, KSS18, BLS12 and BLS24) have q of polynomial form and k composite, so they correspond to the value $c = 32$, which is the lowest in the diagram. Note that the Appendix B of [BD18] gives arguments to support that no variant of NFS can have a lower value of c.

Our main purpose is to analyze the efficiency of the new attack [KB16] when applied to less popular pairings. We identify families where the real-life cost of the Kim-Barbulescu attack is higher

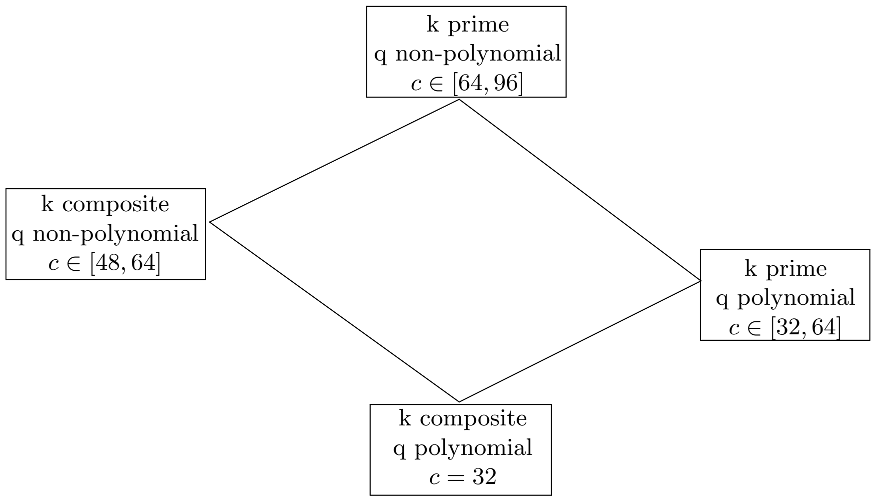

Fig. 1: Representation of the four cases of finite fields $\mathbb{F}_{q^k}$ with respect to the constant c such that the complexity of the fastest NFS attack is $L[c]$.

than for BN, KSS and BLS and hence one can use smaller key sizes for the same security level. Further, smaller key sizes correspond to shorter computation loops and faster real-life timings.

Our contribution

We make an extensive literature inspection to find as many pairing-friendly families as possible. The main reference is the taxonomy [FST10] whose title we copy, but we discovered some families [DCC05],[LZZW08] which weren't included in that work. We also add a small number of families which were published after the taxonomy : [Dry11],[SG18]. Before the key sizes had to be corrected, the BN family was much faster and received much more attention than the other families in the taxonomy, some of which remained to the status of theoretical formulae. Three recent works [FK18,ZX18,FM18] tackle the problem of proposing numerical examples of elliptic curves from each family which correspond to classical levels of security (128, 192 and 256). However, they still make the asymptotic hypothesis that we explained above. We make an extensive analysis of more than 150 families and find the exact parameters for each of them. We emphasize that for some families of high embedding degree it is impossible to find small parameters so one cannot have 128 bits of security without having a larger number, say 150 bits. With the precise key sizes in hand we proposed precise implementation algorithms for all the afore mentioned families. For some of them, for example of prime embedding degree, we are in virgin territory as these families have been considered to be slow; we concluded that they still are. For many families the asymptotic hypothesis gives sizes which are close to being enough and it is no problem to slightly increase the parameters in order to fill the contract of the security level. For other families, like BLS k=27, the corrected key sizes with the practical estimation are smaller than the ones obtained with the RSA hypothesis or the asymptotic hypothesis. This allows us to find a series of families which are faster than BN.

The article is organized as follows. In Section 2, we recall the basic notations on pairings, present the classical optimizations of the implementation and recall the various constructions of pairings. In Section 3, we draw the big lines of the NFS algorithm, recall what are the choices for an attacker and compute the updated key sizes for a large number of families. For each family, we construct pairings and evaluate the cost of Miller's loop, first in arithmetic then in binary operations, at 128 bits (Section 4) and respectively 192 and 256 bits of security (Section 5). Then, in Section 6 we present the final exponentiation complexity for the Optimal Ate pairings in some of proposed curves, and obtain the overall cost. We conclude in Section 7.

2 Some background on pairings

2.1 Definition of pairings

We briefly recall here elementary definition on pairings [Wei40]. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, with q a large prime integer. We denote by $\mathcal{O}$ the neutral element of the additive group law over E. The elliptic curve is described in the Weierstrass model:

$E(\mathbb{F}_q) = \{(x, y), y^2 = x^3 + ax + b, a, b \in \mathbb{F}_q\}$.

Let r be a large prime divisor of the group order $\sharp E(\mathbb{F}_q)$ and k the embedding degree of E with respect to r, i.e. the smallest integer k such that r divides $q^k - 1$.

The Weil [Wei40] and the Tate [Tat63] pairings are constructed using the Miller algorithm [Mil04]. For the Ate, twisted Ate [HSV06], optimal pairing [Ver10] and pairing lattices [Hes08], the most efficient pairings are constructed on the Tate model. Hence, we only recall here the definition of the reduced Tate pairing, a more complete definition being given in [BSS99, §IX.5].

Definition 1 (Tate pairing). *Let $E(\mathbb{F}_q)$ be an elliptic curve over the finite field $\mathbb{F}_q$ for q a large prime number. Let r be a prime divisor of card$(E(\mathbb{F}_q))$. Let k be the embedding degree of E relatively to r. Let $\mathbb{G}_1 = E(\mathbb{F}_q)[r]$, $\mathbb{G}_2 = E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ and $\mathbb{G}_3 = \{\mu \in \mathbb{F}_{q^k}$ such that $\mu^r = 1\}$. The reduced Tate pairing is defined as*

$$e_T : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3,$$
$$(P, Q) \to f_{r,P}(Q)^{\frac{q^k-1}{r}}$$

where $f_{r,P}(Q)$ is the Miller function defined by the divisor $D = r(P) - (rP) - (r - 1)(\mathcal{O})$.

The Miller function is computed through the Miller's algorithm [Mil04], which is constructed on the double and add scheme using the construction of rP and based on the notion of divisors. We only give here the essential elements for the pairing computation.

The Miller algorithm constructs the rational function $f_{r,P}$ associated to the point P, where P is a generator of $\mathbb{G}_1$; and at the same time, it evaluates $f_{r,P}(Q)$ for a point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$.

The final exponentiation is used to ensure the uniqueness of the resulting value of two equal pairing computations (e.g. $e(P, [2]Q) = e([2]P, Q)$). The final exponentiation maps the result of the Miller algorithm into the group formed by the r^{th} roots of unity in $\mathbb{F}_{q^k}^*$.

2.2 Optimizations for pairings

The optimisations of pairings rely on an accurate choice of the embedding degree, the parametrization family of elliptic curves, the use of a twist for $E(\mathbb{F}_{q^k})$, the research for particular curves inside the chosen family.

Choice of the embedding degree The most general optimisations for a pairing implementation are obtained when k is chosen to have only small prime factors, more particularly when k is a product of powers of 2 and 3 [EJ17]. This property allows the extension field $\mathbb{F}_{q^k}$ to be constructed using tower field extensions. The interest of using tower field extensions is an optimization of the arithmetic. In particular, the multiplication over $\mathbb{F}_{q^k}$ can be constructed using intermediate multiplications on the floor of the tower field extension.

The pairing friendly elliptic curves which are the most interesting for implementation purposes are obtained from families, a taxonomy of which was made by Freeman, Scott and Teske in [FST10], to which we add a very recent construction [SG18].

Existence of twisted elliptic curve An important trick when computing a Tate-like pairing is the elimination of denominators. This is possible when k is a multiple of 2 [KM05] or 3 [BELL10] together with the use of a twisted elliptic curve. An elliptic curve $E/\mathbb{F}_q$ of embedding degree k is said to have a twist of degree d if d is a factor of k and there exists an elliptic curve $E'/\mathbb{F}_{q^{k/d}}$ which is $\mathbb{F}_{q^k}$-birationally isomorphic to $E/\mathbb{F}_{q^{k/d}}$. The larger d is, the faster the pairing is because one can replace the operations over $E(\mathbb{F}_{q^k})$ by operations over $E(\mathbb{F}_{q^{k/d}})$ using the embedding map into $E(\mathbb{F}_{q^k})$. The existence of a twist relies on the value of the DM discriminant Δ (if D is the squarefree part of $t^2 - 4q$ we set $\Delta = -D$ if $D \equiv 1 \pmod 4$ and $-4D$ otherwise; D il also call discriminant abusively). If $\Delta = 3$ and 3 (resp. 6) divides k, we can use a twist of degree 3 (resp. 6). If $\Delta = 4$ and 4 divides k, then we can use a quartic twist $d = 4$. Else, if k is even, we can use a quadratic twist $d = 2$.

Choice of parameters inside a family A family of pairing friendly elliptic curves with embedding degree k is given by a triple $(q(x), r(x), t(x))$ of polynomials with coefficients in $\mathbb{Q}$. In this representation, $q(x)$ is the characteristic of the finite field, $r(x)$ a prime factor of $\mathrm{Card}(E(\mathbb{F}_q))$ and $t(x)$ is the trace of the elliptic curve. If u is an integer such that $q(u)$ and $r(u)$ are prime numbers, then there exists an elliptic curve with embedding degree k and parameters $(q(u), r(u), t(u))$. The integer u is used in the exponent in the Miller loop, in the final exponentiation, and it can have a great impact on the $\mathbb{F}_{q^k}$ arithmetic [DEHR18]. For this reason, u should have a NAF weight as small as possible in order to improve the efficiency of the pairing computation. Once we have found an integer u such that $q(u)$ and $r(u)$ are prime integers, we have to construct the equation of the elliptic curve. This can be done thanks to the complex multiplication (CM) method [FST10]. There exists several models for elliptic curves, but the most efficient computation of pairings are obtained using Weierstrass model: $E : y = x^3 + ax + b$ with $a \in \{0, -3\}$ and $b \in \mathbb{F}_q$.

As the final exponentiation is the same for every pairings, the goal is to obtain the shortest Miller loop. In practice, the reduction of Miller's loop is performed using the definition of optimal pairing [Ver10]. For example, the best results of implementation were obtained for the optimal Ate pairing over BN curves and parameters of Hamming weight at most four [UW14,KNG+17,AFG+17]. When different curves made difficult the decision of which one is more efficient, we discuss on $\log_2(q^k)$. Indeed, this value is the size of the extension field in which we perform the final exponentiation, but it is also a rough estimation of the size of the exponent. As a consequence, the smaller size should be the better.

Last but not least, when choosing the elliptic curve, one must take into consideration the subgroup security problem [BCM+15]. This can demand to modify the value of the parameter u and doesn't modify the performances.

2.3 Construction of pairing-friendly elliptic curves

The construction of pairing-friendly elliptic curve is difficult. An elliptic curve $E/\mathbb{F}_q$ is pairing-friendly when the the embedding degree is not too large and $\#E(\mathbb{F}_q)$ admits a large prime divisor. Furthermore, in order to resist to the subgroups attack, the order of the elliptic curve should not admit small prime factors [BCM+15]. Such elliptic curves are rare and needs very specific construction. The article [FST10] is a nice survey and, to our knowledge, the only complements in the literature are [Dry11] and [SG18].

Let us briefly recall the existing constructions. In order to construct pairings of embedding degree k one starts by searching for integers q, r, t and D such that q and r are primes, there exists integers t and y such that $4q = Dy^2 + t^2$ and r divides both $\Phi_k(t + 1)$ and $q + 1 - t$. These integers are used to compute the equation of an elliptic curve $E/\mathbb{F}_q$ which has a point of order r over $\mathbb{F}_q$ and embedding degree k using the CM method [Mor91]. Since the cost of this last step grows rapidly with D one usually fixes D to integer values in $[-3, 3]$. Hence all the pairings in the taxonomy fits in one of the following categories:

- Supersingular curves (Sec. 3 of the taxonomy). $k \leq 2$ or small characteristic and $k \in \{4, 6, 12\}$.
- Cocks-Pinch and Dupont-Enge-Morain (Sec 4. of the taxonomy) One can use it for any pair (k,D), but for security levels between 128 and 256 the number of pairings is small and there might be no pairing for certain values of k. Note also that $\log q \approx 2 \log r$.
- Sparse families (Sec. 5 of the taxonomy and Drylo [Dry11]). One can use it for $k \in \{2, 3, 4, 6, 8, 9, 10, 12, 15, 28, 30\}$ but the values of D are either restricted or are different for each pairing.
- Complete families (Sec. 6 of the taxonomy [FST10] and the work of Scott and Guillevic [SG18]). Any pair (k,D) is possible, the generation is fast. The prime q is equal to $q(u)$ where q is a polynomial. The values u which give pairings become more rare when k increases.

2.4 Existence of twists

As recalled in Section 2.2, twists determine the speed of Miller's algorithm. The number of twists is given by the following rules (cf. Prop 2 in [HSV06]):

- $\Delta = 3$ and 3 (resp. 6) divides k, we can use a twist of degree 3 (resp. 6).
- $\Delta = 4$ and 4 divides k, then we can use a quartic twist $d = 4$.
- when k is even, we can at least use a quadratic twist $d = 2$, otherwise a quartic or sextic.
- For others combination we cannot use a twist, in particular for prime embedding degree.

construction in [FST10]	embedding degree k	CM discriminant	twist degree d
construction 6.2	$k \equiv 2[4]$	$\Delta = 4$	$d = 2$
	$k \equiv 1[4]$	$\Delta = 4$	$d = 1$
construction 6.3	$k \equiv 2[4]$	$\Delta = 4$	$d = 2$
	$k \equiv 1[4]$	$\Delta = 4$	$d = 1$
construction 6.4	$k \equiv 4[8]$	$\Delta = 4$	$d = 4$
construction 6.6(BLS)	$k \equiv 0[6]$	$\Delta = 3$	$d = 6$
	$k \equiv 3[6]$	$\Delta = 3$	$d = 3$
	$k \equiv 2, 4[6]$	$\Delta = 3$	$d = 2$
	$k \equiv 1, 5[6]$	$\Delta = 3$	$d = 1$
construction 6.7	$k \equiv 0[6]$	$\Delta = 8$	$d = 2$
	$k \equiv 3[6]$	$\Delta = 8$	$d = 1$
construction 6.8 (BN)	$k = 12$	$\Delta = 3$	$d = 6$
construction 6.11 (KSS16)	$k = 16$	$\Delta = 4$	$d = 4$
construction 6.12 (KSS16)	$k = 18$	$\Delta = 3$	$d = 6$
construction 6.13 (KSS32)	$k = 32$	$\Delta = 4$	$d = 4$
construction 6.14 (KSS36)	$k = 36$	$\Delta = 3$	$d = 6$
construction 6.15 (KSS40)	$k = 40$	$\Delta = 4$	$d = 4$
Scott-Guillevic (KSS54)	$k = 54$	$\Delta = 3$	$d = 6$
construction 6.20	$k \equiv 1[4]$	$\Delta \notin \{3, 4\}$	$d = 1$
construction 6.24	$k \equiv 0[4]$	$\Delta \notin \{3, 4\}$	$d = 2$
construction 5.3	$k = 10$	$\Delta \notin \{3, 4\}$	$d = 2$
Drylo [Dry11]	$k \in \{10, 12, 28, 30\}$	$\Delta \notin \{3, 4\}$	$d = 2$
	$k \in \{9, 15\}$	$\Delta \notin \{3, 4\}$	$d = 1$

3 Overview of the NFS attacks

The extended tower number field sieve (exTNFS) encompasses all the variants of NFS. Let us present briefly the algorithm with a special care on the choices that can be made by an attacker.

3.1 Big lines of the algorithm

At a high level, exTNFS on $\mathbb{F}_{q^k}$ proceeds as follows. Let κ and η be two divisors of k so that $k = \kappa\eta$. Let $h(t)$ be a polynomial in $\mathbb{Z}[t]$ which is irreducible modulo q of degree η, and call ω a root of $h(t)$ in $\mathbb{F}_q[t]/\langle h \rangle$. Then select two polynomials $f(t, x)$ and $g(t, x)$ in $\mathbb{Z}[t, x]$ such that $f(\omega, x)$ and $g(\omega, x)$ have a common irreducible factor of degree κ in $\mathbb{F}_q(\omega) = \mathbb{F}_{q^\eta}$. This step, called polynomial selection, takes a negligible time but determines the cost of the whole algorithm.

In the sieving stage, for a given parameter A, one considers the pairs $(a(t), b(t)) \in \mathbb{Z}[t]^2$ of degree less than η such that $\max(\|a\|_\infty, \|b\|_\infty) \leq A$. We call norms of (a, b) the integers $N_f(a, b) = \mathrm{Res}_t(\mathrm{Res}_x(a(t) - xb(t), f(t, x)), h(t))$ and $N_g(a, b) = \mathrm{Res}_t(\mathrm{Res}_x(a(t) - xb(t), g(t, x)), h(t))$. Given a parameter B, the sieving stage outputs the list of (almost) all pairs (a, b) such that $N_f(a, b)$ and $N_g(a, b)$ are B-smooth, i.e. all their prime factors are less than B.

In the linear algebra stage, the goal is to solve a linear system having twice as many elements as primes less than B (the number of prime ideals in the number fields of f and g of norm less than B). This is done in two steps : filtering where the size of the matrix is greatly reduced and the proper linear algebra computations where the obtained linear system is solved. Due to heuristic arguments in [BD18], the filtering stage reduces the size of the matrix by a factor $\log_2 B$ and the cost of the linear algebra is $2^7 B^2 / (\log(B) \log_2 B)^2$.

The results of the linear algebra allow to compute any discrete logarithm in $\mathbb{F}_{q^k}$. Since this step is much faster than the sieving and the linear algebra stages, we neglect it in the complexity analysis.

3.2 Identifying the best attacks

According to Barbulescu and Duquesne [BD18], the cost of (S)exTNFS is described by the following equation:

$$\text{cost} = \frac{2B}{\mathcal{A}\log B}\,\rho\left(\frac{\log_2(N_f)}{\log_2 B}\right)^{-1}\rho\left(\frac{\log_2(N_g)}{\log_2 B}\right)^{-1} + 2^7\frac{B^2}{\mathcal{A}^2(\log B)^2(\log_2 B)^2}, \tag{1}$$

where ρ is Dickman's function and $\mathcal{A}$ is the number of automorphisms of h multiplied by the number of commun number of automorphisms of f and g (which can be upper bounded by $\eta\kappa/\gcd(\eta,\kappa)$). The validity condition is that the number of relations is larger than the cardinality of the factor base, which is as follows:

$$\frac{(2A+1)^{2\eta}}{2w}\cdot\rho\left(\frac{\log_2(N_f)}{\log_2 B}\right)\rho\left(\frac{\log_2(N_g)}{\log_2 B}\right) \geq \frac{2B}{\log(B)}, \tag{2}$$

where ω is the half of the number of roots of unity of h.

We are almost dne except that we didn't see how to select f, g and h. The values of $\mathcal{A}$ and ω are a consequence and their choice is explained in [BD18].

Polynomial selection The choice of the polynomials f and g for NFS in $\mathbb{F}_{q^k}$ was the object of many works. When q has a polynomial form one can obtain a product $N_f N_g$ which is much smaller than in the general case. This is emphasized by putting an S, for special, before the name of each version of NFS : SNFS, STNFS or SexTNFS.

