Families of SNARK-friendly 2-chains of elliptic curves

Youssef El Housni, Aurore Guillevic

Families of SNARK-friendly 2-chains of elliptic curves

Youssef El Housni[1,2,3][0000−0003−2873−3479]
and Aurore Guillevic[4,5][0000−0002−0824−7273]

[1] ConsenSys zkTeam, Paris, France
[2] LIX, CNRS, École Polytechnique, Institut Polytechnique de Paris
[3] Inria
youssef.elhousni@consensys.net
[4] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
[5] Aarhus University, Aarhus, Denmark
aurore.guillevic@inria.fr

Abstract. At CANS'20, El Housni and Guillevic introduced a new 2-chain of pairing-friendly elliptic curves for recursive zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs) made of the former BLS12-377 curve (a Barreto–Lynn–Scott curve over a 377-bit prime field) and the new BW6-761 curve (a Brezing–Weng curve of embedding degree 6 over a 761-bit prime field). First we generalise the curve construction, the pairing formulas ($e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$) and the group operations to any BW6 curve defined on top of any BLS12 curve, forming a family of 2-chain pairing-friendly curves.
Second, we investigate other possible 2-chain families made on top of the BLS12 and BLS24 curves. We compare BW6 to Cocks–Pinch curves of higher embedding degrees 8 and 12 (CP8, CP12) at the 128-bit security level. We explicit an optimal ate and optimal Tate pairing on our new CP curves. We show that both for BLS12 and BLS24, the BW6 construction always gives the fastest pairing and curve arithmetic compared to Cocks-Pinch curves. Finally, we suggest a short list of curves suitable for Groth16 and KZG-based universal SNARKs and present an optimized implementation of these curves. Based on Groth16 and PlonK (a KZG-based SNARK) implementations, we obtain that the BLS12-377/BW6-761 pair is optimized for the former while the BLS24-315/BW6-672 pair is optimized for the latter.

1 Introduction

A SNARK [39,44,26,10] is a cryptographic primitive that enables a prover to prove to a verifier the knowledge of a satisfying witness to a non-deterministic (NP) statement by producing a proof π such that the size of π and the cost to verify it are both sub-linear in the size of the witness. If π does not reveal anything about the witness we refer to the cryptographic primitive as a zero-knowledge (zk) SNARK. Today, the most efficient SNARKs require pairing-friendly elliptic curves and trusted setup assumptions as in Groth'16 [29] but in return admit

small, constant-size proofs with a constant-time verification. However, the trusted setup is specific to the NP statement to prove. Hence, Groth'16 is not suitable in applications that need to prove many different statements. Fortunately, SNARKs with a universal or transparent setup are an active area of research and recent polynomial-commitment-based constructions allow very efficient constructions. The most efficient universal constructions such as PlonK [23] and Marlin [14] are based on the KZG polynomial commitment [38], which also requires a pairing-friendly elliptic curve.

A pairing-friendly curve E has a bilinear map $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}_1, \mathbb{G}_2$ are distinct prime-order r subgroups of E, and $\mathbb{G}_T \subset \mathbb{F}_{q^k}$ of the same order r. On the one hand, one requires two different kind of curves: a curve tailored for Groth'16 should be optimized for operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and for pairings while a curve tailored for KZG-based SNARKs should only focus on $\mathbb{G}_1$ and pairings. On the other hand, both constructions make SNARKs appealing for an incrementally verifiable computation (IVC) [50] in which proofs not only attest to the correct execution of a computation but also, by exploiting succinctness, to the validity of a previous proof. The canonical construction of IVC, or proof-carrying data [11] (PCD) as a generalization, is achieved via recursive proof composition which was demonstrated to be practical for pairing-based SNARKs in [8]. In such a setting, a prover encodes the statement in the curve's scalar field $\mathbb{F}_r$ (the $\mathbb{G}_i$ are of order r) and a verifier checks the proof π in an extension $\mathbb{F}_{q^k}$ of the curve base field. To allow recursive proof composition, one needs to encode the verification algorithm (which lies in $\mathbb{F}_{q^k}$) as a statement in $\mathbb{F}_r$. However, this is highly impractical as $r \neq q$ and simulating one field's operations in the other incurs a significant overhead. The authors of [8] sidestep this issue by constructing a 2-cycle of pairing-friendly elliptic curves such that the base field of either curve is the scalar field of the other. Unfortunately, only the MNT4/MNT6 [21, Sec. 5] family of pairing-friendly curves is known to satisfy this property and due to their low embedding degrees, secure curves in this family must be constructed over very large (1024-bit) fields, downgrading the performances. To relax this constraint, authors of [12] constructed a 2-chain of pairing-friendly elliptic curves such that only the base field of one curve is equal to the scalar field of the other, allowing one-layer recursive proof composition. Namely, the inner curve is a BLS12-377 and the outer curve is a CP6-782.

Previous work. In [18], El Housni and Guillevic introduced a 2-chain of curves made of the previous BLS12-377 and a new BW6-761 curve, a Brezing-Weng curve of embedding degree 6 defined over a 761-bit prime field, which they demonstrated to be orders of magnitude faster than CP6-782.

Contributions. First we are interested in families of 2-chains in which the BW6-761 curve would fall. We present a family of BW6 curves from any BLS12 curve and derive generic formulas, in terms of the BLS12 curve seed u, and integer parameters h_t, h_y. We extend this work to a 2-chain family of BW6 curves from BLS24 curves. To achieve higher levels of security in the target finite field of the outer curves, we compare a larger field characteristic thanks to larger parameters

h_t, h_y, to the larger embedding degrees 8 and 12 obtained with Cocks-Pinch curves. Finally, we argue that the BLS12 and BLS24 based families are respectively tailored for Groth'16 and KZG-based SNARKs recursive proof composition, and we present a short list of curves with an optimized implementation along with benchmarks.

