A short-list of pairing-friendly curves resistant to the Special TNFS algorithm at the 192-bit security level

Diego F. Aranha, Georgios Fotiadis, Aurore Guillevic

A short-list of pairing-friendly curves resistant to the Special TNFS algorithm at the 192-bit security level

Diego F. Aranha[1], Georgios Fotiadis[2] and Aurore Guillevic[1,3,4]

[1] Aarhus University, Aarhus N, Denmark
[2] Université du Luxembourg, Esch-sur-Alzette, Luxembourg
[3] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
[4] Univ Rennes, Inria, CNRS, IRISA, Rennes, France

Abstract. For more than two decades, pairings have been a fundamental tool for designing elegant cryptosystems, varying from digital signature schemes to more complex privacy-preserving constructions. However, the advancement of quantum computing threatens to undermine public-key cryptography. Concretely, it is widely accepted that a future large-scale quantum computer would be capable to break any public-key cryptosystem used today, rendering today's public-key cryptography obsolete and mandating the transition to quantum-safe cryptographic solutions. This necessity is enforced by numerous recognized government bodies around the world, including NIST which initiated the first open competition in standardizing post-quantum (PQ) cryptographic schemes, focusing primarily on digital signatures and key encapsulation/public-key encryption schemes. Despite the current efforts in standardizing PQ primitives, the landscape of complex, privacy-preserving cryptographic protocols, e.g., zkSNARKs/zkSTARKs, is at an early stage. Existing solutions suffer from various disadvantages in terms of efficiency and compactness and in addition, they need to undergo the required scrutiny to gain the necessary trust in the academic and industrial domains. Therefore, it is believed that the migration to purely quantum-safe cryptography would require an intermediate step where current classically secure protocols and quantum-safe solutions will co-exist. This is enforced by the report of the Commercial National Security Algorithm Suite version 2.0, mandating transition to quantum-safe cryptographic algorithms by 2033 and suggesting to incorporate ECC at 192-bit security in the meantime. To this end, the present paper aims at providing a comprehensive study on pairings at 192-bit security level. We start with an exhaustive review in the literature to search for all possible recommendations of such pairing constructions, from which we extract the most promising candidates in terms of efficiency and security, with respect to the advanced Special TNFS attacks. Our analysis is focused, not only on the pairing computation itself, but on additional operations that are relevant in pairing-based applications, such as hashing to pairing groups, cofactor clearing and subgroup membership testing. We implement all functionalities of the most promising candidates within the RELIC cryptographic toolkit in order to identify the most efficient pairing implementation at 192-bit security and provide extensive experimental results.

Keywords: pairing-friendly curves · SNARK · TNFS

E-mail: dfaranha@cs.au.dk (Diego F. Aranha), georgios.fotiadis@uni.lu (Georgios Fotiadis), aurore.guillevic@inria.fr (Aurore Guillevic)

1 Introduction

Pairings have been used as a black-box tool in the cryptographic landscape since the first constructive applications have been proposed in 2000 and 2001. Some of the early applications were identity-based encryption [BF01], short signatures [BLS01], and tripartite Diffie-Hellman key exchange [Jou04]. In the following decades, pairings increased in popularity following the explosion in research within the broad area of cryptographic computing, consisting in essential building blocks to modern digital signatures [PS16, TZ23], commitment schemes [KZG10], anonymous credentials [CL04], and zero-knowledge proof systems such as zkSNARKs [Gro16].

Since the practical constructions of cryptographic pairings over elliptic curves are not secure against quantum-computers, an active research area is developing quantum-safe alternatives for pairing-based cryptography. However, migrating to quantum-safe cryptography has disadvantages in terms of efficiency and compactness, and potentially loss of some functionality. Taking for example zkSNARKs, there are promising quantum-safe constructions, such as zkSTARKs [BBHR18] and LaBRADOR [BS23], but they may offer larger proof sizes and/or prover/verification times that are still linear in the size of the witness or statement. Despite the exciting work in progress to develop alternatives, such performance penalties can be prohibitive for many applications.

In this paper, we follow a nuanced approach and argue that practical applications of pairing-based cryptography will need more conservative parameter choices while quantum-safe alternatives are further developed. In order to offer security in the next decade, we envision that parameters at the 192- and 256-bit security level will be needed. For reference, the Commercial National Security Algorithm Suite version 2.0 mandates transition to quantum-safe algorithms by 2033, and establishes that elliptic curve cryptography (ECC) at the 192-bit security level should be used before transition, effectively deprecating ECC at 128-bit security. The rationale behind the latter decision is rather unclear and unlikely to be ever properly substantiated, but there are speculations around computing elliptic curve discrete logs in cube-root time [KM16], without as much storage cost as publicly-known algorithms [BL13].

1.1 Motivation

Concretely, pairings are bilinear maps that are instantiated on algebraic curves. Contrary to other domains of cryptography where the choice of parameters is limited, straightforward, and standardized, the picture is more elaborated with pairings. Genus two pairings do not offer any advantage compared to elliptic curve (genus one) pairings in terms of security, while at the same time the Jacobian arithmetic is by far more costly compared to elliptic curve point addition. Instances of efficient and secure pairings on genus two curves have been recently reported in [AFK24], however they are still less efficient than the well-known elliptic curve pairings. For genus one, Barreto-Naehrig (BN) curves over large characteristic fields [BN06] became the dominant choice at 128-bit security for many years, followed by Barreto-Lynn-Scott (BLS) curves with embedding degree 12 at the 192-bit level [AFK+13]. Supersingular curves over small-characteristic fields were also considered for smaller devices, but efficient algorithms and record computations for discrete logarithms over extension fields completely eliminated their viability.[1] These attacks do not apply in large characteristic, but the Kim-Barbulescu variant of the TNFS attack [KB16] still forced larger parameters, and the widely deployed BN-254/256 curves became outdated. Current deployments of pairing-based schemes are tailored to the updated 128-bit security level, and practical deployments have employed the BLS12-381 elliptic curve designed by the ZCash project [Bow17].

[1] In 2012 happened the first large record computation of discrete logarithm computation https://dldb.loria.fr in $\mathrm{GF}(3^{6\cdot 97})$ (923 bits), and a final milestone in 2019 with $\mathrm{GF}(2^{30750})$ [HSST12, GKL+21].

Scaling the security of pairing-friendly curves under the new constraints is challenging. Some previously obvious choices became undefined: alternative curves are not of prime order, contrary to BN curves, or might not have j-invariant 0 like BN curves. While the choice of parameters at the 256-bit security level seems to be clearer [KIK+17, BMDFAF19], the story is more complicated for the 192-bit security level due to a broader range of candidate parameters. Hence, we focus our efforts exactly on finding and benchmarking parameter choices at 192-bit security, providing a detailed overview in terms of security and efficiency. We discuss which curves to choose, together with precise security estimates; and performance in terms of the cost of pairing computation and group operations including hashing, membership testing, scalar multiplication and exponentiation. There has been a long and dense bibliography on pairing implementation, from designing new curves (see the survey article [FST10]) to dedicated hardware optimizations, which we extend with a study dedicated to the 192-bit security level.

1.2 Our Contributions

We review pairing-friendly curves at the 192-bit security level and provide TNFS-secure short-listed curves according to popular criteria. Some pre-selected curves were sketched in [Gui20, §5]. We also consider Scott–Guillevic (SG), Fotiadis–Martindale (FM), and Gasnier–Guillevic (GG) curves [SG18, FM19, GG23]. Because the *search space* of pairing-friendly curve families is too large for a complete survey, we restrict the evaluated curves w.r.t. the following a-priori criteria:

- Embedding degrees from 15 to 28. We justify this choice in Section 2.5.
- curves with high degree twists: **quartic twist**: $4 \mid k$ and $j(E) = 1728$ ($k = 16, 20, 28$), **sextic twist**: $6 \mid k$ and $j(E) = 0$ ($k = 18, 24$), and also **cubic twist**: $3 \mid k$ odd and $j(E) = 0$ ($k = 15, 21, 27$).
- variable ρ-value from the lowest possible according to [FST10], to $\rho = 2$ thanks to [FK19, FM19]. We observe that at fixed k, a larger ρ can provide better performance.

We provide a full comparison of curves in terms of pairing efficiency, and $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ group operations:

- Formulas for optimal ate pairing, fast $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ cofactor clearing and subgroup membership testing.
- Formulas for hashing into $\mathbb{G}_1$, $\mathbb{G}_2$ with SwiftEC [CSRT22] and Koshelev's generalization [Kos24]. A side-contribution is an implementation-friendly description of hashing, potentially impacting other parameters for 128-bit security (e.g. embedding degree 8 curves [GMT20]).
- SageMath and Magma prototype code. Automated estimated cost in terms of multiplications in the base field GF(p).
- An optimized implementation within the RELIC library, and benchmarks of pairings and group operations for the most promising curves according to the SageMath estimates.

2 Preliminaries

Notations. Denote E an ordinary elliptic curve defined over a prime field $\mathbb{F}_p$. Denote k the *embedding degree* such that the pairing embeds a pair of points of $E(\mathbb{F}_p)$ and $E(\mathbb{F}_{p^k})$ onto $\mathbb{F}_{p^k}^*$. The three pairing groups of prime order r are denoted as usual $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$.

2.1 Pairings

The state-of-the-art implementations of pairings at 128-bit security are for the optimal ate pairing on different types of curves. Below we recall the definition of the optimal ate pairing and the formula to compute the Miller loop. For more information on this pairing type we refer to Vercauteren's paper [Ver10]. Furthermore, we give a brief description of the final exponentiation. For a complete introduction on pairings we refer to Craig Costello's guide *Pairings for beginners* [Cos12].

Optimal ate pairing Miller loop. This is defined as the bilinear, non-degenerate and efficiently computable map $e\colon \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{G}_\mathrm{T}$, with $\mathbb{G}_1 = E[r] \cap \ker(\pi_p - [1])$ and $\mathbb{G}_2 = E[r] \cap \ker(\pi_p - [p])$, where $\pi_p : E \to E$ is the p^{th}-power Frobenius endomorphism.

Let L be the $\varphi(k)$-dimensional lattice defined as:

$$L = \begin{pmatrix} r & 0 & 0 & \dots & 0 \\ -p & 1 & 0 & \dots & 0 \\ -p^2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -p^{\varphi(k)-1} & 0 & 0 & \dots & 1 \end{pmatrix}$$

and let $v = (c_0, c_1, \dots, c_{\varphi(k)-1})$ be the shortest vector of the lattice L. Then the coordinates of the shortest vector v satisfy the following relation:

$$\sum_{i=0}^{\varphi(k)-1} c_i p^i \equiv 0 \bmod r, \tag{1}$$

where the coordinates c_i are functions of the seed u, used to instantiate the polynomials $(p(x), t(x), r(x))$. Based on this setup, the formula for computing the optimal ate pairing is [Ver10]:

$$(Q, P) \to \left[\prod_{i=0}^{\varphi(k)-1} f_{c_i,Q}^{p^i}(P) \cdot \prod_{i=0}^{\varphi(k)-2} \frac{\ell_{R_i,S_i}(P)}{v_{R_i+S_i}(P)} \right],$$

where $R_i = [s_{i+1}]Q$, $S_i = [c_i p^i]Q$ and the scalars s_i are defined as:

$$s_i = \sum_{j=i}^{\varphi(k)-1} c_j p^j,$$

for every $i = 0, \dots, \varphi(k) - 2$. In addition, $\ell_{R_i,S_i}(P)$ denotes the line passing through the points R_i and S_i, which is evaluated at P, and $v_{R_i+S_i}(P)$ is the vertical line through the point $R_i + S_i$, evaluated at P.

The optimal ate pairing formula can be simplified using certain properties of the Miller loop. In particular:

- For every $c_i < 0$, we have:

$$f_{c_i,Q}(P) = \frac{1}{f_{-c_i,Q}(P) \cdot v_{[c_i]Q}(P)}.$$

- By [BKLS02, Theorem 2] the following holds:

$$f_{a+b,Q}(P) = f_{a,Q}(P) \cdot f_{b,Q}(P) \cdot \frac{\ell_{[a]Q,[b]Q}(P)}{v_{[a+b]Q}(P)}.$$

- By [ALH10, Lemma 1] the following relation holds:

$$f_{ab,Q}(P) = f_{b,Q}(P)^a \cdot f_{a,[b]Q}(P) = f_{a,Q}(P)^b \cdot f_{b,[a]Q}(P).$$

- We have: $f_{0,Q}(P) = f_{1,Q}(P) = f_{-1,Q}(P) = 1$.
- Every scalar multiplication $[p^i]Q$ can be efficiently computed by applying the p^i-power Frobenius endomorphism $\pi_i : E \to E$. Therefore we write $[p^i]Q = \pi_i(Q)$, for every $i = 1, \ldots, \varphi(k) - 1$.

Final exponentiation. The final exponentiation is the process of raising an element in $\mathbb{F}_{p^k}^*$ to the exponent $e = (p^k - 1)/r$. When working with pairing groups, this is accomplished by factorizing e as follows:

$$e = \frac{p^k - 1}{\Phi_k(p)} \cdot \frac{\Phi_k(p)}{r},$$

where we refer to $e_{\text{easy}} = (p^k - 1)/\Phi_k(p)$ as the *easy part* and $e_{\text{hard}} = \Phi_k(p)/r$ as the *hard part*. Both parts can be significantly simplified, depending on weather the embedding degree k is even, composite odd, or prime. The formula for computing the easy part of the final exponentiation is the same for all pairing-friendly elliptic curves with the same embedding degree. On the contrary, the formula for computing the hard part of the final exponentiation is different for each family of curves. There are different methods for simplifying the hard part, which we explore in Section 3.

2.2 Constructing pairing-friendly curves and families

Plain ordinary elliptic curves are not pairing-friendly, in particular, their embedding degree k has the magnitude of r so the pairing is impracticable. Pairing-friendly curves should be designed specifically and there have been a long series of papers revealing new curves. The first ordinary curves were MNT ($k = 3, 4, 6$), followed by BLS, Brezing–Weng, the popular BN curves, and the KSS curves [MNT01, BLS03, BW05, BN06, KSS08]. Freeman, Scott and Teske delivered a useful taxonomy [FST10]. The recent works are a generalization of the BN construction [SG18] and the KSS construction [GG23] though without finding new prime-order curves. With the deployment of SNARK and the need of cycles of prime-order pairing-friendly curves, impossibility results are also investigated [BMUS23, CCW19]. We also mention a very recent work on higher genus cycle constructions for SNARK [CRSCN24]. In this work we concentrate on complete families, parameterized by polynomials and of fixed small discriminant.

2.3 Pairings at 128-bit security

2.3.1 Assessing the TNFS-security level of pairing-friendly curves.

In 2016 the Kim–Barbulescu attack downgraded the security level of many pairing-friendly curves [KB16, KJ17], in particular the curves with composite embedding degree such as 12, 16, 18, 24. Because choosing $k = 2^i 3^j$ allows an implementation-friendly towering and efficient field arithmetic, such choices are very common. Menezes, Sarkar, and Singh [MSS16] were the first to generate TNFS-secure parameters, recommending BLS12 curves of about 384 bits. Since then, the curve BLS12-381 is replacing BN-254 curves almost everywhere. We can mention the standardization effort about pairing-friendly curves [SKSW22], [BGW+22] on BLS signatures, [TL23] on BLS-curve-based key representation. Apart from BLS12, there are other specific proposals in other contexts [GMT20, FK19, FM19, Fot21, CDS20].

Analyzing TNFS is a challenging task. Barbulescu and Duquesne made recommendations in a worst-case scenario with a powerful attacker [BD19] (with conjectured hypotheses in the polynomial selection step of the TNFS algorithm). Guillevic and Singh [GS21] refined the TNFS analysis with the computation of the estimator *alpha*, and released an implementation of the simulator. There are very few record computations with TNFS. To the best of our knowledge, we are aware of Oisín Robinson's work in $\mathrm{GF}(p^4)$ [Rob22] and Gabriele De Micheli in $\mathrm{GF}(p^6)$ [DGP21].

In parallel of a better understanding of the TNFS algorithm, new parameter sets are proposed [Gui20, CDS20]. A short-list at the 128-bit security level [Gui20] was published in 2020, it gave hints on possible curves at the 192-bit security level but missed the shortest $\mathbb{G}_1$ constraint [CDS20] and lacked BLS24 at the 128-bit security level. More recently there is Guillevic's blog post [Gui21].

Better pairing implementations continue to be an active area of research, with the long-term software development in C++ RELIC [AGM+] (see for example [APR21] for recent benchmarks) but also newer projects in Golang or Rust for SNARK (see the survey [AHG23]). Latest improvements about pairing computations include a faster final exponentiation for all BLS curves [HHT20] thanks to a decomposition pattern, and for KSS18 curves[CHZ22]; implementation of pairings and group operations on prime-embedding-degree pairing-friendly curves BW13-310 and BW19-286 (with endomorphisms but without twists) [DZZ23a, DZZ23b, FAGA23, DZZZ21] (Yu Dai's github at [Dai23]); Cofactor clearing and subgroup membership testing on pairing-friendly curves [HGP22, DLZZ23]; pairings in arithmetic circuits in the context of SNARK [Hou23].

The recommendations at the 128-bit security level are the following [Gui21]. For a **prime-order curve** e.g. for a hybrid cycle in the context of SNARK, pick a BN curve of 384–448 bits (Freeman and MNT curves are also possible but less efficient). For the **fastest pairing**, choose a BLS12 curve of 384–448 bits. For **smallest** $\mathbb{G}_1$, set a BLS24 curve of ≈320 bits, for even smaller $\mathbb{G}_1$, BW13-P310, BW19-P286 or BLS48-286, SG54-283. Finally for **small embedding degree** k, there are curves with $k = 1$, supersingular curves with $k = 2$, and modified Cocks–Pinch curves with $5 \leq k \leq 8$.

The papers [KIK+17, BMDFAF19] consider the 256-bit security level with embedding degrees up to 48 (BLS48-581) and 54 (SG54-569). While [BD19] recommends a KSS18-1484 or a BLS24-1032 curve, [BEG19] expects a BLS27-559 curve (however this looks undersized). We collected the data in Appendix C Table 22.

2.4 Previous work on 192-bit security

The situation at the 192-bit security level is unclear, for at least two reasons. First there are more parameters to tune in TNFS for higher extension degrees. Second, there are more curve families to investigate (the embedding degree range to consider is larger for example). A nice short-list of best curves at the 192-bit security level is missing. Not only pairing computation but $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ operations are important and need to be assessed. The size of $\mathbb{G}_1$ and $\mathbb{G}_2$ is an important criterion in certain usecases. Barbulescu and Duquesne focused on KSS18 and BLS24 (see Table 1). In 2019, Barbulescu, El Mrabet and Ghammam posted a preliminary report on many curves [BEG19]. Guillevic and Singh [GS21] and Guillevic [Gui20] proposed seeds and keysizes at the 192-bit security level for some families of curves (Tables 2 and 3). We aim at consolidating the knowledge for pairing-friendly curves at the 192-bit security level.

2.5 Justification of our a priori criteria

We listed in Section 1.2 our a priori criteria to restrict the considered curves. First we set boundaries on the embedding degree: $15 \leq k \leq 28$. The earlier work stated large bounds [Gui20, Eq. (8)]: $7168 \leq 384\rho k \leq 14336$. Because we target high degree twist curves, the

Table 1: Curves at the 192-bit security level from [BD19, Tab. 9, §7.6].

k	Curve	NFS variant	seed	p, bits	p^k, bits	$\mathbb{F}_{p^k}$ sec, bits
18	KSS	STNFS	85 bits	678	12200	192
24	BLS	STNFS	56 bits	555	13300	192
18	KSS	STNFS $\kappa = 1$	$u = -2^{85} - 2^{31} - 2^{26} + 2^6$	676	12168	204
24	BLS	STNFS $\kappa = 1$	$u = -2^{56} - 2^{43} + 2^9 - 2^6$	559	13416	204

Table 2: Curves at the 192-bit security level from [Gui20, Tab. 10, §5].

k	Curve	seed, bits	r, bits	p, bits	p^k, bits	$\mathbb{F}_{p^k}$ sec, bits
14	FST 6.3	52	620	928	12979-12992	194
15	BLS	78	620	928	13906-13920	193
20	FST 6.4	56	448	670	13371-13400	192
21	BLS	32	384	511	10691-10719	195
27	BLS	22	384	427	11496-11524	212
28	FST 6.4	32	384	510	14243-14280	208

embedding degree k is always composite. Kim–Barbulescu extended TNFS applies well and we rather set

$$10000 \leq 384\rho k \leq 14336 \ .$$

we scan the curve families of the taxonomy paper plus the newer ones listed in 1.2 (Fotiadis–Martindale, Scott–Guillevic) and obtain these bounds. We provide a Sagemath script to reproduce this result. For RSA keysizes at the 192-bit security level, a modulus of 7680 bits is required according to NIST, while Lenstra Updated corresponds to 12548 bits[2]. For $k = 15$, the work [Gui20] already considered BLS15 and stated that p of 928 bits offers 192 bits of security. Such p is very large and we do not include it in our analysis. However we can take $k = 15$ and $\rho = 2$ as in [FK19]. The parameters are p of 768 bits, r of 384 bits and the security level is about 192 bits, for p^k of about 11520 bits. the paper [Gui20] already established that there is no curve family whose embedding degree is larger than 28 but whose ρ-value is small enough to satisfy $384\rho k \leq 14336$, hence $k \leq 28$.