The special case Let $P \in \mathbb{Z}[x]$ and $u \in \mathbb{Z}$ be such that $q = P(u)$ and $\|P\|_\infty = O(\log(q^k)$. When k is small or prime one can use STNFS [BGK15], i.e. h an irreducible polynomial of degree k, $f = P(x)$ and $g = x - u$, or Joux-Pierrot [JP13], i.e. $h = x$ (no tower), $f = P(x^k + S(x))$ and $g = x^k + S(x) - u$ where $S(x)$ is a polynomial of degree less than k. When k is large and can be written as $k = \kappa\eta$, one can use SexTNFS [KB16]: one choses h to be an irreducible polynomial of degree η, $f(t,x) = P(x^\kappa + S(x) + t)$ and $g(t,x) = x^\kappa + S(x) + t - u$. When $\gcd(\kappa,\eta) = 1$ one can drop t in the definitin of f and g.

The case of arbitrary finite fields All primes q, of polynomial or non-polynomial form, must withstand the variants of NFS for the general case. When k is small or prime one uses either TNFS [BGK15], i.e. h is an irreducible polynomial of degree k and f and g are chosen by the "base m" method or the two algorithms of Kleinjung [Kle06],[Kle08], or one uses a classical variant, i.e. $h = x$ (no tower) and any of the methods of polynomial selection: GJL [BGGM15, Sec. 3.2],[Mat06], JLSV$_1$ [JLSV06, Sec 3.2], JLSV$_2$ [JLSV06, Sec 3.1], Sarkar and Singh's algorithms A,B,C,D [SS16a,SS16c,SS16b] and the Conjugation method [BGGM15, Sec 3.3]. When k is large and can be written as $k = \kappa\eta$, one uses exTNFS [KB16]: one selects f and g adequated for DLP computations in $\mathbb{F}_{q^\kappa}$ using the afore mentioned methods and then sets h equal to an irreducible polynomial of degree η. If $\gcd(\kappa,\eta) \neq 1$, one follows [JK16] and replaces the polynomials with $f(x+t)$ and $g(x+t)$.

Optimizing parameters of for NFS attacks For each construction of pairings and for each of the security levels 128, 192 and 256, we generated pairings which guarantee that security on the curve side. Then, for each possible choice of h, f and g, we solved the optimization problem consisting in minimizing the cost in Equation (1) under the validity condition of Equation (2). For each value of $\log_2 A$ and $\log_2 B$ up to a precision of 0.01 we estimated experimentally N_f and N_g on a sample of 3000 pairs (a, b) chosen randomly in the sieving space. If the field side security is not sufficient, we increase the size of $\log_2 r$ and start over. The complete computations took more than 1 CPU year. We summarize the results in the electronic complement (not included in the submitted version for anonymity reasons), as well as in the next section in the tables associated to each family.

3.3 An example of key size computations : MNT of embedding degree 6

Let us consider the family of Section 3.3 of the taxonomy [FST10] : the base field is $\mathbb{F}_q$ where q is a prime of the form $q(u) = 4u^2 - 1$, the elliptic curve order $\#E(\mathbb{F}_q)$ is $r(u) = 4u^2 - 2u + 1$ and the embedding degree equals 6, so the target of the pairing is the multiplicative group of $\mathbb{F}_{q^6}$. The polynomial form of q is important, and we must compute all the manners to write $q(u)$ as a polynomial with small coefficients. In the case of MNT 6 we take, $v = 2u$ and $P(v) = v^2 + 1$ so that $P(v) \equiv 0 \pmod{q(u)}$. For many families one takes $v = u^2$ or $v = u + \frac{1}{u}$ but the only manner to find all the possibilities is to compute the subfields of the number field of $q(u)$.

Given a security level s, e.g. 128 bits, we compute the real roots of the polynomial $r(u) - 2^s$. For integers u close to such a root, we compute integers $q(u)$ until we find primes. For the families of large embedding degree, the bit size of u might be increased in order to find primes; this is not the case for MNT. Then we test all the possible choices of polynomials f, g, h. For example, at 128 bits of security, we find that the best choice is $h = t^2 - t - 1$, $f = P(x^3)$ and $g = x - v(u)$. For each bit size of $\log_2 A$ and $\log_2 B$ up to a precision of 0.1, we compute the size of $\log_2 N_f$ and $\log_2 N_g$ using a sample of 3000 pairs $(a(t), b(t)) \in Z[x]$ with coefficients bounded by A and degree less than $\deg h$. We obtain that $\log_2 A = 31.2$ and $\log_2 B = 54$ corresponds to $\log_2 N_f = 369.8$ and $\log_2 N_g = 439.8$, which satisfies Equation (2). Plugging everything in Equation (1), we find a cost $2^{95.17}$. Since the security on the field side is not enough, we increase the security level on the curve side until we find a security of 2^{128}. This occurs when the field size $\log_2(q^6)$ equals 4032, or equivalently $\log_2 q = 672$ and $\log_2 u = 334$. This corresponds perfectly with the results in the seminar talk of Guillevic [GS19].

For the larger security levels one can use the same choice of polynomials. One can tune the parameters in an automatic manner and obtain for example that SexTNFS with these polynomials on a field of 9216 bits has a cost of 2^{192} (this is also in accordance with Guillevic and Singh's results). However, one can also use a different choice : $h = t$ (no tower), $f = P(x^6)$ and $g = x^6 - v$ which is a Joux-Pierrot construction. We obtain that a field of 9216 bits has 190.5 bits of security. We need to increase a bit the field size and obtain that 9742 is enough. The situation is once again different for 256 bits of security, where the best choice is the Conjugation method with $\kappa = 6$ and $h = t^2 - t - 1$: the key size is 20770.

Among the more than 150 families studied, almost no two were the same : each has a different combination of polynomials h, f and g to be used. Instead of a blind program to guess the polynomials automatically, we made all the choices manually using our experience on computation records of discrete logarithms. It is a good research project to write a programme which reproduces our choices.

3.4 Security results

We keep the model of security of Barbulescu and Duquesne [BD18] which is conservative in that it assumes perfect conditions for an attacker (sieving in TNFS for which no computation record is available, perfect matrix reduction in the filtering step, no memory limitation, ECM having the same performances for slightly larger smoothness bounds). The results are more precise than these obtained by forgetting the $o(1)$ term in the complexity as in [FK18] and [DGS17] because we don't omit any term in Equation (1). The analysis is also more precise than that of Menezes, Sarkar and Singh [MSS16] because we evaluate numerically the size of the norms N_f and N_g instead of using the mathematical upper bound.

In the following table we list the known families of pairings with $9 \le k \le 54$, which is a safety margin since the choices among BN, BLS and KSS have k between 12 and 24. The labels follow the format k, value of k,m, a two or three digits number which designs the construction number in the taxonomy [FST10], e.g. k9m62 denotes the family having $k = 9$ in the section 6.2 of the taxonomy, whereas k11m620 denotes the family of $k = 11$ of section 6.20 in the taxonomy. The sizes of the Dupont-Enge-Morain (DEM) construction also apply for Cocks-Pinch (CP). To verify the results one has to use Equation 1 and compute the best values of $\log_2 A$ and $\log_2 B$ (we provide our results and scripts on demand and we will maintain an online taxonomy together with the files which determine the security results).

family	security level		
	128 bits	192 bits	256 bits
	$\log_2(q^k)$, field side security when min(field,curve) security level = required level, algorithm, κ		
k9DEM	8622. 185 exTNFS-Conj k=3	9234. 192 exTNFS-Conj k=3	16070. 256 exTNFS-Conj k=3
k10DEM	5100. 161 exTNFS-Conj k=2	7660. 200 exTNFS-Conj k=2	11980. 257 exTNFS-Conj k=2
k11DEM	5610. 179 TNFS-base m k=1	8426. 226 TNFS-base m k=1	11240. 272 TNFS-base m k=1
k12DEM	6120. 163 exTNFS-Conj k=4	10540. 194 exTNFS-Conj k=4	16010. 256 exTNFS-Conj k=3
k13DEM	6630. 200 TNFS-base m k=1	9958. 240 TNFS-base m k=1	13290. 294 TNFS-base m k=1
k14DEM	7140. 195 exTNFS-Conj k=2	10720. 241 exTNFS-Conj k=2	14310. 285 exTNFS-Conj k=2
k15DEM	7650. 182 exTNFS-Conj k=5	11490. 200 exTNFS-Conj k=5	20370. 258 exTNFS-Conj k=5
k16DEM	8160. 193 exTNFS-Conj k=4	12260. 230 exTNFS-Conj k=4	17250. 257 exTNFS-Conj k=4
k17DEM	8670. 243 TNFS-base m k=1	13020. 300 TNFS-base m k=1	17370. 339 TNFS-base m k=1
k18DEM	9180. 211 exTNFS-Conj k=3	13790. 252 exTNFS-Conj k=3	18400. 269 exTNFS-Conj k=6
k19DEM	9690. 261 TNFS-base m k=1	14550. 330 TNFS-base m k=1	19420. 371 TNFS-base m k=1
k20DEM	10200. 219 exTNFS-Conj k=4	15320. 257 exTNFS-Conj k=4	20440. 292 exTNFS-Conj k=4
k9method62	4356. 134 SNFS k=1	13460. 194 SNFS k=1	25340. 257 SNFS k=1
k10method62	4460. 133 SNFS k=1	14400. 196 SexTNFS k=2	27980. 256 SexTNFS k=2
k11method62	3697. 173 SNFS k=1	7128. 192 SNFS k=1	24860. 256 SNFS k=1
k13method62	4265. 325 SNFS k=1	6216. 210 SNFS k=1	16350. 259 SNFS k=1
k14method62	5516. 159 SNFS k=1	9800. 195 SNFS k=1	19120. 256 SNFS k=1
k15method62	8131. 207 SNFS k=1	12210. 263 SNFS k=1	16290. 280 SNFS k=1
k17method62	5152. 254 SNFS k=1	7776. 291 SNFS k=1	10300. 281 SNFS k=1
k18method62	8677. 197 SNFS k=1	12640. 225 SNFS k=1	16990. 304 SNFS k=1
k19method62	6709. 245 SNFS k=1	8740. 329 SNFS k=1	11940. 292 SNFS k=1
k21method62	10680. 257 exTNFS-Conj k=3	15420. 294 exTNFS-Conj k=3	21210. 315 exTNFS-Conj k=3
k22method62	7394. 253 exTNFS-Conj k=2	11400. 284 exTNFS-Conj k=2	14830. 293 TNFS-base m k=1
k23method62	9778. 279 TNFS-base m k=1	10370. 289 TNFS-base m k=1	13770. 305 TNFS-base m k=1
k25method62	11820. 268 exTNFS-Conj k=5	13490. 303 exTNFS-Conj k=5	17590. 309 exTNFS-Conj k=5
k26method62	8528. 228 exTNFS-Conj k=2	12430. 297 exTNFS-Conj k=2	17110. 322 exTNFS-Conj k=2

family	security level		
	128 bits	192 bits	256 bits
	$\log_2(q^k)$, field side security when min(field,curve) security level = required level, algorithm, κ		
k27method62	14810. 289 exTNFS-Conj k=3	17200. 317 exTNFS-Conj k=3	23460. 409 exTNFS-Conj k=3
k29method62	10920. 338 TNFS-base m k=1	15960. 372 TNFS-base m k=1	18580. 406 TNFS-base m k=1
k30method62	16260. 181 exTNFS-GJL k=5	24420. 233 exTNFS-GJL k=5	32580. 398 exTNFS-GJL k=6
k31method62	11870. 273 TNFS-base m k=1	16430. 384 TNFS-base m k=1	18650. 419 TNFS-base m k=1
k33method62	19600. 387 exTNFS-Conj k=3	23490. 389 exTNFS-Conj k=3	30140. 453 exTNFS-Conj k=3
k34method62	10300. 248 exTNFS-Conj k=2	15550. 372 exTNFS-Conj k=2	20610. 430 exTNFS-Conj k=2
k35method62	17250. 374 exTNFS-Conj k=5	24210. 425 exTNFS-Conj k=5	29560. 437 exTNFS-Conj k=5
k37method62	14960. 327 TNFS-base m k=1	19140. 455 TNFS-base m k=1	23660. 499 TNFS-base m k=1
k38method62	13420. 285 exTNFS-Conj k=2	17480. 388 exTNFS-Conj k=2	23880. 465 exTNFS-Conj k=2
k39method62	20530. 427 exTNFS-Conj k=3	29920. 446 exTNFS-Conj k=3	35220. 459 exTNFS-Conj k=3
k41method62	18290. 359 TNFS-base m k=1	18290. 381 TNFS-base m k=1	29050. 515 TNFS-base m k=1
k42method62	21370. 459 exTNFS-Conj k=5	30840. 488 exTNFS-GJL k=6	42420. 503 exTNFS-Conj k=3
k43method62	31020. 477 TNFS-base m k=1	31020. 413 TNFS-base m k=1	31020. 515 TNFS-base m k=1
k45method62	31000. 361 exTNFS-Conj k=5	34740. 448 exTNFS-Conj k=5	47120. 496 exTNFS-Conj k=5
k46method62	19560. 408 exTNFS-Conj k=2	20740. 435 exTNFS-Conj k=2	27540. 472 exTNFS-Conj k=2
k47method62	33070. 510 TNFS-base m k=1	33070. 459 TNFS-base m k=1	33070. 515 TNFS-base m k=1
k49method62	23110. 547 exTNFS-Conj k=7	34720. 574 exTNFS-Conj k=7	42560. 654 exTNFS-Conj k=9
k50method62	23640. 418 exTNFS-Conj k=5	26970. 632 exTNFS-Conj k=5	35180. 519 exTNFS-Conj k=5
k10method63	4460. 134 SexTNFS k=2	12580. 192 SexTNFS k=2	23080. 256 SexTNFS k=2
k14method63	5516. 148 SNFS k=1	8036. 206 SNFS k=1	21640. 258 SexTNFS k=2
k18method63	8676. 294 SexTNFS k=2	12640. 275 SNFS k=1	16990. 292 SexTNFS k=2
k22method63	7409. 387 SexTNFS k=2	11400. 273 exTNFS-Conj k=2	14830. 351 SexTNFS k=2
k26method63	8568. 416 SNFS k=1	12440. 288 exTNFS-Conj k=2	17110. 347 exTNFS-Conj k=2
k30method63	16270. 547 SNFS k=1	24420. 351 exTNFS-GJL k=6	32580. 434 exTNFS-GJL k=6
k34method63	10460. 670 SNFS k=1	15560. 348 exTNFS-Conj k=2	20680. 409 exTNFS-Conj k=2
k38method63	13530. 316 exTNFS-Conj k=2	17560. 393 exTNFS-Conj k=2	23940. 459 exTNFS-Conj k=2
k42method63	21340. 411 exTNFS-Conj k=6	30920. 470 exTNFS-Conj k=6	42500. 515 exTNFS-GJL k=7
k46method63	13900. 332 exTNFS-Conj k=2	21450. 405 exTNFS-Conj k=2	27740. 452 exTNFS-Conj k=2
k50method63	22680. 415 exTNFS-GJL k=5	27080. 462 exTNFS-Conj k=5	35170. 486 exTNFS-GJL k=5
k54method63	25130. 476 exTNFS-Conj k=6	34870. 1880 SNFS k=1	46980. 570 exTNFS-GJL k=9
k12method64	9192. 172 SNFS k=1	24460. 192 SexTNFS k=2	43180. 258 SexTNFS k=2
k20method64	7640. 208 SNFS k=1	11480. 227 SNFS k=1	19160. 257 SNFS k=1
k28method64	9800. 412 SexTNFS k=2	14280. 345 SNFS k=1	19210. 310 SNFS k=1
k36method64	15770. 517 SexTNFS k=2	22970. 368 SNFS k=1	30890. 379 SNFS k=1
k44method64	13650. 412 SNFS k=1	21030. 431 SNFS k=1	27370. 436 SNFS k=1
k52method64	15920. 575 SNFS k=1	23200. 502 exTNFS-Conj k=4	31930. 594 SNFS k=1
k9method66	4810. 129 SNFS k=1	6178. 196 SNFS k=1	20070. 258 SNFS k=1
k10method66	5104. 166 SNFS k=1	12780. 192 SNFS k=1	25420. 261 SNFS k=1
k11method66	3421. 339 exTNFS-GJL k=1	5263. 216 TNFS-base m k=1	6846. 241 SNFS k=1
k12method66	5525. 128 SexTNFS k=2	12580. 192 SexTNFS k=2	26120. 256 SexTNFS k=2
k13method66	4008. 155 TNFS-base m k=1	5806. 229 TNFS-base m k=1	11990. 259 TNFS-base m k=1
k14method66	4906. 175 SNFS k=1	7594. 197 exTNFS-Conj k=7	9610. 232 SNFS k=1
k15method66	5736. 175 SNFS k=1	8616. 192 SNFS k=1	11500. 222 SNFS k=1
k16method66	5608. 258 SNFS k=1	8422. 202 exTNFS-Conj k=4	15810. 256 exTNFS-Conj k=4
k17method66	5914. 202 TNFS-base m k=1	7426. 237 TNFS-base m k=1	9784. 259 exTNFS-Conj k=2
k19method66	6411. 217 TNFS-base m k=1	8397. 233 TNFS-base m k=1	11390. 274 TNFS-base m k=1
k20method66	7013. 331 SNFS k=1	14050. 244 exTNFS-Conj k=4	17130. 257 exTNFS-Conj k=4