Organization of the paper. Section 2 provides the preliminaries and definitions of SNARK-friendly elliptic curves. In section 3, we argue on the choice of BLS family as the inner curve and present faster group operations. The core of the paper are sections 4 and 5. Section 4 exposes the constructions of the outer curves, with optimized pairings and group operations. Finally, section 5 reports on the implementation of the most promising constructions identified in Section 4 and compares the performances in relevant practical settings.

2 Preliminaries

We present a short background on pairing-friendly elliptic curves and propose definitions of a SNARK-friendly chain of curves.

2.1 Background on bilinear pairings

We briefly recall elementary definitions on pairings and present the computation of two pairings used in practice, the modified Tate and ate pairings. All elliptic curves discussed below are *ordinary* (i.e. non-supersingular).

Let E be an elliptic curve defined over a field $\mathbb{F}_q$, where q is a prime power. Let π_q be the Frobenius endomorphism: $(x, y) \mapsto (x^q, y^q)$. Its minimal polynomial is $X^2 - tX + q$ where t is called the *trace*. Let r be a prime divisor of the curve order $\#E(\mathbb{F}_q) = q + 1 - t$. The r-torsion subgroup of E is denoted $E[r] = \{P \in E(\overline{\mathbb{F}_q}), [r]P = \mathcal{O}\}$ and has two subgroups of order r (eigenspaces of π_q in $E[r]$) that are useful for pairing applications. We define the two groups $\mathbb{G}_1 = E[r] \cap \ker(\pi_q - [1])$ with a generator denoted by G_1, and $\mathbb{G}_2 = E[r] \cap \ker(\pi_q - [q])$ with a generator G_2. The group $\mathbb{G}_2$ is defined over $\mathbb{F}_{q^k}$, where the embedding degree k is the smallest integer $k \in \mathbb{N}^*$ such that $r \mid q^k - 1$.

We recall the Tate and ate pairing definitions, based on the same two steps: evaluating a function $f_{s,Q}$ at a point P, the Miller loop step, and then raising it to the power $(q^k - 1)/r$, the final exponentiation step. The function $f_{s,Q}$ has divisor $div(f_{s,Q}) = s(Q) - ([s]Q) - (s - 1)(\mathcal{O})$ and satisfies, for integers i and j,

$$f_{i+j,Q} = f_{i,Q} f_{j,Q} \frac{\ell_{[i]Q,[j]Q}}{v_{[i+j]Q}},$$

where $\ell_{[i]Q,[j]Q}$ and $v_{[i+j]Q}$ are the two lines needed to compute $[i+j]Q$ from $[i]Q$ and $[j]Q$ (ℓ intersecting the two points and v the vertical). We compute $f_{s,Q}(P)$ with the Miller loop presented in Algorithm 1.

Algorithm 1: MillerLoop(s, P, Q)

Output: $m = f_{s,Q}(P)$

$m \leftarrow 1$; $S \leftarrow Q$

for b *from the second most significant bit of* s *to the least* **do**

  $\ell \leftarrow \ell_{S,S}(P)$; $S \leftarrow [2]S$; $v \leftarrow v_{[2]S}(P)$ // Doubling Step

  $m \leftarrow m^2 \cdot \ell / v$

  if $b = 1$ **then**

    $\ell \leftarrow \ell_{S,Q}(P)$; $S \leftarrow S + Q$; $v \leftarrow v_{S+Q}(P)$ // Addition Step

    $m \leftarrow m \cdot \ell / v$

return m

The Tate and ate pairings are defined by

$$\mathrm{Tate}(P, Q) = f_{r,P}(Q)^{(q^k-1)/r}$$
$$\mathrm{ate}(Q, P) = f_{t-1,Q}(P)^{(q^k-1)/r}$$

where $P \in \mathbb{G}_1 \subset E[r](\mathbb{F}_q)$ and $Q \in \mathbb{G}_2 \subset E[r](\mathbb{F}_{q^k})$. The final powering $z \mapsto z^{(q^k-1)/r}$ ensures that the values $\mathrm{Tate}(P, Q)$ and $\mathrm{ate}(Q, P)$ are in the *target* group $\mathbb{G}_T$ of r-th roots of unity in $\mathbb{F}_{q^k}$. It is decomposed into two steps: the easy part $z^{(q^k-1)/\Phi_k(q)}$ with one inversion and some Frobenius powers, and the hard part $z^{\Phi_k(q)/r}$, where Φ_k is the k-th cyclotomic polynomial. In this paper, when abstraction is needed, we denote a pairing as follows: $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

It is also important to recall some results with respect to the complex multiplication (CM) discriminant $-D$. When $D = 3$ (resp. $D = 4$), the curve has CM by $\mathbb{Q}(\sqrt{-3})$ (resp. $\mathbb{Q}(\sqrt{-1})$) so that twists of degrees 3 and 6 exist (resp. 4). If moreover the twist degree d divides k, then $\mathbb{G}_2$ is isomorphic to $E'[r](\mathbb{F}_{q^{k/d}})$ for a d-twist E'. Otherwise, in the general case, E admits a single twist (up to isomorphism) and it is of degree 2.

2.2 zk-SNARKs

In this paper, we focus on preprocessing zkSNARKs for NP languages for which we give a basic explanation. Given a public NP program F, public inputs a and b and private input w, such that the program F satisfies the relation $F(a, w) \coloneqq b$, a zk-SNARK consists in proving this relation succinctly without revealing the private input w. Given a security parameter λ, it consists of the `Setup`, `Prove` and `Verify` algorithms:

$$(\sigma_p, \sigma_v) \leftarrow \texttt{Setup}(F, \tau, 1^\lambda)$$
$$\pi \leftarrow \texttt{Prove}(a, b, w, \sigma_p)$$
$$0/1 \leftarrow \texttt{Verify}(a, b, \pi, \sigma_v)$$

where τ is the setup trapdoor, σ_p the proving key which encodes the program F for the prover, σ_v the verification key that encodes F for the verifier and π the proof.

