Exponentiating in Pairing Groups

Joppe W. Bos, Craig Costello(✉), and Michael Naehrig

Microsoft Research, Redmond, USA
{jbos,craigco,mnaehrig}@microsoft.com

Abstract. We study exponentiations in pairing groups for the most common security levels and show that, although the Weierstrass model is preferable for pairing computation, it can be worthwhile to map to alternative curve representations for the non-pairing group operations in protocols.

1 Introduction

At the turn of the century it was shown that elliptic curves can be used to build powerful cryptographic primitives: bilinear pairings [14,36,49]. *Pairings* are used in a large variety of protocols, and even when considering the recent breakthrough paper which shows how to instantiate multilinear maps using ideal lattices [26], pairings remain the preferred choice for a bilinear map due to their superior performance. Algorithms to compute cryptographic pairings involve computations on elements in all three pairing groups, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, but protocols usually require many additional standalone exponentiations in any of these three groups. In fact, protocols often compute only a single pairing but require many operations in any or all of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ [13,28,47]. In this work, we use such scenarios as a motivation to enhance the performance of group operations that are not the pairing computation.

Using non-Weierstrass models for elliptic curve group operations can give rise to significant speedups (cf. [9,10,31,43]). Such alternative models have not found the same success within pairing computations, since Miller's algorithm [42] not only requires group operations, but also relies on the computation of functions with divisors corresponding to these group operations. These functions are somewhat inherent in the Weierstrass group law, which is why Weierstrass curves remain faster for the pairings themselves [17]. Nevertheless, this does not mean that alternative curve models cannot be used to give speedups in the standalone group operations in pairing-based protocols. The purpose of this paper is to determine which curve models are applicable in the most popular pairing scenarios, and to report the speedups achieved when employing them. In order to obtain meaningful results, we have implemented curve arithmetic in different models that target the 128-, 192- and 256-bit security levels. Specifically, we have implemented group exponentiations and pairings on BN curves [4] (embedding degree $k = 12$), KSS curves [38] ($k = 18$) and BLS curves [3] ($k = 12$ and $k = 24$). We use GLV [25] and GLS [23] decompositions of dimensions 2, 4, 6 and 8 to speed up the scalar multiplication.

T. Lange, K. Lauter, and P. Lisoněk (Eds.): SAC 2013, LNCS 8282, pp. 438–455, 2014.
DOI: 10.1007/978-3-662-43414-7_22,

The goal of this work is *not* to set new software speed records, but to illustrate the improved performance that is possible from employing different curve models in the pairing groups $\mathbb{G}_1$ and $\mathbb{G}_2$. In order to provide meaningful benchmark results, we have designed our library using recoding techniques [21,29] such that all code runs in constant-time, i.e. the run-time of the code is independent of any secret input material. Our implementations use state-of-the-art algorithms for computations in the various groups [24] and for evaluating the pairing [2]. For any particular curve or security level, we assume that the *ratios* between our various benchmark results remain (roughly) invariant when implemented for different platforms or when the bottleneck arithmetic functions are converted to assembly. We therefore believe that our table of timings provides implementers and protocol designers with good insight as to the relative computational expense of operating in pairing groups versus computing the pairing(s).

2 Preliminaries

A cryptographic pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map that relates the three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, each of prime order r. These groups are defined as follows. For distinct primes p and r, let k be the smallest positive integer such that $r \mid p^k - 1$. Assume that $k > 1$. For an elliptic curve $E/\mathbb{F}_p$ such that $r \mid \#E(\mathbb{F}_p)$, we can choose $\mathbb{G}_1 = E(\mathbb{F}_p)[r]$ to be the order-r subgroup of $E(\mathbb{F}_p)$. We have $E[r] \subset E(\mathbb{F}_{p^k})$, and $\mathbb{G}_2$ can be taken as the (order-r) subgroup of $E(\mathbb{F}_{p^k})$ of p-eigenvectors of the p-power Frobenius endomorphism on E. Let $\mathbb{G}_T$ be the group of r-th roots of unity in $\mathbb{F}^*_{p^k}$. The *embedding degree* k is very large (i.e. $k \approx r$) for general curves, but must be kept small (i.e. $k < 50$) if computations in $\mathbb{F}_{p^k}$ are to be feasible in practice – this means that so-called *pairing-friendly* curves must be constructed in a special way. In Sect. 2.1 we recall the best known techniques for constructing such curves with embedding degrees that target the 128-, 192- and 256-bit security levels – k is varied to optimally balance the size of r and the size of $\mathbb{F}_{p^k}$, which respectively determine the complexity of the best known elliptic curve and finite field discrete logarithm attacks.

2.1 Parameterized Families of Pairing-Friendly Curves with Sextic Twists

The most suitable pairing-friendly curves for our purposes come from parameterized families, such that the parameters to find a suitable curve $E(\mathbb{F}_p)$ can be written as univariate polynomials. For the four families we consider, we give below the polynomials $p(x)$, $r(x)$ and $t(x)$, where $t(x)$ is such that $n(x) = p(x)+1-t(x)$ is the cardinality of the desired curve, which has $r(x)$ as a factor. All of the curves found from these constructions have j-invariant zero, which means they can be written in Weierstrass form as $y^2 = x^3 + b$. Instances of these pairing-friendly families can be found by searching through integer values x of an appropriate size until we find $x = x_0$ such that $p = p(x_0)$ and $r = r(x_0)$ are simultaneously

prime, at which point we can simply test different values for b until the curve $E : y^2 = x^3 + b$ has an n-torsion point.

To target the 128-bit security level, we use the BN family [4] ($k = 12$), for which

$$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1, t(x) = 6x^2 + 1, r(x) = p(x) + 1 - t(x). \quad (1)$$

At the 192-bit security level, we consider BLS curves [3] with $k = 12$, for which

$$p(x) = (x-1)^2(x^4 - x^2 + 1)/3 + x, \quad t(x) = x + 1, \quad r(x) = x^4 - x^2 + 1, \quad (2)$$

where $x \equiv 1 \bmod 3$, and KSS curves [38] with $k = 18$, which are given by

$$\begin{aligned} p(x) &= (x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 + 259x^3 + 343x^2 + 1763x + 2401)/21, \\ t(x) &= (x^4 + 16x + 7)/7, \quad r(x) = (x^6 + 37x^3 + 343)/7^3, \end{aligned} \quad (3)$$

with $x \equiv 14 \bmod 42$. At the 256-bit security level, we use curves from the BLS family [3] with embedding degree $k = 24$, which have the parametrization

$$p(x) = (x-1)^2(x^8 - x^4 + 1)/3 + x, \quad t(x) = x + 1, \quad r(x) = x^8 - x^4 + 1, \quad (4)$$

with $x \equiv 1 \bmod 3$.

For the above families, which all have $k = 2^i 3^j$, the best practice to construct the full extension field $\mathbb{F}_{p^k}$ is to use a tower of (intermediate) quadratic and cubic extensions [5,40]. Since $6 \mid k$, we can always use a *sextic twist* $E'(\mathbb{F}_{p^{k/6}})$ to represent elements of $\mathbb{G}_2 \subset E(\mathbb{F}_{p^k})[r]$ as elements of an isomorphic group $\mathbb{G}_2' = E'(\mathbb{F}_{p^{k/6}})[r]$. This shows that group operations in $\mathbb{G}_2$ can be performed on points with coordinates in an extension field with degree one sixth the size, which is the best we can do for elliptic curves [50, Proposition X.5.4].

In all cases considered in this work, the most preferable sextic extension from $\mathbb{F}_{p^{k/6}} = \mathbb{F}_p(\xi)$ to $\mathbb{F}_{p^k} = \mathbb{F}_{p^{k/6}}(z)$ is constructed by taking $z \in \mathbb{F}_{p^k}$ as a root of the polynomial $z^6 - \xi$, which is irreducible in $\mathbb{F}_{p^{k/6}}[z]$. We describe the individual towers in the four cases as follows: the BN and BLS cases with $k = 12$ preferably take $p \equiv 3 \bmod 4$, so that $\mathbb{F}_{p^2}$ can be constructed as $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 + 1)$, and take $\xi = u + 1$ for the sextic extension to $\mathbb{F}_{p^{12}}$. For $k = 18$ KSS curves, we prefer that 2 is not a cube in $\mathbb{F}_p$, so that $\mathbb{F}_{p^3}$ can be constructed as $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^3 + 2)$, before taking $\xi = u$ to extend to $\mathbb{F}_{p^{18}}$. For $k = 24$ BLS curves, we again prefer to construct $\mathbb{F}_{p^2}$ as $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 + 1)$, on top of which we take $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}[v]/(v^2 - (u+1))$ (it is easily shown that $v^2 - u$ cannot be irreducible [18, Proposition 1]), and use $\xi = v$ for the sextic extension. All of these constructions agree with the towers used in the "speed-record" literature [1,2,18,48].