family	security level		
	128 bits	192 bits	256 bits
	$\log_2(q^k)$, field side security when min(field,curve) security level = required level, algorithm, κ		
k21method66	7359. 250 SNFS k=1	10720. 227 exTNFS-Conj k=3	14410. 262 exTNFS-Conj k=3
k22method66	8008. 136 exTNFS-Conj k=2	12320. 269 exTNFS-Conj k=2	16020. 314 exTNFS-Conj k=2
k23method66	9614. 236 TNFS-base m k=1	11160. 293 TNFS-base m k=1	13500. 340 TNFS-base m k=1
k24method66	7642. 171 SNFS k=1	12440. 195 SNFS k=1	24680. 259 SNFS k=1
k25method66	12160. 249 exTNFS-Conj k=5	14220. 257 exTNFS-Conj k=5	16880. 294 exTNFS-Conj k=5
k26method66	7972. 226 exTNFS-Conj k=2	11610. 267 exTNFS-Conj k=2	15980. 319 exTNFS-Conj k=2
k27method66	8062. 249 exTNFS-GJL k=9	11840. 259 exTNFS-Conj k=3	15620. 349 exTNFS-GJL k=3
k28method66	10460. 289 exTNFS-Conj k=7	15190. 261 exTNFS-Conj k=4	20900. 300 exTNFS-Conj k=4
k29method66	18650. 363 TNFS-base m k=1	18650. 382 TNFS-base m k=1	18650. 370 TNFS-base m k=1
k30method66	11470. 216 exTNFS-Conj k=3	17230. 256 exTNFS-Conj k=5	22990. 297 exTNFS-Conj k=5
k31method66	14600. 347 TNFS-base m k=1	14600. 362 TNFS-base m k=1	21780. 410 TNFS-base m k=1
k32method66	8984. 355 exTNFS-Conj k=2	13010. 414 exTNFS-Conj k=2	17360. 305 exTNFS-Conj k=4
k33method66	10260. 267 exTNFS-Conj k=3	15790. 302 exTNFS-Conj k=3	20540. 328 exTNFS-Conj k=3
k34method66	12050. 355 exTNFS-Conj k=2	16280. 328 exTNFS-Conj k=2	21730. 391 exTNFS-Conj k=2
k35method66	20780. 381 exTNFS-Conj k=5	20780. 344 exTNFS-Conj k=5	31060. 390 exTNFS-Conj k=5
k37method66	20320. 404 TNFS-base m k=1	20320. 422 TNFS-base m k=1	22680. 442 TNFS-base m k=1
k38method66	13550. 355 exTNFS-Conj k=2	16800. 339 exTNFS-Conj k=2	22740. 405 exTNFS-Conj k=2
k39method66	12020. 273 exTNFS-Conj k=3	17420. 327 exTNFS-Conj k=3	23960. 361 exTNFS-Conj k=3
k40method66	14780. 371 exTNFS-Conj k=5	22490. 383 exTNFS-Conj k=5	29380. 406 exTNFS-Conj k=5
k41method66	33370. 485 TNFS-base m k=1	33370. 503 TNFS-base m k=1	33370. 522 TNFS-base m k=1
k42method66	14720. 297 exTNFS-Conj k=3	21440. 356 exTNFS-Conj k=3	28830. 398 exTNFS-Conj k=3
k43method66	17050. 399 TNFS-base m k=1	30940. 499 TNFS-base m k=1	30940. 516 TNFS-base m k=1
k44method66	14910. 335 exTNFS-Conj k=2	20640. 379 exTNFS-Conj k=2	26340. 453 exTNFS-Conj k=2
k45method66	15800. 357 exTNFS-Conj k=5	22980. 386 exTNFS-Conj k=5	31610. 431 exTNFS-Conj k=5
k46method66	14590. 335 exTNFS-Conj k=2	22560. 401 exTNFS-Conj k=2	29000. 476 exTNFS-Conj k=2
k47method66	26130. 466 TNFS-base m k=1	26130. 485 TNFS-base m k=1	29360. 523 TNFS-base m k=1
k48method66	13750. 304 exTNFS-Conj k=3	20660. 366 exTNFS-Conj k=3	27570. 404 exTNFS-Conj k=3
k49method66	29930. 496 TNFS-base m k=1	29930. 518 TNFS-base m k=1	34520. 566 TNFS-base m k=1
k50method66	21820. 433 exTNFS-Conj k=5	26420. 445 exTNFS-Conj k=2	33820. 459 exTNFS-Conj k=5
k52method66	20800. 387 exTNFS-Conj k=2	27500. 460 exTNFS-Conj k=2	33620. 491 exTNFS-Conj k=2
k53method66	48570. 582 TNFS-base m k=1	48570. 610 TNFS-base m k=1	48570. 631 TNFS-base m k=1
k9method67	4564. 266 SNFS k=1	6598. 198 exTNFS-GJL k=9	9081. 287 SNFS k=1
k12method67	5340. 148 SNFS k=1	8028. 199 SexTNFS k=2	20120. 256 SexTNFS k=2
k15method67	14520. 217 SNFS k=1	14520. 217 SNFS k=1	15810. 431 SNFS k=1
k18method67	7540. 192 exTNFS-Conj k=3	10900. 273 exTNFS-GJL k=1	14990. 276 exTNFS-Conj k=3
k21method67	12560. 259 exTNFS-Conj k=3	15190. 276 exTNFS-Conj k=3	19910. 312 exTNFS-Conj k=3
k24method67	9144. 324 SexTNFS k=2	13750. 357 SexTNFS k=2	18360. 499 SexTNFS k=2
k27method67	14360. 220 exTNFS-Conj k=5	18360. 315 exTNFS-Conj k=3	24770. 346 exTNFS-Conj k=3
k30method67	16510. 263 exTNFS-Conj k=6	20900. 292 exTNFS-Conj k=6	27760. 375 exTNFS-GJL k=6
k33method67	21880. 310 exTNFS-Conj k=3	21880. 352 exTNFS-Conj k=3	30310. 395 exTNFS-Conj k=3
k36method67	13540. 260 exTNFS-Conj k=3	19480. 348 exTNFS-Conj k=3	26820. 383 exTNFS-Conj k=3
k39method67	29090. 354 exTNFS-Conj k=3	29090. 406 exTNFS-Conj k=3	35570. 438 exTNFS-Conj k=3
k42method67	28040. 365 exTNFS-Conj k=3	28040. 423 exTNFS-Conj k=3	36340. 450 exTNFS-Conj k=3
k45method67	40400. 444 exTNFS-Conj k=3	40400. 522 exTNFS-Conj k=3	67960. 587 exTNFS-GJL k=6
k48method67	16760. 301 exTNFS-Conj k=3	25200. 426 exTNFS-Conj k=3	33650. 451 exTNFS-Conj k=3
k51method67	64050. 553 exTNFS-Conj k=3	64050. 725 exTNFS-Conj k=3	64050. 644 exTNFS-Conj k=3
k54method67	23080. 356 exTNFS-Conj k=3	32600. 499 exTNFS-Conj k=3	46960. 516 exTNFS-Conj k=6

family	security level		
	128 bits	192 bits	256 bits
	$\log_2(q^k)$, field side security when min(field,curve) security level = required level, algorithm, κ		
BN	5534. 128 SexTNFS k=2	13120. 192 SexTNFS k=3	25310. 256 SexTNFS k=3
k16methodKSS	5281. 154 SNFS k=1	8161. 192 SNFS k=1	18240. 257 SNFS k=1
k18methodKSS	6401. 156 SNFS k=1	11730. 195 SNFS k=1	26270. 260 SexTNFS k=2
k32methodKSS	11030. 395 SNFS k=1	14870. 370 SNFS k=1	19470. 394 SNFS k=1
k36methodKSS	11560. 370 exTNFS-GJL k=6	17110. 421 exTNFS-GJL k=6	22150. 521 exTNFS-GJL k=6
k40methodKSS	15070. 411 exTNFS-GJL k=6	22080. 400 exTNFS-GJL k=8	29120. 531 exTNFS-GJL k=6
k11method620	5258. 128 SNFS k=1	16870. 192 SNFS k=1	32980. 256 SNFS k=1
k15method620	7650. 171 SNFS k=1	11490. 209 SNFS k=1	33330. 256 SNFS k=1
k26method624	8546. 191 SNFS k=1	12180. 212 SNFS k=1	17270. 260 SNFS k=1
k34method624	10740. 289 SNFS k=1	15650. 270 SNFS k=1	20490. 315 SNFS k=1
k3MNT	4127. 128 exTNFS-Conj k=3	9191. 192 exTNFS-Conj k=3	16120. 256 SexTNFS k=3
k4MNT	4240. 128 exTNFS-Conj k=4	10520. 192 exTNFS-Conj k=4	19040. 256 exTNFS-Conj k=4
k6MNT	4620. 128 SexTNFS k=3	15000. 192 SexTNFS k=6	20760. 256 exTNFS-Conj k=6
k9methodLZZW	5314. 128 SNFS k=1	11510. 192 SNFS k=1	20650. 256 SNFS k=1
k12methodDCC	10790. 177 SexTNFS k=2	14390. 199 SexTNFS k=2	25910. 262 SexTNFS k=2
k15methodDCC	5745. 285 SNFS k=1	8985. 192 exTNFS-Conj k=3	20140. 256 exTNFS-Conj k=5
k24methodDCC	7656. 196 SNFS k=1	11500. 248 exTNFS-Conj k=3	15340. 269 exTNFS-Conj k=6
k48methodDCC	13780. 352 exTNFS-Conj k=3	20690. 523 exTNFS-Conj k=6	27600. 560 exTNFS-Conj k=6
k2rho1	3460. 129 exTNFS-Conj k=2	7200. 195 exTNFS-Conj k=2	12200. 259 exTNFS-Conj k=2

Our results can be downloaded at:

https://webusers.imj-prg.fr/~razvan.barbaud/Pairings/Pairings.html

4 Complexity of the Miller's algorithm at 128 bits of security

In this section, we make an extensive comparison among a large number of families in the literature. Our comparison is not optimized enough to be directly implemented for each of the over 150 families, but is optimized enough to make apparent the good families of pairings. The criterion of comparison is the binary cost of the Ate pairing computation (Miller loop and final exponentiation.

For each family, we compute parameters u with a small NAF weight, if it is possible. Otherwise, we use random parameters u of the required bit size, but in some cases of large embedding degree even this is impossible. Indeed, some of the families, for example those of prime degree have never been investigated numerically, e.g. BLS-26.

4.1 Notation and arithmetic

In the following we use the classical notations M_q, S_k and I_q for the binary cost of the multiplication, squaring and respectively inversion over $\mathbb{F}_q$. We denote by M_k, S_k and I_k the binary cost of the multiplication, squaring and inversion in the field $\mathbb{F}_{q^k}$. For our level of optimization, the crude estimation $M_q = S_q$ is enough. When a multiplication by an element of $\mathbb{F}_q$ is necessary (for instance a multiplication by a, denoted d_a, in the doubling of points) we make the coarse estimation that $d_a = M_q$.

In any case one can use the estimation $M_k \leq k^2 M_q$, but when q is a prime of 500 to 5000 bits we use the formulae of multiplication in tower fields:

- when $k = 2$ Karatsuba's trick [Knu97] gives $M_2 = 3M_q$;
- when $k = 3$ Toom-Cook's trick [Knu97] gives $M_3 = 5M_q$;
- when $k = 5, 6, 7$, we use the formulae in [EGI11]: $M_5 = 9M_q$, $M_6 = 11M_q$ and $M_7 = 13M_q$ as the implementations in [EGI11] demonstrate that the arithmetic in this article is the more efficient.
- when we use a twist of degree $d = 2$ (resp. 3, 4, 6) we count $M_k = 3M_e$ (resp. $M_k = 5M_e$, $M_k = 9M_e$, $M_k = 11M_e$) for $e = k/d$ [Knu97,EGI11];
- when $k = 22, 26, 34, 46$ and we have a twist of degree $d = 2$, we consider that $M_e = (k/2)^2 M_q$ where $e = k/d$.

We use the

We go from the arithmetic complexity to the binary complexity using the estimate that M_q counts for w^2 word multiplications, where w is the number of machine words of q. We denote by m_{32} (resp. m_{64}) the cost of a word multiplication on a 32-bit (resp 64-bit machine). A comparison of hardware implementation is beyond the scope of this article because it is much more difficult to take into account the dedicated architectures.

4.2 Construction 6.2 from [FST10]

In this metafamily of curves we can construct curves whose embedding degree is either odd or twice an odd. All the curves admit a discriminant $D = -1$ (we abusively replace D in the sequel by its absolute value), so we have a twist of degree $d = 2$ when the embedding degree is even and no twist otherwise ($d = 1$).

The general expression of Ate pairing for construction 6.2 is defined as follow:

$\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3, (P, Q) \to \left(f_{x^2,Q}(P) \times \frac{l_{qQ,x^2Q}(P)}{v_{(x^2+q)Q}(P)} \right)^{\frac{q^k-1}{r}}$

The complexity depends on whether $d = 2$ or $d = 1$.

Curves admitting a twist of degree $d = 2$ Note that $D = 1$ and the equation of the elliptic curve is $y^2 = x^3 + ax$. We use the formulas from [CLN10].

The Ate pairing computation is composed of one execution of the Miller algorithm, which has $\log_2(u^2)$ iterations using the denominator elimination. The vertical line $v_{(x^2+q)Q}(P)$ belongs to $\mathbb{F}_{q^{k/2}}$ and is eliminated by the final exponentiation. The Ate pairing expression is simplified into:

$\left(f_{x^2,Q}(P) \times l_{qQ,x^2Q}(P) \right)^{\frac{q^k-1}{r}}$.

Its complexity is equal to $\log_2(u^2)$ doubling steps, plus $HW(u^2)$ (the Hamming weight of u) addition steps and an extra doubling step for the evaluation of $l_{qQ,x^2Q}(P)$. As we do not need the coordinates of the point $(x^2 + q)Q$, this line evaluation (Le) is cheaper than a full doubling step [BD18][3]. We recall these complexities in Table 2. We use the projective coordinates, which are better than the affine ones at 128 bits of security [CLN10,ZL12].

The Table 3 presents the complexity of the Miller computation at 128 bits of security. Note that the embedding degree $k = 14$ offers the best arithmetic complexity and the smallest target field.

---

[3] We count $5M_e$ in the evaluation of Le instead of $4M_e$ as presented in [BD18] because when we wrote down the equation we do not see how to save one more M_e

Operation	Complexity
Doubling step [CLN10]	$(2k/d)M_q + 2M_e + 8S_e + 1d_a + M_k + S_k$
Addition step [CLN10]	$(2k/d)M_q + 12M_e + 7S_e + M_k$
Mixed addition [CLN10]	$(2k/d)M_q + 9M_e + 5S_e + M_k$
Final line evaluation [BD18]	$5M_e + 2k/dM_q$

Table 2: Complexity of Miller's steps using quadratic twist and $D = 1$

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx$
10	446	$31, 8$	$1+2^4+2^5+2^8+2^9-2^15+2^32$	446	$64DBL+6Madd+Le+M_k$	$10\ 971M_q$
14	394	22	$-1+2^4+2^5+2^6-2^{11}+2^{15}+2^{22}$	394	$44DBL+9Madd+Le+M_k$	$12\ 032M_q$
18	482	22	$1+2^6-2^9+2^{12}+2^{13}+2^{15}+2^{17}+2^{20}+2^{21}$	482	$44DBL+10Madd+Le+M_k$	$23\ 059M_q$
22	336	$12, 9$	$1-2^6+2^9+2^{12}$	314	$24DBL+9Madd+Le+M_k$	$66\ 596M_q$
30	542	$15, 9$	$1+2^4-2^7+2^{12}+2^{13}+2^{15}$	524	$31DBL+9Madd+Le+M_k$	$30\ 781M_q$
38	353	8	$1+2^2+2^5+2^6+2^8+2^9$	408	$20DBL+8Madd+Le+M_k$	$19\ 179M_q$
42	508	11	$1+2-2^3+2^7+2^8+2^{11}$	515	$23DBL+8Madd+Le+M_k$	$34\ 582M_q$
46	425	$8, 5$	$-1+2^3+2^4+2^7+2^{11}$	553	$22DBL+8Madd+Le+M_k$	$263\ 303M_q$
50	473	$8, 7$	$-1+2^3+2^4+2^6+2^9+2^{12}$	657	$24DBL+9Madd+Le+M_k$	$45\ 788M_q$
$26, 34$			no value for u below 2^{11}			

Table 3: Method 6.2, 128 bits of security, quadratic twist, practical value

Curves without twists The Ate pairing computation is composed of one execution of the Miller algorithm for $\log_2(u^2)$ iterations, without the denominator elimination. The Ate pairing expression cannot be simplified:

$$\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3, (P, Q) \to \left(f_{x^2,Q}(P) \times \frac{l_{qQ,x^2Q}(P)}{v_{(x^2+q)Q}(P)} \right)^{\frac{q^k-1}{r}}.$$

Its complexity is $\log_2(u^2)$ doubling step, plus $HW(u^2)$ addition step and an extra doubling step for the evaluation of $\frac{l_{qQ,x^2Q}(P)}{v_{(x^2+q)Q}(P)}$.

To our knowledge, there is no reference in the literature to pairing computations without twists. We computed new formulae and we obtain the arithmetic cost of each step in Table 5.

We use the estimation $M_k = S_k$ and find that the doubling step in projective coordinates has a cost of $3kM_q + 19M_k$. Compare this to that in Jacobian coordinates which is $3kM_q + 18M_k$. For the addition step, the difference between the two types of coordinates is more important : in projective coordinates we obtain $3kM_q + 18M_k$ and in Jacobina ones we get $3kM_q + 33M_k$. As our goal is to give a first estimation of the pairing complexity, we do not search especially for parameters with very small Hamming weight. Note that the affine coordinates could be more interesting than the projective ones if the complexity of the inversion in $\mathbb{F}_{q^k}$ is smaller than $20M_k$. This coarse estimation is obtained by considering that $M_k = S_k$ and $kM_q = M_k$. The expected gain is not important enough, so we don't continue with a precise estimation in this case.

Best choice for the method 6.2 According to the results in Table 6, the curves of embedding degree 9 are the champion among the curves of construction 6.2 without twists. Yet, they are no match for the family of embedding degree $k = 14$. The prime embedding degrees are interesting when one desires a small target field and a short Miller's loop, in this case one might prefer $k = 11$. However, the denominator elimination together with the works improving the arithmetic for the

k	$(\log_2(q))$	m_{32} words	$\approx$ Miller's cost	k	$(\log_2(q))$	m_{64} words	$\approx$ Miller's cost
14	394	13	2 150 316m_{32}	14	394	7	537 579m_{64}
22	314	10	6 659 600m_{32}	22	314	5	1 664 900m_{64}
38	408	13	3 241 251m_{32}	38	408	7	939 771m_{64}

Table 4: Method 6.2, Comparison between $k = 14, 22$ and 38

Operation	Complexity affine	Complexity projective	Complexity jacobien
Doubling step	$2M_k + S_k + I_k$	$3kM_q + 12M_k + 7S_k$	$3kM_q + 10M_k + 8S_k$
Addition step	$5M_k + 2S_k + I_k$	$3kM_q + 16M_k + 2S_k$	$3kM_q + 19M_k + 14S_k$

Table 5: Complexity of Miller's steps without twist

tower field extensions make imply that the best choice for this metafamily should be the curve with embedding degree 14.

4.3 Construction 6.3 from [FST10]

Using this construction, we obtain elliptic curve having an embedding degree $k = 2k'$, for k' an odd number. Those curves have a discriminant $D = 1$, they admit a twist of degree 2.

The expression of the optimal Ate pairing for this family is the following: $\left(f_{x^2,Q}(P) \times l_{-qQ,x^2Q}(P)\right)^{\frac{q^k-1}{r}}$.

The optimal Ate pairing for curves constructed using method 6.3 consists in one Miller's algorithm indexed over x^2, plus an extra line evaluation.

The Table 7 presents the value that we find by a quick research and using very large estimation for the cost of arithmetic in the tower field. We used the estimation cost from Table 2 as we are working on elliptic curve with discriminant 1 and quadratic twist.

The smallest number of iterations for Miller's algorithm could be reached for the curve with $k = 38$, but unfortunately, in practice, we do not find a value of u that makes p and q prime below 15 bits.

The smallest size for $\mathbb{F}_q$ is theoretically obtained for the curve with embedding degree 26, 34 and 46. Together with the theoretically smallest number of iterations during the Miller algorithm. In practice, the less expensive Miller's algorithm corresponds to $k = 14$. For this value we also have the smallest finite field $\mathbb{F}_q$. As a consequence, the best choice for the method 6.3 using a quadratic twist at the 128 bits of security should be the curve with $k = 14$.

4.4 Construction 6.4 from [FST10]

In this metafamily of curves, we construct curves with embedding degrees $4k'$ where k' is an odd integer. The discriminant is $D = 1$, consequently, curves in this family admit a twist of degree 4.