2.3 SNARK-friendly chains

Definition 1. *An m-chain of elliptic curves is a list of distinct curves*

$$E_1/\mathbb{F}_{q_1}, \ldots, E_m/\mathbb{F}_{q_m}$$

where $q_1, \ldots, q_m$ are large primes and

$$q_1 = r_2 \mid \#E_2(\mathbb{F}_{q_2}), \ldots,\ q_{i-1} = r_i \mid \#E_i(\mathbb{F}_{q_i}), \ldots,\ q_{m-1} = r_m \mid \#E_m(\mathbb{F}_{q_m})\ . \tag{1}$$

Definition 2. *An m-chain of SNARK-friendly elliptic curves is an m-chain where each of the $\{E_i/\mathbb{F}_{q_i}\}_{1\leq i\leq m}$ curves*

- *is pairing-friendly;*
- *has a highly 2-adic subgroup, i.e. $r_i - 1 \equiv 0 \mod 2^L$ for a large $L \geq 1$.*

In particular, a SNARK-friendly 2-chain is a pair of two pairing-friendly elliptic curves $E_1/\mathbb{F}_{q_1}$ and $E_2/\mathbb{F}_{q_2}$ where $q_1 = r_2 \mid \#E_2(\mathbb{F}_{q_2})$ and $r_2 - 1 \equiv r_1 - 1 \equiv 0 \mod 2^L$. We call E_1 the inner curve and E_2 the outer curve.

In this paper, we aim at constructing families of SNARK-friendly 2-chains that are suitable respectively for Groth'16 and KZG-based universal SNARKs.

3 Inner curves: Barreto–Lynn–Scott (BLS) curves

We investigate the BLS family as an option for a SNARK-friendly inner curve. We first present our results for a better arithmetic on all BLS curves and then argue on the choice of BLS12 and BLS24 curves for our applications.

3.1 Parameters with a polynomial form

BLS curves were introduced in [7]. This is a family of pairing-friendly elliptic curves of embedding degree k multiple of 3 but not multiple of 18. Well-known families are given with $k = 2^i 3^j$ for $i, j \geq 0$: $k = 9, 12, 24, 27, 48$ (Table 1). The curves have j-invariant 0, discriminant $-D = -3$. Each family has polynomial parameters $q(x), r(x), t(x)$ for characteristic, subgroup order of embedding degree k, and trace. The subgroup order is $r(x) = \Phi_k(x)$ the k-th cyclotomic polynomial. The trace has a simple expression $t(x) = x+1$, so that the ate pairing whose Miller loop computes the function $f_{x,Q}(P)$ is optimal in terms of Vercauteren's paper [51]. The curve order is $q(x)+1-t(x)$ and the CM equation is $4q(x) = t(x)^2 + Dy(x)^2$. We state useful lemmas whose proofs are given in Appendix A.1. The explicit polynomials for BLS curves with $k \leq 99$ are given in Tables 16 and 17.

Lemma 1. *The cofactor $c(x)$ of BLS curves such that $q(x)+1-t(x) = c(x)r(x)$ has the form*

Table 1. Parameters of BLS curves for $k = 2^i 3^j$, $i \geq 0$, $j \geq 1$, $18 \nmid k$.

k	$2^i 3^j,\ i,j \geq 1\ (6, 12, 24, 48, 96, \ldots)$	$3^j,\ j \geq 1\ (3, 9, 27, 81, \ldots)$
$t(x)$	$x+1$	
$y(x)$	$(x-1)(2x^{k/6}-1)/3$	$(x-1)(2x^{k/3}+1)/3$
$r(x)$	$x^{k/3}-x^{k/6}+1$	$x^{2k/3}+x^{k/3}+1$
$q(x)$	$r(x)(x-1)^2/3+x$	$r(x)/3(x-1)^2+x$
$c_2(x)$	1	1
ρ	$1+6/k$	$1+3/k$

1. $(x-1)^2/3 \cdot c_2(x)$ *for odd* k*, where* $c_2(x) = (x^{2k/3}+x^{k/3}+1)/\Phi_k(x) \in \mathbb{Q}[x]$*;*
2. $(x-1)^2/3 \cdot c_2(x)$ *for even* k*, where* $c_2(x) = (x^{k/3}-x^{k/6}+1)/\Phi_k(x) \in \mathbb{Q}[x]$*.*

Lemma 2. *For all BLS curves, the polynomial form of the characteristic* $q(x)$ *is such that* $(x-1)/3$ *divides* $q(x)-1$*.*

Lemma 3. *The parameter* $y(x)$ *of BLS curves has the form*

1. $(x-1)(2x^{k/3}+1)/3$ *for odd* k*;*
2. $(x-1)(2x^{k/6}-1)/3$ *for even* k*.*

Lemma 4. *Any BLS curve has endomorphism ring* $\mathbb{Z}[\omega]$ *where* $\omega = (1+\sqrt{-3})/2$*.*

3.2 Faster co-factor multiplication

Because $\mathbb{G}_1$ is a proper subgroup of $E(\mathbb{F}_q)$, one multiplies a point $P \in E(\mathbb{F}_q)$ by the cofactor $c(x) = (x-1)^2/3$ to map it to $\mathbb{G}_1$, a.k.a. *cofactor clearing*. Wahby and Boneh noted in [52], that it is sufficient to multiply by $x-1$ to clear the cofactor of $\mathbb{G}_1$ for the BLS12-381 curve (also in [46, §2]). Here we generalize and prove that it is true for all BLS curves. Let $\mathrm{End}_{\mathbb{F}_q}(E)$ denotes the ring of $\mathbb{F}_q$-endomorphisms of E, let $\mathcal{O}$ denotes a complex quadratic order of the ring of integers of a complex quadratic number field, and $\mathcal{O}(\Delta)$ denotes the complex quadratic order of discriminant Δ.