2.6 Security Analysis: TNFS and its variants

In [BD19], Barbulescu and Duquesne further investigated the consequences of the extended Tower NFS algorithm of Kim et al. [KB16], after a first analysis in [MSS16]. In particular, they simulated worst-case scenarii from a cryptographer perspective to obtain key-size

[2] https://www.keylength.com/en/compare/

Table 3: Seeds at the 192-bit security level from [GS21, Tables 3, 4, 7].

k	curve	r, bits	p, bits	p^k, bits	seed u	$\mathbb{F}_{p^k}$ sec
12	BN	1022	1022	12255	$-2^{254} + 2^{33} + 2^6$	191
12	BLS12	768	1150	13799	$-2^{192} + 2^{188} - 2^{115} - 2^{110} - 2^{44} - 1$	193
16	KSS16	605	766	12255	$2^{78} - 2^{76} - 2^{28} + 2^{14} + 2^7 + 1$	194
18	KSS18	474	638	11477	$2^{80} + 2^{77} + 2^{76} - 2^{61} - 2^{53} - 2^{14}$	193
24	BLS24	409	509	12202	$-2^{51} - 2^{28} + 2^{11} - 1$ [CLN11]	193

Table 4: Choices of parameters in [BD19, GS21].

parameter		Barbulescu–Duquesne [BD19]	Guillevic–Singh [GS21]
c_{sieve}	cost of sieving, per relation	1	$\log\log B$
c_{linalg}	factor of lin. algebra	1/4	$\lceil \log_2 \ell/64 \rceil$
c_{filter}	filtering factor	$\log_2 B$	20
c_{density}	matrix' weight per row	128	200
$\mathcal{A}$	automorphisms	$k/\gcd(\deg h, k/\deg h)$	$\text{aut}(h)\,\text{aut}(f,g)$

recommendations. We recall their formula [BD19, Eq. (2)]:

$$\begin{aligned}\text{cost} &= c_{\text{sieve}} \frac{1}{\mathcal{A}} \frac{\#\text{ required relations}}{\text{smoothness probability}} + c_{\text{lin. algebra}} c_{\text{density}} \frac{\#\text{ required relations}}{\mathcal{A} \cdot \text{filtering ratio}} \\ &= \frac{2B}{\mathcal{A}\log B} \rho_D\left(\frac{\text{avrg } \log_2 N_f}{\log_2 B}\right)^{-1} \rho_D\left(\frac{\text{avrg } \log_2 N_g}{\log_2 B}\right)^{-1} + \frac{1}{4} 2^7 \left(\frac{2B}{\mathcal{A}\log B \log_2 B}\right)^2\end{aligned}$$

where ρ_D is Dickman's ρ function. Inside the relation collection model of cost, [BD19] assumes that the duplicates due to the roots of unity can be avoided for free in the relation collection, hence a number of obtained relations in a sieving space of dimension $2\deg h$ and samples whose coefficients are bounded by A of

$$\frac{(2A+1)^{2\deg h}}{2w} \rho_D\left(\frac{\text{avrg } \log_2 N_f}{\log_2 B}\right) \rho_D\left(\frac{\text{avrg } \log_2 N_g}{\log_2 B}\right) \geq \frac{2B}{\log B}\,.$$

Guillevic and Singh propose a slightly different model of cost in [GS21] where they evaluate the smoothness bias of the norms in TNFS. For that they need practical curve parameters and polynomials. Their experiments are given as a Sagemath implementation. One requires

$$\begin{aligned}&\frac{(2A+1)^{2\deg h}}{2w} \frac{1}{\zeta_{K_h}(2)} \text{avrg} \rho_D\left(\frac{\log N_f + \alpha(f,h)}{\log B}\right) \rho_D\left(\frac{\log N_g + \alpha(g,h)}{\log B}\right) \\ &\geq 2\text{LogIntegral}(B)\end{aligned}$$

and

$$\text{cost} = \frac{(2A+1)^{2\deg h}}{2\mathcal{A}} c_{\text{sieve}} + c_{\text{linalg}} c_{\text{density}} \left(\frac{2\text{LogIntegral}(B)}{c_{\text{filter}}}\right)^2$$

We summarize in Table 2.6 the different choices. In [GS21], the cost of linear algebra is modeled as the number of 64-bit machine word limbs of ℓ: $\lceil \log_2 \ell/64 \rceil$ times the weight per row, times the matrix size squared.

We incorporated the assumptions of [BD19] in the Sagemath simulator of [GS21] so that we can give the two estimates with the same simulation tool.

3 Pairings at 192-bit security

3.1 Selection of pairing-friendly curves

Our selection of pairing-friendly curves for 192-bit security follows the same strategy as the case of 128-bit security (e.g., see the recent study in [Gui20]). More concretely, the choice of elliptic curves that will be included in our analysis is based on the following criteria:

C1. Use elliptic curves that admit high degree twists $d \in \{3, 4, 6\}$.
C2. Focus on pairing-friendly elliptic curves with embedding degree $15 \leq k \leq 28$.
C3. The selected curves should be TNFS-resistant.

C4. Consider pairing-friendly curves with $\rho = \log p / \log r$ up to 2.
C5. Apply and implement the optimal ate pairing.
C6. Identify efficient candidate curves, based on additional pairing-related operations, beyond the pairing computation itself.

We choose to implement the optimal ate pairing for our selected curves, since the state-of-the-art for 128-bit security suggests that this is the most efficient pairing type. The curves that we consider admit high degree twists i.e., they have j-invariant 1728 ($4 \mid k$) or j-invariant 0 (3 or $6 \mid k$). The use of high-degree twists is crucial for ensuring fast optimal ate pairing instances [CLN10]. We note that extending the fastest curve options in 128-bit security (e.g., BLS12 and BN curves) to the 192-bit setting is not the optimal choice, since this would require the size of the base field prime p to be larger than 1000 bits [Gui21]. The security of the selected curves should be evaluated taking into consideration the progress on the TNFS attacks targeting the DLP in extension fields of composite degree [KB16]. In particular, the security of our selected curves is evaluated using the STNFS software simulator[3] of Guillevic and Singh [GS21].

Furthermore, although the initial requirement for pairing-friendly curves was to look for $\rho \approx 1$ [FST10], sometimes curves with larger ρ can be beneficial. In particular, given a pairing-friendly family of curves with fixed and composite embedding degree k, increasing the size of the base field prime p to counter the TNFS attacks results in the increase of the size of the prime r defining the prime order subgroup of $E(\mathbb{F}_p)$ and hence an increase of the length of the Miller loop. Alternatively, one can choose a pairing-friendly family of curves for the same k with larger ρ, so that the size of p is increased without affecting the size of r. Therefore, in our study we include curves with ρ-value up to 2. Such curve instances at 192-bit security with $\rho = 2$ are studied in [FK19]. In addition, we point out that our results on optimal pairing-friendly curves for 192-bit security depend not only on the fast pairing implementation, but also on efficient exponentiation in the three pairing-groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, hashing to $\mathbb{G}_1, \mathbb{G}_2$, cofactor clearing and subgroup membership testing in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$. Hence, in our quest for the most efficient pairing candidates at 192-bit security, we include an analysis on the efficiency of these additional pairing functionalities.

Based on these criteria we identified the most promising candidate families of pairing-friendly elliptic curves in the literature which can result in efficient pairing-related operations at 192-bit security. In particular, we have concluded to the following families:

- Kachisa–Schaefer–Scott (KSS) [KSS08]. Families KSS16 and KSS18 for embedding degrees 16 and 18 respectively.
- Freeman–Scott–Teske (FST) [FST10]. We follow Construction 6.4 in [FST10] for families with $k = 20$ and $k = 28$ and we refer to these families as FST20 and FST28.
- Barreto–Lynn–Scott [BLS03]. Families BLS12, BLS15, BLS21, BLS24, BLS27, for embedding degrees 12, 15, 21, 24 and 27. Family BLS12 does not comply with the embedding degree selection criteria (C2). However, we include this family to justify our earlier claim that although BLS12 is the most efficient choice for 128-bit security, increasing the size of p to reach 192-bit security is not the optimal strategy to follow. In addition, we include BLS15 to compare to FM15.
- Fotiadis–Martindale [FM19]. Families FM15, FM16, FM18 and FM20, for embedding degrees 15, 16, 18 and 20. These correspond to the families with number #21, #23, #25 and #27 in [FM19, Table 2]. We note that family FM20 coincides with FST20.
- Scott–Guillevic [SG18]. Families SG18 and SG20 for embedding degrees 18 and 20. These types of families are also known as *Aurifeuillean* and they are constructed following the method of Scott and Guillevic [SG18].
- Gasnier–Guillevic [GG23]. Families GG20b and GG28 for $k = 20$ and $k = 28$. As mentioned in [GG23], although they don't improve the ρ-value compared to FST10

[3]Simulation tool in SageMath under MIT license: https://gitlab.inria.fr/tnfs-alpha/alpha

Table 5: Curves of embedding degree 15 to 28 with a high degree twist $d \in \{3, 4, 6\}$ and such that the curve $E(\mathbb{F}_p)$ offers at least 192 bits of security.

k	curve	seed	(a, b)	$p \bmod k$	$\log p$	$\log r$	ρ	$\log p^k$	secu/ref
12	BLS12	$-(2^{192} - 2^{188} + 2^{115} + 2^{110} + 2^{44} + 1)$	$(0, 1)$	7	1150	768	1.50	13800	193 [Gui20]
15	BLS15	$2^{74} + 2^{73} + 2^{62} + 2^{59} + 2^6$	$(0, -2)$	1	894	597	1.50	13410	190
	FM15	$2^{47} + 2^{46} + 2^{34} + 2^{14} + 2^{13} + 2^{11}$	$(0, 1)$	1	762	381	2.00	11430	191
16	KSS16	$2^{78} - 2^{76} - 2^{28} + 2^{14} + 2^7 + 1$	$(1, 0)$	13	766	605	1.25	12255	194 [GS21]
	FM16	$2^{48} - 2^{44} - 2^{38} + 2^{31}$	$(1, 0)$	1	765	384	2.00	12232	196
	AFG16	$2^{48} - 2^{28} - 2^{25} + 2^2$	$(1, 0)$	1	766	384	2.00	12256	196
18	KSS18	$2^{80} + 2^{77} + 2^{76} - 2^{61} - 2^{53} - 2^{14}$	$(0, 6)$	7	638	474	1.33	11477	193 [GS21]
	SG18	$-(2^{63} + 2^{54} + 2^{16})$	$(0, 15)$	7	638	383	1.66	11484	**187**
	SG18	$2^{66} + 2^{60} + 2^{43} + 2^3$	$(0, 17)$	1	669	401	1.66	12027	**191**
	SG18	$2^{70} - 2^{68} - 2^{54} + 2^{28}$	$(0, 23)$	1	704	423	1.66	12668	194
	FM18	$-2^{64} + 2^{33} + 2^{30} + 2^{20} + 1$	$(0, 5)$	7	768	384	2.00	13824	197
20	FM20/FST20	$-2^{48} + 2^{22} - 2^{15} - 1$	$(29, 0)$	1	574	384	1.50	11480	**182**
	FM20/FST20	$-2^{53} - 2^{49} + 2^{10} + 1$	$(1, 0)$	1	636	425	1.50	12701	**189**
	FM20/FST20	$-2^{56} + 2^{44} + 1$	$(1, 0)$	1	670	448	1.50	13400	193
	SG20	$-2^{47} - 2^{45} + 2^{15} + 2^{13}$	$(1, 0)$	1	670	383	1.75	13391	203
	GG20b	$2^{49} + 2^{46} - 2^{41} + 2^{35} + 2^{30} - 1$	$(2, 0)$	1	575	379	1.52	11499	196
24	BLS24	$-2^{51} - 2^{28} + 2^{11} - 1$	$(0, 1)$	19	509	409	1.25	12202	193 [GS21]
	BLS24	$-2^{51} + 2^{49} + 2^{44} + 2^{35} + 1$	$(0, 1)$	1	505	405	1.25	12099	193
	BLS24	$2^{51} + 2^{44} + 2^{41} - 2^{37} + 1$	$(0, 1)$	1	509	409	1.25	12205	193
21	BLS21	$-2^{32} + 2^{25} + 2^6 + 2$	$(0, 16)$	1	511	384	1.33	10715	199
27	BLS27	$-2^{21} - 2^{19} - 2^{15} + 2^{10} + 2^4 + 2^2 + 1$	$(0, 1)$	19	426	383	1.11	11481	212
	BLS27	$2^{21} + 2^{19} + 2^{17} - 2^{11} + 2^5 - 2^3$	$(0, -2)$	7	427	384	1.11	11509	212
28	FST28	$2^{32} - 2^{25} + 2^{22} + 2^{15} + 1$	$(1, 0)$	1	510	384	1.33	14276	208
	GG28	$-2^{32} - 2^{28} + 2^{19} + 2^9 - 2^3 - 1$	$(9, 0)$	1	500	381	1.31	13983	223

and FST28, they are more resistant to the TNFS attack.

FM15, FM16 and FM18 families have $\rho = 2$. In Section 3.4 we compare these families with others which have smaller ρ for the same embedding degree. Furthermore, for completeness we also considered the case $k = 22$ and specifically the families of Freeman–Scott–Teske obtained by Constructions 6.3 and 6.4 (FST22 (6.3), FST22 (6.6)) and the family of Gasnier–Guillevic (GG22). Such examples of pairings do not follow the selection criterion (C1), since such elliptic curves have only quadratic twists, and they don't seem to be competitive to pairings on elliptic curves that admit high degree twists. However, because we already have SageMath implementations of these curves and in order to justify that indeed they are not competitive, we include these examples in Appendix A (see Table 18). As we were finalizing our paper, Lin, Zhao and Zheng posted [LZZ24] on ePrint, on implementing $k = 22$ $D = 7$ GG curves, investigating the strategy of super-optimal pairings [FAGA23]. In addition to the aforementioned families of pairing-friendly elliptic curves, we also include the following family for embedding degree 16 and $\rho = 2$ in our analysis.

Family 1 (Aranha-Fotiadis-Guillevic (AFG16))**.** *The following polynomials* $p(x), t(x), r(x) \in \mathbb{Q}[x]$*, where:*

$$\begin{aligned} p(x) &= (x^{16} + 2x^{13} + x^{10} + 5x^8 + 6x^5 + x^2 + 4)/4 \\ t(x) &= r(x) + x^5 + 1 = x^8 + x^5 + 2 \\ r(x) &= \Phi_{16}(x) = x^8 + 1 \end{aligned}$$

define a polynomial family of pairing-friendly elliptic curves with embedding degree $k = 16$*, CM discriminant* $D = 1$ *and* $\rho = 2$*. The order of the curve in parametric form is* $\#E(\mathbb{F}_p) = h(x)r(x)$*, where* $h(x) = (x(x^3 + 1)/2)^2$*. The family is integer-valued for all* $u \in \mathbb{Z}$*. The polynomial* r *evaluated at integers of the form* $2u$ *is odd, and is even at integers of the form* $2u + 1$*, for* $u \in \mathbb{Z}$*.* □

The above family of elliptic curves is similar to the one proposed in [FK19, Table 5] for $k = 16$. The difference is that the cofactor $h(x)$ is a perfect square and $\sqrt{h(x)}$ divides

Table 6: Optimal ate Miller loop formulas. The Miller functions $f_{u,Q}$ and lines $\ell_{Q,R}$ are evaluated at the point $P \in \mathbb{G}_1$.

k	curve	Equation (1)	Optimal ate formula
12	BLS12	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
15	BLS15	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
	FM15	$u - p^4 \equiv 0 \bmod r$	$f_{u,Q}(P)$
16	KSS16	$2 + up^3 + p^4 \equiv 0 \bmod r$	$\left[f_{u,Q}(P) \cdot \ell_{[u]Q,\pi(Q)}(P)\right]^{p^3} \cdot \ell_{Q,Q}(P)$
	FM16	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
	AFG16	$u + p^5 \equiv 0 \bmod r$	$f_{u,Q}(P)$
18	KSS18	$1 + up^2 + 2p^3 \equiv 0 \bmod r$	$f_{u,Q}(P) \cdot f_{2,Q}(P)^p \cdot \ell_{[u]Q,\pi([2]Q)}(P)$
	SG18	$u + p^2 + up^3 \equiv 0 \bmod r$	$[f_{u,Q}(P)]^{1+p^3} \cdot \ell_{[u]Q,\pi_2(Q)}(P)$
	FM18	$1 + up^2 \equiv 0 \bmod r$	$f_{u,Q}(P)$
20	FM20/FST20	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
	SG20	$2u + p^2 + p^7 \equiv 0 \bmod r$	$f_{2u,Q}(P) \cdot \ell_{[2u]Q,\pi_2(Q)}(P)$
	GG20b	$u - p - 2p^6 \equiv 0 \bmod r$	$f_{u,Q}(P) \cdot f_{2,\pi_6(-Q)}(P) \cdot \ell_{[u]Q,-\pi(Q)}(P)$
21	BLS21	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
24	BLS24	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
27	BLS27	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
28	FST28	$u - p \equiv 0 \bmod r$	$f_{u,Q}(P)$
	GG28	$u - p - 2p^8 \equiv 0 \bmod r$	$f_{u,Q}(P) f_{2,-\pi_8(Q)}(P) \ell_{-2\pi_8(Q),-\pi(Q)}(P)$

$p(x) - 1$, hence the trick of Wahby and Boneh [WB19] for fast cofactor clearing applies. That is, in order to fix a point P on the curve to have order r, one needs to multiply with the scalar $u(u^3 + 1)$ instead of $h(u)$, which can potentially be more efficient compared to other candidates of the same embedding degree. We refer to the new family as AFG16.

The seeds that we have chosen for instantiating these families are given in Table 5. We searched for seed with minimal Hamming weight, which would give curves with either $a = 0$ or $a = 1$ and $p \equiv 1 \pmod 3$ for fast hashing, and a nice towering of extensions with only small non-residues. For some families multiple seeds are reported, however we choose the ones highlighted in grey for our theoretical comparison, as well as for our benchmarking in Section 5. The resulting curves offer security level of at least 192-bit with respect to the TNFS attacks, except for the ones highlighted in red which offer slightly less. The seed $2^{48} + 2^{28} + 2^{26}$ for FM16 [FM19] does not give machine-word aligned parameters: p is 767-bit long but r is 385-bit long, this overflows the 384-bit standard size. We replace it with $2^{48} - 2^{44} - 2^{38} + 2^{31}$ which has heavier Hamming weight but whose r fits in 384 bits. The seed $-(2^{64} + 2^{35} - 2^{11} + 1)$ for FM18 does not give machine-word aligned parameters, we replace it with $-2^{64} + 2^{33} + 2^{30} + 2^{20} + 1$. The seed for KSS16 from [GS21] (see Table 3) does not fulfill all criteria for fast hashing (see 4.2) and we replace it with a new one. For BLS24, we consider the seed given in [GS21] which dates back to [CLN11] and we give two new ones well-suited for SNARK, with a high 2-valuation of $p - 1$ and $r - 1$.

3.2 Pairing Computation: Miller Loop

As stated earlier, for all curves given in Table 5 we apply the optimal ate pairing, introduced by Vercauteren [Ver10], following our selection criterion (C5). In Table 6 we give the form of Equation (1), as well as the corresponding optimal ate pairing formulas, for all families of pairing-friendly elliptic curves considered in Table 5. These formulas are simplified using the properties described in Section 2. We note that for some curves, there exist alternative choices for the short vectors, which however do not offer any advantage in terms of efficiency compared to the ones that we have chosen.