The expression of the optimal Ate pairing for this family is the following:

$OptAte_{6.4} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$,

$(P, Q) \to \left(f_{x,Q}(P) \times \frac{l_{-qQ,x^2Q}(P)}{v_Q^q(P)v_{(x-q)Q}(P)}\right)^{\frac{q^k-1}{r}}$

As we can use a quadratic twist, the denominator $v_Q^q(P)v_{(x-q)Q}(P)$ vanishes during the final exponentiation. Thus the expression of the optimal Ate pairing can be simplified as:

$OptAte_{6.4} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$,

$(P, Q) \to \left(f_{x,Q}(P) \times l_{-qQ,x^2Q}(P)\right)^{\frac{q^k-1}{r}}$.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx$
9	484	22	$-1+2^3+2^4+2^5+2^9+2^{10}+2^{22}$	482	$44DBL+20Add+1DBL+M_k+I_k$	$31\ 155M_q + I_k$
11	336	13	$-1+2^8+2^{14}$	363	$28DBL+4Add+1DBL+M_k+I_k$	$65\ 316M_q + I_k$
13	328		$1+2+2^3+2^4+2^8+2^{10}+2^{14}+2^{20}$	599	$20DBL+14Madd+1DBL+M_k+I_k$	$110\ 085M_q + I_k$
$15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35$				complexity higher than $203\ 985M_q + I_k$		
$37, 39, 43, 45$			no value for u below 2^{11}			

Table 6: Method 6.2, 128 bits of security, no twist, pairing estimation

k	u	$(\log_2(q))$	Miller's cost	$\approx M_q$
10	$1+2^3-2^5+2^{10}+2^{13}+2^{31}$	432	$62DBL+14Madd+Le+M_k$	11 938
14	$1-2^2+2^6+2^9-2^{12}-2^{15}-2^{19}+2^{22}$	390	$22DBL+7Madd+Le+M_k$	6 894
18	$1+2+2^3+2^5+2^7+2^8+2^{10}+2^{12}+2^{13}+2^{22}$	482	$44DBL+11Madd+Le+M_k$	23 458
22	$1+2+2^4+2^{14}+2^{15}$	403	$30DBL+9Madd+Le+M_k$	78 423
26	$1+2^8+2^{12}$	360	$24DBL+5Madd+Le+M_k$	81 248
30	$1+2^2+2^3-2^{10}+2^{14}+2^{16}$	552	$32DBL+11Madd+Le+M_k$	26 687
34	$1-2^4+2^{10}+2^{14}$	533	$28DBL+6Madd+Le+M_k$	165 138
38	$1+2^3+2^9+2^{11}+2^{17}$	713	$34DBL+11Madd+Le+M_k$	268 200
42	$1+2^4+2^7+2^8+2^{10}+2^{11}$	539	$24DBL+7Madd+Le+M_k$	225 150
46	$1+2+2^9+2^{10}+2^{13}$	660	$26DBL+9Madd+Le+M_k$	315 415
50	$1+2^4-2^7+2^{10}+2^{11}+2^{14}$	746	$28DBL+9Madd+Le+M_k$	50 603
54	$1+2+2^3+2^5+2^8+2^9+2^{11}$	664	$23DBL+9Madd+Le+M_k$	74 466

Table 7: Method 6.3, 128 bits of security

The optimal Ate pairing for curves constructed using method 6.4 is composed by one Miller's algorithm indexed over x, plus an extra line evaluation. The Table 8 presents some examples of values for u that minimize the number of addition steps during Miller's algorithm. In this Table, we do not include the column giving the number of bits of u, as it can be deduced by the number of doubling step we count.

We compare the curves with approximately 10 000 M_q ($k = 12, 20, 28$) and the curve with the smallest field $\mathbb{F}_q$ ($k = 44$). On a 32 bits architecture, it seems that the curves constructed by method 6.4 with $k = 28$ provides the most efficient pairing, on a 64 bits architecture, it should be the curve with $k = 20$. Of course, those results highly depends on the architecture and the implementation.

4.5 Construction 6.6 from [FST10]

In this metafamily of curves, also called BLS, we can construct curves with discriminant $D = 3$. Hence, in this case the elliptic curves can admit a twist of degree 3 or 6. The method of construction depends on the residue of k modulo 6, and we studied all the families from $k = 9$ to $k = 53$, all being possible except those for which 18 divides k, i.e. 18, 36 and 54.

Curves admitting a twist of degree 6 When $k = 0 \bmod 6$, then the elliptic curve admits a twist of degree 6. The corresponding embedding degrees are $k \in \{12$ (i.e. BLS12), 24 (i.e. BLS24), $30, 36, 42, 48\}$.

The expression of the optimal Ate pairing is the following:

$OptAte_{6.6d6} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$,

$(P, Q) \to \left(\frac{f_{x,Q}(P) \times l_{-qQ,xQ}(P)}{v_Q(P) v_{(x-q)Q}(P)}\right)^{\frac{q^k-1}{r}}$.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx M_q$
12	510	$63, 7$	$1+2+2^3+2^8+2^9+2^{11}+2^{64}$	510	$64DBL+6Madd+Le+M_k$	10 141
20	382	$31, 8$	$1+2^4+2^{16}+2^{32}$	383	$32DBL+3Madd+Le+M_k$	9 116
28	350	$21, 8$	$1+2+2^3+2^4+2^8+2^9+2^{22}$	350	$22DBL+6Madd+Le+M_k$	10 278
36	438	$21, 9$	$1+2^2+2^{10}+2^{14}+2^{16}+2^{22}$	438	$22DBL+5Madd+Le+M_k$	18 901
44	310	$12, 9$	$1+2^7+2^8+2^{12}+2^{14}$	342	$14DBL+4Madd+Le+M_k$	59480
52	306	$10, 9$	$1-2^6+2^9+2^{12}+2^{13}$	380	$13DBL+4Madd+Le+M_k$	81134

Table 8: Method 6.4, 128 bits of security, twist of degree 4

k	$\log_{32}(q)$	Miller's in m_{32}	$\log_{64}(q)$	Miller's in m_{64}
12	16	2 596 096	8	649 024
20	12	1 312 704	6	328 176
28	11	1 243 638	6	370 008
44	11	7 197 080	6	2 141 280

Table 9: Method 6.4, Comparison of the best candidates

Since these curves admit a twist of degree 6, we can use the denominator elimination in order to simplify the expression of the pairing:
$OptAte_{6.6d6} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$,
$(P, Q) \rightarrow (f_{x,Q}(P) \times l_{-qQ,xQ}(P))^{\frac{q^k-1}{r}}$.

We use the most efficient formulas in the literature in order to estimate the algebraic complexity of a Miller's execution. We recall them in Table 10.

Operation	Complexity
Doubling step [CLN10]	$(2k/d)M_q + 3M_e + 5S_e + M_k + S_k$
Addition step [CLN10]	$(2k/d)M_q + 14M_e + 2S_e + 1d_c + M_k$
Mixed addition [CLN10]	$(2k/d)M_q + 10M_e + 2S_e + 1d_c + M_k$
Final line evaluation	$2k/dM_q + 5M_e$

Table 10: Complexity of Miller's steps using sextic twist

The smallest number of operation over $\mathbb{F}_q$ is obtained for $k = 12$, but the smallest field is obtained for $k = 24$.

In order to compare those two curves, we have to estimate the complexity of the Miller algorithm in terms of machine word. The Table 12 presents our estimation. We consider that a multiplication over $\mathbb{F}_q$ is computed using the schoolbook multiplication.

According to our estimation, the optimal Ate pairing seems to be more efficient on BLS24 than on BLS12 curves.

Curves admitting a twist of degree 3 Among the elliptic curves constructed by method 6.6, those for which $k = 3 \bmod 6$ admit a twist of degree 3. The expression of the optimal Ate pairing depends on the embedding degree. For each embedding degree $k \in \{15, 21, 27, 33, 39, 45, 51\}$, we obtain a different short vector that should be used in order to compute the pairing. The expression

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx M_q$
12	461	64	$-2^{77}+2^{50}+2^{33}$	460	$77DBL{+}2Madd{+}Le{+}M_k$	7 438
24	318	32	$-2^{32}+2^{28}+2^{12}$	319	$32DBL{+}2Madd{+}Le{+}M_k$	9 381
30	383	32	4294971136	383	$32DBL{+}4Madd{+}Le{+}M_k$	9 887
42	350	22	$-2^{22}+2^{18}+2^{6}$	349	$22DBL{+}2Madd{+}Le{+}M_k$	9 738
48	286	16	$2^6+2^{11}+2^{13}+2^{14}+2^{16}$	296	$17DBL{+}4Madd{+}Le{+}M_k$	17 042

Table 11: Method 6.6 (BLS), 128 bits of security, twist of degree 6

k	$\log_{32}(q)$	Miller's in m_{32}	$\log_{64}(q)$	Miller's in m_{64}
12	15	1 673 550	8	476 032
24	10	938 100	5	234 525

Table 12: Method 6.6, Comparison of the best candidates

of the pairing follows a common pattern for $k \in \{15, 33, 51\}$, respectively for $k \in \{27, 45\}$; and for $k \in \{21, 39\}$.

For $k \in \{15, 33, 51\}$ using the construction 6.6, we obtain the same pattern for a short vector: $[x, -1, 0, \ldots 0, -1, 0, \ldots, 0]$.

We give here the definition of an optimal Ate pairing for $k = 15$.

We choose $[x, -1, 0, 0, 0, 0, -1, 0, \ldots, 0]$ as short vector. The expression of the optimal Ate pairing using this vector is the following:
$OptAte_{k15_6.6d3} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$,
$(P, Q) \to \left(\left(\frac{f_{x,Q}}{v_Q^{q+q^6}} \frac{l_{s_1Q,xQ}}{v_{s_0Q}} \frac{l_{s_2Q,-qQ}}{v_{s_1Q}}\right)(P) \right)^{\frac{q^k-1}{r}}$, where $s_0 = x - q - q^6$, $s_1 = -q - q^6$ and $s_2 = -q^6$.

When using a twist of degree 3, the vertical line does not vanish during the final exponentiation. We can however simplify the pairing expression. Zhang and Lin in [ZL12] proposes the latest record for the computation of pairings over curves with a twist of degree 3. They barely improve the result of [CLN10] but the method is very helpful for the simplification of the optimal Ate pairing in our case. We use Zhang and Lin formulas for the complexity of Miller's algorithm's step 13.

Applying the method developed by Zhang and Lin in [ZL12], we can make the following transformation $\frac{1}{(v_Q)}(P) = \frac{X_Q^2 + X_Q Z_Q x_p + x_q^2}{Z_Q^2}$.

Indeed, using the method developed by Zhang and Lin in [ZL12], we can transform the fraction $\frac{l_{s_1Q,xQ}}{v_{s_0Q}}$ into

$$X_{s_0Q}^2 - Z_{s_1Q} Z_{xQ} (Z_{s_1Q} X_{xQ} - X_{s_1Q} Z_{xQ})^2 (Z_{s_1Q} Y_{xQ} - Y_{s_1Q} Z_{xQ})(Y_{s_0Q} - Z_{s_0Q} y_P) +$$

$$X_{s_0Q} Z_{s_0Q} x_P + Z_{s_0Q}^2 x_q^2$$

which correspond to an extra addition step $s_0Q = s_1Q + xQ$. We can apply the same method to the other fraction $\frac{l_{s_2Q,-qQ}}{v_{s_1Q}}$. The Miller algorithm output the point xQ. We remark that $s_1Q = s_2Q + (-Q^q)$, thus the evaluation of $\frac{l_{s_2Q,-qQ}}{v_{s_1Q}}$ correspond to the addition step between s_2Q and $-Q^q$. We also can notice that $s_0Q = s_1Q + xQ$, we then obtain that $\frac{l_{s_1Q,xQ}}{v_{s_0Q}}$ correspond to the addition step between s_1Q and xQ the output of Miller's algorithm.In order to perform these computations, we have to precompute the points $s_2Q = -Q^{q^6}$, $s_1Q = -Q^q + Q^{q^6}$ and $s_0Q = xQ - Q^q + Q^{q^6}$. Those computations correspond to two Frobenius Q^q and Q^{q^6}. We follow the example of [BD18] the coarse estimation that a Frobenius evaluation cost $(k-1)M_q$.

We want to simplify the evaluation of $\frac{1}{(v_Q)^{q+q^6}}$. The power $q+q^6$ could be split into two Frobenius evaluation. We will modify the expression of $\frac{1}{(v_Q)}$ by the following way:

$$\begin{aligned}\frac{1}{(v_Q)}(P) &= \frac{1}{x_Q - x_P} \text{ we begin with affine coordinates} \\ &= \frac{(y_Q^2 - y_q^2)}{(x_Q - x_P)(y_Q^2 - y_q^2)}, \\ &= \frac{x_Q^2 + x_Q x_P + x_q^2}{y_Q^2 - y_q^2}.\end{aligned}$$

Using a twist of degree 3, we have that $y_Q^2 - y_q^2$ belongs to $\mathbb{F}_{q^{k/d}}$ and as a consequence will vanish during the final exponentiation.

In [ZL12], the authors made the assumption that affine coordinates should be more efficient than projective one as long as $I_k \leq 5.6M_k$. In order to be the more general, we will consider only the projective coordinates. We than transform the affine expression into the following projective one:

$$\frac{1}{(v_Q)}(P) = \frac{X_Q^2 + X_Q Z_Q x_p + x_q^2}{Z_Q^2}.$$

When using a twist, the coordinates Z_Q belongs to $\mathbb{F}_{q^{k/d}}$.

As a consequence, the evaluation of $\frac{1}{(v_Q)}$ is composed by $S_q + kM_q + S_{k/d} + M_{k/d}$ operations. We need two Frobenius maps (one by p and one by q^6) plus M_k in order to compute $\frac{1}{(v_Q)^{q+q^6}}$. Finally the total complexity of $(\frac{f_{x,Q}}{v_Q^{q+q^6}} \frac{l_{s_1 Q, xQ}}{v_{s_0 Q}} \frac{l_{s_2 Q, -qQ}}{v_{s_1 Q}})(P)$ is the computation of Miller's algorithm plus $(5k-4)M_q + S_q + S_{k/d} + M_{k/d} + 2Madd + 2M_k$. We present in Table 14 the estimation of the Miller algorithm when $k \in \{15, 33, 51\}$.

Operation	Complexity in projectives coordinates
Doubling step [ZL12]	$M_{3b} + kM_q + 3M_e + 9S_e + M_k + S_k$
Mixed addition [ZL12]	$kM_q + 12M_e + 5S_e + M_k$
Final line evaluation	$(5k-4)M_q + S_q + S_{k/d} + M_{k/d} + 2Madd$

Table 13: Complexity of Miller's steps using twist of degree 3

For $k \in \{27, 45\}$ we obtain a short vector on the pattern $[x, 0, \ldots 0, 1, 0, \ldots, 0]$. The optimal Ate pairing expression is then $\left(f_{x,Q} \frac{l_{q^{10}Q, xQ}}{v_{(x+q^{10}Q)}}(P)\right)^{\frac{q^k-1}{r}}$. An alternative family for the BLS 27 family was proposed by Zhang and Lin [ZL12]. They used a substitution of x by $-1/x$. The optimal Ate pairing expression is simplified into $(f_{x,Q})^{\frac{q^k-1}{r}}$. Another advantage to the Zhang and Lin family for BSL27 is the existence of x such that q and r are both prime.

For $k = 45$, the fraction is $\frac{l_{q^{16}Q, xQ}}{v_{(x+q^{16}Q)}}$.

As a consequence, for $k \in \{27, 45\}$ the pairing complexity is one Miller execution, plus one addition step.

For $k=21$, we obtain this short vector $[0,0,0,0,0,0,x^2,-x,1,0,0,0]$ and for $k=39$ this one $[0,0,0,0,0,0,0,0,0,0,0,0,x^2,-x,1,0,0,0,0,0,0,0,0,0]$.

We obtain the following expressions for the pairings $\left(\frac{f_{x^2,Q}^{q^6}}{f_{x,Q}^{q^7}v_{xQ}^{q^7}}\frac{l_{s_7Q,x^2Q}}{v_{s_6Q}}\frac{l_{s_8Q,-xqQ}}{v_{s_7Q}}\frac{v_Q}{v_{s_8}Q}(P)\right)^{\frac{q^k-1}{r}}$, where $s_6=x^2q^6-xq^7+q^8$, $s_7=-xq^7+q^8$ and $s_8=q^8$ and $\left(\frac{f_{x^2,Q}^{q^{12}}}{f_{x,Q}^{q^{13}}v_{xQ}^{q^{13}}}\frac{l_{s_{13}Q,x^2Q}}{v_{s_{12}Q}}\frac{l_{s_{14}Q,-xqQ}}{v_{s_{13}Q}}\frac{v_Q}{v_{s_{14}Q}}\right)^{\frac{q^k-1}{r}}$, where $s_{12}=x^2q^{12}-xq^{13}+q^{14}$, $s_{13}=-xq^{13}+q^{14}$ and $s_{14}=q^{14}$.

The pairing computation consists in one Miller execution as its result, $f_{x,Q}$, is an intermediate step of the computation of $f_{x^2,Q}$. The point xQ can also be saved during the execution of $f_{x^2,Q}$. The output is the point x^2Q. We must perform 6 Frobenius. The computation of $\frac{l_{s_{13}Q,x^2Q}}{v_{s_{12}Q}}\frac{l_{s_{14}Q,-xQ}}{v_{s_{13}Q}}$ are two extra addition steps. The denominators $v_{s_{13}Q}$ and $v_{s_{14}Q}$ cost $2(S_q+kM_q+S_{k/d}+M_{k/d})$. The complexity of the pairing computation for $k=21$ and $k=39$ is then one Miller execution $f_{x^2,Q}$ plus the extra computations $26(k-1)M_q+2Madd+2(S_q+kM_q+S_{k/d}+M_{k/d})+5M_k+I_k$.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx$
15	$382,4$	$31,8$	$1+2^2+2^{12}+2^{16}+2^{32}$	383	$32DBL+4Madd+Le+M_k$	$8\ 216M_q$
21	$350,4$	$21,9$	$2^4+2^6+2^9+2^{12}+2^{15}+2^{22}$	351	$44DBL+11Madd+$ extra computation	$19160M_q+I_k$
27	$298,5$	$15,1$	$2^3+2^4+2^{11}+2^{15}$	300	$15DBL+3Madd$	$6\ 401M_q$
33	311	13	$1+2+2^7+2^9+2^{14}$	336	$14DBL+4Madd+Le+M_k$	$54\ 320M_q$
39	308	11	$2^4+2^7+2^{10}+2^{11}+2^{13}$	375	$26DBL+9Madd+$ extra computation	$145\ 000M_q+I_k$
45	351	11	$1+2-2^3+2^8+2^{10}+2^{11}$	373	$12DBL+8Madd+Madd+M_k$	$17\ 832M_q$
51			no value for u below 2^{11}			

Table 14: Method 6.6 (BLS), 128 bits of security, twist of degree 3.

The Table 14 presents our results. The best candidates among those curves are for $k=15$ and $k=27$.

Curves admitting a twist of degree 2 The curves constructed using method 6.6 admits a twist of degree 2, when $k \bmod 6 \in \{2,4\}$. This means that $k\in\{14,16,20,22,26,28,32,34,38,40,44,46,50,52\}$.

The optimal pairing expression depends on the value of $k \bmod 6$. For every $k=2 \bmod 6$ we find the same short vector: $[x^2,x,1,0,\dots 0]$. The expression of the optimal Ate pairing is then $\left(f_{x^2,Q}f_{x,Q}^q\frac{l_{s_1Q,x^2Q}}{v_{s_0Q}}\frac{l_{s_2Q,xqQ}}{v_{s_1Q}}\right)^{\frac{q^k-1}{r}}$, where $s_0=x^2+xq+q^2$, $s_1=-xq+q^2$ and $s_2=q^2$.

The denominators are eliminated by the final exponentiation. As the results xQ and $f_{x,Q}$ are computed during the computation of $f_{x^2,Q}$ we count only one Miller evaluation. Two line evaluations plus 3 Frobenius and $3M_k$ are also necessary. The Table 15 presents the cost of the Miller execution.