Theorem 1 ([45, Proposition 3.7]). *Let* E *be an elliptic curve over* $\mathbb{F}_q$ *and* $n \in \mathbb{Z}_{\geq 1}$ *with* $q \nmid n$*. Let* π_q *denotes the Frobenius endomorphism of* E*. Then,*

$$E[n] \subset E(\mathbb{F}_q) \iff \begin{cases} n^2 \mid \#E(\mathbb{F}_q), \\ n \mid q-1 \text{ and} \\ \pi_q \in \mathbb{Z} \text{ or } \mathcal{O}\big(\frac{t^2-4q}{n^2}\big) \subset \mathrm{End}_{\mathbb{F}_q}(E). \end{cases}$$

Proof. Proposition 3.7. in [45].

Corollary 1. *Let* $E(\mathbb{F}_{q(x)})$ *be a BLS curve of order* $c(x)r(x)$ *where* $r(x)$ *is the subgroup prime order and* $c(x) = (x-1)^2/3 \cdot c_2(x)$ *the cofactor. It is sufficient to multiply by* $(x-1)c_2(x)$ *to clear the cofactor.*

Proof. Let $n(x) = (x-1)/3$, we show that the full $n(x)$-torsion is in $E(\mathbb{F}_q)$, that is there is no point of order $n^2(x)$ in $E(\mathbb{F}_q)$ but there are $n^2(x)$ points of order $n(x)$. Thus it is sufficient to multiply by $3n(x)$ to clear the $(x-1)^2/3$ cofactor. According to Lemmas 1, 2, 3, and 4, we have $n(x)^2 \mid \#E(\mathbb{F}_q)$, $n(x) \mid q(x) - 1$. Now $(t^2(x) - 4q(x))/n^2(x) = -3y^2(x)/n^2(x) = -3(2x^{k/6} - 1)^2$ for even k, and $-3(2x^{k/3} + 1)^2$ for odd k. Hence $\mathcal{O}\big(\frac{t(x)^2 - 4q(x)}{n^2(x)}\big) \subset \mathrm{End}_{\mathbb{F}_q}(E)$. Thus, Theorem 1 applies and $E[n(x)] \subset E(\mathbb{F}_q)$.

Theorem 1 applied to BLS curves tells us that the curve endomorphism $\phi\colon E \to E$, $(x, y) \mapsto (\omega x, y)$ with $\omega \in \mathbb{F}_q$ a primitive third root of unity ($\omega^2 + \omega + 1 = 0 \bmod q$) acts has a *distortion map* on $E[n] \simeq \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}$. With a Weil pairing e_W, one can embed a discrete logarithm on $E(\mathbb{F}_q)[n]$ into $\mathbb{F}_q^*$, where subexponential DL computation takes place, although the much larger size of q compared to n seems prohibitive. For $G, P \in E[n]$ in the same subgroup of order n, $\log_G(P) = \log_{e_W(G,\phi(G))} e_W(P, \phi(G))$. See A.2 for more details.

3.3 Subgroup membership testing: $\mathbb{G}_T$

Testing membership in $\mathbb{G}_T$ for candidate elements z of $\mathbb{F}_{q^k}$ is done in two steps. First, one checks that z belongs to the *cyclotomic subgroup* of $\mathbb{F}_{q^k}$ (subgroup of order $\Phi_k(q)$), that is $z^{\Phi_k(q)} = 1$. To avoid inversions, one multiplies the positive terms in q^i on one hand, and the negative terms one the other hand, and check for equality: it costs only Frobenius powers. With $k = 6$ and $\Phi_6(q) = q^2 - q + 1$, it means checking that $z^{q^2+1} = z^q$. Second, it is possible to use a generalisation of Scott's technique first developed for BN curves, where $r = q + 1 - t$ [48, §8.3]. In the BN case, the computation of z^r is replaced by a Frobenius power z^q and an exponentiation z^{t-1}, and the test $z^q = z^{t-1}$. BLS curves are not of prime order, and we use Proposition 1. This trick already appears in [4, §5], but without a proof.

Proposition 1. *Let E be a pairing-friendly curve defined over $\mathbb{F}_q$, of embedding degree k w.r.t. the subgroup order r, and order $\#E(\mathbb{F}_q) = r \cdot c = q + 1 - t$. For $z \in \mathbb{F}_{q^k}^*$, we have this alternative $\mathbb{G}_T$ membership testing:*

$$z^{\Phi_k(q)} = 1 \text{ and } z^q = z^{t-1} \text{ and } \gcd(q + 1 - t, \Phi_k(q)) = r \implies z^r = 1 \ .$$

Proof. If $z^{\Phi_k(q)} = 1$ and $z^{q+1-t} = 1$, then the order of z divides the gcd of the exponents $\gcd(\Phi_k(q), q + 1 - t)$. If this gcd is exactly r, then z is in the subgroup of order r, that is $z^r = 1$.

BLS curves have $c \cdot r = q + 1 - t = q - u$ hence

$$q \equiv u \bmod r \ . \tag{2}$$

As soon as $\gcd(q + 1 - t, \Phi_k(q)) = r$, then the following two tests are enough:

1. test if $z^{\Phi_k(q)} = 1$ with Frobenius maps;

2. test if $z^q = z^u$, using cyclotomic squarings [27] for a faster exponentiation.