2.2 The GLV and GLS Algorithms

The GLV [25] and GLS [23] methods both use an efficient endomorphism to speed up elliptic curve scalar multiplications. The GLV method relies on endomorphisms specific to the shape of the curve E that are unrelated to the Frobenius endomorphism. On the other hand, the GLS method works over extension

fields where Frobenius becomes non-trivial, so it does not rely on E having a special shape. However, if E is both defined over an extension field and has a special shape, then the two can be combined [23, Sect. 3] to give higher-dimensional decompositions, which can further enhance performance.

Since in this paper we have $E/\mathbb{F}_p : y^2 = x^3 + b$ and $p \equiv 1 \bmod 3$, we can use the GLV endomorphism $\phi : (x, y) \mapsto (\zeta x, y)$ in $\mathbb{G}_1$ where $\zeta^3 = 1$ and $\zeta \in \mathbb{F}_p \setminus \{1\}$. In this case ϕ satisfies $\phi^2 + \phi + 1$ in the endomorphism ring $\mathrm{End}(E)$ of E, so on $\mathbb{G}_1$ it corresponds to scalar multiplication by λ_ϕ, where $\lambda_\phi^2 + \lambda_\phi + 1 \equiv 0 \bmod r$, meaning we get a 2-dimensional decomposition in $\mathbb{G}_1$. Since $\mathbb{G}_2'$ is always defined over an extension field herein, we can combine the GLV endomorphism above with the Frobenius map to get higher-dimensional GLS decompositions. The standard way to do this in the pairing context [24] is to use the untwisting isomorphism Ψ to move points from $\mathbb{G}_2'$ to $\mathbb{G}_2$, where the p-power Frobenius π_p can be applied (since E is defined over $\mathbb{F}_p$, while E' is not), before using the twisting isomorphism Ψ^{-1} to move this result back to $\mathbb{G}_2'$. We define ψ as $\psi = \Psi^{-1} \circ \pi_p \circ \Psi$, which (even though Ψ and Ψ^{-1} are defined over $\mathbb{F}_{p^k}$) can be explicitly described over $\mathbb{F}_{p^{k/6}}$. The GLS endomorphism ψ satisfies $\Phi_k(\psi) = 0$ in $\mathrm{End}(E')$ [24, Lemma 1], where $\Phi_k(\cdot)$ is the k-th cyclotomic polynomial, so it corresponds to scalar multiplication by λ_ψ, where $\Phi_k(\lambda_\psi) \equiv 0 \bmod r$, i.e. λ_ψ is a primitve k-th root of unity modulo r. For the curves with $k = 12$, we thus obtain a 4-dimensional decomposition in $\mathbb{G}_2' \subset E'(\mathbb{F}_{p^2})$; for $k = 18$ curves, we get a 6-dimensional decomposition in $\mathbb{G}_2' \subset E'(\mathbb{F}_{p^3})$; and for $k = 24$ curves, we get an 8-dimensional decomposition in $\mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$.

To compute the scalar multiple $[s]P_0$, a $d = 2$ dimensional GLV or $d = \varphi(k)$ dimensional GLS decomposition starts by computing the $d - 1$ additional points $P_i = \psi^i(P_0) = \psi(P_{i-1}) = [\lambda_\psi^i]P_0$, $1 \le i \le d - 1$. One then seeks a vector $(\hat{s}_0, \hat{s}_1) \in \mathbb{Z}^2$ in the "GLV lattice" L_ϕ that is close to $(s, 0) \in \mathbb{Z}^2$, or $(\hat{s}_0, \ldots, \hat{s}_{\varphi(k)-1}) \in \mathbb{Z}^{\varphi(k)}$ in the "GLS lattice" L_ψ that is close to $(s, 0, \ldots, 0) \in \mathbb{Z}^{\varphi(k)}$. The bases B_ϕ and B_ψ (for L_ϕ and L_ψ) are given as (see [22, p. 229–230])

$$B_\phi = \begin{pmatrix} r & 0 \\ -\lambda_\phi & 1 \end{pmatrix}; \qquad B_\psi = \begin{pmatrix} r & 0 \ldots 0 \\ -\lambda_\psi & 1 \ldots 0 \\ \vdots & \vdots \ddots \vdots \\ -\lambda_\psi^{d-1} & 0 \ldots 1 \end{pmatrix}. \tag{5}$$

Finding close vectors in these lattices is particularly easy in the case of BLS $k = 12$ and $k = 24$ curves [24, Example 3,4]. For BN curves, we can use the special routine described by Galbraith and Scott [24, Example 5], which bears resemblance to the algorithm proposed in [46], which is what we use for the GLS decomposition on KSS curves.

To obtain the d mini-scalars $s_0, \ldots, s_{d-1}$ from the scalar s and the close vector $(\hat{s}_0, \ldots, \hat{s}_{d-1})$, we compute $(s_0, \ldots s_{d-1}) = (s, 0, \ldots, 0) - (\hat{s}_0, \ldots, \hat{s}_{d-1})$ in $\mathbb{Z}^d$. We can then compute $[s]P_0$ via the multi-exponentiation $\sum_{i=0}^{d-1} [s_i]P_i$. The typical way to do this is to start by making all of the s_i positive: we simultaneously

negate any (s_i, P_i) pair for which $s_i < 0$ (this can be done in a side-channel resistant way using bitmasks). We then precompute all possible sums $\sum_{i=0}^{d-1}[b_i]P_i$, for the 2^d combinations of $b_i \in \{0, 1\}$, and store them in a lookup table. When simultaneously processing the j-th bits of the d mini-scalars, this allows us to update the running value with only one point addition, before performing a single point doubling. In each case however, this standard approach requires individual attention for further optimization – this is what we describe in Sect. 3.

We aim to create constant-time programs: implementations which have an execution time independent of any secret material (e.g. the scalar). This means that we always execute exactly the same amount of point additions and duplications independent of the input. In order to achieve this in the setting of scalar multiplication using the GLV/GLS method, we use the recoding techniques from [21,29]. This recoding technique not only guarantees that the program performs a constant number of point operations, but that the recoding itself is done in constant time as well. Furthermore, an advantage of this method is that the lookup table size is reduced by a factor of two, since we only store lookup elements for which the multiple of the first point P_0 is odd. Besides reducing the memory, this reduces the time to create the lookup table.

3 Strategies for GLV in $\mathbb{G}_1$ and GLS in $\mathbb{G}_2$

This section presents our high-level strategy for 2-GLV on $\mathbb{G}_1$, 4-GLS in $\mathbb{G}_2$ in the two $k = 12$ families, 6-GLS in $\mathbb{G}_2$ for the KSS curves with $k = 18$, and 8-GLS in $\mathbb{G}_2$ for the BLS curves with $k = 24$. We use the following abbreviations for elliptic curve operations that we require: `DBL` – for the doubling of a projective point, `ADD` – for the addition between two projective points, `MIX` – for the addition between a projective point and an affine point, and `AFF` – for the addition between two affine points to give a projective point.

3.1 2-GLV on $\mathbb{G}_1$

For the 2-GLV routines we compute the multi-exponentiation $[s_0]P_0 + [s_1]P_1$. Recoding our mini-scalars and proceeding in the naive way would give a lookup table consisting of two elements: P_0 and $P_0 + P_1$. However, the number of point additions can be further reduced by using a large window size [16] (see [23,24] for a description in the context of GLV/GLS). Specifically, we can reduce the number of point additions in the scalar processing phase by a factor of w if we generate a lookup table of size 2^{2w-1}. Since computing an element in the lookup table costs roughly one addition, one can compute the optimal window size given the maximum size of the mini-scalars (see Table 3). For 2-GLV in $\mathbb{G}_1$, we found a fixed window size of $w = 3$ to be optimal in all cases except BN curves (where we use $w = 2$ due to the smaller maximum size of the mini-scalars). In Algorithms 1 and 2 we give the algorithms for computing the 2-GLV

Algorithm 1. Generating the lookup table for 2-GLV with window size $w = 2$ (cost: 6 MIX + 1 AFF + 1 DBL).