When $k=4 \bmod 6$, one short vector is $[x^2,0,\dots,0,-x,0,\dots,0,1,0,\dots,0]$. For instance, for $k=16$, the optimal Ate pairing is then

$\left(\frac{f_{x^2,Q}}{f_{x,Q}^{q^3}}l_{s_1Q,x^2Q}l_{s_2Q,-xq^3Q}\right)^{\frac{q^k-1}{r}}$, where $s_0=x^2+xq^3+q^6$, $s_1=-xq^3+q^6$ and $s_2=q^6$. The cost is one Miller execution, plus 3 Frobenius, two line evaluations, $3M_k$ and one inversion over $\mathbb{F}_{q^k}$.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx$
14	350	$21,9$	$-1+2^6+2^7+2^9+2^{10}+2^{13}+2^{17}+2^{22}$	352	$44DBL+11Madd+2Le+3\pi_q+3M_k$	$11\ 173M_q$
16	$350,5$	16	$2^3+2^5+2^6-2^8+2^{11}-2^{14}+2^{17}$	369	$66DBL+4Madd+2Le+3\pi_q+3M_k+I_k$	$28\ 282M_q+I_k$
20	$350,65$	16	$1+2^6+2^{17}$	372	$34DBL+4Madd+2Le+3\pi_q+3M_k$	$15\ 990M_q$
22	364	13	2^5+2^{17}	474	$34DBL+2Madd+2Le+3\pi_q+3M_k+I_k$	$64\ 426M_q+I_k$
26	$306,6$	$10,9$	$2^2+2^3+2^5+2^7+2^{13}+2^{14}$	407	$28DBL+8Madd+2Le+3\pi_q+3M_k$	$91\ 242M_q$
28	373	$10,9$	$-2^2+2^7+2^8+2^{10}+2^{14}$	478	$28DBL+8Madd+2Le+3\pi_q+3M_k+I_k$	$21\ 778M_q+I_k$
32	280	$8,3$	$2+2^4+2^5+2^9$	309	$20DBL+6Madd+2Le+3\pi_q+3M_k$	$32\ 990M_q$
34	354	$8,8$	$2+2^3+2^5+2^{10}$	400	$20DBL+3Madd+2Le+3\pi_q+3M_k+I_k$	$102\ 102M_q+I_k$
38	356	$8,9$	$1+2^2+2^4+2^6+2^7+2^{10}$	409	$20DBL+9Madd+2Le+3\pi_q+3M_k$	$152\ 518M_q$
40	370	8	$-2^3+2^7+2^10$	466	$20DBL+3Madd+2Le+3\pi_q+3M_k+I_k$	$28\ 984M_q+I_k$
$44,46,50,52$			no value for u below 2^{12}			

Table 15: Method 6.6 (BLS), 128 bits of security, twist of degree 2

Curves without twists The remaining elliptic curves ($k=1$ or 5 mod 6) do not admit twists. As we have seen for construction 6.2, even if the theoretical dimension of $\mathbb{F}_{q^k}$ is smaller for prime embedding degree than for not prime embedding degrees, the lack of denominator elimination is a heavy drawback.

The expression of optimal Ate pairing according to the short vector $[x^2,-x,1,0,\dots,0]$ is valuable for $k\in\{11,17,23,29,35,41,47,53\}$.

$OptAte_{k11_{6.6}}:\mathbb{G}_1\times\mathbb{G}_2\rightarrow\mathbb{G}_3$,

$(P,Q)\rightarrow\left(f_{x^2,Q}f^q_{-x,Q}\frac{l_{s_1Q,x^2Q}}{v_{s_0Q}}\frac{l_{q^2Q,-xqQ}}{v_{s_1Q}}\right)^{\frac{q^k-1}{r}}$, where $s_0=x^2-xq+q^2$, $s_1=-xq+q^2$.

The complexity of this computation is one Miller's algorithm execution, two extra addition steps, two Frobenius, hence a total of $5M_k+I_k$ operations.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	$\approx$
11	311	13	$2^4+2^6+2^7+2^9+2^{10}+2^{14}$	338	$84\ 538M_q+I_k$
13	308	11	$2^4+2^7+2^{10}+2^{11}+2^{13}$	376	$125\ 722M_q+I_k$
29	643	$10,7$	$2^4-2^7+2^{10}+2^{11}$	690	$511\ 589M_q+I_k$
$17,19,23,25,31,35,37,41,43,47,49,51,53$			no value for u below 2^{12}		

Table 16: Method 6.6 (BLS), 128 bits of security, without twists

Comparison among the 6.6 (BLS) families of curve We compare in Table 17 the complexity of Miller's algorithm for the curves constructed using method 6.6.

Twist	k	u	$(\log_2(q))$	$\approx$	m_{32}	m_{64}
6	12	$-2^{77}+2^{50}+2^{33}$	460	$7\ 438M_q$	1 673 550	476 032
6	24	$-2^{32}+2^{28}+2^{12}$	319	$9\ 381M_q$	938 100	234 525
3	27	$2^3+2^4+2^{11}+2^{15}$	300	$6\ 401M_q$	640 100	160 025
3	15	$1+2^2+2^{12}+2^{16}+2^{32}$	383	$8\ 216M_q$	1 183 104	295 776
2	14	$-1+2^6+2^7+2^9+2^{10}+2^{13}+2^{17}+2^{22}$	352	$11\ 173M_q$	1 351 933	402 228
none	11	$2^4+2^6+2^7+2^9+2^{10}+2^{14}$	338	$84\ 538M_q+I_k$	−	−

Table 17: Comparison of the best candidates for method 6.6 at 128 bits of security

The curve BLS27 in the version of Zhang and Lin provides the smallest field $\mathbb{F}_q$ and the smallest number of operation over $\mathbb{F}_q$. This curve seems to provide the most efficient choice when considering the Miller loop among the BLS families. We analyse the final exponentiation in Section 6. The curves BLS 24 seems to provide the second most efficient Miller loop. Considering that, the BLS 24 curves have a degree 6 twist and that $\log_2(q_{24}^k) = 7656$ (when $\log_2(q_{27}^k) = 8058$), the comparison with the final exponentiation will decide between this two curves. Potentially, the BLS 15 curves could also be a competitor if a nice arithmetic over $\mathbb{F}_{p^5}$ can be deployed. Indeed, if we compare $\log_2(q_{15}^k) = 5745$ and $\log_2(q_{24}^k) = 7656$, which is roughly the size of the exponent for the final exponentiation, the BLS15 curve provide smaller field but the BLS24 curve can be implemented using the compressed squarings when no practical optimization are available in the literature for $k = 15$. As a conclusion, a precise implementation and analysis is necessary, in order to choose one between those three families.

4.6 Construction 6.7 from [FST10]

In this metafamily, we can construct curves with discriminant $D = 2$. They admit a twist of degree 2 if k is even, and no twist otherwise.

Curves admitting a twist of degree 2 The optimal Ate pairing is different for $k = 12$, $k = 24$ and respectively $k \in \{18, 30\}$.

For $k = 12$, it is $\left(\left(f_{x^2,Q} l_{-qQ,x^2Q}\right)(P)\right)^{\frac{q^k-1}{r}}$.

For $k = 24$, it is $\left((f_{x,Q} l_{-p}Q, xQ)(P)\right)^{\frac{q^k-1}{r}}$.

For $k \in \{18, 30\}$, it is $\left(\left(f_{x^4,Q} l_{-qQ,x^4Q}\right)(P)\right)^{\frac{q^k-1}{r}}$.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx M_q$
12	445	32	$1 + 2^{14} + 2^{17} + 2^{32}$	445	$64DBL{+}9Madd{+}Le{+}M_k$	13 976
24	381	4.4	$1 + 2^2 + 2^8 + 2^9 + 2^{32}$	381	$32DBL{+}4Madd{+}Le{+}M_k$	20 192
30	550	10	$1 + 2 + 2^5 - 2^7 + 2^{12}$	691	$48DBL{+}25Madd{+}Le{+}M_k$	56 133
36	541	16	$1 + 2^3 + 2^5 - 2^8 + 2^{11} + 2^{13} + 2^{16}$	547	$32DBL{+}13Madd{+}Le{+}M_k$	56 963
42	667	9	no value for u below 2^{11}			
48	525	24	$1 + 2^3 + 2^5 + 2^6 + 2^8 + 2^{10} + 2^{14} + 2^{24}$	525	$24DBL{+}7Madd{+}Le{+}M_k$	72 348

Table 18: Method 6.7, 128 bits of security, quadratic twist

Curves without twists The optimal Ate pairing is different for $k = 15$ and $k \in \{9, 21, 27\}$.

For $k = 15$, the shortest vector found is $[x^4 - 1, 1, 0, -1, 1, -1, 0, 1]$, the cost of the optimal Ate pairing in this case is the evaluation of $f_{x^4-1,Q}$, plus 6 addition steps, hence a total of $10M_k + I_k$.

For $k \in \{9, 21, 27\}$, it is $\left(f_{x^4,Q} \frac{l_{q^5Q,x^4Q}}{v_{[x^4+q^5]Q}}\right)^{\frac{q^k-1}{r}}$.

For $k = 21$, there are very few possible values for u, so that we could not provide a realistic example of such pairing,

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	Miller's cost	$\approx$
9	507	11	2607	520	$48DBL + 15Madd + Madd + M_k + I_k$	$31\ 369M_q + I_k$
15	607	9	22165	950	$56DBL + 34Madd + 6Madd + 10M_k + I_k$	$85\ 050M_q + I_k$
21	598	7	7315	1100	$52DBL + 23Madd + Madd + M_k + I_k$	$97\ 135M_q + I_k$
27	465	4	2941	1218	$48DBL + 16Madd + Madd + M_k + I_k$	$157\ 460M_q + I_k$
$33, 39, 45$			no value for u below 2^{10}			

Table 19: Method 6.7, 128 bits of security, without twists

Best candidate for method 6.7 The cost of Miller's loop for the curves without twists is much more expensive than the cost for curve with a quadratic twist. Among the curves with quadratic twists, the curves with $k = 12$ and $k = 24$ are the most promising. With $k = 12$ we have the least number of operation over $\mathbb{F}_q$, with $k = 24$ the smallest field $\mathbb{F}_q$. We compare the two pairings using a coarse estimation on the number of machine word operation in Table 20. According to our estimation, the most efficient pairing for curves constructed with method 6.7 should be implemented over the curve with $k = 12$.

k	$\log_{32}(q)$	Miller's in m_{32}	$\log_{64}(q)$	Miller's in m_{64}
12	14	$2\ 739\ 296m_{32}$	7	$684\ 824m_{64}$
24	12	$2\ 907\ 648m_{32}$	6	$726\ 912m_{64}$

Table 20: Method 6.7, Comparison of the best candidates at 128 bits of security.

4.7 Construction 6.20, 6.24 and "+" from [FST10]

We denote by "+" the construction described in [FST10] that relies on the application of Theorem 6.19 [FST10]. The method is to use one construction among 6.2, 6.3, 6.7, 6.20 or 6.24 and made the substitution $x^2 \to \alpha x^2$ in the definition of q and r, where α is a square free positive integer. The best choices for α are described in the Algorithm for Generating Variable-Discriminant Families [FST10]. The "+" doesn't change the security because (and hence doesn't change the key sizes) because we obtain the same values of k, $\log_2 q$ and polynomials in the SexNFS attacks. Indeed, if the fastest SexTNFS attack against a family uses two polynomials f and g, one could use either the same polynomials or $f(\alpha x^)$ and $g(\alpha x^2)$ for the "+" family. However, the degree of f and g is "too high" for all the families tested, so an attacker is bound to continue to use f and g.

For example , using the "+" method, we generate values of u such that $\log_2(u) = 13$ for $k = 11$ and construction 6.20, but for 128 bits of security u should be at least 20 bits. One can use our results and try to generate curves with nice discriminant. It is very important to remark that using the construction "+", we can construct elliptic curve with any discriminant. For instance, in the construction 6.2, when $k = 3 \bmod 6$, we cannot use any twist, but with construction "6.2+", we can generate curves with discriminant $D = 3$ and then use twists in order to improve the computation. By the same way, when $k = 0 \bmod 6$, the construction 6.2 allows a quadratic twist, while the construction "6.2+" allows a sextic twist.

Using construction 6.20 and 6.24, we obtain elliptic curves with discriminant $D = 1$. As a consequence, if k is even, we have a quadratic twist, otherwise we do not have a twist. For some embedding degrees, $q(x)$ is reducible so we had to apply the "+" construction.

The only drawback of the "+" method is that instead of searching for parameters u of a given bit size b we search for parameters y_0 of approximately $b/2$ bits. This gives less choices and we could not find parameters of low NAF weight for the constructions 6.20+ and 6.24+. We leave it as an open problem the generation of nice parameters and curves using the "+" method.

4.8 KSS families from [FST10]

The KSS families of elliptic curve was introduced by [KSS08]. It is a promising complete family for specific values of k. They are defined for $k = 16, 18, 32, 36, 40$ in [KSS08]. Scott and Guillevic [SG18] found a similar family with $k = 54$.

The KSS16 and KSS18 were already studied in the literature, we use the recent results from [BD18].

For $k = 32$, an expression of the optimal Ate pairing is $f_{x,Q} f_{-3,Q}^{q} f_{2,Q}^{q^8} l_{s_1 Q, xQ} l_{2q^8 Q, -3Q}$, with $s_1 = -3q + 2q^8$. This is almost the same expression for KSS36 curves, the difference is that the power of p is 7 and not 8. For both KSS32 and KSS36 curves, we search for a value u such that the most significant bits are both 1, this will guarantee that the computation of $3Q$ is the first addition step during the computation of $f_{x,Q}$. As a consequence the cost of this optimal Ate pairing is one Miller execution $f_{x,Q}$ plus $3\pi_q + 2Le + 4M_k + I_k$.

For $k = 40$, $f_{x,Q} f_{2,Q}^{q^{11}} l_{s_1 Q, xQ} l_{2q^{11} Q, -Q}$, with $s_1 = -q + 2q^{11}$. The cost is $f_{x,Q}$ plus $2\pi_q + 2Le + 3M_k$.

For $k = 54$, $f_{x,Q}^{q^9+1} l_{q^9 xQ + q^{10} Q, xQ} l_{q^{10} Q, q^9 xQ}$ [SG18].

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	$\approx$ Miller's cost
16	330	33	$-2^{34} + 2^{27} - 2^{23} + 2^{20} - 2^{11} + 1$	340	$7\ 534 M_q$
18	356	44	$2^{44} + 2^{22} - 2^{9} + 2$	352	$9\ 431 M_q$
32	344	19	$2^5 + 2^{10} + 2^{11} + 2^{19} + 2^{20}$	349	$19\ 321 M_q + I_k$
36	321	23	$1 + 2 + 2^4 + 2^9 + 2^{14} + 2^{17} + 2^{23} + 2^{24}$	329	$10\ 771 M_q + I_k$
40	376	17	$1 + 2^4 + 2^7 + 2^8 + 2^{13} + 2^{18}$	377	$18\ 254 M_q$
54	315,9	15,7	$2^3 + 2^7 + 2^{11} + 2^{15} + 2^{16}$	348	$20\ 427 M_q$

Table 21: KSS families, 128 bits of security

4.9 MNT curves

The MNT curves [MNT00] remain the preferred choice when implementing type-I (symmetric) pairings. In [LEMHT19], Phong et al. extended the original construction by Miyaji et al. [MNT00].

We find the following vectors and optimal Ate expression for the MNT curves:

- $k = 3$, $[6x - 2, 1]$, $e_{MNT3}(P, Q) = (f_{6x-2,Q}(P))^{(q^k-1)/r}$;
- $k = 4$, $[x, 1]$, $e_{MNT3}(P, Q) = (f_{x,Q}(P))^{(q^k-1)/r}$;
- $k = 6$, $[2x, 1]$, $e_{MNT3}(P, Q) = (f_{2x,Q}(P))^{(q^k-1)/r}$.

In [PSV06,SB04,LEMHT19], some examples of MNT curves are given. These parameters are more rare than for the complete families and the algorithms to compute them are more costly, so it is beyond the scope of this article to propose numerical values of u. Instead, we estimate the cost of

Miller's loop for this curves in Table 22. We consider that for $k = 3$ we do not have a twsit, but for $k = 4$ and $k = 6$ we consider that we have a degree 2 and 6 twist. The MNT curve with $k = 6$ would provide the most efficient pairing among the MNT families. But when considering Table 12, the MNT family is not at all competitive.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	$\approx$ Miller	$\approx$ Miller	m_{32}	m_{64}
3	1375	687	$687DBL + 10Madd$	$72\ 438M_q$	$> 133.10^6$	$> 34.10^6$
4	1060	530	$530DBL + 10Madd$	$24\ 870M_q$	$> 26.10^6$	$> 6.10^6$
6	770	385	$385DBL + 10Madd$	$15\ 690M_q$	$> 9.10^6$	$> 2.10^6$

Table 22: MNT curves, 128 bits of security

4.10 Other families

The article [FST10] presents a non exhaustive list of pairing-friendly elliptic curve constructions at the beginning of 2010.

There were other constructions like [DCC05,LZZW08] not included in [FST10]. In 2010, the ρ value was important when considering the efficiency of pairings. The curves constructed in [DCC05] have embedding degree already included in [FST10] but with larger ρ. It could be a reason why the results from [DCC05] were not included in [FST10]. However, the curve with embedding degree 15 in [DCC05] resists better the Kim-Barbulescu attack and we choose to evaluate them in our study. In [DCC05], other families are constructed with embedding degree $k = 12, 13, 14, 24, 48$. They do not provided efficient pairings, either because of the lack of discriminant $D = 3$ ($k = 13, 14$) or because the Kim-Barbulescu attack is very efficient and the required bit sizes make the pairing less efficient than others families ($k = 12, 24, 48$).

The k=9 family from [LZZW08] and the k=15 family from [DCC05] were studied in [FMP16], where Fouotsa et al. evaluate the cost the optimal Ate pairing computation for curves with odd embedding degree. The expression of the optimal pairing for this family is nice: $(f_{x,Q})^{\frac{q^k-1}{r}}$. It is the same expression for the family with embedding degree 9 studied by Lin et al. in [LZZW08]. Their results were that the $k = 9$ family is a little bit more expensive than the BN family.

We report in Table 23 the estimation of the Miller loop for those families at the 128 bits security level.

k	$\min(\log_2(q))$	$\min(\log_2(u))$	u	$(\log_2(q))$	$(\log_2(q^k))$	$\approx$ Miller	m_{32}	m_{64}
9	590	73	$2^{74} + 2^{35} - 2^{22}$	590	5 310	$8\ 808M_q$	2 050 048	512 512
15	383	$31, 9$	$2 + 2^{10} + 2^{16} + 2^{19} + 2^{32}$	383	5 745	$6\ 836M_q$	984 384	246 096

Table 23: The k=9 family of [LZZW08] and the k=15 family of [DCC05], 128 bits of security

Between those two curves, the construction from [DCC05] with $k = 15$ is the more efficient when considering the Miller loop. We provide in Section 6 the expression of the final exponentiation in order to decide between those two families.

4.11 Comparison of the best candidates between each constructions

We select one promising family for each method of construction and compare them all together in Table 24. It seems that the BLS 27 curve provides the most efficient Miller's loop. It is mandatory to check if the final exponentiation confirm this prediction.