Proposition 1 came out of email discussions between cryptographers, and appears in Scott's preprint [46, §5].

Remark 1. For BLS-curves of embedding degree k a power of 3 ($k = 3^j$), the cyclotomic polynomial $\Phi_k(x)$ does not generate primes, actually one has $r(x) = \Phi_k(x)/3$. Moreover a BLS curve has points of order 3, hence $\gcd(q+1-t, \Phi_k(q)) = 3r$ for all $k = 3^j$.

Remark 2. For SNARK-friendly 2-chains, $z^u \in \mathbb{G}_T$ can be implemented efficiently using a mix of Granger-Scott's [27] and Karabina's [37] cyclotomic squares. Since $2^L \mid u - 1$, there are $L - 1$ consecutive squarings in the exponentiation. One can use Karabina's method for this series and then switch to Granger-Scott's method for the remaining part. Hence, trading off one inversion in $\mathbb{F}_{q^{k/d}}$ for $2(L-1)$ multiplications in $\mathbb{F}_{q^{k/d}}$. Particularly, for BLS12 and BLS24, this trick yields significant speedups as long as an $\mathbb{F}_q$-inverse costs, respectively, less than $(6L-4)$ and $(18L-16)$ $\mathbb{F}_q$-multiplications, which is the case of curves we are interested in.

3.4 Choosing a curve coefficient $b = 1$

Proposition 2. *Half of BLS curves are of the form $Y^2 = X^3 + 1$, these are the curves with odd seed x.*

Proof. Let $E : Y^2 = X^3 + b$ be a BLS curve over $\mathbb{F}_q$ and g neither a square nor a cube in $\mathbb{F}_q$. One choice of $b \in \{1, g, g^2, g^3, g^4, g^5\}$ gives a curve with the correct order (i.e. $r \mid \#E(\mathbb{F}_q)$) [49, §X.5]. For all BLS curves, $x - 1 \mid \#E(\mathbb{F}_q)$ (cf Lemma 1, Tables 16, 17) and $3 \mid x - 1$ (which leads to all involved parameters being integers). If, additionally, $2 \mid x - 1$ then $2, 3 \mid \#E(\mathbb{F}_q)$ and the curve has points of order 2 and 3. A 2-torsion point is $(x_0, 0)$ with x_0 a root of $x^3 + b$, hence $b = (-x_0)^3$ is a cube. The two 3-torsion points are $(0, \pm\sqrt{b})$ hence b is a square. This implies that b is a square and a cube in $\mathbb{F}_q$ and therefore $b = 1$ is the only solution in the set $\{g^i\}_{0 \leq i \leq 5}$ for half of all BLS curves: those with odd x.

3.5 SNARK-friendly inner BLS curves

This paper focuses on inner SNARK-friendly BLS curves as in Def. 1 at the 128-bit security level and suitable for the Groth'16 and KZG-based universal SNARKs. On the one hand, a Groth'16-tailored curve should optimize $\mathbb{G}_1$ and $\mathbb{G}_2$ operations, and the pairing computation: the proving algorithm involves multi-scalar multiplications (MSM) in $\mathbb{G}_1$ and $\mathbb{G}_2$, and the verification algorithm involves multi-pairings. On the other hand, KZG polynomial commitments only need multi-scalar multiplications in $\mathbb{G}_1$ and multi-pairings.

According to [31], an efficient non-conservative choice of a Groth'16-tailored curve at the 128-bit security level is a BLS12 curve of roughly 384 bits. A conservative but efficient alternative is a BLS12 curve of 440 to 448 bits. Then to

fulfill SNARK-friendliness, it is sufficient to choose a seed x s.t. $x \equiv 1 \mod 3 \cdot 2^L$ with the desired 2-adicity $L \geq 1$. Consequently, Prop. 1 and 2, and Cor. 1 apply: such an inner BLS12 is always of the form $Y^2 = X^3 + 1$; multiplying by $x - 1$ is sufficient to clear the cofactor on $\mathbb{G}_1$, and the efficient $\mathbb{G}_T$ membership testing applies. In fact, for all BLS12 curves, $\gcd(q(x) + 1 - t(x), \Phi_{12}(q(x)))$ is always equal to $r(x)$ and the membership testing boils down to $z^q = (z^q)^q \cdot z$ and $z^q = z^u$ for $z \in \mathbb{G}_T$.

KZG-based SNARKs require a 128-bit secure curve with efficient $\mathbb{G}_1$ operations and fast pairing. For a faster $\mathbb{G}_1$ arithmetic, we consider a BLS24 curve of roughly 320 bits, that meets the 128-bit level security [33] and gives the best tradeoff between small $\rho = \log_2 q / \log_2 r$ value ($\rho = 1.25$) and fast pairing. For SNARK-friendliness, cofactor clearing and curve equation ($Y^2 = X^3 + 1$), the same observations as for BLS12 apply. For $\mathbb{G}_T$ membership testing, $\gcd(q(x) + 1 - t(x), \Phi_{24}(q(x)))$ is always equal to $r(x)$ for the BLS24 curves and the test boils down to $z^{q^2} = (z^{q^2})^{q^2} \cdot z$ and $z^q = z^u$ for $z \in \mathbb{G}_T$.