Input: $P_0, P_1 \in \mathbb{G}_1$.
Output: The 2-GLV lookup table, T, for window size $w = 2$.

$t_0 \leftarrow \texttt{DBL}(P_0)$, $T[0] \leftarrow P_0$, $T[1] \leftarrow \texttt{MIX}(t_0, P_0)$,
$T[2] \leftarrow \texttt{AFF}(P_0, P_1)$, $T[3] \leftarrow \texttt{MIX}(T[1], P_1)$, $T[4] \leftarrow \texttt{MIX}(T[2], P_1)$,
$T[5] \leftarrow \texttt{MIX}(T[3], P_1)$, $T[6] \leftarrow \texttt{MIX}(T[4], P_1)$, $T[7] \leftarrow \texttt{MIX}(T[5], P_1)$.

lookup tables using $w = 2$ and $w = 3$, respectively. Algorithm 1 outlines how to compute $T[\lfloor \frac{a}{2} \rfloor + 2 \cdot b] = [a]P_0 + [b]P_1$ for $a \in \{1, 3\}$ and $b \in \{0, 1, 2, 3\}$, where T consists of eight elements. Algorithm 2 computes $T[\lfloor \frac{a}{2} \rfloor + 4 \cdot b] = [a]P_0 + [b]P_1$ for $a \in \{1, 3, 5, 7\}$ and $b \in \{0, 1, 2, 3, 4, 5, 6, 7\}$, where T consists of 32 elements.

For both BLS families and the KSS family, we get a simple GLV scalar decomposition and obtain the mini-scalars by writing s as a linear function in λ_ϕ. This has the additional advantage that both s_0 and s_1 are positive. For BN curves, we use the algorithm from [46] for the decomposition. In this setting, the mini-scalars can be negative, so we must ensure that they become positive (see Sect. 2.2) before using Algorithm 1 to generate the lookup table.

3.2 4-GLS on $\mathbb{G}_2$ for BN and BLS Curves with $k = 12$

In the BLS case, we have $\lambda_\psi(x) = x$, which means $|\lambda_\psi| \approx r^{1/4}$, so we get a 4-dimensional decomposition in $\mathbb{G}_2$ by writing the scalar $0 \leq s < r$ in base $|\lambda_\psi|$ as $s = \sum_{i=0}^{3} s_i |\lambda_\psi|^i$, with $0 \leq s_i < |\lambda_\psi|$ [24, Example 3]. On the other hand, the mini-scalars resulting from the decomposition on BN curves in [24, Example 5] can be negative.

Deciding on the best window size for 4-GLS is trivial since a window size of $w = 2$ requires a lookup table of 128 entries, where generating each entry costs an addition. This is far more than the number of additions saved from using this larger window. In Algorithm 3, we state how to generate the lookup table for $w = 1$ of size eight, which consists of the elements $T[\sum_{i=1}^{3} b_i 2^{i-1}] = P_0 + \sum_{i=1}^{3} [b_i] P_i$, for all combinations of $b_i \in \{0, 1\}$.

Algorithm 2. Generating the lookup table for 2-GLV with window size $w = 3$ (cost: 29 MIX + 2 ADD + 1 DBL).

Input: $P_0, P_1 \in \mathbb{G}_1$.
Output: The 2-GLV lookup table, T, for window size $w = 3$.

$t_0 \leftarrow \texttt{DBL}(P_0)$, $T[0] \leftarrow P_0$, $T[1] \leftarrow \texttt{MIX}(t_0, P_0)$,
$T[2] \leftarrow \texttt{ADD}(t_0, T[1])$, $T[3] \leftarrow \texttt{ADD}(t_0, T[2])$,
for $i = 1$ to 7 **do**
  for $j = 0$ to 3 **do**
    $T[4i + j] \leftarrow \texttt{MIX}(T[4(i-1) + j], P_1)$

Algorithm 3. Generating the lookup table for 4-GLS with window size $w = 1$ (cost: 4 MIX + 3 AFF).

Input: $P_0, P_1, P_2, P_3 \in \mathbb{G}_2$.
Output: The 4-GLS lookup table T.

$T[0] \leftarrow P_0$, $T[1] \leftarrow \texttt{AFF}(T[0], P_1)$, $T[2] \leftarrow \texttt{AFF}(T[0], P_2)$,
$T[3] \leftarrow \texttt{MIX}(T[1], P_2)$, $T[4] \leftarrow \texttt{AFF}(T[0], P_3)$, $T[5] \leftarrow \texttt{MIX}(T[1], P_3)$,
$T[6] \leftarrow \texttt{MIX}(T[2], P_3)$, $T[7] \leftarrow \texttt{MIX}(T[3], P_3)$.

3.3 6-GLS on $\mathbb{G}_2$ for KSS Curves with $k = 18$

To decompose the scalar for 6-GLS on $\mathbb{G}_2$ for KSS curves, we use the technique[1] from [46], after which we must ensure all the s_i are non-negative according to Sect. 2.2. In this case, the decision of the window size (being $w = 1$) is again trivial, since a window of size $w = 2$ requires a lookup table of size 2^{11}. On input of P_i corresponding to $s_i > 0$, for $0 \leq i \leq 5$, we generate the 32 elements of the lookup table as follows. We use Algorithm 3 to produce $T[0], \ldots, T[7]$ (using $P_0, \ldots, P_3$). We compute $T[8] \leftarrow \texttt{AFF}(T[0], P_4)$ and $T[i] \leftarrow \texttt{MIX}(T[i-8], P_4)$ for $9 \leq i \leq 15$. Next, we compute $T[16] \leftarrow \texttt{AFF}(T[0], P_5)$ and $T[i] \leftarrow \texttt{MIX}(T[i-16], P_5)$ for $17 \leq i \leq 31$.

3.4 8-GLS on $\mathbb{G}_2$ for BLS Curves $k = 24$

BLS curves with $k = 24$ have $\lambda_\psi(x) = x$, which means $|\lambda_\psi| \approx r^{1/8}$, so one can compute an 8-dimensional decomposition in $\mathbb{G}_2$ by writing the scalar $0 \leq s < r$ in base $|\lambda_\psi|$ as $s = \sum_{i=0}^{7} s_i |\lambda_\psi|^i$, with $0 \leq s_i < |\lambda_\psi|$ [24, Example 4]. We use the 8-dimensional decomposition strategy studied in [15]: the idea is to split the lookup table (a single large lookup table would consist of 128 entries) into two lookup tables consisting of eight elements each. In this case, we need to compute twice the amount of point additions when simultaneously processing the mini-scalars (see Table 3), but we save around 120 point additions in generating the lookup table(s). Let T_1 be the table consisting of the 8 entries $P_0 + \sum_{i=1}^{3} [b_i] P_i$, for $b_i \in \{0, 1\}$, which is generated using Algorithm 3 on $P_0, \ldots, P_3$. The second table, T_2, consists of the 8 entries $P_4 + \sum_{i=5}^{7} [b_i] P_i$ for $b_i \in \{0, 1\}$, and can be pre-computed as $T_2[j] \leftarrow \psi^4(T_1[j])$, for $j = 0, \ldots, 7$. With the specific tower construction for $k = 24$ BLS curves (see Sect. 2.1), the map $\psi^4 : \mathbb{G}_2 \rightarrow \mathbb{G}_2$ significantly simplifies to $\psi^4 : (x, y) \mapsto (c_x x, c_y y)$, where the constants c_x and c_y are in $\mathbb{F}_p$.

[1] We note that for particular KSS $k = 18$ curves, large savings may arise in this algorithm due to the fact that the $\alpha = \sum_{i=0}^{5} a_i \psi^i$ (from Sect. 5.2 of [46]) have some of the a_i being zero. In the case of the KSS curve we use, around 2/3 of the computations vanish due to $a_2 = a_4 = a_5 = 0$ and $a_1 = 1$.