Method	k	u	$(\log_2(q))$	$(\log_2(q^k))$	$\approx M_q$	m_{32}	m_{64}
6.2	14	$-1-2^4+2^7-2^{11}+2^{15}+2^{22}$	394	5 516	12 228	2 066 532	599 172
6.3	14	$1-2^2+2^6+2^9-2^{12}-2^{15}-2^{19}+2^{22}$	390	5 460	6 894	1 165 086	248 184
6.4	20	$1+2^4+2^{16}+2^{32}$	383	7 660	9 116	1 312 704	328 176
6.4	28	$1+2+2^3+2^4+2^8+2^9+2^{22}$	350	9 800	10 278	1 243 638	370 008
6.6	12	$-2^{77}+2^{50}+2^{33}$	460	5 520	7 438	1 673 550	476 032
6.6	15	$1+2^2+2^{12}+2^{16}+2^{32}$	383	5 745	8 216	1 183 104	295 776
6.6	24	$-2^{32}+2^{28}+2^{12}$	319	7 656	9 381	938 100	234 525
6.6	27	$2^3+2^4+2^{11}+2^{15}$	300	8 058	6 401	640 100	160 025
6.7	12	$1+2^{14}+2^{17}+2^{32}$	445	5 340	13 976	2 739 296	684 824
KSS	16	$-2^{34}+2^{27}-2^{23}+2^{20}-2^{11}+1$	340	5 540	7 534	911 614	271 224
DCC	15	$2+2^{10}+2^{16}+2^{19}+2^{32}$	383	5 745	6 836	984 384	246 096

Table 24: Comparison of the best candidates for 128 bits of security

We select one promising family by each method of construction and compare them all together in Table 24. The cost of Miller's algorithm together with the bit size of the target field offer a good hint about the fastest ones, but we cannot declare a clear winner. Efficient arithmetic is available for $k = 12, 16, 24$, but there is room for improvement for $k = 14, 15, 20, 27, 28$. At a higher level of refinement, BLS24, BLS27 and KSS16 were less studied than the other families. This makes a short list of fast pairings. Specific implementations are necessary to decide the overall champion and, depending on the application of pairings (short signature, identity based encryption, etc.) there might be several champions.

5 Complexity of Miller's loop at higher levels of security

In this section, we search for nice parameters for the optimal Ate pairing in order to make a comparison between all families at the 192 and 256 bits security level.

5.1 Complexity of the Miller's algorithm at 192 bits security level

We only provide here our most efficient curves for each construction.

We select one promising family by method of construction and compare them all together in Table 25.

It seems that the curve with $k = 27$ and construction 6.6 version Zhang Lin could provide the most efficient Miller's algorithm at the 192 bits security level. Other good candidates could be BLS 15, BLS 24 $k = 28$ construction 6.4 and DCC 15. The final exponentiation could shuffle this ranking. In Section 6 we compare the cost of the final exponentiation in order to determine which curve will provide the most efficient optimal Ate pairing.

Method	k	u	$(\log_2(q))$	$\approx M_q$	m_{32}	m_{64}
6.2	14	$1-2^3+2^7+2^8+2^{11}+2^{40}$	718	21 940	11 606 260	3 159 360
6.4	28	$-2^{31}-2^1-2^{13}-1$	494	13 250	3 392 000	848 000
6.6	15	$1+2^7+2^8+2^{12}+2^{15}+2^{48}$	574	11 649	3 774 276	943 569
6.6(ZL)	27	$2^{22}+2^{14}+2^9+2^8+2^4+2^3+2$	438, 5	16 178	2 734 082	792 722
6.6	24	$-2^{56}-2^{43}+2^9-2^6$	518	16 368	4 730 352	1 325 808
6.7	24	$-2^{48}+2^{12}+2^{42}+1$	572	38 871	12 594 204	3 148 551
KSS	16	$2^2+2^5-2^9+2^{22}-2^{23}+2^{51}$	500	24 795M_q	6 347 520	1 586 880
KSS	18	$2-2^5+2^9+2^{11}+2^{14}+2^{82}$	652	13 488M_q	5 948 208	1 632 048
DCC	15	$2^{50}-2^{40}+2^{15}+2^{13}+2^{11}+2^{10}$	598	8 975	3 239 975	897 500

Table 25: Comparison of the best candidates for 192 bits security level

5.2 Miller's complexity at 256 bit security level

We choose to give the estimation of the pairing computation for the curves such that $\log_2(q^k)$ is not greater then 15 000 and of course to the curves that provide efficient pairing implementation at 128 and 192 bits security level.

The curves providing $\log_2(q^k) \leq 15000$, are curves without twist and/or expensive pairing computation. We found out that even if the extension field $\mathbb{F}_{q^k}$ is not very large, the estimation cost for the Miller loop (see Table 26) is much more expensive than curves admitting twists reported in Table 27.

Method	k	u	$(\log_2(q))$	$\approx M_q$	m_{32}	m_{64}
6.2	17	$1+2+2^4-2^7-2^{11}+2^{15}$	564	387 184	125.10^6	31.10^6
6.2	19	$-1+2^4+2^5-2^7+2^{11}+2^{15}$	631	$> 190.10^3$	$> 76.10^6$	$> 19.20^6$
6.3	22	of ≈ 26 bits	of ≈ 674 bits	108 955	$> 103.10^6$	$> 13.10^6$
6.6	11	of $\approx 47,9$ bits	of ≈ 622 bits	134 046	$> 53.10^6$	$> 13.10^6$
6.7	9	$1+2^3-2^5-2^{10}+2^{13}+2^{14}+2^{20}+2^{21}$	990	61 373	58 979 453	15 711 488

Table 26: Comparison of the possibles exotic candidates for 256 bits security level

According to Table 27, the most efficient Miller's loop would be for the curves $k = 28$ construction 6.4 in [FST10], BLS15 and BLS27. Those curves correspond to the families such that $\log(q)$ is smaller than 1 000 bits.

6 The Computation of the final exponentiation

The computation of Tate pairing and its derivatives requires two steps. After computing Miller's loop as described in Sections 4 and 5, we have to carry out an extra step for the overall cost. This second step is called the final exponentiation where the result of Miller loop must be raised to the power $\frac{q^k-1}{r}$.

In this section, we present the complexity of computing the final exponentiation. Since we are considering Optimal Ate pairings [Ver10], which decrease the length of the Miller loop, then its complexity. Indeed the final exponentiation has become a significant component of the global

Method	k	u	$(\log_2(q))$	$\approx M_q$	m_{32}	m_{64}
6.2	14	$1{+}2{+}2^4{+}2^7{-}2^{12}{+}2^{15}{-}2^{73}{+}2^{76}$	$1362 bits$	38 523	$> 71.10^6$	18 645 132
6.3	14	of $\approx 85, 8$ bits	of ≈ 1545 bits	35 959	$> 82.10^6$	$> 20.10^6$
6.4	20	$1{+}2^3{-}2^6{+}2^{10}{-}2^{12}{+}2^{15}{+}2^{77}{+}2^{78}{+}2^{79}$	956	22 480	$20\ 232.10^3$	$5\ 058.10^3$
6.4	28	$1{+}2{+}2^6{+}2^8{+}2^9{-}2^{12}{+}2^{15}{+}2^{40}{+}2^{41}{+}2^{42}$	683	18 759	9 079 356	2 269 839
6.6	15	$1{+}2^5{+}2^7{+}2^9{+}2^{11}{+}2^{14}{+}2^{64}$	766	11 796	6 794 496	1 698 624
6.6	24	$2^{103}{-}2^{101}{+}2^{68}{+}2^{50}$	1024	37 126	38 017 024	9 504 256
6.6	27	$1{+}2{+}2^4{+}2^6{+}2^7{+}2^9{+}2^{10}{+}2^{12}{+}2^{29}$	578	20 800	6 739 200	1 684 800
6.7	12	$-1{+}2^4{+}2^9{-}2^{12}{+}2^{15}{+}2^{119}$	1663	57 279	$> 154.10^6$	$> 38.10^6$
KSS	16	$2{+}2^2{-}2^{12}{+}2^{15}{+}2^{114}$	1132	30 750	39 852 000	9 963 000
KSS	18	$2^{186}{-}2^{75}{-}2^{22}{+}2^4$	1484	37 437	82 698 333	21 563 712
DCC	15	$1{+}2{+}2^3{+}2^5{+}2^6{-}2^8{+}2^{15}{+}2^{112}$	1342	22 435	39 575 340	10 775 835

Table 27: Comparison of the best candidates for 256 bits security level

computation. Thanks to the cyclotomic polynomial, the final exponentiation can be broken down into two components as follows:

$$\frac{q^k-1}{r} = \frac{q^k-1}{\phi_k(q)} \times \frac{\phi_k(q)}{r}.$$

In this work, we are only interested in the computation of the second part of the final exponentiation. This part is called the hard part since computing $\frac{q^k-1}{\phi_k(q)}$ is the easy part of the final exponentiation and its computation requires some Frobenius (2 if k is even), some multiplications and an inversion in $\mathbb{F}_{q^k}$. Due to the results given in Sections 4, we will not consider all elliptic curves in our computation of the final exponentiation. Therefore, we focus elliptic curves of embedding degree $k = 9$, 15, 12, 16; 20; 24 and 28 for the 128 bits security level. For the security levels 192 and 256, we use the same method presented below, we have just to change the parameter u. Recall that the embedding degree is the most significant complexity parameter of a pairing friendly elliptic curve.

Through this part, we denote by d the hard part of the final exponentiation, i.e $d = \frac{\phi_k(q)}{r}$ and d' a multiple of d with r not dividing d'.

We keep the notations M_q, S_q, I_q for the cost of the multiplication, of the squaring and of the inversion in $\mathbb{F}_q$ and similarly M_k, S_k and I_k for the operations in $\mathbb{F}_{q^k}$ as they were introduced in Section 4.1. When it is clear from the context we drop the k index and write M, S and I for M_k, S_k and I_k. We add the notations E_u for an exponentiation by the parameter u and F_k for the cost of a Frobenius map in $\mathbb{F}_{q^k}$.

6.1 The case of $k = 9$

The elliptic curves of embedding degree $k = 9$ have not been intensively examined in the literature. The final exponentiation can be presented as follows:

$$\frac{q^9-1}{r} = (q^3-1) \times \frac{q^6+q^3+1}{r}.$$

In this paragraph, we will only consider construction presented in [FMP16] (recalled in Section 4.10) since it gives the most efficient computation of the final exponentiation by comparing with

constructions 6.2 and 6.7.
This family of elliptic curves is defined by the following q and r.

$$q = ((u+1)^2 + ((u-1)^2(2u^3+1)^2)/3)/4 \text{ and } r = (u^6+u^3+1)/3.$$

The representation of the hard part of the final exponentiation ,i.e, $\frac{q^6+q^3+1}{r}$ in basis q does not give the optimal vector. Therefore, we tried to find a new multiple of the second part of the final exponentiation that can be computed easily. For computational efficiency, a linear combination with a maximum number of zero coefficients is desired. By applying the LLL [LLL82] algorithm to the 8×48 integer matrix M, constructed as in [CKH11]). The most efficient vector is $u^3 d$, which is the same as the one given in [FMP16]. This is illustrated as follows:

$$d'(u) = \lambda_0 + \lambda_1 q + \lambda_2 q^2 + \lambda_3 q^3 + \lambda_4 q^4 + \lambda_5 q^5 \text{ with}$$

$$\begin{array}{ll}
\lambda_0 = -u^4 + 2u^3 - u^2 & \lambda_1 = -u^3 + 2u^2 - u \\
\lambda_2 = -u^2 + 2u - 1 & \lambda_3 = u^7 - 2u^6 + u^5 + 3 \\
\lambda_4 = u^6 - 2u^5 + u^4 & \lambda_5 = u^5 - 2u^4 + u^3.
\end{array}$$

The computation of the hard part of the final exponentiation requires 2 exponentiations by $(u-1)$ (since $\lambda_2 = -(x-1)^2$), 5 exponentiations by u, 7 multiplications, one squaring, q, q^2, q^3, q^4, q^5-Frobenius maps and two inversions. All these operations are performed in the cyclotomic subgroup of $\mathbb{F}_{p^9}$. When considering the parameter $u = 2^{74} + 2^{35} - 2^{22}$, the overall cost of the final exponentiation is then 519 S+ 25 M_9+ 3 I_{cyc}+ I_9+ q, q^2, q^3, q^4, q^5. In terms of multiplications in $\mathbb{F}_q$, the overall cost of the final exponentiation is $14043M + I$.

6.2 The case of $k = 12$

We showed in Section 4 that for computing Miller loops in the case of elliptic curves of embedding degree $k = 12$, it is better to consider BLS12 than BN curves. In this paragraph, we compare the cost of the final exponentiation of Optimal Ate pairing in both curves. Recall that

$$\frac{q^{12}-1}{r} = (q^6-1) \times (q^2+1) \times \frac{q^4+q^2+1}{r}.$$

The computation of the first part of the final exponentiation, i.e: the result of Miller loop raised to power $(q^6-1) \times (q^2+1)$, has almost the same cost for the two families (2 $q-$Frobenius, 2 multiplications and one inversion in $\mathbb{F}_{q^k}$ a finite field of 5535 bits for BN curves and respectively 5532 bits for BLS curves).

We present now the cost of computing the second part.

BN curves: We briefly present the BN elliptic curve [BN05] which is defined over $\mathbb{F}_q$ by $E: \ y^2 = x^3 + b$, where $b \neq 0$ is neither a square nor a cube and by a parameter u such that

$$r = 36u^4 + 36u^3 + 18u^2 + 6u + 1 \quad \text{and} \quad q = 36u^4 + 36u^3 + 24u^2 + 6u + 1.$$

The parameter u is chosen such that both q and r are prime numbers, we consider the parameter suggested in [BD18]: $u = 2^{114} + 2^{101} - 2^{14} - 1$.

From the given expressions of q and r, the hard part of the final exponentiation can be written as a function of u:

$$\frac{q^4-q^2+1}{r} = \lambda_0 + \lambda_1 q + \lambda_2 q^2 + \lambda_3 q^3 \quad \text{with} \begin{cases} \lambda_0 = -36u^3 - 30u^2 - 18u - 2 \\ \lambda_1 = -36u^3 - 18u^2 - 12u + 1 \\ \lambda_2 = 6u^2 + 1 \\ \lambda_3 = 1 \end{cases}.$$

There are many efficient methods for computing the hard part of the final exponentiation presented in [SBC+09], [DSD07], [CKH11] and in [DG16]. In this paragraph we present our new development of the multiple of this part presented by Fuentes et *al.* in [CKH11], which makes the computation of the part in question more efficient (we know that an exponent of a pairing is a pairing). So we give the following presentation.

$$\begin{aligned} 2u\left(6u^2+3u+1\right)\frac{q^4(u)+q^2(u)+1}{r(u)} &= \left(12u^2(u+1)-6u^2+4u-1\right)q^3 + \left(12u^2(u+1)-6u^2+6u\right)q^2 \\ &\quad + \left(12u^2(u+1)-6u^2+4u\right)q + \left(12u^2(u+1)+6u+1\right). \\ &=_3' q^3 + \lambda_2' q^2 + \lambda_1' q + \lambda_0' \end{aligned}$$

$$\text{with,} \begin{cases} \lambda_0' = (12u^2(u+1)+6u)+1 = c+1 \\ \lambda_1' = (\alpha_2 - 2u) \\ \lambda_2' = c - 6u^2 \\ \lambda_3' = \alpha_1 - 1 \end{cases}$$

Since the parameter u is odd, an exponentiation by $u+1$ is more efficient than by u since $WH(u+1) < WH(u)$. Therefore, our algorithm for computing the hard part of the final exponentiation, is more efficient than the methods presented in [DG16] and [BD18]. Our algorithm requires $2E_u + E_{u+1} + 9M_{12} + 3S_{12} + 3F_{12}$. The overall cost of the final exponentiation is $3E_u + 10M_{12} + 3S_{12} + 5F_{12}$. In term of complexity in $\mathbb{F}_q$, our method for computing the final exponentiation requires $7381M + I$ when we use the cyclotomic squaring and $5598M + 4I$ in the case of considering the compressed squaring in the cyclotomic subgroup.

BLS12 curves: BLS12 [BLS02] are defined over $\mathbb{F}_q$ by $E : y^2 = x^3 + b$ and by a parameter $u \in \mathbb{Z}$ such that:

$$\begin{cases} q = (u-1)^2(u^4-u^2+1)/3 + u \\ r = u^4 - u^2 + 1 \\ t = u + 1 \end{cases}$$

For computing the hard part of the final exponentiation, we refer to the algorithm presented in [GF18] and we adapted it to the parameter $u = -2^{77} + 2^{50} + 2^{33}$. Then, in terms of complexity in $\mathbb{F}_q$, the final exponentiation requires $8151M + I$ when we use the cyclotomic squaring and $6188M + 6I$ in the case of considering the compressed squaring in the cyclotomic subgroup.
Then comparing the final exponentiation complexity when considering these two curves is presented in the following table.

Curve	Using Cyclotomic squarings	Using Compressed squarings
BN	$7\ 381M + I$	$5\ 598M + 4I$
(m_{32})	$1\ 660\ 725 + I$	$1\ 259\ 550 + 4I$
(m_{64})	$472\ 384 + I$	$358\ 272 + 4I$
$BLS12$	$8\ 151M + I$	$6\ 188M + 6I$
(m_{32})	$1\ 833\ 975 + I$	$1\ 392\ 300 + 6I$
(m_{64})	$521\ 664 + I$	$396\ 032 + 6I$

Table 28: Final Exponentiation Complexity for $k = 12$

6.3 The case of $k = 14$

Computing the Optimal Ate pairing over elliptic curves of embedding degree $k = 14$ is not studied in literature. We showed in Section 4 that such curves are a good candidate for the 128 security level when considering constructions 6.2 and 6.3. Therefore, for the final exponentiation we will present the computation of these two methods. The final exponentiation consists of computing $\frac{q^{14}-1}{r} = (q^7 - 1) \times (q + 1) \times \frac{q^6-q^5+q^4-q^3+q^2-q+1}{r}$. We are interested in computing the hard part of the final exponentiation since computing $(q^7 - 1) \times (q + 1)$ is considered easy. Then, we write it in basis q as follows:

$$\frac{q^6 - q^5 + q^4 - q^3 + q^2 - q + 1}{r} = \lambda_0 + \lambda_1 q + \lambda_2 q^2 + \lambda_3 q^3 + \lambda_4 q^4 + \lambda_5 q^5.$$

Considering the construction 6.2 An elliptic curve of embedding degree $k = 14$ is defined over $\mathbb{F}_p$ by $E:\ y^2 = x^3 + ax + b$ and by a parameter u such that:

$$\begin{cases} p = (u^{16} + u^{15} + u^{14} - u^9 + 2u^8 - u^7 + u^2 - 2u + 1)/3 \\ r = u^{12} + u^{11} - u^9 - u^8 + u^6 - u^4 - u^3 + u + 1 \\ t = u^8 - u + 1. \end{cases}$$

The hard part of the final exponentiation is given by the following λ_i, $0 \le i \le 5$:

$$\lambda_0 = u^{16} + 2u^{14} + u^{12} - u^4 - 2u^2 - 5 \qquad \lambda_1 = u^{14} + 2u^{12} + u^{10} + u^4 + 2u^2 + 1$$
$$\lambda_2 = u^{12} + 2u^{10} + u^8 - u^4 - 2u^2 - 1 \qquad \lambda_3 = u^{10} + 2u^8 + u^6 + u^4 + 2u^2 + 1$$
$$\lambda_4 = u^8 + 2u^6 - 2u^2 - 1 \qquad \lambda_5 = u^6 + 3u^4 + 3u^2 + 1.$$

By doing some simplifications, we get the following expressions:

$$\lambda_0 = u^2 \times \lambda_{1,0} - \lambda_{5,1} - 4 \qquad \lambda_1 = u^2 \times \lambda_{2,0} + \lambda_{5,1} = \lambda_{1,0} + \lambda_{5,1}$$
$$\lambda_2 = u^2 \times \lambda_{3,0} + \lambda_{5,1} = \lambda_{2,0} - \lambda_{5,1} \qquad \lambda_3 = u^2 \times \lambda_{4,0} + \lambda_{5,1} = \lambda_{3,0} + \lambda_{5,1}$$
$$\lambda_4 = u^2 \times \lambda_{5,0} - \lambda_{5,1} = \lambda_{4,0} - \lambda_{5,1} \qquad \lambda_5 = u^2 \times (u^2 + 1)^2 + (u^2 + 1)^2 = \lambda_{5,0} + \lambda_{5,1}$$
$$\lambda_{5,0} = u^2 \times (u^2 + 1)^2 \qquad \lambda_{5,1} = (u^2 + 1)^2.$$

For computing the hard part of the final exponentiation, we need two exponentiations by $(u+1)$, an exponentiation by u^2 and 1 multiplication for computing λ_5. Then, for computing each of λ_1, λ_2, λ_3 and λ_4 we need an exponentiation by u^2 and one multiplication. The computation of λ_0 requires one exponentiation by u^2, 2 squarings and 2 multiplications in $\mathbb{F}_{q^{14}}$. We need also q, q^2, q^3, q^4 and q^5 Frobenius maps and 5 multiplications to multiply terms together to get the coherent result. When we consider the parameter $u = -1 - 2^{+}2^7 - 2^{11} + 2^{15} + 2^{22}$ proposed in Section 4, the computation of the final exponentiation requires then 17702 multiplications in $\mathbb{F}_q$.