Remark 1 (Concerning BLS curves with odd embedding degrees $k = 3 \bmod 6$, examples

Table 7: Miller loop cost in Weierstrass model from [CSB05, CLN10].

k	curve	`Dbl step, tangent line` `Add step, chord line`	`sparse-sparse-`$\mathbf{m}_k$ `full-sparse-`$\mathbf{m}_k$	reference
$2 \mid k$	$y^2 = x^3 + ax + b$ quadratic twist	$5\mathbf{m}_{k/2} + 6\mathbf{s}_{k/2} + k\mathbf{m}$ $10\mathbf{m}_{k/2} + 3\mathbf{s}_{k/2}$	$\mathbf{m}_k$ $\mathbf{m}_k$	[CSB05]
$2 \mid k$	$y^2 = x^3 - 3x + b$ quadratic twist	$6\mathbf{m}_{k/2} + 4\mathbf{s}_{k/2} + k\mathbf{m}$ $10\mathbf{m}_{k/2} + 3\mathbf{s}_{k/2}$	$\mathbf{m}_k$ $\mathbf{m}_k$	[CSB05]
$2 \mid k$	$y^2 = x^3 + b$ quadratic twist	$2\mathbf{m}_{k/2} + 7\mathbf{s}_{k/2} + k\mathbf{m}$ $10\mathbf{m}_{k/2} + 2\mathbf{s}_{k/2} + k\mathbf{m}$	$\mathbf{m}_k$ $\mathbf{m}_k$	[CLN10, §5,Tab.3]
$2 \mid k$	$y^2 = x^3 + ax$ quadratic twist	$2\mathbf{m}_{k/2} + 8\mathbf{s}_{k/2} + k\mathbf{m}$ $9\mathbf{m}_{k/2} + 5\mathbf{s}_{k/2} + k\mathbf{m}$	$\mathbf{m}_k$ $\mathbf{m}_k$	[CLN10, §4,Tab.3]
$3 \mid k$	$y^2 = x^3 + b$ cubic twist	$6\mathbf{m}_{k/3} + 7\mathbf{s}_{k/3} + k\mathbf{m} + \mathbf{m}_b$ $13\mathbf{m}_{k/3} + 5\mathbf{s}_{k/3} + k\mathbf{m}$	$\mathbf{m}_k$ $\mathbf{m}_k$	[CLN10, §6]
$6 \mid k$	$y^2 = x^3 + b$ sextic twist	$2\mathbf{m}_{k/6} + 7\mathbf{s}_{k/6} + (k/3)\mathbf{m}$ $10\mathbf{m}_{k/6} + 2\mathbf{s}_{k/6} + (k/3)\mathbf{m}$	$6\mathbf{m}_{k/6}$ $13\mathbf{m}_{k/6}$	[CLN10, §5]
$6 \mid k$	$y^2 = x^3 + b$ sextic twist	$3\mathbf{m}_{k/6} + 6\mathbf{s}_{k/6} + (k/3)\mathbf{m}$ $11\mathbf{m}_{k/6} + 2\mathbf{s}_{k/6} + (k/3)\mathbf{m}$	$6\mathbf{m}_{k/6}$ $13\mathbf{m}_{k/6}$	[AKL+11, §4,6]
$4 \mid k$	$y^2 = x^3 + ax$ quartic twist	$2\mathbf{m}_{k/4} + 8\mathbf{s}_{k/4} + (k/2)\mathbf{m}$ $9\mathbf{m}_{k/4} + 5\mathbf{s}_{k/4} + (k/2)\mathbf{m}$	$6\mathbf{m}_{k/4}$ $8\mathbf{m}_{k/4}$	[CLN10, §4]

are BLS15, BLS21, and BLS27 in Table 6). For these curves, the verticals are not in a subfield and there is no easy *denominator elimination* as for even embedding degrees. Nevertheless the optimal ate Miller loop simplifies to $f_{u,Q}(P)$. One starts expanding the Miller loop of length $u-p$: $f_{u-p,Q}(P) = f_{u,Q}(P)f_{-p,Q}(P)\ell_{uQ,-[p]Q}(P)$. Now observe that $f_{-p,Q}(P) = f_{p,Q}^{-1}(P)/v_{[p]Q}(P)$ with a vertical at $[p]Q$ evaluated at P. Also the line through $[u]Q$ and $-[p]Q$ is a vertical since $[u-p]Q = \mathcal{O}$. Finally $1/v_{[p]Q}(P)$ and $\ell_{[u]Q,-[p]Q}(P)$ cancel themselves, and $f_{p,Q}(P)$ is a pairing and can be taken out. The final formula is indeed $f_{u,Q}(P)$ like for BLS curves of even embedding degrees.

In Table 7 we report the cost for the different elliptic curve operations that are relevant to the pairing computation and especially the Miller loop. Specifically, we include the cost for computing the point addition and point doubling (column 3), as well as the cost of the line computation (column 3) and result accumulation (column 4) that occur in the Miller loop. The elliptic curve model that we have considered is the Weierstrass model, where points are represented in the projective coordinate system, following the works of Chatterjee, Sarkar, Barua [CSB05] and Costello, Lange, Naehrig [CLN10]. In addition, these costs refer to elliptic curves admitting degree 2, 3, 4 and 6 twists. In Table 10 we denote by $\mathbf{m}_i$ and $\mathbf{s}_i$ the multiplication and squaring, respectively, in the finite field $\mathbb{F}_{p^i}$, for $i > 1$ and by $\mathbf{m}$ we denote multiplication over the base field $\mathbb{F}_p$.

In column 4 of Table 7 we report the cost in terms of $\mathbb{F}_p$ multiplications for computing a product of two elements in $\mathbb{F}_{q^d}$, where $q = p^{k/d}$ and $d \in \{2, 3, 4, 6\}$ is the degree of the twist. More concretely, we write $a, b \in \mathbb{F}_{q^d}$, such that:

$$a = \sum_{i=0}^{d-1} a_i w^i \quad \text{and} \quad b = \sum_{i=0}^{d-1} b_i w^i,$$

where $a_i, b_i \in \mathbb{F}_q$ and w is the generator of the degree d extension $\mathbb{F}_{q^d}$ of $\mathbb{F}_q$. The term `sparse-sparse-`$\mathbf{m}_k$ refers to the case where in the representations of a and b, some of the coefficients a_i and b_i are zero (sparse representation), while on the other hand, `full-sparse-`$\mathbf{m}_k$ is the case where one of the two representations is sparse. Such types of $\mathbb{F}_{q^d}$-multiplications appear when updating the Miller function after the doubling and addition steps in the Miller loop when sextic or quartic twists are applied. Note that when quadratic or cubic twists are employed, the `sparse-sparse-`$\mathbf{m}_k$ and `full-sparse-`$\mathbf{m}_k$ multiplications are equivalent to an $\mathbf{m}_k$ multiplication, as reported in Table 7.

Table 8: Exponents for the final exponentiation, easy and hard parts.

k	final exp. easy part	final exp. hard part
12	$(p^6-1)(p^2+1)$	$(p^4-p^2+1)/r$
15	$(p^5-1)(p^2+p+1)$	$(p^8-p^7+p^5-p^4+p^3-p+1)/r$
16	p^8-1	$(p^8+1)/r$
18	$(p^9-1)(p^3+1)$	$(p^6-p^3+1)/r$
20	$(p^{10}-1)(p^2+1)$	$(p^8-p^6+p^4-p^2+1)/r$
21	$(p^7-1)(p^2+p+1)$	$(p^{12}-p^{11}+p^9-p^8+p^6-p^4+p^3-p+1)/r$
24	$(p^{12}-1)(p^4+1)$	$(p^8-p^4+1)/r$
27	p^9-1	$(p^{18}+p^9+1)/r$
28	$(p^{14}-1)(p^2+1)$	$(p^{12}-p^{10}+p^8-p^6+p^4-p^2+1)/r$

3.3 Pairing Computation: Final Exponentiation

Table 8 shows the formulas for the easy and hard parts of the final exponentiation for each embedding degree we consider in this paper. The computation of the easy part is straightforward, while for the hard part we follow the state of the art approaches to obtain an optimized formula for the final exponentiation in the examples that we present here. That is, according to [HHT20], there are three distinct techniques for the efficient computation of the hard part: 1. a base-p expansion to exploit cheap Frobenius powers, 2. a lattice-reduction technique on the exponent in base p to reduce further the size of the coefficients of the base-p expansion, and 3. a cyclotomic decomposition of the exponent. The latter technique is better suited when the trace of Frobenius of the curve is represented as a polynomial of degree one, which is the case for all BLS curves.

In the following we give the formulas for the optimized hard part of the final exponentiation for all families of Table 5. In each case we present the theoretical cost in terms of the number of required operations. Section 3.3.1 decomposes the hard part exponent $\Phi_k(p)/r$ with the techniques in [HHT20] while Section 3.3.2 considers lattice reduction [FKR12] before a fine tuning, resulting in a final exponentiation by a multiple of the hard part exponent $\Phi_k(p)/r$.

3.3.1 With Cyclotomic Decomposition

BLS12. The optimal formula [HHT20, §5 p.14] is:

$$3\frac{\Phi_{12}(p)}{r} = (u-1)^2(p+u)(p^2+u^2-1)+3$$

with cost: $2\exp(|u-1|)+3\exp(|u|)+5\mathbf{m}_{12}+1\mathbf{s}_{12}+2\mathbf{f}_{12}+1\mathbf{f}_6$.

BLS15. The formula in [HHT20, §5 p.14] is:

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

where $\lambda_7=1$ and the rest of the λ_i are defined as follows:

$$\lambda_6 = u\lambda_7-1, \lambda_5 = u\lambda_6, \lambda_4 = u\lambda_5+1, \lambda_3 = u\lambda_4-1, \lambda_2 = u\lambda_3+1, \lambda_1 = u\lambda_2, \lambda_0 = u\lambda_1-1.$$

The cost is: $1\exp(|u-1|)+1\exp(|u^3-1|)+7\exp(|u|)+14\mathbf{m}_{15}+1\mathbf{s}_{15}+7\mathbf{f}_{15}+1\mathbf{inv\text{-}cyclo}_{15}$, where $\mathbf{inv\text{-}cyclo}_{15}$ corresponds to cyclotomic inversion in $\mathbb{F}_{p^{15}}$ and is equivalent to $f^{-1} = f^{p^{10}}f^{p^5}$ with cost $\mathbf{inv\text{-}cyclo}_{15} = \mathbf{f}_{10}+\mathbf{f}_5+\mathbf{m}_{15}$.

FM16. We obtain the formula:

$$\frac{\Phi_{16}(p)}{r} = \left[\frac{u^2}{4}(u^6+1)+1\right](p+u)(p^2+u^2)(p^4+u^4)+1$$

with cost: $13\exp(|u|) + 2\exp(|u|/2) + 6\mathbf{m}_{16} + 3\mathbf{f}_{16}(+\mathbf{cj}_{16})$. We note that the seed u for this family is always even, in order to produce primes p and r [FM19]. In addition, our implementation works for negative seeds as well, in which case one conjugation is needed in the operation count ($\mathbf{cj}_{16}$).

AFG16. We compute this formula for Family 1 for $k = 16$:

$$\frac{\Phi_{16}(p)}{r} = \left[\left(\frac{u^2}{4}(u^3+1)^2+1\right)\left(p^5+u(u^3p(1+up^3)-1)\right)+up(1+up^3)\right]+1$$

with cost: $2\exp(u/2) + 13\exp(u) + 9\mathbf{m}_{16} + 4\mathbf{f}_{16}$.

SG18. For the hard part of SG18 we apply the formula of Scott–Guillevic [SG18], namely:

$$\frac{\Phi_{18}(p)}{r} = (3u^2-1)^2\left[(p^2+3pu^2+9u^4+3u)(1-3up+p^3)-3p^2\right]+(1-3up+p^3)$$

with total cost: $9\exp(|u|)+14\mathbf{m}_{18}+5\mathbf{s}_{18}+5\mathbf{f}_{18}+3\mathbf{cj}_{18}(+\mathbf{cj}_{18})$. The additional conjugation is added to include the case of negative seeds.

FM18. Our formula for the hard part of the final exponentiation is the following:

$$\begin{aligned}\frac{\Phi_{18}(p)}{r} = (p-u)\Bigg[&\left(u^6+\frac{(u-1)^2}{3}+1\right)\left(p^4+p^3u-p-u^3(p^3u-p-u)\right)\\ &+u(p^3u-p-u)\Bigg]+1\end{aligned}$$

with cost: $7\exp(|u|)+\exp(|u-1|)+\exp(|u-1|/3)+2\exp(|u+1|)+13\mathbf{m}_{18}+2\mathbf{s}_{18}+4\mathbf{f}_{18}+4\mathbf{cj}_{18}$. We note also that $(u-1)^2 \equiv 0 \bmod 3$. Alternatively, one can multiply by 3 the exponent to avoid a possibly non-sparse $(u-1)/3$:

$$\begin{aligned}\frac{3\Phi_{18}(p)}{r} = (p-u)\Bigg[&\left(2(2(u^6+1)-u)+u^2-u^6\right)\left(p^4+p^3u-p-u^3(p^3u-p-u)\right)\\ &+3u(p^3u-p-u)\Bigg]+3\end{aligned}$$

with cost: $11\exp(|u|) + 12\mathbf{m}_{18} + 3\mathbf{s}_{18} + 4\mathbf{f}_{18} + 6\mathbf{cj}_{18}$.

FM20/FST20. The hard part of the final exponentiation for FM20 [FM19, Family #27, Table 2] is equivalent to FST6.4 [FST10, Construction 6.4] for $k = 20$. The formula we considered is:

$$\frac{\Phi_{20}(p)}{r} = \frac{(u-1)^2}{4}(u^2+1)(p+u)\left[(p^2+u^2-1)(p^4+u^4+1)-u^2p^2\right]+1,$$

with total cost: $2\exp(|u-1|/2) + 9\exp(|u|) + 8\mathbf{m}_{20} + 1\mathbf{f}_{20} + 2\mathbf{f}_2 + 1\mathbf{f}_4 + 2\mathbf{cj}_{20}(+1\mathbf{cj}_{20})$, where the one additional conjugation is required when the seed is negative.

BLS21. For this type of curves, we apply [HHT20, Theorem 1] to obtain the following formula for the hard part of the final exponentiation:

$$3\frac{\Phi_{21}(p)}{r} = (u-1)(u^3-1)\left(\sum_{i=0}^{11}\lambda_i p^i\right) + 3,$$

where $\lambda_{11} = 1$, $\lambda_i = u\lambda_{i+1} + c_i$ and c_i is defined as:

$$c_i = \begin{cases} 1, & \text{for } i = 2,5,8 \\ 0, & \text{for } i = 1,4,6,9 \\ -1, & \text{for } i = 0,3,7,10 \end{cases}$$

with cost: $15\exp(|u|) + 15\mathbf{m}_{21} + \mathbf{s}_{21} + 11\mathbf{f}_{21} + 2\mathbf{inv\text{-}cyclo}_{21}$.

BLS24. The hard part of the final exponentiation is written as [HHT20, §5 p.16]:

$$3\frac{\Phi_{24}(p)}{r} = (u-1)^2(u+p)(u^2+p^2)(u^4+p^4-1) + 3$$

with cost: $2\exp(|u-1|) + 7\exp(|u|) + 6\mathbf{m}_{24} + \mathbf{s}_{24} + 3\mathbf{f}_{24} + \mathbf{cj}_{24}$.

BLS27. We apply the formula of Hayashida et al. [HHT20, §5 p.16] for computing the hard part of the final exponentiation for BLS27. We have:

$$\frac{\Phi_{27}(p)}{r} = (u-1)^2(p^2+pu+u^2)(p^6+p^3u^3+u^6)(p^9+u^9+1) + 3$$

The total cost is: $2\exp(|u-1|) + 17\exp(|u|) + 8\mathbf{m}_{27} + \mathbf{s}_{27} + \mathbf{f}_{27} + \mathbf{f}_{27}^2 + \mathbf{f}_{27}^3 + \mathbf{f}_{27}^6 + \mathbf{f}_{27}^9$ $(+3(\mathbf{m}+2\mathbf{f})$ if $u < 0)$.

FST28. Our optimal formula for the hard part of the final exponentiation is:

$$\frac{\Phi_{28}(p)}{r} = \frac{(u-1)^2}{4}(u^2+1)(p+u)e + 1,$$

where the value e is determined as follows:

$$e = \Big(u^2(u^2+p^2)+p^4\Big)\Big(u^2\Big(u^2(u^2-1)+1\Big)+p^6-1\Big) + \Big(1-p^6\Big)\Big((u^2-1)+p^2\Big).$$

The total cost is: $2\exp(|u-1|/2) + 13\exp(|u|) + 12\mathbf{m}_{28} + 1\mathbf{f}_{28} + 2\mathbf{f}_2 + 1\mathbf{f}_4 + 2\mathbf{f}_6 + 2\mathbf{cj}_{28}(+1\mathbf{cj}_{28})$, where the extra conjugation appears in the case where the seed is negative.

3.3.2 With Lattice Reduction

FM15. We use the following formula for the hard part in the case of FM15:

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

where the λ_i are defined by the following relations:

$$\begin{aligned}
\lambda_6 &= -3 - 3u^2 + 2u^4 + u^5 + 2u^6 - 3u^8, & \lambda_0 &= -(u^6 - u^4 + u^3)\lambda_6 + 3u^2 + 3u \\
\lambda_1 &= u^7\lambda_6 - (3u^3 + 3u^3 + 3u), & \lambda_2 &= u^3\lambda_6 \\
\lambda_3 &= -(u^7 + u^2 - 1)\lambda_6 + 3u^3 + 3u^2 + 3u, & \lambda_4 &= (u^7 - u^5 - 1)\lambda_6 - (3u^3 + 3u^2 - 3) \\
\lambda_5 &= u^4\lambda_6 - 3, & \lambda_7 &= -(u^4 - u^3 + u)\lambda_6 + 3
\end{aligned}$$

Then the total cost is: $15\exp(|u|) + 37\mathbf{m}_{15} + 6\mathbf{s}_{15} + 11\mathbf{f}_{15} + 2\mathbf{f}_5 + 2\mathbf{inv\text{-}cyclo}_{15}$ when $u > 0$, while one less multiplication in $\mathbb{F}_{p^{15}}$ is required for negative seeds.

KSS18. For the computation of the hard part of the final exponentiation, the most efficient method is presented in [CHZ22]. We write

$$\frac{3u^2}{49} \cdot \frac{\Phi_{18}(p)}{r} = \sum_{i=0}^{5} \lambda_i p^i$$

where the λ_i are defined in terms of the cofactor $c = 3h(u)/49 = u^2 + 5u + 7$:

$$\begin{aligned} &\lambda_5 = u^2c + 3, && \lambda_4 = -3u\lambda_5 - 49c, && \lambda_3 = 2u^2\lambda_5 + 35uc, \\ &\lambda_1 = 2\lambda_4 + u\lambda_5, && \lambda_0 = 2\lambda_3 + u\lambda_4, && \lambda_2 = -u\lambda_0 + 2\lambda_5 \end{aligned}$$

The cost for the hard part in this case is: $7\exp(u) + 24\mathbf{m}_{18} + 11\mathbf{s}_{18} + 7\mathbf{f}_9 + 5\mathbf{f}_{18}$.

SG20. For the hard part of the final exponentiation for the SG20 family we apply the lattice reduction to obtain the formula $\Phi_{20}(p)/r = \sum_{i=0}^{7} \lambda_i p^i$. The λ_i in this representation are defined as $\lambda_6 = uh + 1$, $\lambda_7 = uy$, with $y = 2u\lambda_6 - 1$, $\lambda_0 = -uy'$, with $y' = 2u\lambda_7$, $\lambda_1 = -uy''$, with $y'' = -2u\lambda_0$ and for the rest of the λ_i we have:

$$\lambda_4 = \lambda_1 - \lambda_6 - y, \lambda_3 = -\lambda_1 - \lambda_6 + y'', \lambda_2 = -\lambda_3 + y' + \lambda_7 - h, \lambda_5 = -\lambda_3 + \lambda_0 - h$$

The total cost is: $13\exp(|u|) + 19\mathbf{m}_{20} + 6\mathbf{s}_{20} + 7\mathbf{f}_{20} + 7\mathbf{cj}_{20}(+\mathbf{cj}_{20})$, where the additional conjugation is to account for the case of negative seeds.

3.3.3 Base-p Expansion

KSS16. For the hard part of the KSS16 family we obtained the following formula:

$$\frac{2u^7 + 48u^3}{125} \cdot \frac{\Phi_{16}(p)}{r} = (u^3c + 56)e_1 + ce_2 - 1540p^7,$$

where $c = 2h(u)/125 = u^2 + 2u + 5$ and the values e_1, e_2 are defined as:

$$\begin{aligned} e_1 &= u^4p^7 + u^3(1 - 2p^4) + u^2p(3 + 4p^4) + up^2(-11 + 2p^4) + p^3(7 + 24p^4) \\ e_2 &= 125p^2(p^4 - 2) + 25up(3p^4 + 4) + 5u^2(-11p^4 + 2) \end{aligned}$$

The total cost for computing the hard part for KSS16 is: $2\exp(u-1) + 7\exp(u) + 29\mathbf{s}_{16} + 30\mathbf{m}_{16} + 10\mathbf{f}_4 + 4\mathbf{f}_{16}$, improving on the previous best count from [Gha16, §4.3 p.107, Eq.(4.9) p.114] (also at [GF16]) of $2\exp(u+1) + 7\exp(u) + 37\mathbf{s}_{16} + 35\mathbf{m}_{16} + 4\mathbf{f}_{16} + 2\mathbf{f}_2 + \mathbf{f}_4$.

GG20a and GG20b. For the hard part of the final exponentiation we use the fact that $p^8 \equiv (p^6 - p^4 + p^2 - 1) \bmod r$ and we set the following values: $c = u^2 - 2u + 5$, $\alpha_\pm = 2 \pm p^5$, $\beta_\pm = 4 \pm 3p^5$, $\gamma_\pm = 7 \pm 24p^5$ and $\delta_\pm = 11 \pm 2p^5$. Then we obtain the following formula for the hard part of the GG20a family, where s is a scaling factor coprime to r:

$$\begin{aligned} s\frac{\Phi_{20}(p)}{r} &= (u^4c + 328)(-41p^2 + up\gamma_- + u^2\delta_- + u^3p^4\beta_- + u^4p^3\alpha_+ + u^5p^7) \\ &\quad + c(625p\alpha_- + 125u\beta_+ + 25u^2p^4\delta_+ + 5u^3p^3\gamma_+ + 38u^4p^7) + 6724p^7 \end{aligned}$$

with a total cost: $2\exp(u-1) + 9\exp(u) + 51\mathbf{s}_{20} + 42\mathbf{m}_{20} + 4\mathbf{f}_5 + 9\mathbf{f}_{20}$.