4 Outer curves: Brezing–Weng, Cocks–Pinch

This section presents the families of 2-chains with a BW6 curve on top of a BLS12 curve (Sec. 4.2), and on top of a BLS24 curve (Sec. 4.3). Cocks-Pinch curves (CP) are addressed in Sec. 4.4. For BW6, all parameters and formulas are given as polynomials in the variable x, with integer parameters h_t, h_y that are the lifting cofactors of the Brezing-Weng construction. We use subscripts $q_{\text{bls}}, q_{\text{bw}}, q_{\text{cp}}$ to identify parameters of BLS, BW and CP curves. BW and CP constructions follow the same recipe, but CP deals with integers, while BW deals with polynomials [21, §4.1,§6]. They start from the subgroup order $r_{\text{bw}}(x) = q_{\text{bls}}(x)$, $r_{\text{cp}}(u) = q_{\text{bls}}(u)$, and look for k-th roots of unity ζ_k mod q_{bls} to set the trace value $t = \zeta_k + 1$. For CP, the existence of ζ_k requires $q_{\text{bls}}(u) \equiv 1 \mod k$: for $k = 6, 12, 8$ resp., this means $u \equiv 1 \mod 3$, $1, 10 \mod 12$, and $1, 10 \mod 24$ resp. For BW, the number field defined by $q_{\text{bls}}(x)$ only contains $\zeta_k(x)$ for $k \mid 6$, limiting the BW construction to $k = 6$ at most.

4.1 Generic BW6 curve parameters

To satisfy Def. 1, a BW curve chained to a BLS curve has a subgroup of prime order $r_{\text{bw}}(x) = q_{\text{bls}}(x)$. To get an embedding degree $k = 6$, a primitive 6-th root of unity ζ_6 modulo $r_{\text{bw}}(x)$ is required, the trace of the curve modulo r_{bw} is then $t_{\text{bw},3} = \zeta_6 + 1 \mod r_{\text{bw}}$. Alternatively $t_{\text{bw},0} = \overline{\zeta_6} + 1 \mod r_{\text{bw}}$ with $\zeta_6 = -\overline{\zeta_6} + 1$. With $D = -3$ and $1/\sqrt{-3} = (2\overline{\zeta_6} - 1)/3 \mod r_{\text{bw}}$, then $y_{\text{bw},0} = (t_{\text{bw},0} - 2)/\sqrt{-3} = (\overline{\zeta_6} + 1)/3 = -t_{\text{bw},0}/3$. Or with $1/\sqrt{-3} = -(2\zeta_6 - 1)/3 \mod r_{\text{bw}}$, one has $y_{\text{bw},3} = (t_{\text{bw},3} - 2)/\sqrt{-3} = (\zeta_6 + 1)/3 = t_{\text{bw},3}/3$. Any BW6 curve will have parameters of the form $t_i = t_{\text{bw},i} \pm h_t r$, $y_i = y_{\text{bw},i} \pm h_y r$, where h_t, h_y are integer lifting cofactors. We label the two cases according to the constant coefficient of the polynomial defining the trace modulo r_{bw}: this is either 0 or 3.

One denotes $q_{\mathrm{bw},0}(x,h_t,h_y) = ((t_{\mathrm{bw},0}+h_t r)^2 + 3(y_{\mathrm{bw},0}+h_y r)^2)/4$. We have

$$q_{\mathrm{bw},0} = t_{\mathrm{bw},0}^2/3 + t_{\mathrm{bw},0}\cdot r_{\mathrm{bw}}(h_t-h_y)/2 + r_{\mathrm{bw}}^2(h_t^2+3h_y^2)/4\ , \tag{3}$$
$$q_{\mathrm{bw},3} = t_{\mathrm{bw},3}^2/3 + t_{\mathrm{bw},3}\cdot r_{\mathrm{bw}}(h_t+h_y)/2 + r_{\mathrm{bw}}^2(h_t^2+3h_y^2)/4\ . \tag{4}$$

The curve cofactor $c_{\mathrm{bw},i}(x,h_t,h_y)$ such that $c_{\mathrm{bw},i}r_{\mathrm{bw}} = q_{\mathrm{bw},i}+1-t_{\mathrm{bw},i}$ is

$$c_{\mathrm{bw},0} = (h_t^2+3h_y^2)/4r_{\mathrm{bw}} + (h_t-h_y)/2t_{\mathrm{bw},0} + (t_{\mathrm{bw},0}^2/3 - t_{\mathrm{bw},0}+1)/r_{\mathrm{bw}} - h_t \tag{5}$$
$$c_{\mathrm{bw},3} = (h_t^2+3h_y^2)/4r_{\mathrm{bw}} + (h_t+h_y)/2t_{\mathrm{bw},3} + (t_{\mathrm{bw},3}^2/3 - t_{\mathrm{bw},3}+1)/r_{\mathrm{bw}} - h_t \tag{6}$$

where $(t_{\mathrm{bw},i}^2/3 - t_{\mathrm{bw},i}+1)/r_{\mathrm{bw}} = \Phi_6(t_{\mathrm{bw},i}-1)/(3r_{\mathrm{bw}})$ is a polynomial in $\mathbb{Q}[x]$ since by construction r_{bw} divides $\Phi_6(t_{\mathrm{bw},i}-1)$. Tables 3 and 4 give the explicit values of the polynomials for BLS12 and BLS24 inner curves.