In this section we will only consider construction 6.2 since it is more efficient than the other constructions. We studied all of them but in this paper we give only the efficient method for computing the Optimal Ate pairing.

6.4 The case of $k = 15$

In this paragraph, we give the cost of computing the final exponentiation of the Optimal Ate pairing on elliptic curves of embedding degree $k = 15$. That is, it is better to present the final exponentiation as follows since it is more efficient than considering the cyclotomic polynomial in our development:

$$\frac{q^{15}-1}{r} = (q^5 - 1) \times \frac{q^{10}+q^5+1}{r}.$$

We are interested on computing the hard part of the final exponentiation, i.e the computation of $\frac{q^{10}+q^5+1}{r}$, since the first one consists only on one inversion, on multiplication and a q^5-Frobenius map.
Considering construction given in Section 4.10, the parametrization of these elliptic curves is given by the following q and r polynomials of $u \in \mathbb{Z}$:

$$\begin{cases} q = q = (u^{12} - 2u^{11} + u^{10} + u^7 - 2u^6 + u^5 + u^2 + 2u + 1)/3 \\ r = u^8 - u^7 + u^5 - u^4 + u^3 - u + 1. \end{cases}$$

The decomposition of a multiple of the hard part in basis q is given by $\lambda_0 + \lambda_1 q + \lambda_2 q^2 + \cdots\cdots + \lambda_9 q^9$, where the λ_i, $0 \leq i \leq 9$ are presented in [FMP16] as follows:

$$\begin{cases} \lambda_2 = (((u-1)^2)(u^2+u+1)); & \lambda_1 = \lambda_2 u; & \lambda_0 = \lambda_1 u \\ \lambda_9 = \lambda_0 u; & \lambda_8 = \lambda_9 u; & \lambda_7 = \lambda_8 u \\ \lambda_6 = \lambda_7 u; & \lambda_5 = \lambda_6 u + 3; & \lambda_4 = M - \lambda_1 - \lambda_7; \\ \lambda_3 = M - \lambda_0 - \lambda_6 - \lambda_9; & \text{with, } \; M = \lambda_2 + \lambda_5 + \lambda_8 & \end{cases}$$

Then, the final exponentiation of the Optimal Ate pairing over this construction of elliptic curve of embedding degree $k = 15$ requires 2 exponentiations by $(u-1)$, 9 exponentiations by u, 20 multiplications, one cyclotomic squaring, 4 inversions in the cyclotomic subgroup of in $\mathbb{F}_{q^{15}}$ and q, q^2, q^3, q^4, q^5, q^6, q^7, q^8 and q^9 Frobenius maps. By using the parameter $u = 2^3 + 2^4 + 2^{15} + 2^{18} + 2^{32}$, we have to perform 288 S_{cyc}, 56 M_{15}, 5 I_{cyc}, I_{15} and q, q^2, q^3, q^4, q^5, q^6, q^7, q^8, q^9 Frobenius maps. By using the arithmetic results given in [MGI09] and in [FMP16], the overall cost of the final exponentiation is then 19190 multiplications in $\mathbb{F}_q$.

6.5 The case of $k = 16$

As showed in [BD18] and in [KNG+17], the elliptic curves of embedding degree $k = 16$ are a good candidate for computing Optimal Ate pairing for 128-bits security level. In this paragraph, we just recall the cost of the final exponentiation given in [GF16]. Before that, recall that an elliptic curve of embedding degree $k = 16$ is defined over $\mathbb{F}_p$ by the equation of the form $y^2 = x^3 + ax$ and by the parameter u such that

$$\begin{cases} t = 1/35\left(2u^5 + 41u + 35\right) \\ r = u^8 + 48u^4 + 625 \\ q = \frac{1}{980}\left(u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 + 625u^2 + 2398u + 3125\right). \end{cases}$$

The final exponentiation is based on computing $\frac{q^{16}-1}{r} = (f_1^{q^8-1})^{\frac{q^8+1}{r}}$. In [GF16], authors suggested to compute the following multiple of the hard part of the final exponentiation:

$$u^3/125 \times \frac{q^8+1}{r} = \sum_{i=0}^{\phi(16)-1} \lambda_i p^i = \lambda_0 + \lambda_1 q + \lambda_2 q^2 + \cdots + \lambda_7 q^7 \text{ with}$$

$$\begin{cases} \lambda_0 = 2u^3A + 55u^2B; & \lambda_4 = u^3A + 10u^2B \\ \lambda_1 = -4u^2A - 75uB; & \lambda_5 = 3u^2A + 100uB \\ \lambda_2 = -2uA - 125B; & \lambda_6 = -11uA - 250B \\ \lambda_3 = -u^4A - 24u^3B + 196; & \lambda_7 = 7A \end{cases}$$

and

$$A = u^3B + 56; \qquad\qquad B = (u+1)^2 + 4.$$

So for computing the hard part of the final exponentiation we need to perform 7 exponentiations by u, 2 exponentiations by $(u+1)$, 34 cyclotomic squarings in $G_{\phi_2(q^8)}$, 32 multiplications in $\mathbb{F}_{q^{16}}$, 3 cyclotomic cubings in $\mathbb{F}_{q^{16}}$ and q, q^2, q^3, q^4, q^5, q^6, q^6, q^7, q^8-Frobenius maps.
By considering the arithmetic given in [ZL12], the overall cost of the final exponentiation is then $18514M + I$ when considering the cyclotomic squaring.

6.6 The case of $k = 20$

The elliptic curves of embedding degree $k = 20$ have not been considered before in literature. In our estimation for Miller loop's costs, we showed that this curve may be a good candidate for computing the Optimal Ate pairing.
Using Construction 6.4 of the taxonomy we have:

$$\begin{cases} q = (u^{12} - 2u^{11} + u^{10} + u^2 + 2u + 1)/4 \\ r = u^8 - u^6 + u^4 - u^2 + 1 \end{cases}$$

In this case, the final exponentiation consists of raising the result of the Miller loop to power $\frac{q^{20}-1}{r}$. This can be simplified thanks to the cyclotmic polynomial as follows.

$$\frac{q^{20}-1}{r} = (q^{10} - 1) \times (q^2 + 1) \times \frac{q^8 - q^6 + q^4 - q^2 + 1}{r}.$$

In our computation of the Miller loop in Sections 4 and 5 we considered constructions 6.4, 6.6 and 6.7. But for the final exponentiation we will only consider construction 6.4 since it yields the most efficient decomposition in basis q of the hard part of the final exponentiation which is presented as follows:

$$\frac{q^8 - q^6 + q^4 - q^2 + 1}{r} = \sum_{i=0}^{\phi(20)-1} \lambda_i q^i = \lambda_0 + \lambda_1 q + \lambda_2 q^2 + \cdots + \lambda_7 q^7,$$

where

$$\begin{cases}\lambda_0 = u^{11} - 3u^{10} + 4u^9 - 3u^8 + 4u^6 - 6u^5 + 4u^4 - u^3 - u^2 + 2u + 3\\ \lambda_1 = u^{10} - 3u^9 + 4u^8 - 3u^7 + 4u^5 - 6u^4 + 5u^3 - 3u^2 + u\\ \lambda_2 = u^9 - 3u^8 + 4u^7 - 3u^6 + 4u^4 - 6u^3 + 5u^2 - 3u + 1\\ \lambda_3 = u^8 - 3u^7 + 4u^6 - 3u^5 + 3u^3 - 4u^2 + 3u - 1\\ \lambda_4 = u^7 - 3u^6 + 4u^5 - 3u^4 + u^3 + u^2 - 2u + 1\\ \lambda_5 = u^6 - 3u^5 + 4u^4 - 4u^3 + 3u^2 - u\\ \lambda_6 = u^5 - 3u^4 + 4u^3 - 4u^2 + 3u - 1\\ \lambda_7 = u^4 - 2u^3 + 2u^2 - 2u + 1.\end{cases}$$

For more efficiency for computing the hard part of the final expression, we propose the following development:

$$\begin{array}{ll}\lambda_0 = \lambda_1 u - \lambda_7 + 4; & \lambda_1 = \lambda_2 u\\ \lambda_2 = \lambda_3 u + \lambda_7 & \lambda_3 = \lambda_4 u - \lambda_7\\ \lambda_4 = \lambda_5 u - \lambda_7 & \lambda_5 = \lambda_6 u\\ \lambda_6 = \lambda_7 u - \lambda_7 & \lambda_7 = (u-1)^2 + \left(u(u-1)\right)^2\end{array}$$

By this development, the hard part of the final exponentiation then requires 9 E_u, $2E_{u-1}$, 2 squarings, 7 Frobenius maps, 14 multiplications and one inversion in the cyclotomic subgroup of $\mathbb{F}_{q^{20}}$. In terms of multiplications in $\mathbb{F}_q$ and by using the parameter $u = 1 + 2^4 + 2^{16} + 2^{32}$ proposed in Section 4, the final exponentiation requires 29250 multiplications in $\mathbb{F}_q$.

6.7 The case of $k = 24$

BLS curves of embedding degree 24 are important candidates for computing Optimal Ate pairing for both of the 128 and 192 security levels [BD18]. Recall that BLS24 curves are families of elliptic curves defined over $\mathbb{F}_q$ by the parametrization:

$$\begin{cases} q = (u-1)^2(u^8 - u^4 + 1)/3 + u\\ r = u^8 - u^4 + 1\\ t = u + 1\end{cases}$$

The final exponentiation for BLS24 curves is decomposed into two parts thanks to the cyclotomic polynomial

$$\frac{q^{24}-1}{r} = \left(q^{12}-1\right)\left(q^4+1\right)\frac{q^8-q^4+1}{r}.$$

The hard part of the final exponentiation can be decomposed in basis q [SBC+09] as:

$$\frac{q^8-q^4+1}{r} = \sum_{i=0}^{\phi(24)-1} \lambda_i q^i = \lambda_0 + \lambda_1 q + \lambda_2 q^2 + \cdots + \lambda_7 q^7,$$

where

$$\begin{cases}\lambda_0 = u^9 - 2u^8 + u^7 - u^5 + 2u^4 - u^3 + 3\\ \lambda_1 = u^8 - 2u^7 + u^6 - u^4 + 2u^3 - u^2\\ \lambda_2 = u^7 - 2u^6 + u^5 - u^3 + 2u^2 - u\\ \lambda_3 = u^6 - 2u^5 + u^4 - u^2 + 2u - 1\\ \lambda_4 = u^5 - 2u^4 + u^3\\ \lambda_5 = u^4 - 2u^3 + u^2\\ \lambda_6 = u^3 - 2u^2 + u\\ \lambda_7 = u^2 - 2u + 1.\end{cases}$$

The best result in the literature to our knowledge is the one presented in [GF18]. In their work, the hard part of the final exponentiation is presented as follows:

$$\begin{array}{ll}
\lambda_0 = \lambda_1 u + 3 & \lambda_1 = \lambda_2 u \\
\lambda_2 = \lambda_3 u & \lambda_3 = \lambda_4 u - \lambda_7 \\
\lambda_4 = \lambda_5 u & \lambda_5 = \lambda_6 u \\
\lambda_6 = \lambda_7 u & \lambda_7 = u^2 - 2u + 1
\end{array}$$

The overall cost of the hard part of the final exponentiation is then 8 exponentiations bu u, one exponentiation by $u/2$ (since u is even), one squaring, 10 multiplications and 7-Frobenius operations in $\mathbb{F}_{q^{24}}$. Then, we need to add two Frobenius operations, two multiplications and one inversion in $\mathbb{F}_{q^{24}}$ to compute the final exponentiation. By using the arithmetic presented in [AFK$^+$12] and the parameter $u = -2^{32} + 2^{28} + 2^{12}$ proposed in Section 4 the final exponentiation requires 18732 multiplications and 10 Inversions in $\mathbb{F}_q$ when considering the compressed squaring and 23400 multiplications and one inversion when considering the cyclotomic squaring.

6.8 The case of $k = 27$

Elliptic curves of embedding degree $k = 27$ are suitable for computing Miller loop. In this paragraph, we give the computation of the final exponentiation on this category of curves which is defined bu the parameter u as follow [ZL12]

$$\begin{cases} q = 1/3(u-1)^2(u^{18} + u^9 + 1) + u \\ r = 1/3(u^{18} + u^9 + 1) \\ t = u + 1. \end{cases}$$

In this case, the final exponentiation consists on computing

$$\frac{q^{27} - 1}{r} = (q^9 - 1)\frac{q^{18} + q^9 + 1}{r}$$

Then, the representation of the hard part of the final exponentiation can be given as described in [ZL12] as follow.

$$(u1)^2 \times (q^9 + u^9 + 1) \times (q^8 + uq^7 + u^2q^6 + u^3q^5 + \cdots + u^7q + u^8) + 3$$

This decomposition requires one inversion in $\mathbb{F}_{q^{27}}$, 17 exponentiations by u, 2 exponentiations by $(u - 1)$, 11 multiplications, 2 q^9, q, q^2, q^3, q^4, q^5, q^6, q^7 and q^8 Frrobenius maps. When considering our parameter $u = 2^3 + 3^4 + 2^{11} + 2^{15}$ given in Section 4 the overall cost of the final exponentiation is then 76980 multiplications and one inversion in $\mathbb{F}_q$.

6.9 The case of $k = 28$

In Sections 4 and 5 we obtained elliptic curves of embedding degree $k = 28$ have an efficient computation of the Miller loop. In this paragraph, we are interested in the final exponentiation. These elliptic curves are defined by the parameter u such that q and r are two polynomials of u. In the case of construction 6.4, q and r are defined as follows:

$$\begin{cases} q = (u^{16} - 2u^{15} + u^{14} + u^2 + 2u + 1)/4 \\ r = u^{12} - u^{10} + u^8 - u^6 + u^4 - u^2 + 1 \\ t = u + 1. \end{cases}$$

Note that we consider only this construction since it is more efficient than the others.
The final exponentiation in this case is based on computing

$$\frac{q^{28} - 1}{r} = \left(q^{14} - 1\right)\left(q^2 + 1\right)\frac{q^{12} - q^{10} + q^8 - q^6 + q^4 - q^2 + 1}{r}$$

The representation of the hard part in basis p gives:

$$\frac{q^{12} - q^{10} + q^8 - q^6 + q^4 - q^2 + 1}{r} = \sum_{i=0}^{11} \lambda_i q^i = \lambda_0 + \lambda_1 q + \lambda_2 q^2 \ldots \cdots + \lambda_{11} q^{11} \text{ with}$$

$\lambda_0 = u^{15} - 2u^{14} + u^{13} - u^3 + 2u^2 - u + 4$ $\quad$ $\lambda_1 = u^{14} - 2u^{13} + u^12 - u^2 + 2u - 1$
$\lambda_2 = u^{13} - 2u^{12} + u^{11} + u^3 - 2u^2 + u$ $\quad$ $\lambda_3 = u^{12} - 2u^{11} + u^{10} + u^2 - 2u + 1$
$\lambda_4 = u^{11} - 2u^{10} + u^9 - u^3 + 2u^2 - u$ $\quad$ $\lambda_5 = u^{10} - 2u^9 + u^8 - u^2 + 2u - 1$
$\lambda_6 = u^9 - 2u^8 + u^7 + u^3 - 2u^2 + u$ $\quad$ $\lambda_7 = u^8 - 2u^7 + u^6 + u^2 - 2u + 1$
$\lambda_8 = u^7 - 2u^6 + u^5 - u^3 + 2u^2 - u$ $\quad$ $\lambda_9 = u^6 - 2u^5 + u^4 - u^2 + 2u - 1$
$\lambda_{10} = u^5 - 2u^4 + 2u^3 - 2u^2 + u$ $\quad$ $\lambda_{11} = u^4 - 2u^3 + 2u^2 - 2u + 1.$

To have a more efficient computation, we present the following presentation of the λ_i with $0 \leq i \leq 11$.

$\lambda_0 = \lambda_1 u + 4$ $\quad$ $\lambda_1 = \lambda_2 u - \lambda_{11}$ $\quad$ $\lambda_2 = \lambda_3 u$
$\lambda_3 = \lambda_4 u + \lambda_{11}$ $\quad$ $\lambda_4 = \lambda_5 u$ $\quad$ $\lambda_5 = \lambda_6 u - \lambda_{11}$
$\lambda_6 = \lambda_7 u$ $\quad$ $\lambda_7 = \lambda_8 u + \lambda_{11}$ $\quad$ $\lambda_8 = \lambda_9 u$
$\lambda_9 = \lambda_{10} u - \lambda_{11}$ $\quad$ $\lambda_{11} = (u-1)^2 + \left(u(u-1)\right)^2$

This computation requires 13 exponentiations by u, 2 exponentiations by $(u-1)$ 17 multiplications, one squaring, q, q^2, q^3, q^4, q^5, q^6; q^7, q^8, q^9, q^{10} and $q^{11}-$ Frobenius in $\mathbb{F}_{q^{28}}$. For the overall cost of the final exponentiation, we add the cost of the easy part which is one inversion, 2 multiplications and q^2 and $q^{14}-$Frobenius in $\mathbb{F}_{q^{28}}$. In terms of multiplications in the finite field $\mathbb{F}_q$, the computation of the final exponentiation, when considering the parameter $u = 1 + 2 + 2^3 + 2^4 + 2^8 + 2^9 + 2^{22}$, requires 50302 multiplications.

In the following Table, we summarize the cost of the final exponentiation of the Optimal Ate pairing in the target elliptic curves.