4 Alternate Curve Models for Exponentiations in Groups $\mathbb{G}_1$ and $\mathbb{G}_2$

An active research area in ECC involves optimizing elliptic curve arithmetic through the use of various curve models and coordinate systems (see [9,31] for an overview). For example, in ECC applications the fastest arithmetic to realize a group operation on Weierstrass curves of the form $y^2 = x^3 + b$ requires 16 field multiplications [9], while a group addition on an Edwards curve can incur as few as 8 field multiplications [33]. While alternative curve models are not favorable over Weierstrass curves in the pairing computation itself [17], they can still be used to speed up the elliptic curve operations in $\mathbb{G}_1$ and $\mathbb{G}_2$.

4.1 Three Non-Weierstrass Models

Unlike the general Weierstrass model which covers all isomorphism classes of elliptic curves over a particular field, the non-Weierstrass elliptic curves usually only cover a subset of all such classes. Whether or not an elliptic curve E falls into the classes covered by a particular model is commonly determined by the existence of a Weierstrass point with a certain order on E. In the most popular scenarios for ECC, these orders are either 2, 3 or 4. In this section we consider the fastest model that is applicable in the pairing context in each of these cases.

- $\mathcal{W}$ - **Weierstrass**: all curves in this paper have j-invariant zero and Weierstrass form $y^2 = x^3 + b$. The fastest formulas on such curves use Jacobian coordinates [8].
- $\mathcal{J}$ - **Extended Jacobi quartic**: if an elliptic curve has a point of order 2, then it can be written in (extended) Jacobi quartic form as $\mathcal{J}\colon y^2 = dx^4 + ax^2+1$ [11, Sect. 3] – these curves were first considered for cryptographic use in [11, Sect. 3]. The fastest formulas work on the corresponding projective curve given by $\mathcal{J}\colon Y^2Z^2 = dX^4 + aX^2Z^2 + Z^4$ and use the 4 extended coordinates $(X\colon Y\colon Z\colon T)$ to represent a point, where $x = X/Z$, $y = Y/Z$ and $T = X^2/Z$ [34].
- $\mathcal{H}$ - **Generalized Hessian**: if an elliptic curve (over a finite field) has a point of order 3, then it can be written in generalized Hessian form as $\mathcal{H}\colon x^3+y^3+c = dxy$ [20, Theorem 2]. The authors of [37,51] studied Hessian curves of the form $x^3 + y^3 + 1 = dxy$ for use in cryptography, and this was later generalized to include the parameter c [20]. The fastest formulas for `ADD/MIX/AFF` are from [7] while the fastest `DBL` formulas are from [32] – they work on the homogeneous projective curve given by $\mathcal{H}\colon X^3 + Y^3 + cZ^3 = dXYZ$, where $x = X/Z$, $y = Y/Z$. We note that the j-invariant zero version of $\mathcal{H}$ has $d = 0$ (see Sect. 4.3), so in Table 1 we give updated costs that include this speedup.
- $\mathcal{E}$ - **Twisted Edwards**: if an elliptic curve has a point of order 4, then it can be written in twisted Edwards form as $\mathcal{E}\colon ax^2 + y^2 = 1 + dx^2y^2$ [6, Theorem 3.3]. However, if the field of definition, K, has $\#K \equiv 1 \bmod 4$, then $4 \mid E$ is enough to write E in twisted Edwards form [6, Sect. 3] (i.e. we do not necessarily need a point of order 4). Twisted Edwards curves [19] were introduced to cryptography in [6,10] and the best formulas are from [33].

Table 1. The costs of necessary operations for computing group exponentiations on four models of elliptic curves. Costs are reported as $\mathbf{T}_{\mathbf{M},\mathbf{S},\mathbf{d},\mathbf{a}}$, where **M** is the cost of a field multiplication, **S** is the cost of a field squaring, **d** is the cost of multiplication by a curve constant, **a** is the cost of a field addition (we have counted multiplications by 2 as additions), and **T** is the total number of multiplications, squarings, and multiplications by curve constants.

Model/ coords	Requires	DBL cost	ADD cost	MIX cost	AFF cost
$\mathcal{W}$/Jac.	-	$\mathbf{7}_{2,5,0,14}$	$\mathbf{16}_{11,5,0,13}$	$\mathbf{11}_{7,4,0,14}$	$\mathbf{6}_{4,2,0,12}$
$\mathcal{J}$/ext.	pt. of order 2	$\mathbf{9}_{1,7,1,12}$	$\mathbf{13}_{7,3,3,19}$	$\mathbf{12}_{6,3,3,18}$	$\mathbf{11}_{5,3,3,18}$
$\mathcal{H}$/proj.	pt. of order 3	$\mathbf{7}_{6,1,0,11}$	$\mathbf{12}_{12,0,0,3}$	$\mathbf{10}_{10,0,0,3}$	$\mathbf{8}_{8,0,0,3}$
$\mathcal{E}$/ext.	pt. of order 4, or $4 \mid E$ and $\#K \equiv 1 \bmod 4$	$\mathbf{9}_{4,4,1,7}$	$\mathbf{10}_{9,0,1,7}$	$\mathbf{9}_{8,1,0,7}$	$\mathbf{8}_{7,0,1,7}$

For each model, we summarize the cost of the required group operations in Table 1. The total number of field multiplications are reported in bold for each group operation – this includes multiplications, squarings and multiplications by constants. We note that in the context of plain ECC these models have been studied with small curve constants; in pairing-based cryptography, however, we must put up with whatever constants we get under the transformation to the non-Weierstrass model. The only exception we found in this work is for the $k = 12$ BLS curves, where $\mathbb{G}_1$ can be transformed to a Jacobi quartic curve with $a = -1/2$, which gives a worthwhile speedup [34].

4.2 Applicability of Alternative Curve Models for $k \in \{12, 18, 24\}$

In this section we prove the existence or non-existence of points of orders 2, 3 and 4 in the groups $E(\mathbb{F}_p)$ and $E'(\mathbb{F}_{p^{k/6}})$ for the pairing-friendly families considered in this work. These proofs culminate in Table 2, which summarizes the alternative curve models that are available for $\mathbb{G}_1$ and $\mathbb{G}_2$ in the scenarios we consider. We can study $\#E(\mathbb{F}_p)$ directly from the polynomial parameterizations in Sect. 2.1, while for $\#E'(\mathbb{F}_{p^e})$ (where $e = k/6$) we do the following. With the explicit recursion in [12, Corollary VI.2] we determine the parameters t_e and f_e which are related by the CM equation $4p^e = t_e^2 + 3f_e^2$ (since all our curves have CM discriminant $D = -3$). This allows us to compute the order of the correct sextic twist, which by [30, Proposition 2] is one of $n'_{e,1} = p^e + 1 - (3f_e + t_e)/2$ or $n'_{e,2} = p^e + 1 - (-3f_e + t_e)/2$. For $k = 12$ and $k = 24$ BLS curves, we assume that $p \equiv 3 \bmod 4$ so that $\mathbb{F}_{p^2}$ can be constructed (optimally) as $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 + 1)$. Finally, since $p \equiv 3 \bmod 4$, $E(\mathbb{F}_p)$ must contain a point of order 4 if we are to write E in twisted Edwards form; however, since E' is defined over $\mathbb{F}_{p^e}$, if e is even then $4 \mid E'$ is enough to write E' in twisted Edwards form (see Sect. 4.1).

Proposition 1. *Let $E/\mathbb{F}_p$ be a BN curve with sextic twist $E'/\mathbb{F}_{p^2}$. The groups $E(\mathbb{F}_p)$ and $E'(\mathbb{F}_{p^2})$ do not contain points of order 2, 3 or 4.*

Proof. From (1) we always have $\#E(\mathbb{F}_p) \equiv 1 \bmod 6$. Remark 2.13 of [44] shows that we have $\#E'(\mathbb{F}_{p^2}) = (p+1-t)(p-1+t)$, which from (1) gives that $\#E'(\mathbb{F}_{p^2}) \equiv 1 \bmod 6$. □

Proposition 2. *For $p \equiv 3$ mod 4, let $E/\mathbb{F}_p$ be a $k = 12$ BLS curve with sextic twist $E'/\mathbb{F}_{p^2}$. The group $E(\mathbb{F}_p)$ contains a point of order 3 and can contain a point of order 2, but not 4, while the group $E'(\mathbb{F}_{p^2})$ does not contain a point of order 2, 3 or 4.*