Using the same logic as for GG20a and changing only $\alpha_\pm = \pm 2 + p^5$ and $\beta_\pm = \pm 4 + 3p^5$, the GG20b final exponent is written as follows, with exactly the same cost as GG20a:

$$\begin{aligned} s'\frac{\Phi_{20}(p)}{r} &= (u^4c - 328)(-41p^2 + up\gamma_+ + u^2\delta_+ + u^3p^4\beta_+ + u^4p^3\alpha_- + u^5p^7) \\ &\quad + c(-625p\alpha_+ + 125u\beta_- + 25u^2p^4\delta_- + 5u^3p^3\gamma_- - 38u^4p^7) + 6724p^7 \ . \end{aligned}$$

Table 9: Relative cost of multiplication $\mathbf{m}_k$, squaring $\mathbf{s}_k$, Frobenius $\mathbf{f}_k$, and inversion $\mathbf{i}_k$ in finite field extensions assuming $p \equiv 1 \bmod k$ for fast f_k.

k	$\mathbf{m}_k$	$\mathbf{s}_k$	$\mathbf{f}_k$	$\mathbf{s}_k^{\text{cyclo}}$	$\mathbf{i}_k - \mathbf{i}_1$	$\mathbf{i}_k$, with $\mathbf{i}_1 = 25\mathbf{m},\ \mathbf{s} = \mathbf{m}$
1	$\mathbf{m}$	$\mathbf{s}$	0	-	0	$25\mathbf{m}$
2	$3\mathbf{m}$	$2\mathbf{m}$	0	$2\mathbf{s}$	$2\mathbf{m} + 2\mathbf{s}$	$29\mathbf{m}$
3	$6\mathbf{m}$	$2\mathbf{m} + 3\mathbf{s}$ [CH07]	$2\mathbf{m}$	-	$9\mathbf{m} + 3\mathbf{s}$	$37\mathbf{m}$
4	$9\mathbf{m}$	$2\mathbf{m}_2 = 6\mathbf{m}$	$2\mathbf{m}$	$2\mathbf{s}_2 = 4\mathbf{m}$	$12\mathbf{m} + 2\mathbf{s}$	$39\mathbf{m}$
5	$13\mathbf{m}$	$13\mathbf{s}$ [Mon05]	$4\mathbf{m}$	-	$48\mathbf{m}$	$73\mathbf{m}$
6	$18\mathbf{m}$	$2\mathbf{m}_2 + 3\mathbf{s}_2 = 12\mathbf{m}$	$4\mathbf{m}$	$6\mathbf{m}$ [GS10]	$34\mathbf{m}$	$59\mathbf{m}$
7	$22\mathbf{m}$	$22\mathbf{s}$	$6\mathbf{m}$	-	$104\mathbf{m}$	$129\mathbf{m}$
8	$27\mathbf{m}$	$2\mathbf{m}_4 = 18\mathbf{m}$	$6\mathbf{m}$	$2\mathbf{s}_4 = 12\mathbf{m}$	$44\mathbf{m}$	$69\mathbf{m}$
9	$36\mathbf{m}$	$2\mathbf{m}_3 + 3\mathbf{s}_3 = 18\mathbf{m} + 9\mathbf{s}$	$8\mathbf{m}$	-	$69\mathbf{m} + 12\mathbf{s}$	$106\mathbf{m}$
10	$39\mathbf{m}$	$2\mathbf{m}_5 = 26\mathbf{m}$	$8\mathbf{m}$	$2\mathbf{s}_5 = 26\mathbf{s}$	$74\mathbf{m} + 26\mathbf{s}$	$125\mathbf{m}$
12	$54\mathbf{m}$	$2\mathbf{m}_6 = 36\mathbf{m}$	$10\mathbf{m}$	$6\mathbf{m}_2 = 18\mathbf{m}$	$97\mathbf{m}$	$119\mathbf{m}$
15	$78\mathbf{m}$	$2\mathbf{m}_5 + 3\mathbf{s}_5 = 26\mathbf{m} + 39\mathbf{s}$	$14\mathbf{m}$	-	$165\mathbf{m} + 39\mathbf{s}$	$229\mathbf{m}$
16	$81\mathbf{m}$	$2\mathbf{m}_8 = 54\mathbf{m}$	$14\mathbf{m}$	$2\mathbf{s}_8 = 36\mathbf{m}$	$134\mathbf{m}$	$159\mathbf{m}$
18	$108\mathbf{m}$	$2\mathbf{m}_9 = 72\mathbf{m}$	$16\mathbf{m}$	$6\mathbf{m}_3 = 36\mathbf{m}$	$232\mathbf{m}$	$257\mathbf{m}$
20	$117\mathbf{m}$	$2\mathbf{m}_{10} = 78\mathbf{m}$	$18\mathbf{m}$	$2\mathbf{s}_{10} = 52\mathbf{m}$	$255\mathbf{m}$	$280\mathbf{m}$
21	$132\mathbf{m}$	$110\mathbf{m}$	$20\mathbf{m}$	-	$393\mathbf{m}$	$418\mathbf{m}$
24	$162\mathbf{m}$	$2\mathbf{m}_{12} = 108\mathbf{m}$	$22\mathbf{m}$	$6\mathbf{m}_4 = 54\mathbf{m}$	$318\mathbf{m}$	$343\mathbf{m}$
27	$216\mathbf{m}$	$153\mathbf{m}$	$26\mathbf{m}$	-	$511\mathbf{m}$	$536\mathbf{m}$
28	$198\mathbf{m}$	$132\mathbf{m}$	$26\mathbf{m}$	$88\mathbf{m}$	$437\mathbf{m}$	$462\mathbf{m}$

GG28. With a strategy similar to GG20, we use the fact that $p^{12} \equiv (p^{10} - p^8 + p^6 - p^4 + p^2 - 1) \bmod r$ and we set the following values: $c = u^2 - 2u + 5$, $\alpha_\pm = -2 \pm p^7$, $\beta_\pm = -4 \pm 3p^7$, $\gamma = 11p^7 + 2$, $\overline{\gamma} = 11 + 2p^7$, $\delta = 24 + 7p^7$, $\overline{\delta} = 24p^7 + 7$, $\epsilon_\pm = -41 \pm 38p^7$, and $\zeta_\pm = -117 \pm 44p^7$. Then we obtain the following formula for the hard part, where s is a scaling factor coprime to r:

$$s\frac{\Phi_{28}(p)}{r} = (u^6c - 232)e_1 + ce_2 + 3364p^{11}$$

where the values e_1, e_2 are defined as follows:

$$e_1 = -29p^4 + up^3\zeta_- + u^2p^2\epsilon_+ + u^3p\overline{\delta} + u^4\overline{\gamma} + u^5p^6\beta_- + u^6p^5\alpha_+ + u^7p^{11}$$
$$e_2 = 5^6p^3\alpha_- + 5^5up^2\beta_+ + 5^4u^2p\gamma + 5^3u^3\delta + 5^2u^4p^6\epsilon_- + 5u^5p^5\zeta_+ + 278u^6p^{11}$$

The total cost is: $2\exp(u-1) + 13\exp(u) + 68\mathbf{s}_{28} + 83\mathbf{m}_{28} + 12\mathbf{f}_7 + 14\mathbf{f}_{28}$.

3.4 Pairing Computation: Theoretical Comparison

We present a theoretical analysis of the total cost for computing the different pairing instances of Table 5. More concretely, the usual practice in estimating the cost of a pairing is to express the Miller loop and the final exponentiation in terms of $\mathbb{F}_p$-multiplications. Then we aim at presenting a theoretical comparison of the different pairings Table 5, based on the total number of $\mathbb{F}_p$-multiplications required. This theoretical comparison is outlined in Table 10 and will serve as the baseline for selecting the most promising pairing candidates to be included in our benchmark experiments in Section 5.

Such a theoretical comparison necessitates that all operations needed in the Miller loop and in the final exponentiation are translated to $\mathbb{F}_p$-multiplications. This conversion is described in Table 9, following the usual estimates, e.g. [GMT20, Table 4] for a recent presentation. We denote $\mathbf{m}_i$, resp. $\mathbf{s}_i$ a multiplication, resp. squaring in $\mathbb{F}_{p^i}$. Furthermore, we denote $\mathbf{f}_i$, $\mathbf{s}_k^{\text{cyclo}}$ and $\mathbf{i}_i$ the Frobenius p-power, the cyclotomic squaring and inversion

in $\mathbb{F}_{p^i}$. When it is clear from the context, $\mathbf{f}_j$ denotes a p^j-th power in $\mathbb{F}_{p^i}$. Based on the theoretical comparison in Table 10, we can extract the most promising curves to be benchmarked. We discuss the results of Table 10 and justify our selections of pairing-friendly curves to be benchmarked.

Remark 2 (Inversions in extension fields). For inversions in $\mathbb{F}_{p^k}$, we use the classical formula:

$$x^{-1} = \frac{x^{p+p^2+\ldots+p^{k-1}}}{x^{1+p+\ldots+p^{k-1}}},$$

where the denominator is the norm $\mathrm{Norm}_{\mathbb{F}_{p^k}/\mathbb{F}_p}(x)$ in $\mathbb{F}_p$. It costs the computation of the numerator with Frobenius powers and multiplications in $\mathbb{F}_{p^k}$, the denominator computation knowing that it is in $\mathbb{F}_p$, one inversion in $\mathbb{F}_p$, and a coefficient-wise multiplication. One can factor the exponent to minimize the multiplications. For $\mathbb{F}_{p^5}$: $p+p^2+p^3+p^4 = p(p+1)(p^2+1)$ and the numerator costs $3\mathbf{f}_5 + 2\mathbf{m}_5 = 38\mathbf{m}$. The denominator costs $5\mathbf{m}$. The total cost is $48\mathbf{m}+\mathbf{i}$. For $\mathbb{F}_{p^7}$ see [Mas20, page ix]: $104\mathbf{m}+\mathbf{i}$. $p+p^2+p^3+p^4+p^5+p^6 = p(p^3+1)(1+p+p^2)$ and the numerator costs $4\mathbf{f}_7 + 3\mathbf{m}_7 = 90\mathbf{m}$. The denominator costs $7\mathbf{m} + \mathbf{i}$. The result costs 7 more $\mathbf{m}$. Finally $\mathbf{i}_7 = 104\mathbf{m} + \mathbf{i}$.

In Table 10 we observe that of all curve instances, BLS12-1150 requires the smallest number of $\mathbb{F}_p$-multiplications, however the base field prime p is 1150-bit long. At this point it is not clear how BLS12-1150 compares to the other candidate curves, however this will become apparent in Section 5 where we discuss the benchmarking results. We now compare the curves of same embedding degree. For $k = 15$, FM15-762 clearly offers a more efficient pairing computation than its competitor BLS15-894 as it requires much less $\mathbb{F}_p$-multiplications and operates over a much smaller prime field. Recall also that FM15-762 has $\rho = 2$, while BLS15-894 has $\rho = 1.5$ showing that in this case, a larger ρ offers better performances. However, we exclude both curves from our benchmark comparison, since both curves are not competitive to other embedding degrees.

For $k = 16$, all three curves KSS16-766, FM16-765 and AFG16-766 are defined over a prime field of almost the same size. The difference in the three families is that FM16-765 and AFG16-766 have $\rho = 2$, while on the contrary, KSS16-766 has $\rho \approx 1.25$. Both FM16-765 and AFG16-766 require almost the same number of $\mathbb{F}_p$-multiplications, which is approximately 20% less than the case of KSS16-766. This justifies our earlier claim that for some embedding degrees, having a larger ρ is beneficial, improving the total pairing computation. We include all three curves for $k = 16$ to verify that the 20% improvement of FM16-765 and AFG16-766 over KSS16-766 is also captured in the benchmarking results.

In the case of $k = 18$, the curves KSS18-638 and SG18-638 operate on a smaller prime field compared to FM18-768. In this case, the theoretical analysis shows that SG18-638 requires less $\mathbb{F}_p$-multiplications for the pairing computation than the two competitors KSS18-638 and FM18-768 and in particular, it requires 15% less $\mathbb{F}_p$-multiplications than KSS18-638. In Section 5 we benchmark all three curves to have more conclusive evidence on their actual performance.

In the case of $k = 20$ the theoretical analysis is less inconclusive as to which curve is the best option for this embedding degree. Note that FM20/FST20-670 and SG20-670 operate on a prime field of the same size and require comparable number of $\mathbb{F}_p$-multiplications, with FM20/FST20-670 having a slight advantage (approximately 3% less $\mathbb{F}_p$-multiplications). On the other hand, although GG20b-575 requires $5879\mathbf{m}$ more compared to FM20/FST20-670, the prime field size is significantly smaller in the case of GG20b-575. Nevertheless, we do not provide benchmarks for these curves, since they are certainly not competitive to other candidates for different embedding degrees. Note that these curves admit degree 4 twists, forcing elliptic curve point operations in the Miller loop to be executed over $\mathbb{F}_{p^5}$.

The curves BLS21-511, BLS24-509, BLS27-426, FST28-510 and GG28-500 offer the smallest primes p, compared to the previous candidate curves. Of these five curves, BLS24-509 is the most promising since it admits degree 6 twists and hence point additions

Table 10: Optimal ate pairing and final exponentiation cost estimates in terms of finite field multiplications. The name of each curve is derived from the name of the corresponding family in Table 5, plus the size of the prime p.

Curve name	r bits	Miller loop opt. ate (**m**)	final exp (**m**)			pairing
			easy	hard	total	total (**m**)
BLS12-1150	768	19288	245	14176	14421	33709
BLS15-894	597	25259	501	58014	58515	83774
FM15-762	**381**	16731	501	55321	55822	72553
KSS16-766	605	16784	240	32826	33066	49850
FM16-765	**384**	10020	255	30024	30279	40299
AFG16-766	**384**	10020	255	30282	30537	40557
KSS18-638	474	17433	480	27008	27488	44921
SG18-638	**383**	13351	480	24308	24788	38139
FM18-768	**384**	13410	464	33184	33648	47058
FM/FST20-670	448	18416	507	35276	35783	54199
SG20-670	**383**	16427	507	39152	39659	56086
GG20b-575	**379**	17554	507	42017	42524	60078
BLS24-509	409	15345	658	24310	24968	40313
BLS21-511	**384**	19321	717	62426	63143	82464
BLS27-426	**383**	22.212	1185	88.438	89.907	112.119
FST28-510	**384**	18940	859	56080	56939	75879
GG28-500	**381**	20326	859	78474	79333	99659

and doublings in the Miller loop are executed over $\mathbb{F}_{p^3}$. In BLS21-511, FST28-510 and GG28-500 additions and doublings are executed over $\mathbb{F}_{p^7}$, while in the case of BLS27-426 over $\mathbb{F}_{p^9}$. Furthermore, Table 10 shows that BLS21-511, FST28-510 and GG28-500 require almost 2× more $\mathbb{F}_p$-multiplications than BLS24-509 and BLS27-426 requires almost 3× more $\mathbb{F}_p$-multiplications than BLS24-509. Therefore, of these five curves we only benchmark BLS24-509.

To summarize, the curves that we have chosen to include in our optimized implementation step are, BLS12-1150, KSS16-766, FM16-765, AFG16-766, KSS18-638, SG18-638, FM18-768 and BLS24-509.

Normalized arithmetic cost. We compared the estimated cost of the Miller loop and final exponentiation in Table 10. However when the base field size is different, the comparison is not always obvious. For example, how do the $j = 1728$ curve families compare to each other in terms of Miller loop? Assuming an architecture with limbs of 64 bits, the fields need 12, 11, 9, and 8 machine-words for 768, 672, 576, and 512 bits resp. We apply the methodology from Aranha et al. [AFK+13, Sect. 8]. Elements of $\mathbb{F}_p$ are assumed to be represented with $\ell = 1 + \lfloor \log_2 p \rfloor$ bits, packed in $w = \lceil \ell/64 \rceil$ 64-bit machine-words. If the Montgomery representation is implemented, a multiplication with reduction in $\mathbb{F}_p$ has complexity $O(2w^2 + w)$. The authors deduce that one can estimate the ratio $m_{640} = 210/136 = 1.544 m_{512}$ (for $10w$). In the same way, we estimate $m_{576} = 171/136 = 1.257 m_{512}$ (for $9w$), $m_{704} = 253/136 = 1.860 m_{512}$ (for $11w$), $m_{768} = 300/136 = 2.205 m_{512}$ (for $12w$). We finally obtain Table 11 and Figure 1.

4 Other pairing operations

In this section, we collect notes about how to implement other operations in pairing groups. We start by briefly surveying scalar multiplications and exponentiation algorithms, with

Table 11: Optimal ate pairing and final exponentiation cost estimates in terms of finite field multiplications, normalized w.r.t. a multiplication in $\mathbb{F}_p$ of 512 bits ($\mathbf{m} = \mathbf{m}_{512}$), with the ratios $m_{448} = 0.772\mathbf{m}$, $m_{576} = 1.257\mathbf{m}$, $m_{640} = 1.544\mathbf{m}$, $m_{704} = 1.860\mathbf{m}$, $m_{768} = 2.205\mathbf{m}$, $m_{896} = 2.986\mathbf{m}$, $m_{1152} = 4.897\mathbf{m}$.

Curve name	r bits	Miller loop opt. ate ($\mathbf{m}_{512}$)	final exp ($\mathbf{m}_{512}$)			pairing
			easy	hard	total	total ($\mathbf{m}_{512}$)
BLS12-1150	768	94455	1200	69421	70621	165075
BLS15-894	597	75406	1496	173189	174685	250091
FM15-762	381	36907	1106	122032	123137	160044
KSS16-766	605	37024	530	72411	72940	109964
FM16-765	384	22103	563	66230	66792	88895
AFG16-766	384	22103	563	66799	67362	89464
KSS18-638	474	26919	742	41704	42445	69364
SG18-638	383	20616	742	37535	38276	58892
FM18-768	384	29581	1024	73200	74224	103805
FST20-670	448	34260	944	65624	66567	100827
SG20-670	383	30560	944	72835	73778	104337
GG20b-575	379	22072	638	52831	53468	75540
BLS21-511	384	19321	717	62426	63143	82464
BLS24-509	409	15345	658	24310	24968	40313
BLS27-426	383	17149	915	68280	69414	86563
FST28-510	384	18940	859	56080	56939	75879
GG28-500	381	20326	859	78474	79333	99659

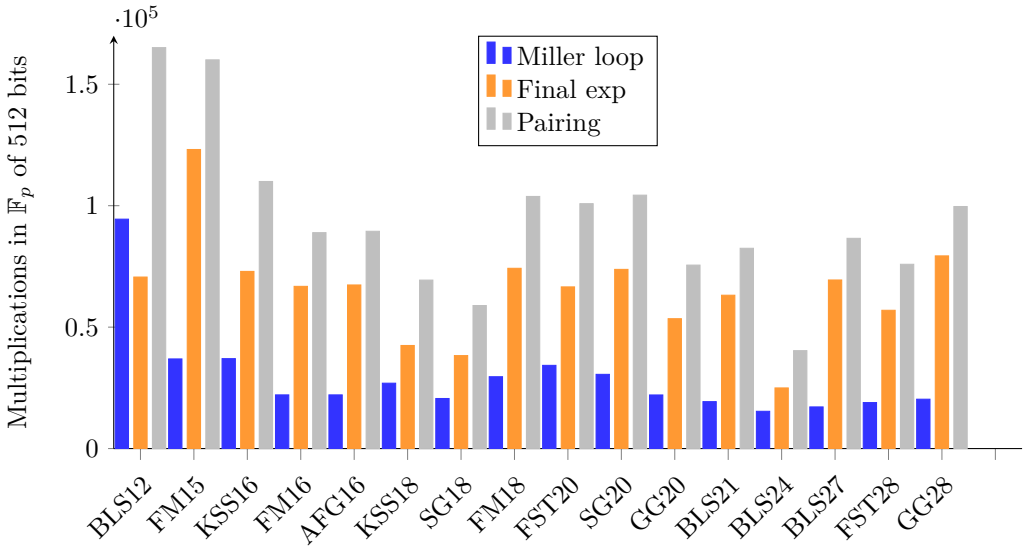

Figure 1: Estimates of Miller loop in terms of normalized multiplications in $\mathbb{F}_p$ of 512 bits.

their constant-time counterparts. We then continue by giving closed formulas for hashing and cofactor-clearing according to the latest state-of-the-art, and finish by enumerating techniques for subgroup membership testing.

4.1 Scalar multiplication and exponentiation in pairing groups

Pairing-friendly curves are typically equipped with additional efficient endomorphisms, such that the widely known Gallant-Lambert-Vanstone (GLV) [GLV01] and Galbraith-Lin-Scott (GLS) [GLS11] scalar multiplication and exponentiation algorithms are the most efficient.

Endomorphisms on $\mathbb{G}_1$ and GLV. All the curves we consider have either $D = 1$ and j-invariant 1728, or $D = 3$ and j-invariant 0. The curves of j-invariant 1728 (e.g. KSS16) have an endomorphism ϕ of the form $(x, y) \mapsto (-x, iy)$ where $i^2 = -1 \in \mathbb{F}_p$ as $p \equiv 1 \bmod 4$, of characteristic polynomial $\chi_\phi(X) = X^2 + 1$. The curves of j-invariant 0 (e.g. KSS18, BLS) have an endomorphism ϕ of the form $(x, y) \mapsto (\omega x, y)$ where ω is a primitive third root of unity in $\mathbb{F}_p$, $\omega^2 + \omega + 1 = 0 \bmod p$, as $p \equiv 1 \bmod 3$, of characteristic polynomial $\chi_\phi(X) = X^2 + X + 1$. In the case of $\mathbb{G}_1$, it is well-known that the GLV technique can be used together with the endomorphism ϕ to decompose a scalar $\ell \in \mathbb{Z}_r$ into half-sized subscalars (ℓ_0, ℓ_1) such that $\ell \equiv \ell_0 + \ell_1\lambda_\phi \bmod r$, with $\lambda_\phi \in \mathbb{Z}_r$ the eigenvalue of the ϕ map. The scalar multiplication can then proceed through the formula $[\ell]P = \ell_0 P + \ell_1\phi(P)$, for $P \in \mathbb{G}_1$. A widely used algorithm to compute scalar multiplication combines the w-NAF scalar recoding algorithm and interleaving of the two smaller scalar multiplications to save point doublings. Because applying ϕ is very efficient in our pairing-friendly curves, we can compute a single precomputation table with 2^{w-2} points for P and apply ϕ to points obtained from the table. Constant-time versions of the algorithm typically replace the recoding process with a regular w-NAF expansion [JT09], which increases the density of non-zero digits from $\frac{1}{w+1}$ to $\frac{1}{w-1}$ and force secure table lookups to prevent leaking what points from the table are being used in the main loop [OLAR13]. From the point of view of efficiency, this approach compares favorably to alternatives [FLS15], by computing fewer point additions at the same cost in precomputed storage.