Cofactor of $\mathbb{G}_2$. The group $\mathbb{G}_2$ of order r_{bw} is a subgroup of one of the two sextic twists of E, defined over $\mathbb{F}_q$. Generically, the orders of the two sextic twists are $q+1-(t+3y)/2$ and $q+1-(t-3y)/2$, where y satisfies $t^2-4q=-3y^2$. One of the orders is a multiple of r_{bw}, and has cofactor $c'_{\mathrm{bw},i}$. Observe that $(t_{\mathrm{bw},0}-3y_{\mathrm{bw},0})/2 = t_{\mathrm{bw},0}$ since $y_{\mathrm{bw},0} = -t_{\mathrm{bw},0}/3$. The correct sextic twist has order

$$\begin{aligned}
&q_{\mathrm{bw},0}+1-(t_{\mathrm{bw},0}+h_t r_{\mathrm{bw}} - 3(y_{\mathrm{bw},0}+h_y r_{\mathrm{bw}}))/2\\
&= q_{\mathrm{bw},0}+1-\underbrace{(t_{\mathrm{bw},0}-3y_{\mathrm{bw},0})/2}_{=t_{\mathrm{bw},0}} - h_t r_{\mathrm{bw}}/2 + 3h_y r_{\mathrm{bw}}/2\\
&= \underbrace{q_{\mathrm{bw},0}+1-t_{\mathrm{bw},0}-h_t r_{\mathrm{bw}}}_{=\#E(\mathbb{F}_q)=r_{\mathrm{bw}}\cdot c_{\mathrm{bw},0}} + (h_t r_{\mathrm{bw}} + 3h_y r_{\mathrm{bw}})/2\\
&= r_{\mathrm{bw}}\cdot \underbrace{(c_{\mathrm{bw},0}+(h_t+3h_y)/2)}_{c'_{\mathrm{bw},0}}
\end{aligned}$$

hence

$$c'_{\mathrm{bw},0} = c_{\mathrm{bw},0} + (h_t+3h_y)/2\ . \tag{7}$$

For the other trace, $(t_{\mathrm{bw},3}+3y_{\mathrm{bw},3})/2 = t_{\mathrm{bw},3}$ and the correct sextic twist has order $q_{\mathrm{bw},3}+1-(t_{\mathrm{bw},3}+h_t r_{\mathrm{bw}}+3(y_{\mathrm{bw},3}+h_y r_{\mathrm{bw}}))/2$ a multiple of r_{bw}, and cofactor

$$c'_{\mathrm{bw},3} = c_{\mathrm{bw},3} + (h_t-3h_y)/2\ . \tag{8}$$

Congruences of cofactors h_t, h_y. One requires $q_{\mathrm{bw},i}$ (Eqs. (3), (4)) to be an integer and a prime. Because $t_{\mathrm{bw},i}$ is always multiple of 3, $t_{\mathrm{bw},i}^2/3$ is an integer. We need $(h_t \pm h_y)/2t_{\mathrm{bw},i} + (h_t^2+3h_y^2)/4r_{\mathrm{bw}}$ to be an integer. We now look at $(h_t \pm h_y)$, $(h_t^2+3h_y^2)$. We have $t_{\mathrm{bw},0}$ always even, then $(h_t-h_y)t_{\mathrm{bw},0}/2$ is an integer and we require $4 \mid (h_t^2+3h_y^2)$. For that we need $h_t - h_y \equiv 0 \bmod 2$ (see Table 2). We have $t_{\mathrm{bw},3}$ always odd. If (h_t+h_y) is odd, then $(h_t+h_y)t_{\mathrm{bw},3}$ is odd but at the same time (see Table 2), $(h_t^2+3h_y^2)$ is odd, and the condition

is not satisfied. Hence we need $(h_t - h_y)$ to be even, and consequently we have $(h_t^2 + 3h_y^2)/4$ an integer. Finally, for both $t_{\mathrm{bw},0}$ and $t_{\mathrm{bw},3}$, we need $2 \mid (h_t - h_y)$ and consequently we have $4 \mid h_t^2 + 3h_y^2$, to ensure q_{bw} to be an integer. Note also that because $x \equiv 1 \bmod 3$, one has $t_{\mathrm{bw}} = 0 \bmod 3$, and Eqs. (3), (4) give $4q_{\mathrm{bw}} = h_t^2 \bmod 3$. Because q_{bw} needs to be prime, h_t is not multiple of 3, and $3 \nmid (h_t^2 + 3h_y^2)$.

Table 2. Are $2(h_t \pm h_y)$, $h_t^2 + 3h_y^2$ multiple of 4?

h_t mod2	h_y mod2	$h_t \pm h_y$ mod2	$h_t^2 + 3h_y^2$ mod4	$2(h_t \pm h_y)t_{\mathrm{bw},i} + (h_t^2 + 3h_y^2)r_{\mathrm{bw}} \bmod 4$	
				$t_{\mathrm{bw},0} = 0 \bmod 2$	$t_{\mathrm{bw},3} = 1 \bmod 2$
0	0	0	0	0	0
0	1	1	3	$3r_{\mathrm{bw}} \neq 0$	$2 + 3r_{\mathrm{bw}} \neq 0$
1	0	1	1	$r_{\mathrm{bw}} \neq 0$	$2 + r_{\mathrm{bw}} \neq 0$
1	1	0	0	0	0

Subgroup membership testing: $\mathbb{G}_T$. We apply the technique of Sec. 3.3. BW6 curves over their base field have order $c_{\mathrm{bw},i} \cdot r_{\mathrm{bw}} = q_{\mathrm{bw},i} + 1 - t_{\mathrm{bw},i} - h_t r_{\mathrm{bw}}$, hence

$$q_{\mathrm{bw},i} \equiv t_{\mathrm{bw},i} - 1 \bmod r_{\mathrm{bw}} \ . \tag{9}$$

As soon as $\gcd(q_{\mathrm{bw},i} + 1 - t_{\mathrm{bw},i}, \Phi_k(q_{\mathrm{bw},i})) = r_{\mathrm{bw}}$, then the following two tests are enough:

1. test if $z^{\Phi_k(q_{\mathrm{bw},i})} = 1$ with Frobenius maps;
2. test if $z^{q_{\mathrm{bw},i}} = z^{t_{\mathrm{bw},i}-1}$ with cyclotomic squarings.