Proof. From [12, Corollary VI.2] we have $t_2(x) = t(x)^2 - 2p(x)$, which with (2) and $4p(x)^2 = t_2(x)^2 + 3f_2(x)^2$ allows us to deduce that the correct twist order is $n'_{2,2}$, which gives $n'_{2,2}(x) \equiv 1 \bmod 12$ for $x \equiv 1 \bmod 3$, i.e. E' does not have points of order 2, 3 or 4. For E, (2) reveals that $3 \mid \#E$, and furthermore that $x \equiv 4 \bmod 6$ implies $\#E$ is odd, while for $x \equiv 1 \bmod 6$ we have $4 \mid \#E$. The assumption $p \equiv 3 \bmod 4$ holds if and only if $x \equiv 7 \bmod 12$, which actually implies $p \equiv 7 \bmod 12$. Now, to have a point of order 4 on $E/\mathbb{F}_p : y^2 = x^3 + b$, the fourth division polynomial $\psi_4(x) = 2x^6 + 40bx^3 - -8b^2$ must have a root $\alpha \in \mathbb{F}_p$, which happens if and only if $\alpha^3 = -10b \pm 6b\sqrt{3}$. However, [35, Sect. 5, Theorem 2-(b)] says that 3 is a quadratic residue in $\mathbb{F}_p$ if and only if $p \equiv \pm b^2 \bmod 12$, where b is co-prime to 3, which cannot happen for $p \equiv 7 \bmod 12$, so E does not have a point of order 4. □

Proposition 3. *Let $E/\mathbb{F}_p$ be a $k = 18$ KSS curve with sextic twist $E'/\mathbb{F}_{p^3}$. The group $E(\mathbb{F}_p)$ does not contain a point of order 2, 3 or 4, while the group $E'(\mathbb{F}_{p^3})$ contains a point of order 3 but does not contain a point of order 2 or 4.*

Proof. From [12, Corollary VI.2] we have $t_3(x) = t(x)^3 - 3p(x)t(x)$. With (3) and $4p(x)^3 = t_3(x)^2 + 3f_3(x)^2)$ it follows that $n'_{3,1}(x)$ is the correct twist order. We have $n'_{3,1}(x) \equiv 3 \bmod 12$ for $x \equiv 14 \bmod 42$, i.e. E' has a point of order 3 but no points of order 2 or 4. For E we have $\#E \equiv 1 \bmod 6$ from (3), which means there are no points of order 2, 3, or 4. □

Proposition 4. *For $p \equiv 3$ mod 4, let $E/\mathbb{F}_p$ be a BLS curve with $k = 24$ and sextic twist $E'/\mathbb{F}_{p^4}$. The group $E(\mathbb{F}_p)$ can contain points of order 2 or 3 (although not simultaneously), but not 4, while the group $E'(\mathbb{F}_{p^4})$ can contain a point of order 2, but does not contain a point of order 3 or 4.*

Proof. Again, [12, Corollary VI.2] gives $t_4(x) = t(x)^4 - 4p(x)t(x)^2 + 2p(x)^2$, and from (4) and $4p(x)^4 = t_4(x)^2 + 3f_4(x)^2$ we get $n'_{4,1}(x)$ as the correct twist order. For $x \equiv 1 \bmod 6$ we have $n'_{4,1}(x) \equiv 1 \bmod 12$ (so no points of order 2, 3, or 4), while for $x \equiv 4 \bmod 6$ we have $n'_{4,1}(x) \equiv 4 \bmod 12$. Recall from the proof of Proposition 2 that $(\alpha, \beta) \in E'(\mathbb{F}_{p^4})$ is a point of order 4 if we have $\alpha \in \mathbb{F}_{p^4}$ such that $\alpha^3 = (-10 \pm 6\sqrt{3})b'$. The curve equation gives $\beta^2 = (-9 \pm 6\sqrt{3})b'$, i.e. b' must be a square in $\mathbb{F}_{p^4}$, which implies that $(0, \pm\sqrt{b'})$ are points of order 3 on $E'(\mathbb{F}_{p^4})$, which contradicts $n'_{4,1}(x) \equiv 1 \bmod 3$. Thus, $E'(\mathbb{F}_{p^4})$ cannot have points of order 3 or 4. For E, from (4) we have $\#E(\mathbb{F}_p) \equiv 3 \bmod 12$ if $x \equiv 4 \bmod 6$, but $\#E \equiv 0 \bmod 12$ if $x \equiv 1 \bmod 6$. Thus, there is a point of order 3 on E,

Table 2. Optional curve models for $\mathbb{G}_1$ and $\mathbb{G}_2$ in popular pairing implementations.

	$\mathbb{G}_1$		$\mathbb{G}_2$		
Family-k	Algorithm	Models avail.	Algorithm	Models avail.	Follows from
BN-12	2-GLV	$\mathcal{W}$	4-GLS	$\mathcal{W}$	Proposition 1
BLS-12	2-GLV	$\mathcal{H}, \mathcal{J}, \mathcal{W}$	4-GLS	$\mathcal{W}$	Proposition 2
KSS-18	2-GLV	$\mathcal{W}$	6-GLS	$\mathcal{H}, \mathcal{W}$	Proposition 3
BLS-24	2-GLV	$\mathcal{H}, \mathcal{J}, \mathcal{W}$	8-GLS	$\mathcal{E}, \mathcal{J}, \mathcal{W}$	Proposition 4

as well as a point of order 2 if $x \equiv 1 \bmod 6$. So it remains to check whether there is a point of order 4 when $x \equiv 1 \bmod 6$. Taking $x \equiv 1 \bmod 12$ gives rise to $p \equiv 1 \bmod 4$, so take $x \equiv 7 \bmod 12$. This implies that $p \equiv 7 \bmod 12$, and the same argument as in the proof of Proposition 2 shows that there is no point of order 4. □

In Table 2 we use the above propositions to summarize which (if any) of the non-Weierstrass models from Sect. 4.1 can be applied to our pairing scenarios.

4.3 Translating Endomorphisms to the Non-Weierstrass Models

In this section we investigate whether the GLV and GLS endomorphisms from Sect. 2.2 translate to the Jacobi quartic and Hessian models. Whether the endomorphisms translate desirably depends on how efficiently they can be computed on the non-Weierstrass model. It is not imperative that the endomorphisms do translate desirably, but it can aid efficiency: if the endomorphisms are not efficient on the alternative model, then our exponentiation routine also incurs the cost of passing points back and forth between the two models – this cost is small but could be non-negligible for high-dimensional decompositions. On the other hand, if the endomorphisms are efficient on the non-Weierstrass model, then the groups $\mathbb{G}_1$ and/or $\mathbb{G}_2$ can be defined so that all exponentiations take place directly on this model, and the computation of the pairing can be modified to include an initial conversion back to Weierstrass form.

We essentially show that the only scenario in which the endomorphisms are efficiently computable on the alternative model is the case of the GLV endomorphism ϕ on Hessian curves.

Endomorphisms on the Hessian Model. We modify the maps given in [20, Sect. 2.2] to the special case of j-invariant zero curves, where we have $d = 0$ on the Hessian model. Assume that $(0 : \pm\alpha : 1)$ are points of order 3 on $\mathcal{W}: Y^2Z = X^3 + \alpha^2 Z^3$, which is birationally equivalent to $\mathcal{H}: U^3 + V^3 + 2\alpha Z^3 = 0$. We define the constants $h_0 = \zeta - 1$, $h_1 = \zeta + 2$, $h_2 = -2(2\zeta + 1)\alpha$, where $\zeta^3 = 1$ and $\zeta \neq 1$. The map $\tau : \mathcal{W} \to \mathcal{H}$, $(X : Y : Z) \mapsto (U : V : W)$ is given as

$$U \leftarrow h_0 \cdot (Y + \alpha Z) + h_2 \cdot Z, \qquad V \leftarrow -U - 3(Y + \alpha Z), \qquad W \leftarrow 3X, \quad (6)$$

where $\tau(0\colon \pm\alpha\colon 1) = \mathcal{O} \in \mathcal{H}$. The inverse map $\tau^{-1} : \mathcal{H} \to \mathcal{W}$, $(U\colon V\colon W) \mapsto (X\colon Y\colon Z)$ is

$$X \leftarrow h_2 \cdot W, \qquad Z \leftarrow h_0 \cdot V + h_1 \cdot U, \qquad Y \leftarrow -h_2 \cdot (U+V) - \alpha \cdot Z. \quad (7)$$

It follows that the GLV endomorphism $\phi_{\mathcal{W}} \in \mathrm{End}(\mathcal{W})$ translates into $\phi_{\mathcal{H}} \in \mathrm{End}(\mathcal{H})$, where $\phi_{\mathcal{W}} : (X\colon Y\colon Z) \mapsto (\zeta X\colon Y\colon Z)$ becomes $\phi_{\mathcal{H}} : (U\colon V\colon W) \mapsto (U\colon V\colon \zeta W)$. However, we note that when computing $\phi_{\mathcal{H}}$ on an affine point, it can be advantageous to compute $\phi_{\mathcal{H}}$ as $\phi_{\mathcal{H}} : (u\colon v\colon 1) \mapsto (\zeta^2 u\colon \zeta^2 v\colon 1)$, where ζ^2 is the (precomputed) other cube root of unity, which produces an affine result.