Endomorphisms on $\mathbb{G}_2$ and $\mathbb{G}_T$, and GLS. In the case of $\mathbb{G}_2$ and $\mathbb{G}_T$, the GLS technique is used instead, assuming again that an efficient endomorphism is available. For $\mathbb{G}_2$, the map ψ is constructed as $(\tau^{-1} \circ \pi \circ \tau)$, where τ is the twisting isomorphism $\tau : E'(\mathbb{F}_{p^{k/d}}) \to E(\mathbb{F}_{p^k})$ and π is the Frobenius map over $E(\mathbb{F}_{p^k})$. This endomorphism ψ has characteristic polynomial $X^2 - tX + p$. It is well-known that ψ has eigenvalue $p \equiv t - 1 \bmod r$ on $\mathbb{G}_2$, by definition of $\mathbb{G}_2 = \ker(\pi - [p])$ [HSV06, GS08, DLZZ23]. For $\mathbb{G}_T$, we choose ϕ as a small multiple or power of the Frobenius π_p in $\mathbb{F}_{p^k}$. In both cases ($\mathbb{G}_2$ and $\mathbb{G}_T$), we can represent the scalar or exponent in base p (the eigenvalue) by decomposing it in $\varphi(k)$ subscalars, where φ is Euler's totient function, such that $\ell \equiv \sum_{0 \le i < \varphi(k)} \ell_i p^i \bmod r$. We can then use the Frobenius map to evaluate a scalar multiplication in $\mathbb{G}_2$ as $[\ell]Q = \sum_{0 \le i < \varphi(k)} [\ell_i]\psi^i(Q)$ in interleaved w-NAF fashion, or the analogue in $\mathbb{G}_T$ after translating to multiplicative notation. The endomorphisms ψ in $\mathbb{G}_2$ and π_p $\mathbb{G}_T$ are quite efficient to evaluate, but it is typical to precompute the various tables over $\psi^i(Q)$ to avoid the cost of applying ψ^i dynamically in succession. The same notes about constant-time implementation apply, noting that the endomorphisms are more expensive to evaluate than in the case of $\mathbb{G}_1$ and now there are $\varphi(k)$ subscalars involved.

One simple implementation trick is useful when supporting multiple pairing-friendly curves simultaneously. Decomposing the scalar into base p can be performed by finding a suitable relationship between the prime subgroup order r, the curve generation seed u and the prime modulus p, such that we can exploit the fact that p and u are related modulo r to decompose in base u instead. After writing the seed u as an expression involving

a few powers of p for all curves, then we can decompose scalar ℓ in base u instead, and perform the scalar recoding with successive (constant-time) integer divisions. Many suitable relationships come directly from the Optimal ate formulas in Table 6, but some require further algebraic manipulation. For AFG16 curves, we exploit $u \equiv -p^5 \bmod r$, to define ψ as the inverse p^5-power Frobenius. For KSS16 curves, we exploit $u \equiv (2p^5 - p) \bmod r$ instead. For KSS18 curves, we exploit $u = (p^3 - 3)p \bmod r$, and for FM18 curves we exploit $u \equiv (p^4 - p) \bmod r$. For SG18 curves, we exploit $-3u = (2p^2 - p^5) \bmod r$ to recode in base $3u$ and define ψ as the $(p^5 - 2p^2)$-power Frobenius, while noting that inversion in both $\mathbb{G}_2$ and $\mathbb{G}_T$ can be efficiently computed.

4.2 Hashing to pairing source groups

Many cryptographic protocols involve hashing arbitrary bit strings to a group of points in an elliptic curve. The security requirements are generally the same for an ideal collision-resistant hash function, such that protocols can rely on the hash function to behave as closely as possible to a conventional random oracle. This notion is formalized as *indifferentiability* to a random oracle, or the infeasibility of differentiating the hash function from a random oracle [BCI+10] with non-trivial probability, given a polynomially-bounded number of queries. Given an indifferentiable hash function h bit strings as inputs/outputs, we can hash to curves by initially constructing a function to hash to a field $\mathbb{F}_q$ and then an encoding function to map field elements to points in the curve, followed by a scalar multiplication by a cofactor to map the output to the right subgroup. From an implementation security standpoint, the full process should also be efficient when evaluated in constant time [WB19, AHST23].

The state-of-the-art approach for indifferentiable hashing to pairing-friendly elliptic curves is the SwiftEC algorithm [CSRT22] and its generalization [Kos24] to broader classes of elliptic curves. The former is restricted to curves with $q \equiv 1 \bmod 3$, with either odd order or order divisible by 4, having $a = 0$ (or j-invariant 0) as a special simpler case. Koshelev's generalization expands the constructions to all curves with $q \equiv 1 \bmod 3$ with non-zero j-invariant, which covers our remaining curves with $b = 0$ (or j-invariant 1728).

Let $\ell = \lceil \log_2 q \rceil$ be the bit length required to encode an element from $\mathbb{F}_q$. Function $H : \{0,1\}^* \to E(\mathbb{F}_q)$ can be constructed by the composition of functions $H = [c] \circ f \circ \eta \circ h$, where $h : \{0,1\}^* \to \{0,1\}^{2\ell+1}$ is a deterministic indifferentiable hash function, $\eta : \{0,1\}^{2\ell+1} \to \mathbb{F}_q^2 \times \{0,1\}$ is an encoding to the field (with an additional sign bit), $f : \mathbb{F}_q^2 \times \{0,1\} \to E(\mathbb{F}_q)$ is an admissible encoding to the group of points (the additional bit is to choose the y-coordinate), and $[c]$ corresponds to scalar multiplication by the cofactor. We select h as the message expansion function XMD [FHSS+23] instantiated with SHA256 as the underlying hash function, and elaborate on the other choices below.

Given $(t_1, t_2) \in \mathbb{F}_q^2$ and a sign bit s, define the constants $\tau = \sqrt{-3} \in \mathbb{F}_q$ and $\omega = \frac{\tau - 1}{2}$, and respectively the denominators d_j and numerators n_j for fractions $X_i = \frac{n_j}{d_j}$ with $j \in \{1, 2, 3\}$:

$$
\begin{aligned}
d_1 &= -2\tau\omega(t_1^6 + 2^3 3\tau b t_1^3 + 2^6 a^3 + 2^3 3\tau t_1^3 t_2^2)t_1 \\
d_2 &= d_1 \cdot \omega \\
d_3 &= -2^4 3^3 (t_1^2 - 2^2 a)^2 t_1^4 t_2^2 \\
n_1 &= t_1^8 + 2^2\omega^2 a t_1^6 + 2^3 3\tau b t_1^5 + 2^5 3\tau\omega^2 a b t_1^3 + 2^6 a^3 t_1^2 + 2^8\omega^2 a^4 + 2^3 3\tau(\omega^2 t_1^2 + 2^2 a)t_1^3 t_2^2 \\
n_2 &= t_1^8 + 2^2\omega a t_1^6 + 2^3 3\tau b t_1^5 + 2^5 3\tau\omega a b t_1^3 + 2^6 a^3 t_1^2 + 2^8\omega a^4 + 2^3 3\tau(\omega t_1^2 + 2^2 a)t_1^3 t_2^2 \\
n_3 &= t_1^{12} + 2^4 3\tau b t_1^9 + 2^6(2a^3 - 3^3 b^2)t_1^6 + 2^{10} 3\tau a^3 b t_1^3 + 2^{12} a^6 \\
&\quad - 2^3 3\tau(t_1^6 - 2^2 3 a t_1^4 - 2^4 3\tau b t_1^3 - 2^4 3 a^2 t_1^2 + 2^6 a^3 - 2^3 3\tau t_1^3 t_2^2)t_1^3 t_2^2.
\end{aligned}
$$

When all of the X_j fractions are defined ($d_j \neq 0$), we select X_j with highest j as the x-coordinate, such that $g(X_j) = X_j^2 + aX_j + b$ is a quadratic residue, to compute $Y = \sqrt{g(X_j)}$. We then take the additional output bit of η to select one of the square roots, and the output is (X_j, y). Otherwise, we output the point at infinity $\mathcal{O}$, which happens only with negligible probability under the assumption that h is indifferentiable from a random oracle. More formally,

$$f(t_1, t_2, s) = \begin{cases} \mathcal{O}, & \text{if any } d_j = 0, \text{ with } j \in \{1,2,3\}. \\ (X_i, (1-2s)Y), & \text{with } i = \max\left\{ j \;\middle|\; \left(\frac{g(X_j)}{q}\right) = 1 \right\} \text{otherwise.} \end{cases}$$

The baseline cost for computing X_j is one square-root extraction to compute Y, two tests for squaredness, one squaring and one multiplication in $\mathbb{F}_q$ to compute each $g(X_j)$ for pairing-friendly curves. Testing for squaredness in $\mathbb{F}_{p^k}$ costs 1 Legendre symbol computation, $(k-1)$ Frobenius p^k-powers and $(k-1)$ multiplications in $\mathbb{F}_{p^k}$ [AR14]. Multiplications by ω or τ consist in k multiplications in $\mathbb{F}_p$, since the constants lie in the base field. Under the guarantee that at least one of the $g(X_j)$ is a square, we can initially assume that $g(X_1)$ is a square and update the choice of X_j after testing the other two values with one symbol computation each. For curves with either $a = 0$ or $b = 0$, we can evaluate the curve equation with one squaring and one multiplication in $\mathbb{F}_q$. Below we discuss the various cases in more details and compute the total operation counts.

The case $b = 0$. We have two subcases in our curve selection. For $\mathbb{G}_2$, we have that multiplicative twists with coefficient $a' = \xi$, for some small non-residue ξ in a subfield, allowing for multiplications by a' to be cheap and evaluated with additions only. By substituting $b = 0$ in the expressions above, after optimizing for common subexpressions and using Montgomery's simultaneous inversion technique, we can evaluate the composition $(f \circ \eta \circ h)$ at the cost of 1 square-root, 2 Legendre symbols and $5\mathbf{s}_{k/d} + 19\mathbf{m}_{k/d} + 5(k/d)\mathbf{m} + \mathbf{i}_{k/d} + ((k/d)-1)(\mathbf{f}_{k/d} + \mathbf{m}_{k/d})$ while adding the baseline costs to evaluating the formulas:

$$\begin{aligned} &h_0 = t_1^2,\ h_1 = h_0^2,\ h_2 = 4a',\ h_3 = 64a'^3,\ h_4 = h_0h_1 + h_3,\ h_5 = t_2^2,\ h_6 = \tau t_1 \\ &h_7 = 24h_0h_5h_6,\ h_8 = h_0h_3,\ h_9 = h_1^2,\ h_{10} = \omega(4a'h_4 + h_0h_7) \\ &d_1 = -2h_6\omega(h_4 + h_7),\ d_2 = d_1\omega,\ d_3 = -432(h_0 - h_2)^2h_1h_5 \\ &n_1 = h_9 + h_8 + 4a'h_7 + h_{10}\omega,\ n_2 = h_9 + h_8 + 4a'h_7 + h_{10} \\ &n_3 = h_1(h_9 + 2h_8) + 4096a'^6 - h_7(h_4 - 12a'(h_1 + 4a'h_0) - h_7). \end{aligned}$$

By further specializing to the case $a = 1$ for $\mathbb{G}_1$, the formulas cost 1 square-root, 2 Legendre symbols and $5\mathbf{s} + 23\mathbf{m} + \mathbf{i}$:

$$\begin{aligned} &h_0 = t_1^2,\ h_1 = h_0^2,\ h_4 = h_1h_0 + 64,\ h_5 = t_2^2,\ h_6 = \tau t_1 \\ &h_7 = 24h_0h_5h_6,\ h_9 = h_1^2,\ h_{10} = \omega(4h_4 + h_0h_7) \\ &d_1 = -2h_6\omega(h_4 + h_7),\ d_2 = d_1\omega,\ d_3 = -432(h_0 - 4)^2h_1h_5 \\ &n_1 = h_9 + 4(16h_0 + h_7) + h_{10}\omega,\ n_2 = h_9 + 4(16h_0 + h_7) + h_{10} \\ &n_3 = h_1(h_9 + 8(16h_0)) + 4096 - h_7(h_4 - 3(4h_1 + 16h_0) - h_7). \end{aligned}$$

The case $a = 0$. For this case, we turn instead to the SwiftEC algorithm, which can be evaluated through 1 square-root extraction, 2 Legendre symbol computations and the

additional costs of $7\mathbf{s}_{k/d} + 11\mathbf{m}_{k/d} + (k/d)\mathbf{m} + \mathbf{i}_{k/d} + ((k/d)-1)(\mathbf{f}_{k/d} + \mathbf{m}_{k/d})$ in $\mathbb{G}_2$, and $7\mathbf{s} + 12\mathbf{m} + \mathbf{i}$ in $\mathbb{G}_1$:

$$h_0 = t_1^3,\ h_1 = t_2^2,\ h_2 = h_0 + b - h_1,\ h_3 = 2h_1 + h_2,\ h_6 = \tau t_1,\ h_7 = h_0 h_6,\ h_8 = 2h_6 t_2$$
$$d_1 = 2h_3 h_8,\ n_1 = h_8(h_7 - t_1 h_3),\ n_2 = (2h_3)^2$$
$$X_1 = n_1/d_1,\ X_2 = -t_1 - n_1/d_1,\ X_3 = t_1 + (n_2/d_1)^2.$$

4.2.1 Cofactor clearing for $\mathbb{G}_1$.

As in Section 2, we denote c_1 the cofactor of the curve, that is, $\#E(\mathbb{F}_p) = c_1 \cdot r$, and define the *cofactor clearing* operation as

$$\begin{array}{rcl} E(\mathbb{F}_p) & \to & \mathbb{G}_1 \\ P & \mapsto & [c_1]P \end{array}$$

In Appendix B Table 21 we give the parameterized formulas for the cofactor c_1 of $\mathbb{G}_1$. Because c_1 might be quite large, there are two strategies to speed-up the multiplication-by-c_1 map $[c_1]$. First, the curves admit an endomorphism for fast GLV scalar multiplication. Second, when the cofactor c_1 has some square factor n_1^2 satisfying the properties of Schoof's [Sch87, Proposition 3.7] (also in [EHG22, Theorem 1]), the strategy of Wahby–Boneh applies [WB19] (multiplying by n_1 instead of n_1^2).

It is important to note that the two techniques (GLV and Wahby–Boneh) are *orthogonal* w.r.t. the $\mathbb{F}_p$-rational curve endomorphism ϕ. In other words, we cannot *combine* the two techniques, we apply them respectively on disjoint subgroups. Following [HGP22], we aim at obtaining the structure of $E(\mathbb{F}_p)$ so as to identify the respective subgroups where to apply each technique:

$$E(\mathbb{F}_p) \simeq \mathbb{Z}/n_1\mathbb{Z} \oplus \mathbb{Z}/n_2\mathbb{Z},\ n_1 \mid n_2 \tag{2}$$

so that $E[n_1] \subset E(\mathbb{F}_p)$, $n_1^2 \mid E(\mathbb{F}_p)$. In our context, we know that $\boldsymbol{r}$ **is prime**, and r^2 does not divide the curve order over $\mathbb{F}_p$, so that r divides n_2 in Eq. (2) that we rewrite as

$$E(\mathbb{F}_p) \simeq \mathbb{Z}/n_1\mathbb{Z} \oplus \mathbb{Z}/n_1 c_1' r\mathbb{Z},\ n_1^2 c_1' = c_1\ . \tag{3}$$

We apply the Wahby–Boneh technique to clear the square factor n_1^2, followed by the GLV technique to clear the factor $c_1' = \#E(\mathbb{F}_p)/(n_1^2 r)$.

Faster co-factor clearing with the GLV method. To improve the cofactor clearing step, one would like to use the endomorphism ϕ. For that one needs to identify the cyclic subgroup of $E(\mathbb{F}_p)$ stable under ϕ, so that ϕ acts as a multiplication by an eigenvalue λ_ϕ. With the notations above, this subgroup has order $c_1' = \#E(\mathbb{F}_p)/(n_1^2 r)$. We compute in SageMath the parameters: eigenvalue λ_ϕ modulo c_1', short scalars (a_0, a_1) so that $a_0 + a_1\lambda_\phi$ is a multiple of c_1'. The multiplication-by-c_1' map becomes

$$\begin{array}{rcl} E(\mathbb{F}_p) & \to & E(\mathbb{F}_p) \\ P & \mapsto & [a_0]P + [a_1]\phi(P)\ . \end{array}$$

Faster co-factor clearing with the Wahby–Boneh technique and Schroof's theorem. We apply the Wahby–Boneh technique to the appropriate subgroup of order n_1 (Eq. (2)), which satisfies the conditions of Theorem 1. We compute in SageMath the parameter n_1 for each curve.

Theorem 1 ([Sch87, Proposition 3.7]). *Let E be an elliptic curve over $\mathbb{F}_p$ and $n_1 \in \mathbb{Z}_{\geq 1}$ with $p \nmid n_1$. Let π_p denote the Frobenius endomorphism of E and t its trace. Then,*

$$E[n_1] \subset E(\mathbb{F}_p) \iff \begin{cases} n_1^2 \mid \#E(\mathbb{F}_p), \\ n_1 \mid p-1 \text{ and} \\ \pi_p \in \mathbb{Z} \text{ or } \mathcal{O}\left(\frac{t^2-4p}{n_1^2}\right) \subset \mathrm{End}_{\mathbb{F}_p}(E). \end{cases}$$

To conlude, we are looking for a decomposition of the curve order so that the Wahby–Boneh technique applies to n_1 and the GLV technique applies to c_1':

$$\#E(\mathbb{F}_p) = \underbrace{n_1^2 \cdot c_1'}_{=c_1} \cdot r \ . \tag{4}$$

4.2.2 Cofactor clearing for $\mathbb{G}_2$

Recently Yu Dai et al. [DLZZ23] extended on finding efficient formulas and provided a Magma script [Dai23]. We rely on their work to obtain the formulas for our curves. As the formulas are quite long for $\mathbb{G}_2$, we implemented them in SageMath to validate the equations. The general strategy is the same as for $\mathbb{G}_1$, but with ψ instead. Note that [FAG20] investigated SG curves.

4.3 Subgroup membership testing

Yu Dai et al. [DLZZ23] consider subgroup membership testing on pairing-friendly curves and generalize Scott's technique [Sco21]. The strategy dates back to GLV idea and for $\mathbb{G}_1$ it consists in finding a short vector (a_0, a_1) such that $a_0 + a_1\lambda_\phi \equiv 0 \bmod r$, then testing if $[a_0]P + [a_1]\phi(P) = \mathcal{O}$. Dai et al. solve the problem for KSS16 and KSS18 curves where the textbook formula of short vector $(a_0, a_1) = (1, \lambda_\phi \bmod r)$, resp. $(1, \lambda_\phi + 1 \bmod r)$ induces a multiplication by a multiple of r *but which is not coprime to the cofactor* c_1. Moreover they obtain a generic technique that can be applied to all curves for $\mathbb{G}_1$ and $\mathbb{G}_2$. We apply their criterion [DLZZ23, Theorem 1] to our curves in Table 12 and solve the problem of cofactors for $\mathbb{G}_1$ in Table 15.

4.3.1 Subgroup membership testing in $\mathbb{G}_1$

Theorem 2 ([DLZZ23, Theorem 3] $\mathbb{G}_1$ for $j = 0$ or $j = 1728$ curves). *Let E be an ordinary elliptic curve defined over a finite field $\mathbb{F}_q$ with j-invariant 0 or 1728, and r a large prime such that $r \| \#E(\mathbb{F}_q)$. Let ϕ be a GLV endomorphism on E, and act as multiplication by an integer λ_ϕ in $\mathbb{G}_1$. Let $(a_0, a_1) \in \mathbb{Z}^2$ with $a_0 + a_1 \cdot \lambda_\phi \equiv 0 \bmod r$. Assume*

- $\gcd(a_0^2 - a_0a_1 + a_1^2, \#E(\mathbb{F}_q)) = r$, *if* $j(E) = 0$,
- $\gcd(a_0^2 + a_1^2, \#E(\mathbb{F}_q)) = r$, *if* $j(E) = 1728$.

Given a non-zero point $P \in E(\mathbb{F}_q)$, then

$$P \in \mathbb{G}_1 \text{ if and only if } [a_0]P + [a_1]\phi(P) = \mathcal{O} \ .$$

For $\mathbb{G}_1$ membership testing there is a problem for KSS-like curves (KSS16, KSS18, GG20, GG28). The naive vector is $(a_0, a_1) = (1, \lambda_\phi \bmod r)$ $(D = 1)$, resp. $(a_0, a_1) = (1, \lambda_\phi + 1 \bmod r)$ $(D = 3)$ or the half-gcd of (r, λ_ϕ). To apply the criteron of Theorem 2, one computes $a_0^2 + a_1^2 = 1 + \lambda_\phi^2$, resp. $a_0^2 - a_0a_1 + a_1^2 = 1 + \lambda_\phi + \lambda_\phi^2$ but this value is not coprime to the cofactor c_1 (see Table 12). In other words if there is a common divisor c_i of $\mathrm{Res}(\chi_\phi(X), a_0 + a_1X)$ and c_1, the test returns true for points of order a divisor c_ir

Table 12: Naive $\mathbb{G}_1$ subgroup membership testing. The eigenvalue of the endomorphism ϕ is denoted λ_ϕ. It assumes testing if $[a_0]P + [a_1]\phi(P)$ is the point at infinity. The (a_0, a_1) are obtained with a half-gcd on $(r(x), \lambda_\phi(x))$ if $2\deg(\lambda_\phi(x)) > \deg(r(x))$.