Method	k	u	$(\log_2(q))$	$(\log_2(q^k))$	$\approx M_q$	m_{32}	m_{64}
6.2	14	$-1-2^4+2^7-2^{11}+2^{15}+2^{22}$	394	5 516	$17\ 702+I$	2 991 638	867 398
6.4	20	$1+2^4+2^{16}+2^{32}$	383	7 660	$29\ 250+I$	4 212 000	1 053 000
6.4	28	$1+2+2^3+2^4+2^8+2^9+2^{22}$	350	9 800	$50\ 302+I$	6 086 542	1 810 302
6.76	12	$-2^{77}+2^{50}+2^{33}$	460	5 520	$6\ 188+6I$	1 833 975	396 032
6.6	24	$-2^{32}+2^{28}+2^{12}$	319	7 656	$18\ 732+10I$	1 873 200	674 352
6.6	27	$2^3+2^4+2^{11}+2^{15}$	300	8 058	$76\ 980+I$	7 698 000	2 771 280
KSS	16	$-2^{34}+2^{27}-2^{23}+2^{20}-2^{11}+1$	340	5 540	$18\ 514+I$	2 240 194	666 504
DCC	15	$2+2^{10}+2^{16}+2^{19}+2^{32}$	383	5 745	$19\ 190+I$	2 763 360	690 840
BN	12	$2^{114}+2^{101}-2^{14}-1$	462	5 532	$5\ 598+4I$	1 259 550	358 272

Table 29: Comparison of the best candidates for 128 bits of security for the Final Exponentiation

6.10 The Overall cost

In this section we compare the cost of the Optimal Ate pairing in several elliptic curves for the 128 bits security level. For the 192 and 256 security, it is sufficient to consider the appropriate parameters u.

Method	k	u	Miller cost	Final.Expo	$\approx M_q$	m_{32}	m_{64}
6.2	14	$-1-2^4+2^7-2^{11}+2^{15}+2^{22}$	12 228	$17\ 702+I$	$29\ 931+I$	$5\ 058\ 339+I$	$1\ 466\ 619+I$
$6.4m_{64}$	20	$1+2^4+2^{16}+2^{32}$	9 116	$29\ 250+I$	$38\ 366+I$	$5\ 524\ 704+I$	$1\ 381\ 176+I$
$6.4m_{32}$	28	$1+2+2^3+2^4+2^8+2^9+2^{22}$	10 278	$50\ 302+I$	$60\ 580+I$	$7\ 330\ 180+I$	$2\ 968\ 420+I$
6.6	12	$-2^{77}+2^{50}+2^{33}$	7438	$6\ 188+6I$	$13\ 626+6I$	$2\ 887\ 182+6I$	$779\ 538+6I$
6.6	24	$-2^{32}+2^{28}+2^{12}$	9 381	$18\ 732+10I$	$28\ 113+10I$	$2\ 811\ 300+10I$	$908\ 877+10I$
6.6	27	$2^3+2^4+2^{11}+2^{15}$	6 401	$76\ 980+I$	$83\ 381+I$	$8\ 338\ 100+I$	$2\ 931\ 305+I$
KSS	16	$-2^{34}+2^{27}-2^{23}+2^{20}-2^{11}+1$	7 534	$18\ 514+I$	$26\ 048+I$	$3\ 151\ 808+I$	$937\ 728+I$
DCC	15	$2+2^{10}+2^{16}+2^{19}+2^{32}$	6 836	$19\ 190+I$	$26\ 026+I$	$3\ 747\ 744+I$	$936\ 936+I$
BN	12	$2^{114}+2^{101}-2^{14}-1$	12 068	$5\ 598+4I$	$17600+4I$	$3\ 960\ 900+4I$	$1\ 126\ 604+4I$

Table 30: Overall cost of the Optimal Ate pairing for 128 bits of security

7 Conclusion

In this work we extended the work of Barbulescu and Duquesne [BD18]. We give the parameters sizes for the Kim-Barbulescu attack for more than 150 families of pairing friendly elliptic curves from the literature. We highlight some families which used to be ignored and which became interesting after the attack. A precise implementation is necessary in order to determine the new champion depending on the application of pairings.

Among our candidates for an efficient pairing implementation at 128, 192 or 256 bits security levels, several present an unbalanced complexity between the Miller part and the final exponentiation. So according to the application of pairings (identity-based cryptography or signature scheme) the choice of the family could be very different. Given our estimation for Miller and final exponentiation complexity, for instance for a signature scheme one could use a BLS12 or BLS24 curve, whereas for an application of multipairing scheme one could use the BLS 27 curve (see Table 30). If a symmetric pairing is necessary, we provide the suitable parameters sizes in order to resist to the Kim-Barbulescu attack, in this case, the MNT 6 curve seems to provide the most efficient pairing.

We note that one should not restrict the search to pairings with $\rho := \log_2 q / \log_2 r$ equal to 1. Indeed, at 128 bits of security, the bit size of $\log_2(q^k)$ required to have a safe field side is often in the range 4000-5500. If $\rho = 1$ this requires k to be in the range 16-22. Some families like BN and Freeman's k=10 sparse family, cannot have k in this range, which gives the possibility to have a better speed with families of small k and large ρ.

We also note that for many families with embedding degree larger than 30 we have to increase the size of the pairings not for security reasons, but because no parameters u of the required size can guarantee that both $q(u)$ and $r(u)$ are prime. If new families are explored for 192 bits, they would lead to have k between 24 and 54, divisible by 6 and have $\deg(q)$ between 10 and 20. Indeed, for small values of q the SexTNFS algorithm has a small estimated cost while, for large values, the parameters are rare.

References

AFG+17. Reza Azarderakhsh, Dieter Fishbein, Gurleen Grewal, Shi Hu, David Jao, Patrick Longa, and Rajeev Verma. Fast software implementations of bilinear pairings. *IEEE Trans. Dependable Sec. Comput.*, 14(6), 2017.

AFGH06. Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1), February 2006.

AFK+12. Diego F. Aranha, Laura Fuentes-Castañeda, Edward Knapp, Alfred Menezes, and Francisco Rodríguez-Henríquez. Implementing pairings at the 192-bit security level. In *Pairing-Based Cryptography - Pairing 2012 - 5th International Conference, Cologne, Germany, May 16-18, 2012, Revised Selected Papers*, pages 177–195, 2012.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, 2004.

BCM+15. Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon. Subgroup security in pairing-based cryptography. In *Progress in cryptology –LATINCRYPT 2015*, volume 9230 of *Lecture Notes in Computer Science*, 2015.

BD18. Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *J. of Cryptology. Published online at https://link.springer.com/article/10.1007%2Fs00145-018-9280-5*, 2018.

BELL10. John Boxall, Nadia El Mrabet, Fabien Laguillaumie, and Duc-Phong Le. A variant of Miller's formula and algorithm. In *Pairing*, volume 6487 of *Lecture Notes in Computer Science*, 2010.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *LNCS*, 2001.

BGDM+10. Jean-Luc Beuchat, Jorge E. González-Díaz, Shigeo Mitsunari, Eiji Okamoto, Francisco Rodríguez-Henríquez, and Tadanori Teruya. High-speed software implementation of the optimal Ate pairing over Barreto–Naehrig curves. In *Pairing-Based Cryptography – Pairing 2010*, volume 6487 of *Lecture Notes in Computer Science*, 2010.

BGGM15. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In *Advances in Cryptology - Eurocrypt 2015*, volume 9056 of *Lecture Notes in Computer Science*, 2015.

BGJT14. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology - Eurocrypt 2014*, volume 8441 of *Lecture Notes in Computer Science*, 2014.

BGK15. Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Towed Number Field Sieve. In *Advances in Cryptology – Asiacrypt 2015*, volume 9453 of *Lecture Notes in Computer Science*, 2015.

BGLS03. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology - Eurocrypt 2003*, volume 2656 of *Lecture Notes in Computer Science*, 2003.

BGW05. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, 2005.

BLM$^+$09. Jean-Luc Beuchat, Emmanuel López-Trejo, Luis Martínez-Ramos, Shigeo Mitsunari, and Francisco Rodríguez-Henríquez. Multi-core implementation of the Tate pairing over supersingular elliptic curves. In *Cryptology and Network Security – CANS 2009*, volume 5888 of *Lecture Notes in Computer Science*, 2009.

BLS02. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing elliptic curves with prescribed embedding degrees. In *Security in Communication Networks*, 2002.

BLS04. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *J. of Cryptology*, 17(4), 2004.

BN05. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography - SAC 2005*, 2005.

BSS99. I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999.

CDK$^+$17. Sébastien Canard, Aïda Diop, Nizar Kheir, Marie Paindavoine, and Mohamed Sabt. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic. In *Asia Conference on Computer and Communications Security*. ACM, 2017.

Cha08. Steve Chang. Trend micro. http://www.trendmicro.fr, 2008.

CKH11. Laura Fuentes Castaneda, Edward Knapp, and Francisco Rodrguez Henrquez. Faster hashing to $\mathbb{G}_2$. In *Selected Areas in Cryptography – SAC 2011*, volume 2259 of *Lecture Notes in Computer Science*, 2011.

CLN10. Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, 2010.

DCC05. Pu Duan, Shi Cui, and Choong Wah Chan. Special polynomial families for generating more suitable elliptic curves for pairing-based cryptosystems. In *The 5th WSEAS International Conference on Electronics, Hardware, Wireless and Optimal Communications*, 2005.

DEHR18. Sylvain Duquesne, Nadia El Mrabet, Safia Haloui, and Franck Rondepierre. Choosing and generating parameters for low level pairing implementation on BN curves. *Appl. Algebra Eng. Commun. Comput.*, 29(2), 2018.

DG16. Sylvain Duquesne and Loubna Ghammam. Memory-saving computation of the pairing final exponentiation on BN curves. *Groups Complexity Cryptology*, 8(1), 2016.

DGS17. Quentin Deschamps, Aurore Guillevic, and Shashank Singh. Estimating size requirements for pairings: Simulating the tower-NFS algorithm in GF(p^n), 2017. Slides are available at `https://ecc2017.cs.ru.nl/slides/ecc2017-guillevic.pdf`.

Dry11. Robert Dryło. On constructing families of pairing-friendly elliptic curves with variable discriminant. In *Progress in cryptology – INDOCRYPT 2011*, volume 7107 of *Lecture Notes in Computer Science*, 2011.

DSD07. Augusto Jun Devegili, Michael Scott, and Ricardo Dahab. Implementing cryptographic pairings over Barreto-Naehrig curves. In *Pairing-Based Cryptography – Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, 2007.

EGI11. Nadia El Mrabet, Aurore Guillevic, and Sorina Ionica. Efficient multiplication in finite field extensions of degree 5. In *Progress in Cryptology – AFRICACRYPT 2011*, volume 6737 of *Lecture Notes in Computer Science*, 2011.

EJ17. N. El Mrabet and M. Joye. *Guide to Pairing-Based Cryptography.* Chapman & Hall/CRC Cryptography and Network Security Series. CRC Press, 2017.

Eur13. European Union Agency of Network and Information Security (ENISA). Algorithms, key sizes and parameters report, 2013 recommandations, version 1.0, October 2013. Publucation available at https://www.enisa.europa.eu/publications/algorithms-key-sizes-and-parameters-report.

FK18. Georgios Fotiadis and Elisavet Konstantinou. Generating pairing-friendly elliptic curve parameters using sparse families. *Journal of Mathematical Cryptology*, 12(2), 2018.

FM18. Georgios Fotiadis and Chloe Martindale. Optimal tnfs-secure pairings on elliptic curves with even embedding degree. Cryptology ePrint Archive, Report 2018/969, 2018. https://eprint.iacr.org/2018/969.

FMP16. Emmanuel Fouotsa, Nadia El Mrabet, and Aminatou Pecha. Computing optimal ate pairings on elliptic curves with embedding degree 9, 15 and 27. *IACR Cryptology ePrint Archive*, 2016.

FST10. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. of Cryptology*, 23(2), 2010.

GAL$^+$12. Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. Efficient implementation of bilinear pairings on ARM processors. In *Selected Areas in Cryptography – SAC 2012*, volume 7707 of *Lecture Notes in Computer Science*, 2012.

GF16. Loubna Ghammam and Emmanuel Fouotsa. Adequate elliptic curves for computing the product of n pairings. In *Arithmetic of Finite Fields – WAIFI 2016*, 2016.

GF18. Loubna Ghammam and Emmanuel Fouotsa. Improving the computation of the optimal ate pairing for a high security level. *J. Appl. Math. Comput.*, 1, 2018.

GMT19. Aurore Guillevic, Simon Masson, and Emmanuel Thom. Cocks-Pinch curves of embedding degrees five to eight and optimal ate pairing computation. Cryptology ePrint Archive, Report 2019/431, 2019. https://eprint.iacr.org/2019/431.

GS19. Aurore Guillevic and Shashank Singh. A comparison of pairing-friendly curves at the 192-bit security level, 2019. Available online at https://members.loria.fr/AGuillevic/files/talks/19_Roscoff_STNFS.pdf.

Hes08. Florian Hess. Pairing lattices. In *Pairing-based cryptography – Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, 2008.

HSV06. Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The Eta pairing revisited. *IEEE Trans. Information Theory*, 52(10), 2006.

JK16. J. Jeong and T. Kim. Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. Cryptology ePrint Archive, Report 2016/526, 2016. http://eprint.iacr.org/2016/526.

JLSV06. Antoine Joux, Reynald Lercier, Nigel P. Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, 2006.

JN09. Marc Joye and Gregory Neven, editors. *Identity-Based Cryptography*, volume 2 of *Cryptology and Information Security Series.* IOS press, 2009.

Jou00. Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory (ANTS-IV)*, volume 1838 of *Lecture Notes in Computer Science*, 2000.

JP13. Antoine Joux and Cécile Pierrot. The special number field sieve in $\mathbb{F}_{p^n}$ – application to pairing-friendly constructions. In *Pairing-Based Cryptography - Pairing 2013*, volume 8365 of *Lecture Notes in Computer Science*, 2013.

KB16. T. Kim and R. Barbulescu. The extended tower number field sieve: A new complexity for the medium prime case. In *Advances in Cryptology – CRYPTO 2016*, volume 9814 of *Lecture notes in computer science*, 2016.

Kle06. Thorsten Kleinjung. On polynomial selection for the general number field sieve. *Math. Comp.*, 75(256), 2006.

Kle08. Thorsten Kleinjung. Polynomial selection, 2008. In CADO workshop on integer factorization, INRIA Nancy, 2008. http://cado.gforge.inria.fr/workshop/slides/kleinjung. pdf.

KM05. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, 2005.

KNG+17. Md. Al-Amin Khandaker, Yuki Nanjo, Loubna Ghammam, Sylvain Duquesne, Yasuyuki Nogami, and Yuta Kodera. Efficient optimal ate pairing at 128-bit security level. In *Progress in Cryptology - INDOCRYPT 2017*, volume 10698 of *Lecture Notes in Computer Science*, 2017.

Knu97. Donald E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1997.

KSS08. Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In *Pairing-Based Cryptography – Pairing 2008*, 2008.

LEMHT19. Duc-Phong Le, Nadia El Mrabet, Safia Haloui, and Chik How Tan. On the near prime-order mnt curves. *Applicable Algebra in Engineering, Communication and Computing*, 30(2), 2019.

Len01. Arjen K Lenstra. Unbelievable security: Matching AES security using public key systems. In *Advances in cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, 2001.

LLL82. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4), Dec 1982.

LZZW08. Xibin Lin, Chang-An Zhao, Fangguo Zhang, and Yanming Wang. Computing the ate pairing on elliptic curves with embedding degree k= 9. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91(9), 2008.

Mat06. Dmitrii Viktorovich Matyukhin. Effective version of the number field sieve for discrete logarithm in a field $GF(p^k)$. *Trudy po Diskretnoi Matematike*, 9, 2006.

MC11. Dustin Moody and Lily Chen. Pairing project. https://csrc.nist.gov/Projects/Pairing-Based-Cryptography, 2011.

MGI09. Nadia El Mrabet, Nicolas Guillermin, and Sorina Ionica. A study of pairing computation for elliptic curves with embedding degree 15. *IACR Cryptology ePrint Archive*, 2009:370, 2009.

Mil04. Victor S. Miller. The Weil pairing, and its efficient calculation. *J. Cryptology*, 17(4), 2004.

MNT00. Atsuko Miyaji, Masaki Nakabayashi, and Shunzo Takano. Characterization of elliptic curve traces under FR-reduction. In *ICISC*, volume 2015 of *Lecture Notes in Computer Science*, 2000.

Mor91. François Morain. Building cyclic elliptic curves modulo large primes. In *Advances in Cryptology – Eurocrypt 91*, volume 547 of *Lecture Notes in Computer Science*, 1991.

MSS16. Alfred Menezes, Palash Sarkar, and Shashank Singh. Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In *Paradigms in Cryptology - Mycrypt 2016*, volume 10311 of *Lecture Notes in Computer Science*, 2016.

PSV06. D. Page, N. P. Smart, and F. Vercauteren. A comparison of MNT curves and supersingular curves. *Applicable Algebra in Engineering, Communication and Computing*, 17(5), 2006.

SB04. Michael Scott and Paulo S. L. M. Barreto. Compressed pairings. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*. Springer, 2004.

SBC+09. Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa. On the final exponentiation for calculating pairings on ordinary elliptic curves. In *Pairing-Based Cryptography – Pairing 2009*, 2009.

SG18. Michael Scott and Aurore Guillevic. A new family of pairing-friendly elliptic curves. In *Finite Fields arithmetic – WAIFI 2018*, Lecture Notes in Computer Science, 2018.

She10. Caroline Sheedy. *Privacy Enhancing Protocols using Pairing Based Cryptography.* PhD thesis, Dublin City University, 2010.

SLPR15. Justine Sherry, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. Blindbox: Deep packet inspection over encrypted traffic. *ACM SIGCOMM Computer communication review*, 45(4), 2015.

SS16a. P. Sarkar and S. Singh. Fine tuning the function field sieve algorithm for the medium prime case. *IEEE Transactions on Information Theory*, 62(4), 2016.

SS16b. P. Sarkar and S. Singh. A generalisation of the conjugation method for polynomial selection for the extended tower number field sieve algorithm. Cryptology ePrint Archive, Report 2016/537, 2016.

SS16c. Palash Sarkar and Shashank Singh. New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. In *Advances in Cryptology – Eurocrypt 2016*, volume 9665 of *Lecture Notes in Computer Science*, 2016.

Tat63. John Tate. Duality theorems in galois cohomology over number fields. In *International Congress of Mathematicians Stockholm 1962*, Djursholm: Inst. Mittag-Leffler. MR 0175892, 1963.

Tea05. Executive Team. Voltage security. https://www.voltage.com, 2005.

UW14. T. Unterluggauer and E. Wenger. Practical attack on bilinear pairings to disclose the secrets of embedded devices. In *2014 Ninth International Conference on Availability, Reliability and Security*, Sept 2014.

Ver10. Frederik Vercauteren. Optimal pairings. *IEEE Trans. Information Theory*, 56(1), 2010.

Wei40. André Weil. Sur les fonctions algebriques à corps de constantes fini. *Les Comptes rendus de l'Academie des sciences*, 210(MR 0002863), 1940.

ZL12. Xusheng Zhang and Dongdai Lin. Analysis of optimum pairing products at high security levels. In *Progress in Cryptology – INDOCRYPT 2012*, volume 7668 of *Lecture Notes in Computer Science*, 2012.

ZX18. Meng Zhang and Maozhi Xu. Generating pairing-friendly elliptic curves using parameterized families. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 101(1), 2018.