For GLS on Hessian curves, there is no obvious or simple way to perform the analogous untwisting or twisting isomorphisms directly between $\mathcal{H}'(\mathbb{F}_{p^{k/6}})$ and $\mathcal{H}(\mathbb{F}_{p^k})$, which suggests that we must pass back and forth to the Weierstrass curve/s to determine the explicit formulas for the GLS endomorphism on $\mathcal{H}'$. The composition of these maps $\psi_{\mathcal{H}'} = \tau \circ \Psi_{\mathcal{W}}^{-1} \circ \pi_p \circ \Psi_{\mathcal{W}} \circ \tau^{-1}$ does not appear to simplify to be anywhere near as efficient as the GLS endomorphism is on the Weierstrass curve. Consequently, our GLS routine will start with a Weierstrass point in $\mathcal{W}'(\mathbb{F}_{p^{k/6}})$, where we compute $d-1$ applications of $\psi \in \mathrm{End}(\mathcal{W}')$, before using (6) to convert the d points to $\mathcal{H}'(\mathbb{F}_{p^{k/6}})$, where the remainder of the routine takes place (save the final conversion back to $\mathcal{W}'$). Note that since we are converting affine Weierstrass points to $\mathcal{H}'$ via (6), this only incurs two multiplications each time. However, the results are now projective points on $\mathcal{H}'$ meaning that the more expensive full addition formulas must be used to generate the remainder of the lookup table.

Endomorphisms on the Jacobi Quartic Model. Unlike the Hessian model where the GLV endomorphism was efficient, for the Jacobi quartic model it appears that neither the GLV nor GLS endomorphisms translate to be of a similar efficiency as they are on the Weierstrass model. Thus, in all cases where Jacobi quartic curves are a possibility, we start and finish on $\mathcal{W}$, and only map to $\mathcal{J}$ after computing all applications of ϕ or ψ on the Weierstrass model. We adapt the maps given in [31, p. 17] to our special case as follows. Let $(-\theta\colon 0\colon 1)$ be a point of order 2 on $\mathcal{W} : Y^2Z = X^3 + \theta^3 Z^3$ and let $a = 3\theta/4$ and $d = -3\theta^2/16$. The curve $\mathcal{W}$ is birationally equivalent to the (extended) Jacobi quartic curve $\mathcal{J} : V^2W^2 = dU^4 + 2aU^2W^2 + W^4$, with the map $\tau : \mathcal{W} \to \mathcal{J}$, $\tau\colon (X\colon Y\colon Z) \mapsto (U\colon V\colon W)$ given as

$$U \leftarrow 2YZ, \quad W \leftarrow X^2 - X\theta Z + \theta^2 Z^2, \quad V \leftarrow 6XZ\theta + W - 4aZ(\theta Z + X), \quad (8)$$

where $\tau((-\theta\colon 0\colon 1)) = (0\colon -1\colon 1) \in \mathcal{J}$. The inverse map $\tau^{-1} : \mathcal{J} \to \mathcal{W}$, $\tau^{-1}\colon (U\colon V\colon W) \mapsto (X\colon Y\colon Z)$, is given by

$$X \leftarrow (2V+2)U + 2aU^3 - \theta U^3, \qquad Y \leftarrow (4V+4) + 4aU^2, \qquad Z \leftarrow U^3, \quad (9)$$

where $\tau^{-1}((0\colon -1\colon 1)) = (-\theta\colon 0\colon 1) \in \mathcal{W}$ and the neutral point on $\mathcal{J}$ is $\mathcal{O}_{\mathcal{J}} = (0\colon 1\colon 1)$.

Endomorphisms on the Twisted Edwards Model. Similarly to the Jacobi-quartic model, endomorphisms on $\mathcal{E}$ are not nearly as efficiently computable as they are on $\mathcal{W}$, so we only pass across to $\mathcal{E}$ after the endomorphisms are applied on $\mathcal{W}$. Here we give the back-and-forth maps that are specific to our case(s) of interest. Namely, since we are unable to use twisted Edwards curves over the ground field (see Table 2), let $\mathcal{W}/\mathbb{F}_{p^e} : Y^2Z = X^3 + b'Z^3$ for $p \equiv 3 \bmod 4$ and e being even. Since we have a point of order 2 on $\mathcal{W}$, i.e. $(\alpha : 0 : 1)$ with $\alpha = \sqrt[3]{-b'} \in \mathbb{F}_{p^e}$, then take $s = 1/(\alpha\sqrt{3}) \in \mathbb{F}_{p^e}$. The twisted Edwards curve $\mathcal{E} : aU^2W^2 + V^2W^2 = W^4 + dU^2V^2$ with $a = (3\alpha s + 2)/s$ and $d = (3\alpha s - 2)/s$ is isomorphic to $\mathcal{W}$, with the map $\tau : \mathcal{W} \to \mathcal{E}, (X : Y : Z) \mapsto (U : V : W)$ given as

$$U \leftarrow s(X - \alpha Z)(sX - s\alpha Z + Z),\ V \leftarrow sY(sX - s\alpha Z - Z),\ W \leftarrow sY(sX - s\alpha Z + Z),$$

with inverse map $\tau : \mathcal{E} \to \mathcal{W}, (U : V : W) \mapsto (X : Y : Z)$, given as

$$X \leftarrow -U(-W - V - \alpha s(W - V)), \quad Y \leftarrow (W + V)W, \quad Z \leftarrow sU(W - V).$$

4.4 Curve Choices for Pairings at the 128-, 192- and 256-bit Security Levels

The specific curves we choose in this section can use any of the alternative models that are available in the specific cases as shown in Table 2. The only exception occurs for $k = 24$, for which we are forced to choose between having a point of order 2 or 3 (see Proposition 4) in $\mathbb{G}_1$ – we opt for the point of order 3 and the Hessian model, as this gives enhanced performance. Note that these curves do not sacrifice any efficiency in the pairing computation compared to previously chosen curves in the literature (in terms of the field sizes, hamming-weights and towering options).

The $k = 12$ BN Curve. Since no alternative models are available for the BN family, we use the curve that was first seen in [45] and subsequently used to achieve speed records at the 128-bit security level [2], which results from substituting $x = -(2^{62} + 2^{55} + 1)$ into (1), and taking $E/\mathbb{F}_p : y^2 = x^3 + 2$ and $E'/\mathbb{F}_{p^2} : y^2 = x^3 + (1 - u)$, where $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 + 1)$.

The $k = 12$ BLS Curve. Setting $x = 2^{106} - 2^{72} + 2^{69} - 1$ in (2) gives a 635-bit prime p and a 424-bit prime r. Let $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 + 1)$ and let $\xi = u + 1$. The Weierstrass forms corresponding to $\mathbb{G}_1$ and $\mathbb{G}_2$ are $\mathcal{W}/\mathbb{F}_p : y^2 = x^3 + 1$ and $\mathcal{W}'/\mathbb{F}_{p^2} : y^2 = x^3 + \xi$. Only $\mathbb{G}_1$ has options for alternative models (see Table 2): the Hessian curve $\mathcal{H}/\mathbb{F}_p : x^3 + y^3 + 2 = 0$ and the Jacobi quartic curve $\mathcal{J}/\mathbb{F}_p : y^2 = \frac{-3}{16}x^4 + \frac{3}{4}x^2 + 1$ are both isomorphic to $\mathcal{W}$ over $\mathbb{F}_p$.