Curve	$\lambda_\phi \bmod r(x)$	(a_0, a_1)	DLZZ criterion [DLZZ23]
BLS12	$x^2 - 1$	$(1, x^2)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = r$
BLS15	x^5	$(x^3 - x^2 + 1, x^4 - x^2 + x)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = r$
FM15	x^5	$(x^3 - x^2 + 1, x^4 - x^2 + x)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = r$
KSS16	$(x^4 + 24)/7$	$(1, (x^4 + 24)/7)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = 1250r$
FM16	x^4	$(1, x^4)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = r$
AFG16	x^4	$(1, x^4)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = r$
KSS18	$x^3 + 18$	$(1, x^3 + 19)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = 343r$
SG18	$9x^3 + 1$	$(1, 9x^3 + 2)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = 3r$
FM18	$x^3 - 1$	$(1, x^3)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = r$
FST20	x^5	$(x^4 - x^2 + 1, x^3 - x)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = r$
SG20	$8x^5 + 1$	$(2x^3 + 2x^2 + x, 4x^4 + 2x^3 - x - 1)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = r$
GG20b	$(x^5 - 38)/41$	$(x^4 - 2x^3 + 3x^2 - 2x - 7,$ $x^3 - 4x^2 + 11x - 24)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = 41 \cdot 5^4 r$
BLS21	x^7	$(x^6 - x^4 + x^3 - x + 1, x^5 - x^4 + x^2 - x)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = r$
BLS24	$x^4 - 1$	$(1, x^4)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = r$
BLS27	x^9	$(1, x^9 + 1)$	$\gcd(a_0^2 - a_0a_1 + a_1^2, r \cdot h) = 3r$
FST28	x^7	$(x^5 - x^3 + x, x^6 - x^4 + x^2 - 1)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = r$
GG28	$(x^7 + 278)/29$	$(x^5 + 4x^4 + 11x^3 + 24x^2 + 41x + 44,$ $x^6 + 2x^5 + 3x^4 + 2x^3 - 7x^2 - 38x - 117)$	$\gcd(a_0^2 + a_1^2, r \cdot h) = 29r$

(e.g. $1250r$, resp. $343r$ for KSS16 and KSS18), but it should returns true only for points of order exactly r. For KSS16 curves, testing for $P + [\lambda_\phi \bmod r]\phi(P)$ being $\mathcal{O}$ only answers if P has order a divisor of $1250r$, and for KSS18 curves, testing for $P + [\lambda_\phi + 1 \bmod r]\phi(P)$ being $\mathcal{O}$ only answers if P has order a divisor of $343r$. The problem is similar with GG20 and GG28 curves. It means that the naive vector (a_0, a_1) is not as short as possible. There are two strategies to reduce the basis, one experimental involving LLL in [DLZZ23] and one theoretical involving shrinking a basis of a vector space in [Smi15]. We summarise our results in Table 15.

4.3.2 Subgroup membership testing for $\mathbb{G}_2$

Theorem 3 ([DLZZ23, Theorem 1] $\mathbb{G}_2$ for curves with a twist)**.** *Let E be an ordinary elliptic curve over $\mathbb{F}_p$ and t the trace of the Frobenius endomorphism π. Let $\phi\colon E' \to E$ be the twisting isomorphism, where E' is defined over $\mathbb{F}_{p^e}$. Let r be a large prime such that $r \| \#E(\mathbb{F}_p)$ and $r \| \#E'(\mathbb{F}_{p^e})$. Define $\psi = \phi^{-1} \circ \pi \circ \phi$ with the characteristic polynomial $g(\psi) = \psi^2 - t \cdot \psi + p$. Let $\eta = \sum_{i=0}^{s} c_i \cdot p^i$ be a multiple of r and $f(\psi) = \sum_{i=0}^{s} c_i \psi^i$ a polynomial with respect to ψ. Denote by $b_0 + b_1\psi$ the remainder for $f(\psi)$ divided by $g(\psi)$, i.e.,*

$$b_0 + b_1\psi = f(\psi) \bmod g(\psi) \ . \tag{5}$$

Assume

$$\gcd(b_0^2 + b_0b_1t + b_1^2p, \#E'(\mathbb{F}_{p^e})) = r \ . \tag{6}$$

Given a non-identity point $Q \in E'(\mathbb{F}_{p^e})$, then $Q \in \mathbb{G}_2 = E'(\mathbb{F}_{p^e})[r]$ if and only if $f(\psi)(Q) = \mathcal{O}_{E'}$.

As noted by Yu Dai et al. [DLZZ23], the formulas for optimal ate pairing computation and fast $\mathbb{G}_2$ membership testing are very similar. For the optimal ate pairing, the formula parameters should satisfy

$$\sum_{i=0}^{\ell} c_i q^i = m \cdot r \text{ and } mkq^{k-1} \not\equiv \frac{q^k - 1}{r} \sum_{i=0}^{\ell} i c_i q^{i-1} \bmod r \tag{7}$$

Table 13: $\mathbb{G}_1$ cofactor clearing, where λ_ϕ is the eigenvalue of the endomorphism ϕ, $\lambda_\phi \bmod c_1'$ is given such that $\lambda_\phi \bmod r$ matches the data in Table 12, the short vector (a_0, a_1) means evaluating $R \leftarrow [a_0]P + [a_1]\phi(P)$.

Curve	$\lambda_\phi \bmod c_1'(x)$	short vector (a_0, a_1)	criterion	cofactor clearing
BLS12	–	–	–	$R \leftarrow (x-1)P$
BLS15	$-x-1$	$(1, -x)$	$a_0^2 - a_0a_1 + a_1^2 = c_1'$	$Q \leftarrow (x-1)P$; $R \leftarrow Q - [x]\phi(Q)$
FM15	$(6x^4+3x^3+2x^2-4x-8)/7$	$((3x^3+2x^2-4x)/3,$ $(4x^2-2x-3)/3)$	$a_0^2 - a_0a_1 + a_1^2 = c_1'$	$Q \leftarrow [x]P$; $R \leftarrow [a_0]Q + [a_1]\phi(Q)$
KSS16	$\pm(x+1)/2$	$(1, \lambda_\phi \bmod c_1)$	$a_0^2 + a_1^2 = c_1/250$	$Q \leftarrow P + [\lambda_\phi \bmod c_1]\phi(P)$; $R \leftarrow 1250Q$
FM16	x^3	$(1, \lambda_\phi \bmod c_1')$	$a_0^2 + a_1^2 = c_1'$	$Q \leftarrow [x/2]P$; $R \leftarrow Q + [x^3]\phi(Q)$
AFG16	–	–	–	$R \leftarrow [x(x^3+1)/2]P$
KSS18	$x+2, -x-3$	$(1, \lambda_\phi + 1 \bmod c_1)$	$a_0^2 - a_0a_1 + a_1^2 = 3c_1/49$	$Q \leftarrow P + [\lambda_\phi + 1 \bmod c_1]\phi(P)$; $R \leftarrow 343Q$
SG18	–	–	–	$R \leftarrow [3x^2-1]P$
FM18	$-(3x^5+3x^4+3x^3+x)/2$	$(2(x-1)/3,$ $x^3+(x-1)/3)$	$a_0^2 - a_0a_1 + a_1^2 = c_1$	$Q \leftarrow [(x-1)/3]P$; $R \leftarrow [3(x^2+x+1)]Q + Q + P + [2]\phi(Q)$
FST20	x	$(1, x)$	$a_0^2 + a_1^2 = c_1'$	$Q \leftarrow [(x-1)/2]P$; $R \leftarrow Q + [x]\phi(Q)$
SG20	$-4x^3+x$	$(2x^2-1, x)$	$a_0^2 + a_1^2 = c_1'$	$Q \leftarrow [2x]P$; $R \leftarrow [2x^2-1]Q + \phi([x]Q)$
GG20b	$x+2 \bmod x^2+4x+5$ $(1-x)/2 \bmod (x^2-2x+5)/4$	$(1, x+2)$ $(1, (1-x)/2)$	$a_{i0}^2 + a_{i1}^2 = c_{i1}'$	$Q_1 \leftarrow [20]P; Q_2 \leftarrow [8]([2]Q_1+Q_1+\phi(Q_1)) - \phi(Q_1)$; $Q_3 \leftarrow Q_2+[x+2]\phi(Q_2); R \leftarrow Q_3+[(1-x)/2]\phi(Q_3)$
BLS21	x	$(1, x+1)$	$a_0^2 - a_0a_1 + a_1^2 = 3c_1'$	$Q \leftarrow [x-1]P$; $R \leftarrow Q + [x+1]\phi(Q)$
BLS24	–	–	–	$R \leftarrow (x-1)P$
BLS27	–	–	–	$R \leftarrow (x-1)P$
FST28	$-x$	$(1, -x)$	$a_0^2 + a_1^2 = c_1'$	$Q \leftarrow [(x-1)/2]P$; $R \leftarrow Q - [x]\phi(Q)$
GG28	$2-x \bmod x^2-4x+5$ $(1-x)/2 \bmod (x^2-2x+5)/4$	$(1, 2-x)$ $(1, (1-x)/2)$	$a_{i0}^2 + a_{i1}^2 = c_{i1}'$	$Q \leftarrow P + [2-x]\phi(P)$; $R \leftarrow Q + [(1-x)/2]\phi(Q)$

Table 14: From [DLZZ23, Table 2]. Note that for KSS18 $\mathbb{G}_2$, we translate the short vector $(2x/7, 1, 0, x/7, 0, 0)$ to $(0, 0, 2x/7, 1, 0, x/7)$ otherwise $\sum_i a_i\mu_r^i$ produces 0, not a multiple of $r(x)$. For KSS16, we use the data from [DLZZ23, §5.2.2].

Curve	$\mathbb{G}_i$	short vector (a_i)	criteria
BN	$\mathbb{G}_2$	$(x+1, x, x, -2x)$	
BLS12	$\mathbb{G}_2$	$(x, -1, 0, 0)$	
KSS16	$\mathbb{G}_1$	$((31x^4+625)/8750, -(17x^4+625)/8750)$ $= ((31(x/5)^4+1)/14, -(17(x/5)^4+1)/14$	$a_0^2 + a_1^2 = r$, $a_0 + a_1(\lambda_\phi \bmod r) = -17r$ $-31a_1 - 1 + 17a_1(\lambda_\phi \bmod r) = -17^2r$
	$\mathbb{G}_2$	$1/70(-11x+5, 9x+15, -3x-5, -3x-5,$ $13x-25, -7x+35, -x-25, -11x+5)$	$\sum_i a_i\mu^i = 127r$
KSS18	$\mathbb{G}_1$	$((x/7)^3, -18(x/7)^3 - 1)$	$a_0^2 - a_0a_1 + a_1^2 = r$, $a_0 + a_1(\lambda_\phi \bmod r) = -18r$
	$\mathbb{G}_2$	$(0, 0, 2x/7, 1, 0, x/7)$	$a_2\mu^2 + a_3\mu^3 + a_5\mu^5 = -18r$

Table 15: $\mathbb{G}_1$ membership testing, solving the issues in Table 12. the endomorphism ϕ has eigenvalue λ_ϕ, the short vector (a_0, a_1) satisfies $a_0^2+a_1^2 = r$ $(j = 1728)$, resp. $a_0^2-a_0a_1+a_1^2 = r$ $(j = 0)$, where a_0, a_1 are integers. Note that for KSS16, $x \equiv 25, 45 \bmod 70$ hence $5 \mid x$; for KSS18, $x \equiv 14 \bmod 21$ hence $7 \mid x$; for GG20b, $x \equiv 1465, 1565 \bmod 2050$ hence $5 \mid x$.

Curve	$\lambda_\phi \bmod r$	(a_0, a_1) s.t. $a_0^2 + a_1^2 = r$, resp. $a_0^2 - a_0a_1 + a_1^2 = r$	observation
KSS16	$(x^4+24)/7$	$((-443(x/5)^4 - 17)/14, ((x/5)^4 + 5)/14)$	$a_0 = -443a_1 + 157$
KSS18	x^3+18	$(19(x/7)^3 + 1, (x/7)^3)$	$a_0 = 19a_1 + 1$
SG18	$9x^3+1$	$(-3x^3, 3x^3+1)$	$a_1 = -a_0 + 1$
GG20b	$(x^5-38)/41$	$((61(x/5)^4 - 54(x/5)^3 + 31(x/5)^2 - 14(x/5) + 5)/41,$ $(148(x/5)^4 - 47(x/5)^3 + 8(x/5)^2 + 3(x/5) - 4)/41)$	$u = 30 \bmod 41$
		$((92(x/5)^4 - 63(x/5)^3 + 32(x/5)^2 - 13(x/5) + 4)/41),$ $(131(x/5)^4 - 34(x/5)^3 + (x/5)^2 + 6(x/5) - 5)/41$	$u = 7 \bmod 41$
BLS27	x^9	$((-x^9+1)/3, (x^9+2)/3)$	$a_1 = -a_0 + 1$
GG28	$(x^7+278)/29$	$((2x^6 + 9x^5 + 26x^4 + 59x^3 + 106x^2 + 129x - 14)/29,$ $(5x^6 + 8x^5 + 7x^4 - 12x^3 - 83x^2 - 272x - 673)/29)$	$u = 14 \bmod 29$
		$((5x^6 + 12x^5 + 23x^4 + 32x^3 + 13x^2 - 108x - 497)/29,$ $(2x^6 - x^5 - 14x^4 - 51x^3 - 134x^2 - 281x - 454)/29)$	$u = 19 \bmod 29$

for some non-zero integer m and short integers c_i [Ver10, Theorem 1].

As explained in [Smi15, HGP22, DLZZ23], the points on the d-th twist $E'(\mathbb{F}_{p^{k/d}})$ containing $\mathbb{G}_2$ have eigenvalue λ_ψ under Galbraith–Scott endomorphism $\psi = \tau^{-1} \circ \pi \circ \tau$, where λ_ψ is a root of the characteristic polynomial $\chi_\psi(X) = X^2 - tX + p$, with the trace t of $E(\mathbb{F}_p)$. A point in $\mathbb{G}_2$ has eigenvalue $\lambda_\psi \bmod r = p \bmod r = t - 1 \bmod r$ under ψ. But for subgroup membership testing, one cannot assume that the point being tested has eigenvalue $t-1$, one relies on the generic formula of λ_ψ the root of $\chi_\psi(X)$ modulo the curve order which maps to $t-1$ when reduced modulo r. One checks that the resultant of $\chi_\psi(X)$ and the polynomial formula $f(X) = c_0 + c_1X + \ldots + c_\ell X^\ell$ (see eq. (7)) has no common factor with the cofactor c_2 of $\mathbb{G}_2$, that is, $\mathrm{Res}_x(\mathrm{Res}_X(\chi_\psi(X), f(X)), c_2(x)) \neq 0$ [DLZZ23, Theorem 1]. The result is a rational number. The final step is to check that $c_2(x)$ has no root modulo each prime divisor of the numerator of that result.

We start from the data in Table 6 and check if Yu Dai et al. Theorem 3 is satisfied. Yu Dai et al. already solved the case for KSS16 with $x = 45 \bmod 70$, and KSS18, see Table 14.

BLS curves. All BLS curves have a trace $t = u + 1$, so that the curve order is $p + 1 - t = p - u$, and this number is a multiple of the prime order r of $\mathbb{G}_1$. The optimal ate pairing formula is based on the equation $u - p = 0 \bmod r$. For BLS12 and BLS24 curves, this equation directly gives a $\mathbb{G}_2$ membership test: as soon as $r = \Phi_k(u)$ is prime (it corresponds to $\gcd(c_1, c_2) = 1$), $Q \in \mathbb{G}_2 \iff [u]Q - \psi(Q) = \mathcal{O}$ that is, $Q \in \mathbb{G}_2 \iff [u]Q = \psi(Q)$. This is not the case for BLS15, BLS21 nor BLS27 curves.

BLS12, BLS24 curves. Solutions appear already in previous works [Sco21, HGP22, DLZZ23]. Assuming that the order r of $\mathbb{G}_1, \mathbb{G}_2$ is prime and $\gcd(c_1, c_2) = 1$ (the cofactors of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively are coprime), the test is $[u]Q = \psi(Q)$.

BLS15, BLS21 curves. This case is different as the `gcd` of c_1 and c_2 is $x^2 + x + 1$. For BLS15, note that $\Phi_{15}(p) = p^8 - p^7 + p^5 - p^4 + p^3 - p + 1$. Then $p^7(x - p) = xp^7 - p^8 = 1 - p + p^3 - p^4 + p^5 + (x - 1)p^7 \bmod \Phi_{15}(p)$ and this formula satisfies Theorem 3. That is, the $\mathbb{G}_2$ membership test is $Q - \psi(Q) + \psi^3(Q) - \psi^4(Q) + \psi^5 + [u - 1]\psi^7(Q) = \mathcal{O}$. When $7 \mid h_2$, this test still does not distinguish points of order $7r$. For BLS21, note that $\Phi_{21}(p) = p^{12} - p^{11} + p^9 - p^8 + p^6 - p^4 + p^3 - p + 1$. Then $p^{11}(x - p) = xp^{11} - p^{12} = 1 - p + p^3 - p^4 + p^6 - p^8 + p^9 + (x - 1)p^{11} \bmod \Phi_{21}(p)$ and this formula succeeds in Theorem 3 when c_2 is odd. The $\mathbb{G}_2$ membership test is $Q - \psi(Q) + \psi^3(Q) - \psi^4(Q) + \psi^6 - \psi^8(Q) + \psi^9(Q) + [u - 1]\psi^{11}(Q) = \mathcal{O}$.

BLS27 curves. The `gcd` of the resultant and c_2 is 3 with the formula $(u, -1)$. The formula $p^{17}(u - p) = 1 + p^9 + up^{17} \bmod \Phi_{27}(p)$ does not solve the problem of the cofactor 3. With Smith technique we obtain that $(1 + p + p^2 + \ldots + p^8)(1 - p^9)(1 - u)/3 + p^9$ is a multiple of r and avoids the cofactor 3 issue. The $\mathbb{G}_2$ membership test is $[(1 - u)/3](S - \psi^9(S)) + \psi^9(Q) = \mathcal{O}$ where $S \leftarrow Q + \psi(Q) + \psi^2(Q) + \ldots + \psi^8(Q)$.

FM15 curves. The formula for optimal ate pairing is $u - p^4$ (Table 6). The `gcd` in $\mathbb{Q}[u]$ of $\mathrm{Res}_X(X^2 - t(u)X + p(u), u - X^4)$ and $\#E'(\mathbb{F}_{p^5}) = h_2(u)r(u)$ is $r(u)$. Following [HGP22], we then compute

$$\mathrm{Res}_u(\mathrm{Res}_X(X^2 - t(u)X + p(u), u - X^4)/r(u), h_2(u))$$

and obtain a very large number whose small factors are $2^4 \cdot 1372471$. We observe that $\mathrm{Res}_X(u)$ and $h_2(u)$ share the irreducible factor $(u^2 + u + 1)$ modulo 2 but do not share a common root. Actually, the resultant $\mathrm{Res}_X(u)$ cannot be even. This ensures that the test based on the optimal ate pairing formula: whether $[u]Q - \psi^4(Q) = \mathcal{O}$ is a valid $\mathbb{G}_2$ subgroup membership test.

KSS16 curves. An optimal pairing formula can be $x/5p(x) + p^2(x) + 2x/5p^5(x)$ but the derived subgroup membership testing formula does not distinguish points of order $2r$. With the same technique as [DLZZ23], we obtain the following formula for $\mathbb{G}_2$ membership testing when $x = 25 \bmod 70$ ([DLZZ23] addresses $x = 45 \bmod 70$).

$$1/70(11x + 5, 11x + 5, -19x - 15, 3x - 5, -3x + 5, -13x - 25, 7x - 35, x + 45) \quad (8)$$

We suggest this formula with larger scalars that works for any $x = 25, 45 \bmod 70$:

$$(u_5 + 1, u_5, u_5, u_5, -3u_5 - 2, -3u_5 - 2, -3u_5 - 2, -3u_5 - 2),\ u_5 = (u - 5)/10 \quad (9)$$

FM16 curves. The formula $u - p$ gives the equation $[u]Q - \psi(Q) = \mathcal{O}$ and satisfies Theorem 3.

AFG16 curves. The formulas $-1+up^3$ and $u+p^5$ give the equations $-Q+[u]\psi^3(Q) = \mathcal{O}$ and $[u]Q + \psi^5(Q) = \mathcal{O}$ that both satisfy Theorem 3.

KSS18 curves. For KSS18 curves, the usual formulas of optimal ate pairing are [Ver10, Section 4]

$$2x/7 + p(x) + x/7p^3(x) \text{ or } 1 + xp^2(x) + 2p^3(x)\ . \quad (10)$$

One notes that $2x/7+(p \bmod r)+x/7(p^3 \bmod r) = 0$ and $1+x(p^2 \bmod r)+2(p^3 \bmod r) = -5 \cdot 7 \cdot r(x)$.

SG18 curves. The optimal pairing formula $u + p^2 + up^3$ gives the equation $[u]Q + \psi^2(Q) + [u]\psi^3(Q) = \mathcal{O}$ and satisfies Theorem 3.