The $k = 18$ KSS Curve. Setting $x = 2^{64} - 2^{51} + 2^{47} + 2^{28}$ in (3) gives a 508-bit prime p and a 376-bit prime r. Let $\mathbb{F}_{p^3} = \mathbb{F}_p[u]/(u^3 + 2)$. The Weierstrass forms for $\mathbb{G}_1$ and $\mathbb{G}_2$ are $\mathcal{W}/\mathbb{F}_p : y^2 = x^3 + 2$ and $\mathcal{W}'/\mathbb{F}_{p^3} : y^2 = x^3 - u^2$. Only $\mathbb{G}_2$ allows for an alternative model (see Table 2): the Hessian curve $\mathcal{H}'/\mathbb{F}_{p^3} : x^3 + y^3 + 2u\sqrt{-1} = 0$ is isomorphic to $\mathcal{W}'$ over $\mathbb{F}_{p^3}$.

The $k = 24$ BLS Curve. Setting $x = 2^{63} - 2^{47} + 2^{38}$ in (3) gives a 629-bit prime p and a 504-bit prime r. Let $\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2+1)$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}[v]/(v^2-(u+1))$. The Weierstrass forms corresponding to $\mathbb{G}_1$ and $\mathbb{G}_2$ are $\mathcal{W}/\mathbb{F}_p : y^2 = x^3 + 4$ and $\mathcal{W}'/\mathbb{F}_{p^4} : y^2 = x^3 + 4v$. This gives us the option of a Hessian model in $\mathbb{G}_1$: the curve $\mathcal{H}/\mathbb{F}_p : x^3 + y^3 + 4 = 0$ is isomorphic to $\mathcal{W}$ over $\mathbb{F}_p$. In $\mathbb{G}_2$ we have both the Jacobi quartic and twisted Edwards models as options. Let $\theta = (u+1)v$ and set $a = -3\theta/4$ and $d = (4A^2 - 3\theta^2)/4$. The curve $\mathcal{J}/\mathbb{F}_{p^4} : y^2 = dx^4 + ax^2 + 1$ is isomorphic to $\mathcal{W}'$ over $\mathbb{F}_{p^4}$. For the twisted Edwards model, we take $\alpha = \theta = (u+1)v$, $s = 1/(\alpha\sqrt{3}) \in \mathbb{F}_{p^4}$, $a' = (3\alpha s + 2)/s$ and $d' = (3\alpha s - 2)/s$; the curve $\mathcal{E}/\mathbb{F}_{p^4} : a'x^2 + y^2 = 1 + d'x^2y^2$ is then isomorphic to W'.

5 Exponentiations in $\mathbb{G}_T$

For the scenarios in this paper, Galbraith and Scott [24] remark that the best known method for exponentiations in $\mathbb{G}_T \subset \mathbb{F}_{p^k}$ is to use the same $\varphi(k)$-dimensional decomposition that is used for GLS in $\mathbb{G}_2$. This means the same techniques for multi-exponentiation can be applied directly. The recoding technique (see Sect. 2.2) also carries across analogously, since inversions of $\mathbb{G}_T$-elements (which are conjugations over $\mathbb{F}_{p^{k/2}}$) are almost for free [24, Sect. 7], just as in the elliptic curve groups. For example, while the GLS map ψ on curves with $k = 12$ gives $\psi^4(Q') - \psi^2(Q') + Q' = \mathcal{O}$ for all $Q' \in \mathbb{G}'_2$, in $\mathbb{G}_T$ we use the p-power Frobenius map π_p, which gives $f \cdot \pi_p^4(f)/\pi_p^2(f) = 1$ for all $f \in \mathbb{G}_T$. Finally, $\mathbb{G}_T$ is contained in the cyclotomic subgroup of $\mathbb{F}^*_{p^k}$, in which much faster squarings are available [27,39]. The optimal choices of window sizes for the multi-exponentiation in $\mathbb{G}_T$ remain equal to those in $\mathbb{G}_2$ (see Sect. 3).

6 Results

In Table 3 we summarize the optimal curve choices in each scenario. We first note that Jacobi quartic curves were unable to outperform the Weierstrass, Hessian or twisted Edwards curves in any of the scenarios. This is because the small number of operations saved in a Jacobi quartic group addition were not enough to outweigh the slower Jacobi quartic doublings (see Table 1), and because of

Table 3. Optimal scenarios for group exponentiations. For both GLV on $\mathbb{G}_1$ and GLS on $\mathbb{G}_2$ in all four families, we give the decomposition dimension d, the maximum sizes of the mini-scalars $||s_i||_\infty$, the optimal window size w, and the optimal curve model.

Sec. level	Family-k	Exp. in $\mathbb{G}_1$				Exp. in $\mathbb{G}_2$			
		d	$\|\|s_i\|\|_\infty$	w	Curve	d	$\|\|s_i\|\|_\infty$	w	Curve
128-bit	BN-12	2	128	2	Weierstrass	4	64	1	Weierstrass
192-bit	BLS-12	2	212	3	Hessian	4	106	1	Weierstrass
	KSS-18	2	192	3	Weierstrass	6	63	1	Hessian
256-bit	BLS-24	2	252	3	Hessian	8	63	1	twisted Edwards

Table 4. Benchmark results for an optimal ate pairing and group exponentiations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ in millions (M) of clock cycles for the best curve models. These results have been obtained on an Intel Core i7-3520M CPU averaged over thousands of random instances.

Sec. level	Family-k	Pairing e	Exp. in $\mathbb{G}_1$	Exp. in $\mathbb{G}_2$	Exp. in $\mathbb{G}_T$
128-bit	BN-12	7.0	0.9 ($\mathcal{W}$)	1.8 ($\mathcal{W}$)	3.1
192-bit	BLS-12	47.2	4.4 ($\mathcal{H}$)	10.9 ($\mathcal{W}$)	17.5
	KSS-18	63.3	3.5 ($\mathcal{W}$)	9.8 ($\mathcal{H}$)	15.7
256-bit	BLS-24	115.0	5.2 ($\mathcal{H}$)	27.6 ($\mathcal{E}$)	47.1

the extra computation incurred by the need to pass back and forth between $\mathcal{J}$ and $\mathcal{W}$ to compute the endomorphisms (see Sect. 4.3). On the other hand, while employing the Hessian and twisted Edwards forms also requires us to pass back and forth to compute the endomorphisms, the group law operations on these models are significantly faster than Weierstrass operations across the board, so Hessian and twisted Edwards curves reigned supreme whenever they were able to be employed – we give the concrete comparisons below. In Table 3 we also present the bounds we used on the maximum sizes of the mini-scalars resulting from a d-dimensional decomposition. In some cases, like those where decomposing s involves writing s in base λ_ϕ or λ_ψ, these bounds are trivially tight. However, in both the GLV and GLS on BN curves, and in the GLS on KSS curves, the bounds presented are those we obtained experimentally from hundreds of millions of scalar decompositions, meaning that the theoretical bounds could be a few bits larger – determining such bounds could be done using similar techniques to those in [41].

In Table 4 we present our timings for pairing computations and exponentiations in the three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, for the four families considered. We note that for 2-GLV on $k = 12$ BLS curves, Hessian curves gave a factor 1.23 speedup over Weierstrass curves (4.4M versus 5.4M cycles); for 6-GLS on $k = 18$ KSS curves, using Hessian curves gave a factor 1.11 speedup (9.8M versus 10.9M cycles); for 2-GLV on $k = 24$ BLS curves, Hessian curves gave a factor 1.19 speedup (5.2M versus 6.2M cycles); lastly, for 8-GLS on $k = 24$ BLS curves, twisted Edwards curves gave a factor 1.16 speedup (27.6M versus 31.9M cycles). The Hessian and twisted Edwards timings include the conversion from the Weierstrass model after the endomorphisms have been computed, and to the Weierstrass model at the end of the scalar multiplication routine.

In [1] it was first proposed to use $k = 12$ BLS curves for the 192-bit security level, by showing that pairings on these curves are significantly faster than pairings on $k = 18$ KSS curves. Our pairing timings add further weight to their claim. However, our timings also show that KSS curves are slightly faster for exponentiations in all three groups. There are many circumstances where Table 4 could guide implementers to make more efficient decisions when deploying a protocol. As one example, we refer to Boneh and Franklin's original identity-based encryption scheme [14, Sect. 4.1], where the sender computes a pairing between a public element P_{pub} and an identities' public key Q_{ID}, i.e. the sender computes

$g_{\mathtt{ID}} = e(P_{\text{pub}}, Q_{\mathtt{ID}})$. The sender then chooses a random exponent s and computes $g_{\mathtt{ID}}^s$ (which is hashed to become part of a ciphertext). In this case Table 4 shows that the sender would be much better off computing the scalar multiplication $[s]P_{\text{pub}}$ (assuming $P_{\text{pub}} \in \mathbb{G}_1$, or else we could compute $[s]Q_{\mathtt{ID}}$) before computing the pairing $e([s]P_{\text{pub}}, Q_{\mathtt{ID}}) = g_{\mathtt{ID}}^s$.

Acknowledgment. We thank the reviewer who pointed out that having $4 \mid \#E(K)$ and $\#K \equiv 1 \bmod 4$ is sufficient to write E/K in twisted Edwards form.

References

1. Aranha, D.F., Fuentes-Castañeda, L., Knapp, E., Menezes, A., Rodríguez-Henríquez, F.: Implementing pairings at the 192-bit security level. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 177–195. Springer, Heidelberg (2013)
2. Aranha, D.F., Karabina, K., Longa, P., Gebotys, C.H., López, J.: Faster explicit formulas for computing pairings over ordinary curves. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 48–68. Springer, Heidelberg (2011)
3. Barreto, P.S.L.M., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 257–267. Springer, Heidelberg (2003)
4. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006)
5. Benger, N., Scott, M.: Constructing tower extensions of finite fields for implementation of pairing-based cryptography. In: Hasan, M.A., Helleseth, T. (eds.) WAIFI 2010. LNCS, vol. 6087, pp. 180–195. Springer, Heidelberg (2010)
6. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 389–405. Springer, Heidelberg (2008)
7. Bernstein, D.J., Kohel, D., Lange, T.: Twisted Hessian curves. http://www.hyperelliptic.org/EFD/g1p/auto-hessian-standard.html#addition-add-2001-jq
8. Bernstein, D.J., Lange, T.: Analysis and optimization of elliptic-curve single-scalar multiplication. In: Mullen, G., Panario, D., Shparlinski, I. (eds.) Finite Fields and Applications. Contemporary Mathematics Series, vol. 461, pp. 1–20. AMS, Providence (2007)
9. Bernstein, D.J., Lange, T.: Explicit-formulas database. http://www.hyperelliptic.org/EFD (2007)
10. Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 29–50. Springer, Heidelberg (2007)
11. Billet, O., Joye, M.: The Jacobi model of an elliptic curve and side-channel analysis. In: Fossorier, M.P.C., Høholdt, T., Poli, A. (eds.) AAECC 2003. LNCS, vol. 2643, pp. 34–42. Springer, Heidelberg (2003)
12. Blake, I., Seroussi, G., Smart, N.: Elliptic Curves in Cryptography, vol. 265. Cambridge University Press, New York (1999)
13. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)

14. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
15. Bos, J.W., Costello, C., Hisil, H., Lauter, K.: High-performance scalar multiplication using 8-dimensional GLV/GLS decomposition. In: Bertoni, G., Coron, J.-S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 331–348. Springer, Heidelberg (2013)
16. Brauer, A.: On addition chains. Bull. Am. Math. Soc. **45**, 736–739 (1939)
17. Costello, C., Lange, T., Naehrig, M.: Faster pairing computations on curves with high-degree twists. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 224–242. Springer, Heidelberg (2010)
18. Costello, C., Lauter, K., Naehrig, M.: Attractive subfamilies of BLS curves for implementing high-security pairings. In: Bernstein, D.J., Chatterjee, S. (eds.) INDOCRYPT 2011. LNCS, vol. 7107, pp. 320–342. Springer, Heidelberg (2011)
19. Edwards, H.M.: A normal form for elliptic curves. Bull. Am. Math. Soc. **44**(3), 393–422 (2007)
20. Farashahi, R.R., Joye, M.: Efficient arithmetic on Hessian curves. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 243–260. Springer, Heidelberg (2010)
21. Faz-Hernandez, A., Longa, P., Sanchez, A.H.: Efficient and secure algorithms for GLV-based scalar multiplication and their implementation on GLV-GLS curves. Cryptology ePrint Archive, Report 2013/158. http://eprint.iacr.org/ (2013). CT-RSA 2014, DOI:10.1007/978-3-319-04852-9_1
22. Galbraith, S.D.: Mathematics of Public Key Cryptography. Cambridge University Press, Cambridge (2012)
23. Galbraith, S.D., Lin, X., Scott, M.: Endomorphisms for faster elliptic curve cryptography on a large class of curves. J. Cryptol. **24**(3), 446–469 (2011)
24. Galbraith, S.D., Scott, M.: Exponentiation in pairing-friendly groups using homomorphisms. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 211–224. Springer, Heidelberg (2008)
25. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 190–200. Springer, Heidelberg (2001)
26. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)
27. Granger, R., Scott, M.: Faster squaring in the cyclotomic subgroup of sixth degree extensions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 209–223. Springer, Heidelberg (2010)
28. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340. Springer, Heidelberg (2010)
29. Hamburg, M.: Fast and compact elliptic-curve cryptography. Cryptology ePrint Archive, Report 2012/309. http://eprint.iacr.org/ (2012)
30. Hess, F., Smart, N.P., Vercauteren, F.: The Eta pairing revisited. IEEE Trans. Inf. Theor. **52**(10), 4595–4602 (2006)
31. Hisil, H.: Elliptic curves, group law, and efficient computation. Ph.D. thesis, Queensland University of Technology (2010)
32. Hisil, H., Carter, G., Dawson, E.: New formulae for efficient elliptic curve arithmetic. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 138–151. Springer, Heidelberg (2007)

33. Hisil, H., Wong, K.K.-H., Carter, G., Dawson, E.: Twisted Edwards curves revisited. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 326–343. Springer, Heidelberg (2008)
34. Hisil, H., Wong, K.K.-H., Carter, G., Dawson, E.: Jacobi quartic curves revisited. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 452–468. Springer, Heidelberg (2009)
35. Ireland, K., Rosen, M.: A Classical Introduction to Modern Number Theory. Graduate Texts in Mathematics, vol. 84. Springer, New York (1990)
36. Joux, A.: A one round protocol for tripartite Diffie-Hellman. J. Cryptol. **17**(4), 263–276 (2004)
37. Joye, M., Quisquater, J.-J.: Hessian elliptic curves and side-channel attacks. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 402–410. Springer, Heidelberg (2001)
38. Kachisa, E.J., Schaefer, E.F., Scott, M.: Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 126–135. Springer, Heidelberg (2008)
39. Karabina, K.: Squaring in cyclotomic subgroups. Math. Comput. **82**(281), 555–579 (2013)
40. Koblitz, N., Menezes, A.: Pairing-based cryptography at high security levels. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 13–36. Springer, Heidelberg (2005)
41. Longa, P., Sica, F.: Four-dimensional Gallant-Lambert-Vanstone scalar multiplication. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 718–739. Springer, Heidelberg (2012)
42. Miller, V.S.: The Weil pairing, and its efficient calculation. J. Cryptol. **17**(4), 235–261 (2004)
43. Montgomery, P.L.: Speeding the Pollard and elliptic curve methods of factorization. Math. Comput. **48**(177), 243–264 (1987)
44. Naehrig, M.: Constructive and computational aspects of cryptographic pairings. Ph.D. thesis, Eindhoven University of Technology (2009)
45. Nogami, Y., Akane, M., Sakemi, Y., Kato, H., Morikawa, Y.: Integer variable χ–based Ate pairing. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 178–191. Springer, Heidelberg (2008)
46. Park, Y.-H., Jeong, S., Lim, J.-I.: Speeding up point multiplication on hyperelliptic curves with efficiently-computable endomorphisms. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 197–208. Springer, Heidelberg (2002)
47. Parno, B., Gentry, C., Howell, J., Raykova, M.: Pinocchio: nearly practical verifiable computation. In: Proceedings of the IEEE Symposium on Security and Privacy. IEEE (2013)
48. Pereira, G.C.C.F., Simplício Jr, M.A., Naehrig, M., Barreto, P.S.L.M.: A family of implementation-friendly BN elliptic curves. J. Syst. Softw. **84**(8), 1319–1326 (2011)
49. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, pp. 135–148 (2000)
50. Silverman, J.H.: The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, vol. 106, 2nd edn. Springer, New York (2009)
51. Smart, N.P.: The Hessian form of an elliptic curve. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 118–125. Springer, Heidelberg (2001)