FM18 curves. The formulas $1+up^2$ and $u+p-p^4$ give the equations $Q+[u]\psi^2(Q) = \mathcal{O}$ and $[u]Q + \psi(Q) - \psi^4(Q) = \mathcal{O}$ and satisfy Theorem 3.

FST20, FST28 curves. The optimal pairing formula $u-p$ does not give a valid equation, the resultant of $u - X$ and $X^2 - t(u)X + p(u)$ is $(u^2 + 1)r(u)$. We apply the same trick as for BLS: $p^{\varphi(k)-1}(u-p) \bmod \Phi_k(p)$ gives $1-p^2+p^4-p^6+up^7$ for $k = 20$ and $1-p^2+p^4-p^6+p^8-p^{10}+up^{11}$ for $k = 28$. The equations $Q-\psi^2(Q)+\psi^4(Q)-\psi^6(Q)+[u]\psi^7(Q) = \mathcal{O}$ for $k = 20$ and $Q - \psi^2(Q) + \psi^4(Q) - \psi^6(Q) + \psi^8(Q) - \psi^{10}(Q) + [u]\psi^{11}(Q) = \mathcal{O}$ satisfy Theorem 3.

SG20 curves. The optimal pairing formulas $u - up^5 - p^7$ and $u + p^2 + up^5$ give the equations $[u]Q - [u]\psi^5(Q) - \psi^7(Q) = \mathcal{O}$ and $[u]Q + \psi^2(Q) + [u]\psi^5(Q) = \mathcal{O}$ and satisfy Theorem 3.

GG20b curves. The optimal pairing formulas $u-p-2p^6$ and $2+up^4-p^5$ do not satisfy Theorem 3 as the gcd is $(u^2 + 4u + 5)r(u)$. We set $p^{\varphi(k)-5}(2 + up^4 - p^5) \bmod \Phi_{20}(p) = 1 - p^2 + p^4 - p^6 + 2p^3 + up^7$ and obtain a formula that satisfies Theorem 3: $(\mathrm{Id} -\psi^2 + \psi^4 - \psi^6 + [2]\psi^3 + [u]\psi^7)(Q) = \mathcal{O}$.

GG28 curves. The optimal pairing formulas $u-p-2p^8$ and $2+up^6-p^7$ do not satisfy Theorem 3 as the gcd is $(u^2 - 4u + 5)r(u)$. We set $p^{\varphi(k)-7}(2 + up^6 - p^7) \bmod \Phi_{28}(p) = 1 - p^2 + p^4 - p^6 + p^8 - p^{10} + 2p^5 + up^{11}$ and obtain a formula that satisfies Theorem 3: $(\mathrm{Id} -\psi^2 + \psi^4 - \psi^6 + \psi^8 - \psi^{10} + [2]\psi^5 + [u]\psi^{11})(Q) = \mathcal{O}$.

5 Experimental results

We implemented the most promising curves from Section 3.4 within the RELIC cryptographic toolkit [AGM+] due to its strong support to pairing-based cryptography. RELIC already implemented several pairing-friendly curves (such as BLS) with state-of-the-art performance, so extending the library to include new curve families is natural and favors a fair comparison between candidates. The library is implemented in the C programming language, with handwritten ASM acceleration for finite field arithmetic in multiple sizes of the prime moduli. For each of the new or previously supported curve families, we included group operations and pairing computation, by either speeding-up existing routines or including entirely new ones. The resulting source code can be found at our anonymized repository [4], with prebuilt binaries to facilitate reproduction of the timings. Our SageMath and MAGMA scripts can also be found at the same repository.

Benchmarking measurements were taken by computing the average latency of running the operation for 10^4 consecutive executions on an Intel Kaby Lake Core i7-7700 CPU running at 3.60GHz. The main compiler used was GCC version 13.2.1, with optimization flags `-O3 -funroll-loops -march=native -mtune=native`; but we also verified that `clang` would exhibit similar performance. Following best practices[5], the TurboBoost and HyperThreading features were disabled in the benchmarking machine for higher stability.

Table 16 shows the cycle counts observed for executing scalar multiplications or exponentiaton in pairing groups using a w-NAF algorithm with $w = 4$ and Jacobian coordinates in the Weierstrass model. The constant-time version for the source groups was implemented with almost-complete exception-free homogeneous projective coordinates [RCB16] using a regular w-NAF algorithm with $w = 5$ to compensate the performance loss. The table also shows timings for hashing to source groups, testing group elements for subgroup membership and pairing computation, split in the Miller loop and Final Exponentiation. From the table, it is clear that BLS24-509 has superior performance for all benchmarked operations, with speedups ranging from to 3.2% to 56.1%. We also highlight the fastest operations in curves with embedding degree 16 or 18, and show that AFG16 is a competitive candidate in comparison to other curves with the same embedding degree.

When checking for consistency between the timings for pairing computation and the normalized operation counts from Table 11, we can conclude that most figures are consistent, showing strong alignment between the implementation effort and the performance estimates. There are two exceptions, curve BLS12-1150 and FM16-765. The former deviates from the other implementations in the sense that it ends up more efficient than estimated, because it uses the low-level interface of the GMP library with Karatsuba splitting, leading to a slightly different cost model than the Schoolbook method assumed during normalization. For FM16-765, the observed performance is worse than AFG16 due to a less efficient choice of towering, that particular affects the timings for the final exponentiation. This is why the pairing latency looks closer to KSS16 than AFG16. Furthermore, we do not claim that all of our timings are the best possible for all parameters. Our emphasis was to fairly optimize all parameters with similar effort, such that meaningful comparisons could be made. For this reason, we decided to not implement techniques that could favor one parameter over the others [Lon23, BCN14] due to availability of more efficient code, although we leave to implement them as future work, as long as we feel confident that the same optimization level can be achieved across all parameters.

[4] https://github.com/cic-pairing192/suppl-material
[5] https://bench.cr.yp.to/supercop.html

Table 16: Latency in 10^3 clock cycles for performing scalar multiplication or exponentiation in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$, possibly with constant-time (CT) countermeasures.

Curve	BLS12	FM16	KSS16	AFG16	FM18	SG18	KSS18	BLS24
Size of p	1150	765	766	766	768	638	638	509
$[\ell]P$ in $\mathbb{G}_1$	6803	2216	3282	*2157*	2079	*1368*	1718	**1066** (22.1%)
$[\ell]Q$ in $\mathbb{G}_2$	15776	12193	18433	*11227*	8196	*6101*	7450	**5199** (14.8%)
g^ℓ in $\mathbb{G}_T$	23639	8867	13375	*8451*	12372	*9437*	11748	**7129** (15.6%)
CT $[\ell]P$ in $\mathbb{G}_1$	9168	2805	4288	*2802*	2764	*1956*	2394	**1275** (34.8%)
CT $[\ell]Q$ in $\mathbb{G}_2$	20332	19068	25795	*17952*	13524	*9622*	11416	**7548** (21.6%)
CT g^ℓ in $\mathbb{G}_T$	27754	12359	16485	*11426*	15760	*12938*	15085	**11058** (03.2%)
Hash to $\mathbb{G}_1$	6280	2845	*1759*	2495	3970	1490	*1115*	**489** (56.1%)
Hash to $\mathbb{G}_2$	18662	27573	22907	*22052*	27350	15130	*8894*	**5788** (34.9%)
Testing in $\mathbb{G}_1$	4547	1652	3060	*1639*	1494	*1018*	1808	**797** (21.7%)
Testing in $\mathbb{G}_2$	4935	3029	6547	*2536*	2057	*1634*	1927	**1061** (35.0%)
Testing in $\mathbb{G}_T$	5351	2579	5895	*2125*	4225	9878	*2359*	**1294** (39.1%)
Miller Loop	28484	8188	11871	*6950*	9749	*7135*	9327	**5429** (21.9%)
Final Exp.	29317	31694	28533	*25666*	27176	*13628*	15607	**9670** (29.0%)
Pairing	57802	39882	40404	*32617*	36925	*20763*	24971	**15100** (27.3%)

6 Conclusion

We performed an extensive comparison of several candidate families of pairing-friendly curves at the 192-bit level, both from theoretical and practical perspectives. Our recommended choices of pairing-friendly curves are the following. For a **prime-order curve**, choose a BN curve of ≈1152 bits. For **the fastest pairing**, take a BLS24 curve of ≈512 bits. For **smallest** $\mathbb{G}_1$, a BLS27 curve can have p of ≈427 bits. While we focus exclusively on the asymmetric pairings that offer the best performance, curves with embedding degree $k = 1$ and supersingular curves with $k = 2$ can be used to instantiate a symmetric pairing, at high performance penalty due to the very large base fields.

References

[AFK+13] Diego F. Aranha, Laura Fuentes-Castañeda, Edward Knapp, Alfred Menezes, and Francisco Rodríguez-Henríquez. Implementing pairings at the 192-bit security level. In Michel Abdalla and Tanja Lange, editors, *PAIRING 2012*, volume 7708 of *LNCS*, pages 177–195. Springer, Berlin, Heidelberg, May 2013. doi:10.1007/978-3-642-36334-4_11.

[AFK24] Mónica P. Arenas, Georgios Fotiadis, and Elisavet Konstantinou. Special TNFS-secure pairings on ordinary genus 2 hyperelliptic curves. In Serge Vaudenay and Christophe Petit, editors, *AFRICACRYPT 2024*, volume 14861 of *LNCS*, pages 285–310, Douala, Cameroon, July 10-12 2024. Springer. doi:10.1007/978-3-031-64381-1_13.

[AGM+] Diego F. Aranha, Conrado P. L. Gouvêa, Tobias Markmann, Riad S. Wahby, and K. Liao. RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic.

[AHG23] Diego F. Aranha, Youssef El Housni, and Aurore Guillevic. A survey of elliptic curves for proof systems. *Des. Codes Cryptography*, 91(11):3333–3378, 2023. doi:10.1007/s10623-022-01135-y.

[AHST23] Diego F. Aranha, Benjamin Salling Hvass, Bas Spitters, and Mehdi Tibouchi. Faster constant-time evaluation of the Kronecker symbol with application

to elliptic curve hashing. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *CCS*, pages 3228–3238. ACM, 2023. ePrint 2023/1261. doi:10.1145/3576915.3616597.

[AKL+11] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio Cesar López-Hernández. Faster explicit formulas for computing pairings over ordinary curves. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 48–68. Springer, Berlin, Heidelberg, May 2011. doi:10.1007/978-3-642-20465-4_5.

[ALH10] Diego F. Aranha, Julio Cesar López-Hernández, and Darrel Hankerson. High-speed parallel software implementation of the η_T pairing. In Josef Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 89–105. Springer, Berlin, Heidelberg, March 2010. doi:10.1007/978-3-642-11925-5_7.

[APR21] Diego F. Aranha, Elena Pagnin, and Francisco Rodríguez-Henríquez. LOVE a pairing. In Patrick Longa and Carla Ràfols, editors, *LATINCRYPT 2021*, volume 12912 of *LNCS*, pages 320–340. Springer, Cham, October 2021. doi:10.1007/978-3-030-88238-9_16.

[AR14] Gora Adj and Francisco Rodríguez-Henríquez. Square root computation over even extension fields. *IEEE Trans. Computers*, 63(11):2829–2841, 2014. ePrint 2012/685. doi:10.1109/TC.2013.145.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. ePrint 2018/046, 2018.

[BCI+10] Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 237–254. Springer, Berlin, Heidelberg, August 2010. doi:10.1007/978-3-642-14623-7_13.

[BCN14] Joppe W. Bos, Craig Costello, and Michael Naehrig. Exponentiating in pairing groups. In Tanja Lange, Kristin Lauter, and Petr Lisonek, editors, *SAC 2013*, volume 8282 of *LNCS*, pages 438–455. Springer, Berlin, Heidelberg, August 2014. doi:10.1007/978-3-662-43414-7_22.

[BD19] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 32(4):1298–1336, October 2019. doi:10.1007/s00145-018-9280-5.

[BEG19] Razvan Barbulescu, Nadia El Mrabet, and Loubna Ghammam. A taxonomy of pairings, their security, their complexity. ePrint 2019/485, rev. Sept. 24, 2019, 2019.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Berlin, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_13.

[BGW+22] Dan Boneh, Sergey Gorbunov, Riad S. Wahby, Hoeteck Wee, Christopher A. Wood, and Zhenfei Zhang. Bls signatures. https://datatracker.ietf.org/doc/draft-irtf-cfrg-bls-signature/, Jun 2022. IETF draft.

[BKLS02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 354–368. Springer, Berlin, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_23.

[BL13] Daniel J. Bernstein and Tanja Lange. Non-uniform cracks in the concrete: The power of free precomputation. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 321–340. Springer, Berlin, Heidelberg, December 2013. doi:10.1007/978-3-642-42045-0_17.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Berlin, Heidelberg, December 2001. doi:10.1007/3-540-45682-1_30.

[BLS03] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing elliptic curves with prescribed embedding degrees. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02*, volume 2576 of *LNCS*, pages 257–267. Springer, Berlin, Heidelberg, September 2003. doi:10.1007/3-540-36413-7_19.

[BMDFAF19] Narcise Bang Mbiang, Diego De Freitas Aranha, and Emmanuel Fouotsa. Computing the optimal ate pairing over elliptic curves with embedding degrees 54 and 48 at the 256-bit security level. *International Journal of Applied Cryptography (IJACT)*, 4(1):45–59, 2019. doi:10.1504/IJACT.2020.107167.

[BMUS23] Marta Bellés-Muñoz, Jorge Jiménez Urroz, and Javier Silva. Revisiting cycles of pairing-friendly elliptic curves. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 3–37. Springer, Cham, August 2023. doi:10.1007/978-3-031-38545-2_1.

[BN06] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 319–331. Springer, Berlin, Heidelberg, August 2006. doi:10.1007/11693383_22.

[Bow17] Sean Bowe. BLS12-381: New zk-SNARK elliptic curve construction. Zcash blog, March 11 2017. https://electriccoin.co/blog/new-snark-curve/.

[BS23] Ward Beullens and Gregor Seiler. LaBRADOR: Compact proofs for R1CS from module-SIS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 518–548. Springer, Cham, August 2023. doi:10.1007/978-3-031-38554-4_17.

[BW05] Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37(1):133–141, 2005. ePrint 2003/143. doi:10.1007/s10623-004-3808-4.

[CCW19] Alessandro Chiesa, Lynn Chua, and Matthew Weidner. On cycles of pairing-friendly elliptic curves. *SIAM Journal on Applied Algebra and Geometry*, 3(2):175–192, 2019. doi:10.1137/18M1173708.

[CDS20] Rémi Clarisse, Sylvain Duquesne, and Olivier Sanders. Curves with fast computations in the first pairing group. In Stephan Krenn, Haya Shulman,

and Serge Vaudenay, editors, *CANS 20*, volume 12579 of *LNCS*, pages 280–298. Springer, Cham, December 2020. doi:10.1007/978-3-030-65411-5_14.

[CH07] Jaewook Chung and M. Anwar Hasan. Asymmetric squaring formulae. In *18th IEEE Symposium on Computer Arithmetic (ARITH-18 2007), 25-27 June 2007, Montpellier, France*, pages 113–122. IEEE Computer Society, 2007. https://www.lirmm.fr/arith18/papers/Chung-Squaring.pdf. doi:10.1109/ARITH.2007.11.

[CHZ22] Shi Ping Cai, Zhi Hu, and Chang-An Zhao. Faster final exponentiation on the KSS18 curve. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E105.A(8):1162–1164, 2022. ePrint 2021/1309. doi:10.1587/transfun.2021EAL2086.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Berlin, Heidelberg, August 2004. doi:10.1007/978-3-540-28628-8_4.

[CLN10] Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 224–242. Springer, Berlin, Heidelberg, May 2010. doi:10.1007/978-3-642-13013-7_14.

[CLN11] Craig Costello, Kristin Lauter, and Michael Naehrig. Attractive subfamilies of BLS curves for implementing high-security pairings. In Daniel J. Bernstein and Sanjit Chatterjee, editors, *INDOCRYPT 2011*, volume 7107 of *LNCS*, pages 320–342. Springer, Berlin, Heidelberg, December 2011. doi:10.1007/978-3-642-25578-6_23.

[Cos12] Craig Costello. Pairings for beginners. https://www.craigcostello.com.au/s/PairingsForBeginners.pdf, 2012.

[CRSCN24] Maria Corte-Real Santos, Craig Costello, and Michael Naehrig. On cycles of pairing-friendly abelian varieties. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO*, LNCS, Santa Barbara, CA, Aug 19-22 2024. Springer-Verlag. to appear, ePrint 2024/869.

[CSB05] Sanjit Chatterjee, Palash Sarkar, and Rana Barua. Efficient computation of Tate pairing in projective coordinate over general characteristic fields. In Choonsik Park and Seongtaek Chee, editors, *ICISC 04*, volume 3506 of *LNCS*, pages 168–181. Springer, Berlin, Heidelberg, December 2005. doi:10.1007/11496618_13.

[CSRT22] Jorge Chávez-Saab, Francisco Rodríguez-Henríquez, and Mehdi Tibouchi. SwiftEC: Shallue-van de Woestijne indifferentiable function to elliptic curves - faster indifferentiable hashing to elliptic curves. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 63–92. Springer, Cham, December 2022. doi:10.1007/978-3-031-22963-3_3.

[Dai23] Yu Dai. smt-magma. https://github.com/eccdaiy39/smt-magma, 2023.

[DGP21] Gabrielle De Micheli, Pierrick Gaudry, and Cécile Pierrot. Lattice enumeration for tower NFS: A 521-bit discrete logarithm computation. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 67–96. Springer, Cham, December 2021. doi:10.1007/978-3-030-92062-3_3.

[DLZZ23] Yu Dai, Kaizhan Lin, Chang-An Zhao, and Zijian Zhou. Fast subgroup membership testings for $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ on pairing-friendly curves. *Designs, Codes and Cryptography*, 91(10):3141–3166, Oct 2023. ePrint 2022/348. doi:10.1007/s10623-023-01223-7.

[DZZ23a] Yu Dai, Fangguo Zhang, and Chang-An Zhao. Don't forget pairing-friendly curves with odd prime embedding degrees. *IACR TCHES*, 2023(4):393–419, 2023. doi:10.46586/tches.v2023.i4.393-419.

[DZZ23b] Yu Dai, Fangguo Zhang, and Chang-An Zhao. Fast hashing to $\mathbb{G}_2$ on pairing-friendly curves with the lack of twists. *Finite Fields and Their Applications*, 91:102263, 2023. ePrint 2022/996. doi:10.1016/j.ffa.2023.102263.

[DZZZ21] Yu Dai, Zijian Zhou, Fangguo Zhang, and Chang-An Zhao. Software implementation of optimal pairings on elliptic curves with odd prime embedding degrees. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E105.A(5):858–870, 2021. ePrint 2021/1162. doi:10.1587/transfun.2021EAP1115.

[EHG22] Youssef El Housni and Aurore Guillevic. Families of SNARK-friendly 2-chains of elliptic curves. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 367–396. Springer, Cham, May / June 2022. doi:10.1007/978-3-031-07085-3_13.

[FAG20] Emmanuel Fouotsa and Laurian Azebaze Guimagang. Fast hashing to $\mathbb{G}_2$ on aurifeuillean pairing-friendly elliptic curves. *SN Comput. Sci.*, 1(1):51, 2020. doi:10.1007/S42979-019-0053-5.

[FAGA23] Emmanuel Fouotsa, Laurian Azebaze Guimagang, and Raoul Ayissi. x-superoptimal pairings on elliptic curves with odd prime embedding degrees: BW13-P310 and BW19-P286. *Applicable Algebra in Engineering, Communication and Computing (AAECC)*, pages 1–19, Feb 2023. ePrint 2022/716. doi:10.1007/s00200-023-00596-5.

[FHSS+23] Armando Faz-Hernandez, Sam Scott, Nick Sullivan, Riad S. Wahby, and Christopher A. Wood. Hashing to Elliptic Curves. RFC 9380, August 2023. URL: https://www.rfc-editor.org/info/rfc9380, doi:10.17487/RFC9380.

[FK19] Georgios Fotiadis and Elisavet Konstantinou. TNFS resistant families of pairing-friendly elliptic curves. *Theoretical Computer Science*, 800:73–89, 31 December 2019. ePrint 2018/1017. arXiv:2018/1017, doi:10.1016/j.tcs.2019.10.017.

[FKR12] Laura Fuentes-Castañeda, Edward Knapp, and Francisco Rodríguez-Henríquez. Faster hashing to $\mathbb{G}_2$. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 412–430. Springer, Berlin, Heidelberg, August 2012. doi:10.1007/978-3-642-28496-0_25.

[FLS15] Armando Faz-Hernández, Patrick Longa, and Ana H. Sánchez. Efficient and secure algorithms for GLV-based scalar multiplication and their implementation on GLV-GLS curves (extended version). *Journal of Cryptographic Engineering*, 5(1):31–52, April 2015. doi:10.1007/s13389-014-0085-7.

[FM19] Georgios Fotiadis and Chloe Martindale. Optimal TNFS-secure pairings on elliptic curves with composite embedding degree. ePrint 2019/555, 2019.

[Fot21] Georgios Fotiadis. Constructing efficient and STNFS–secure pairings. Slides at https://caramba.loria.fr/sem-slides/202102161400.pdf, February 2021. Talk at Inria Nancy seminar.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, April 2010. doi:10.1007/s00145-009-9048-z.

[GF16] Loubna Ghammam and Emmanuel Fouotsa. Adequate elliptic curves for computing the product of n pairings. In Sylvain Duquesne and Svetla Petkova-Nikova, editors, *Arithmetic of Finite Fields - 6th International Workshop, WAIFI 2016, Ghent, Belgium, July 13-15, 2016, Revised Selected Papers*, volume 10064 of *LNCS*, pages 36–53, 2016. ePrint 2016/472. doi:10.1007/978-3-319-55227-9_3.

[GG23] Jean Gasnier and Aurore Guillevic. An algebraic point of view on the generation of pairing-friendly curves. preprint available at https://hal.science/hal-04205681, September 2023.

[Gha16] Loubna Ghammam. *Utilisation des Couplages en Cryptographie asymétrique pour la micro-électronique*. thèse de doctorat (PhD thesis), Université de Rennes 1, France, 12 2016. https://hal.science/tel-01469981v1.

[GKL+21] Robert Granger, Thorsten Kleinjung, Arjen K. Lenstra, Benjamin Wesolowski, and Jens Zumbrägel. Computation of a 30750-bit binary field discrete logarithm. *Math. Comp.*, 90(332):2997–3022, 2021. ePrint 2020/965.

[GLS11] Steven D. Galbraith, Xibin Lin, and Michael Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. *Journal of Cryptology*, 24(3):446–469, July 2011. doi:10.1007/s00145-010-9065-y.

[GLV01] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 190–200. Springer, Berlin, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_11.

[GMT20] Aurore Guillevic, Simon Masson, and Emmanuel Thomé. Cocks–Pinch curves of embedding degrees five to eight and optimal ate pairing computation. *Des. Codes Cryptography*, 88:1047–1081, March 2020. ePrint 2019/431. doi:10.1007/s10623-020-00727-w.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_11.

[GS08] Steven D. Galbraith and Michael Scott. Exponentiation in pairing-friendly groups using homomorphisms. In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 211–224. Springer, Berlin, Heidelberg, September 2008. doi:10.1007/978-3-540-85538-5_15.

[GS10] Robert Granger and Michael Scott. Faster squaring in the cyclotomic subgroup of sixth degree extensions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 209–223. Springer, Berlin, Heidelberg, May 2010. doi:10.1007/978-3-642-13013-7_13.

[GS21] Aurore Guillevic and Shashank Singh. On the alpha value of polynomials in the tower number field sieve algorithm. *Mathematical Cryptology*, 1(1):1–39, Feb. 2021. URL: https://journals.flvc.org/mathcryptology/article/view/125142.

[Gui20] Aurore Guillevic. A short-list of pairing-friendly curves resistant to special TNFS at the 128-bit security level. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 535–564. Springer, Cham, May 2020. doi:10.1007/978-3-030-45388-6_19.

[Gui21] Aurore Guillevic. Pairing-friendly curves. https://members.loria.fr/AGuillevic/pairing-friendly-curves/, Feb 2021. Last updated February 22, 2021.

[HGP22] Youssef El Housni, Aurore Guillevic, and Thomas Piellard. Co-factor clearing and subgroup membership testing on pairing-friendly curves. In Lejla Batina and Joan Daemen, editors, *AFRICACRYPT 22*, volume 2022 of *LNCS*, pages 518–536. Springer, Cham, July 2022. doi:10.1007/978-3-031-17433-9_22.

[HHT20] Daiki Hayashida, Kenichiro Hayasaka, and Tadanori Teruya. Efficient final exponentiation via cyclotomic structure for pairings over families of elliptic curves. ePrint 2020/875, 2020.

[Hou23] Youssef El Housni. Pairings in rank-1 constraint systems. In Mehdi Tibouchi and Xiaofeng Wang, editors, *ACNS 23International Conference on Applied Cryptography and Network Security, Part I*, volume 13905 of *LNCS*, pages 339–362. Springer, Cham, June 2023. doi:10.1007/978-3-031-33488-7_13.

[HSST12] Takuya Hayashi, Takeshi Shimoyama, Naoyuki Shinohara, and Tsuyoshi Takagi. Breaking pairing-based cryptosystems using η_T pairing over GF(3^{97}). In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 43–60. Springer, Berlin, Heidelberg, December 2012. doi:10.1007/978-3-642-34961-4_5.

[HSV06] F. Hess, N.P. Smart, and F. Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006. ePrint 2006/110. doi:10.1109/TIT.2006.881709.

[Jou04] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, September 2004. doi:10.1007/s00145-004-0312-y.

[JT09] Marc Joye and Michael Tunstall. Exponent recoding and regular exponentiation algorithms. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 334–349. Springer, Berlin, Heidelberg, June 2009. doi:10.1007/978-3-642-02384-2_21.

[KB16] Taechan Kim and Razvan Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 543–571. Springer, Berlin, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_20.

[KIK+17] Yutaro Kiyomura, Akiko Inoue, Yuto Kawahara, Masaya Yasuda, Tsuyoshi Takagi, and Tetsutaro Kobayashi. Secure and efficient pairing at 256-bit security level. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17International Conference on Applied Cryptography and Network Security*, volume 10355 of *LNCS*, pages 59–79. Springer, Cham, July 2017. doi:10.1007/978-3-319-61204-1_4.

[KJ17] Taechan Kim and Jinhyuck Jeong. Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. In Serge Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 388–408. Springer, Berlin, Heidelberg, March 2017. doi:10.1007/978-3-662-54365-8_16.

[KM16] Neal Koblitz and Alfred Menezes. A riddle wrapped in an Enigma. *IEEE Security & Privacy*, 14(6):34–42, 2016. ePrint 2015/1018. doi:10.1109/MSP.2016.120.

[Kos24] Dmitrii Koshelev. Simultaneously simple universal and indifferentiable hashing to elliptic curves. ePrint 2024/085, Jan 2024.

[KSS08] Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 126–135. Springer, Berlin, Heidelberg, September 2008. doi:10.1007/978-3-540-85538-5_9.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Berlin, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_11.

[Lon23] Patrick Longa. Efficient algorithms for large prime characteristic fields and their application to bilinear pairings. *IACR TCHES*, 2023(3):445–472, 2023. doi:10.46586/tches.v2023.i3.445-472.

[LZZ24] Jianming Lin, Chang-An Zhao, and Yuhao Zheng. Efficient implementation of super-optimal pairings on curves with small prime fields at the 192-bit security level. ePrint 2024/1195, Jul 2024.

[Mas20] Simon Masson. *Algorithmic of curves in the context of bilinear and post-quantum cryptography*. Doctorat, Université de Lorraine, Nancy, France, December 2020. https://tel.archives-ouvertes.fr/tel-03052499.

[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001. https://dspace.jaist.ac.jp/dspace/bitstream/10119/4432/1/73-48.pdf.

[Mon05] P. L. Montgomery. Five, six, and seven-term Karatsuba-like formulae. *IEEE Transactions on Computer*, 54:362–369, March 2005. doi:10.1109/TC.2005.49.

[MSS16] Alfred Menezes, Palash Sarkar, and Shashank Singh. Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In Raphael C.-W. Phan and Moti Yung, editors, *Mycrypt Conference*, volume 10311 of *LNCS*, pages 83–108, Kuala Lumpur, Malaysia, December 1-2 2016. Springer. arXiv:2016/1102, doi:10.1007/978-3-319-61273-7_5.

[OLAR13] Thomaz Oliveira, Julio Cesar López-Hernández, Diego F. Aranha, and Francisco Rodríguez-Henríquez. Lambda coordinates for binary elliptic curves. In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES 2013*, volume 8086 of *LNCS*, pages 311–330. Springer, Berlin, Heidelberg, August 2013. doi:10.1007/978-3-642-40349-1_18.

[PS16] David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 111–126. Springer, Cham, February / March 2016. doi:10.1007/978-3-319-29485-8_7.

[RCB16] Joost Renes, Craig Costello, and Lejla Batina. Complete addition formulas for prime order elliptic curves. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 403–428. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49890-3_16.

[Rob22] Oisin Robinson. An implementation of the extended tower number field sieve using 4d sieving in a box and a record computation in $\mathbb{F}_{p^4}$, 2022. arXiv:2212.04999 https://arxiv.org/abs/2212.04999.

[Sch87] René Schoof. Nonsingular plane cubic curves over finite fields. *Journal of Combinatorial Theory, Series A*, 46(2):183–211, 1987. doi:10.1016/0097-3165(87)90003-3.

[Sco21] Michael Scott. A note on group membership tests for G_1, G_2 and G_T on BLS pairing-friendly curves. Cryptology ePrint Archive, Report 2021/1130, 2021. URL: https://eprint.iacr.org/2021/1130.

[SG18] Michael Scott and Aurore Guillevic. A new family of pairing-friendly elliptic curves. In Lilya Budaghyan and Francisco Rodríguez-Henríquez, editors, *Arithmetic of Finite Fields*, pages 43–57, Cham, 2018. Springer. ePrint 2018/193. doi:10.1007/978-3-030-05153-2_2.

[SKSW22] Yumi Sakemi, Tetsutaro Kobayashi, Tsunekazu Saito, and Riad S. Wahby. Pairing-friendly curves. https://datatracker.ietf.org/doc/draft-irtf-cfrg-pairing-friendly-curves/11/, Nov 2022. IETF draft.

[Smi15] Benjamin Smith. Easy scalar decompositions for efficient scalar multiplication on elliptic curves and genus 2 Jacobians. *Contemporary mathematics*, 637:15, May 2015. https://hal.inria.fr/hal-00874925. URL: https://hal.inria.fr/hal-00874925, doi:10.1090/conm/637/12753.

[TL23] Michael B. Jones Tobias Looker. Barreto–lynn–scott elliptic curve key representations for JOSE and COSE. https://datatracker.ietf.org/doc/draft-ietf-cose-bls-key-representations/, Oct 2023. IETF draft.

[TZ23] Stefano Tessaro and Chenzhi Zhu. Revisiting BBS signatures. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 691–721. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_24.

[Ver10] F. Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, Jan 2010. ePrint 2008/096. doi:10.1109/TIT.2009.2034881.

[WB19] Riad S. Wahby and Dan Boneh. Fast and simple constant-time hashing to the BLS12-381 elliptic curve. *IACR TCHES*, 2019(4):154–179, Aug. 2019. doi:10.13154/tches.v2019.i4.154-179.

A Discarded curves

In Table 18 we give the seeds for evaluating the families of pairing-friendly elliptic curves with embedding degree $k = 22$. We focus on Freeman–Scott–Teske (FST) families obtained by construction 6.3 (FST6.3) and construction 6.6 (FST6.6) in [FST10]. In addition we consider the family of curves of Gasnier–Guillevic [GG23], GG22 with CM-discriminant $D = 7$. Further, in Table 19 we present the formula of the optimal ate pairing for the three families of pairing-friendly curves with embedding degree 22. The formula of the easy part of the final exponentiation is given in Table 17.

Table 17: Exponents for the final exponentiation, easy and hard parts.

k	final exp. easy part	final exp. hard part
22	$(p^{11}-1)(p+1)$	$(p^{10}-p^9+p^8-p^7+p^6-p^5+p^4-p^3+p^2-p+1)/r$

Table 18: Additional families of pairing-friendly elliptic curves and seeds at 192-bit security.

k	curve	seed	(a,b)	$p \bmod k$	$\log p$	$\log r$	ρ	$\log p^k$	secu/ref
22	FST6.3	$2^{21}-2^{13}+2^6+2^3+1$	$(1,0)$	1	544	420	1.30	11965	194
	FST6.6	$2^{19}+2^{15}+2^{13}+2^{11}+2^3$	$(0,-2)$	1	534	383	1.40	11740	221
	GG22, $D=7$	$-2^{20}+2^{18}+2^{13}-2^{10}-2^8-2^2+1$	$(\frac{-5}{7},\frac{-2}{7})$	3	457	383	1.2	10052	220

Table 19: Optimal ate Miller loop formulas. The Miller functions $f_{u,Q}$ and lines $\ell_{Q,R}$ are evaluated at the point $P \in \mathbb{G}_1$.

k	curve	Equation (1)	Optimal ate formula
22	FST6.3	$u^2 - p \equiv 0 \bmod r$	$f_{u^2,Q}(P)$
	FST6.6	$u^2 - up^4 + p^8 \equiv 0 \bmod r$	$f_{u^2,Q}(P) \cdot f_{u,Q}(P)^{-p^4} \cdot \ell_{[u^2]Q,-[u]\pi_4(Q)}(P)$
	GG22	$u^2 - up + 2p^2 \equiv 0 \bmod r$	$f_{u^2,Q}(P) \cdot f_{u,Q}(P)^{-p} \cdot \ell_{[u^2]Q,-[u]\pi(Q)} \cdot \ell_{Q,Q}(P)^{2p^2}$

Table 20: Optimal ate pairing and final exponentiation cost estimates in terms of finite field multiplications. The name of each curve is derived from the name of the corresponding family in Table 5, plus the size of the prime p.

curve	p bits	r bits	Miller loop optimal ate	final exp easy	final exp hard	final exp total	pairing total
FST6.3-544, $k=22$	544	420	39.707**m**	789**m**	65.604**m**	66.393**m**	106.100**m**
FST6.6-534, $k=22$	534	**383**	33.955**m**	789**m**	64.200**m**	64.989**m**	98.944**m**
GG22-457, $D=7$	457	**383**	41.154**m**	789**m**	72.352**m**	73.141**m**	114.295**m**

Table A shows the number of $\mathbb{F}_p$-multiplications required for the Miller loop and the final exponentiation for the three types of curves with $k = 22$, FST6.3, FST6.6 and GG22. We discarded these curves from our study because they do not seem to be competitive candidates compared to the ones presented in Section 3. We note that such curves admit quadratic twists and hence the point addition and doubling operations in the Miller loop are executed over $\mathbb{F}_{p^{11}}$.

For group operations using GLV, the curve of j-invariant -3375 (discriminant $D = 7$) has an endomorphism ϕ of the form $(x, y) \mapsto (\phi_x(x), y\phi_y(x))$ with some $s = \sqrt{-7} \in \mathbb{F}_p$, of characteristic polynomial $\chi_\phi(X) = X^2 + X + 2$.

FST 6.3 $k = 22$. Here is a formula for the hard part of the final exponentiation on input m:

$$\begin{aligned}
&\bar{m} \leftarrow m^{p^{11}};\\
&a \leftarrow m^{u^2} \cdot \bar{m};\ f \leftarrow f^p \cdot a;\ a \leftarrow a^{u^2} \cdot m;\ f \leftarrow f^p \cdot a;\\
&a \leftarrow m^{u^2} \cdot \bar{m};\ f \leftarrow f^p \cdot a;\ a \leftarrow a^{u^2} \cdot m;\ f \leftarrow f^p \cdot a;\\
&a \leftarrow m^{u^2} \cdot \bar{m};\ f \leftarrow f^p \cdot a;\ a \leftarrow a^{u^2} \cdot m;\ f \leftarrow f^p \cdot a;\\
&a \leftarrow m^{u^2} \cdot \bar{m};\ f \leftarrow f^p \cdot a;\ a \leftarrow a^{u^2} \cdot m;\ f \leftarrow f^p \cdot a;\\
&a \leftarrow m^{u^2} \cdot \bar{m};\ f \leftarrow f^p \cdot a;\\
&f \leftarrow f^{u^2} \cdot f^{p^{11}};\ f \leftarrow f^{u^4} \cdot f^{p^{11}};\ f \leftarrow f \cdot m^4
\end{aligned}$$

which costs $24 \exp(u) + 2\mathbf{s}_{22} + 21\mathbf{m}_{22} + 9\mathbf{f}_{22}$.

FST 6.6 $k = 22$. Here is a formula for the hard part of the final exponentiation:

$$(u^3 - 1)^2((u^{10}(u + q^4) + uq^7 - 1)(u^3 + q)(u^6 + q^2) - u^{10}q^7) + 3(u + q^4)$$

which costs $26 \exp(u) + 11\mathbf{m}_{22} + \mathbf{s}_{22} + 6\mathbf{f}_{22}$.

GG22D7. We obtained a final exp hard formula which costs $22 \exp(u) + 20\mathbf{f}_{22} + 41\mathbf{m}_{22} + 50\mathbf{s}_{22}$.

B Formulas and parameters for group operations

In Table 21 we give polynomial description for families of curves considered in this paper. Specifically, this table lists the CM-discriminant for each type of curves, the polynomial $r(x)$ dividing the order of the curve, the polynomial representation of the cofactor $c_1(x)$, such that $E(\mathbb{F}_{p(x)}) = c_1(x)r(x)$, as well as the congruences that the seeds much satisfy in order for the polynomial family to produce integer values.

C Curves at the 256-bit security level

In Table 22 we give known instantiations in the literature for existing families of pairing-friendly elliptic curves aiming at 256-bit security level.

Table 21: Curve parameters, k is the embedding degree, r is the prime order of $\mathbb{G}_1$, c_1 is the cofactor so that the curve over $\mathbb{F}_p$ has order $r \cdot c_1$, and x is the seed.

k	D	Curve	$r(x)$	$c_1(x)$	x s.t. r, c_1 are integers
12	3	BLS12	$x^4 - x^2 + 1$	$(x-1)^2/3$	1 mod 3
15	3	BLS15	$x^8 - x^7 + x^5 - x^4 + x^3 - x + 1$	$(x-1)^2/3(x^2 + x + 1)$	1, 7 mod 15
		FM15	$x^8 - x^7 + x^5 - x^4 + x^3 - x + 1$	$x^2(3x^6 - 2x^4 - x^3 - 2x^2 + 3)/3$	0, 3 mod 15
16	1	KSS16	$(x^8 + 48x^4 + 625)/61250$	$125/2(x^2 + 2x + 5)$, $1250 \mid c_1$	25, 45 mod 70
		FM16	$\Phi_{16} = x^8 + 1$	$x^2/4(x^6 + 1)$	0 mod 2
		AFG16	$\Phi_{16} = x^8 + 1$	$x^2(x^3 + 1)^2/4$	any
18	3	KSS18	$(x^6 + 37x^3 + 343)/343$	$49/3(x^2 + 5x + 7)$, $343 \mid c_1$	14 mod 21
		SG18	$27x^6 + 9x^3 + 1$	$(3x^2 - 1)^2$	any
		FM18	$\Phi_{18} = x^6 - x^3 + 1$	$(3x^6 + x^2 - 2x + 1)/3$	1 mod 3
20	1	FM20	$\Phi_{20} = x^8 - x^6 + x^4 - x^2 + 1$	$(x-1)^2(x^2 + 1)/4$	1 mod 2
		SG20	$16x^8 + 16x^7 + 8x^6 - 4x^4 + 2x^2 + 2x + 1$	$2x^2(4x^4 - 3x^2 + 1)$	any
		GG20a	$(x^8 + 4x^7 + 11x^6 + 24x^5 + 41x^4 + 120x^3 + 275x^2 + 500x + 625)/(5^4 \cdot 41)$	$125(x^2 - 4x + 5) \cdot (x^2 - 2x + 5)/164$	$1715, 1815 \mod 2050$
		GG20b	$(x^8 - 4x^7 + 11x^6 - 24x^5 + 41x^4 - 120x^3 + 275x^2 - 500x + 625)/(5^4 \cdot 41)$	$125(x^2 + 4x + 5) \cdot (x^2 - 2x + 5)/164$	$1465, 1565 \mod 2050$
21	3	BLS21	$\Phi_{21} = x^{12} - x^{11} + x^9 - x^8 + x^6 - x^4 + x^3 - x + 1$	$(x-1)^2/3 (x^2 + x + 1)$, $9 \mid c_1$	1 mod 3
24	3	BLS24	$\Phi_{24} = x^8 - x^4 + 1$	$(x-1)^2/3$, $3 \mid c_1$	1 mod 3
27	3	BLS27	$\Phi_{27}/3 = (x^{18} + x^9 + 1)/3$	$(x-1)^2$, $3 \mid c_1$	1 mod 3
28	1	FST28	$\Phi_{28} = x^{12} - x^{10} + x^8 - x^6 + x^4 - x^2 + 1$	$(x-1)^2/4(x^2 + 1)$, $2 \mid c_1$	1 mod 2
		GG28	$(x^{12} + 4x^{11} + 11x^{10} + 24x^9 + 41x^8 + 44x^7 - 29x^6 + 220x^5 + 1025x^4 + 3000x^3 + 6875x^2 + 12500x + 15625)/29$	$(x^2 - 4x + 5) \cdot (x^2 - 2x + 5)/580$	$309, 449, 1759, 1899 \mod 2030$

Table 22: Curves at 256-bit security level from [BD19], [KIK+17, Table 6], and [BMDFAF19]

curve	seed	r (bits)	p (bits)	p^k (bits)
[BD19]				
KSS18	$2^{186} - 2^{75} - 2^{22} + 2^4$	1108	1484	26712
BLS24	$-2^{103} - 2^{101} + 2^{68} + 2^{50}$	827	1032	24768
[KIK+17]				
BLS24	$2^{109} - 2^{75} + 2^{65} - 1$	872	1089	26122
KSS32	$2^{49} - 2^{30} + 2^{18} + 2^{14} - 2^{12} - 2^2 - 1$	738	861	27536
KSS36	$-2^{58} + 2^{45} + 2^{40} + 2^{34} + 2^5$	669	798	28699
BLS42	$2^{43} - 2^8 + 2^2 - 1$	516	687	28830
BLS48	$-2^{32} - 2^{30} - 2^{10} + 2^7 - 1$	518	581	27851
[BMDFAF19]				
BLS48	$2^{32} - 2^{18} - 2^{10} - 2^4$	512	575	27600
SG54	$2^{27} + 2^{26} + 2^{22} + 2^{14} + 2^6 + 2$	512	569	30726