Easy part of the final exponentiation. the final exponentiation raises the Miller loop output f to the power

$$(q^6 - 1)/r = (q^6 - 1)/\Phi_6(q) \cdot \Phi_6(q)/r = (q^3 - 1)(q + 1)(q^2 - q + 1)/r \ .$$

The easy part $(q^3 - 1)(q + 1)$ costs one conjugation (q^3-Frobenius power), one inversion in $\mathbb{F}_{q^6}$, one q-Frobenius power and two multiplications. We optimise the hard part $(q^2 - q + 1)/r$ in Sec. 4.2, 4.3.

Optimal Pairing Computation. In [18], the authors presented an optimal ate pairing formula that can be generalized as follows: write

$$a_0 + a_1(t_{\mathrm{bw},i} - 1) = 0 \bmod r_{\mathrm{bw}} \tag{10}$$

with shortest possible scalars a_0, a_1. On $\mathbb{G}_2$, the Frobenius π_q has eigenvalue $t_{\mathrm{bw},i} - 1$. The optimal ate Miller loop is computed with the formula

$$f_{a_0,Q}(P) f_{a_1,\pi_q(Q)}(P) = f_{a_0,Q}(P) f_{a_1,Q}^q(P) \ . \tag{11}$$

Moreover, it turned out that $(a_1 - 1) \mid a_2$, and some of the computations were shared. We now introduce another optimisation. We consider Eq. (10) with a new point of view. BW6 curves have an endomorphism $\phi\colon (x, y) \mapsto (\omega x, -y)$ on $\mathbb{G}_1$ of eigenvalue $\lambda = t_{\mathrm{bw},i} - 1 = q_{\mathrm{bw},i} \bmod r_{\mathrm{bw}}$, and characteristic polynomial $\chi^2 - \chi + 1 = 0$. The (bilinear) twisted ate pairing [35, §6] has precisely Miller loop $f_{\lambda,P}(Q)$. However, λ is too large so instead, we consider a multiple of the Tate pairing $f_{hr,P}(Q) = f_{a_0+a_1\lambda,P}(Q)$ for some h (e.g. Eqs.(20), (29)). Instead of decomposing the Miller function $f_{a_0+a_1\lambda,P}(Q)$ into sub-functions $f_{a_0,P}(Q)f_{a_1\lambda,P}(Q)$, we use Lemma 5 to get shared squares in $\mathbb{F}_{q^k}$ and shared doubling steps in $\mathbb{G}_1$ (Tate), resp. $\mathbb{G}_2$ (ate), in the same idea as a multi-scalar multiplication. This gives us Alg. 2. We are in the very particular case of $k/d = 1$, ϕ on $\mathbb{G}_1$ and π_q on $\mathbb{G}_2$ both have eigenvalue $q_{\mathrm{bw},i} \bmod r_{\mathrm{bw}}$, and our variant of the twisted ate pairing is competitive with the ate pairing.

Lemma 5. *Let E be a pairing-friendly curve with the usual order-r subgroups $\mathbb{G}_1, \mathbb{G}_2$, two points $P \in \mathbb{G}_i$, $Q \in \mathbb{G}_{1-i}$ of order r, and an endomorphism ϕ of eigenvalue λ over $\mathbb{G}_i$: $\phi(P) = [\lambda]P$, $\lambda = q^e \bmod r$ for some $1 \le e \le k-1$. The Miller function can be decomposed as follows.*

$$f_{2(u+v\lambda),P}(Q) = f^2_{u+v\lambda,P}(Q)\ell_{(u+v\lambda)P,(u+v\lambda)P}(Q) \tag{12}$$
$$f_{u+1+v\lambda,P}(Q) = f_{u+v\lambda,P}(Q)\ell_{(u+v\lambda)P,P}(Q) \tag{13}$$
$$f_{u+(v+1)\lambda,P}(Q) = f_{u+v\lambda,P}(Q)\ell_{(u+v\lambda)P,\lambda P}(Q) \tag{14}$$
$$f_{u+1+(v+1)\lambda,P}(Q) = f_{u+v\lambda,P}(Q)\ell_{P,\lambda P}(Q)\ell_{(u+v\lambda)P,(1+\lambda)P}(Q) \tag{15}$$

where $\lambda P = \phi(P)$, $(1+\lambda)P = P + \phi(P)$, and $\ell_{P,\lambda P}(Q)$ can be precomputed.

Proof (of Lemma 5). The usual Miller formulas give (see e.g. [51])

$$f_{2(u+v\lambda),P}(Q) = f^2_{u+v\lambda,P}(Q) \quad \underbrace{f_{2,[u+v\lambda]P}(Q)}_{=\text{ tangent at } (u+v\lambda)P}$$
$$f_{u+1+v\lambda,P}(Q) = f_{u+v\lambda,P}(Q)\underbrace{f_{1,P}(Q)}_{=1}\ell_{(u+v\lambda)P,P}(Q)$$
$$f_{u+(v+1)\lambda,P}(Q) = f_{u+v\lambda,P}(Q) \quad \underbrace{f_{\lambda,P}(Q)}_{\text{bilinear pairing}} \quad \ell_{(u+v\lambda)P,\lambda P}(Q)$$
$$f_{u+1+(v+1)\lambda,P}(Q) = f_{u+v\lambda,P}(Q) \quad \underbrace{f_{1+\lambda,P}(Q)}_{f_{1,P}(Q)f_{\lambda,P}(Q)\ell_{P,\lambda P}(Q)} \quad \ell_{(u+v\lambda)P,(1+\lambda)P}(Q)$$

The term $f_{1,P}(Q) = 1$ can disapear. The term $f_{\lambda,P}(Q)$ is a bilinear pairing as $\lambda \equiv q^e \bmod r$, and then can be removed. Finally $f_{1+\lambda,P}(Q)$ simplifies to $\ell_{P,\lambda P}(Q)$ which can be precomputed.

Remark 3. Alg. 2 shares the squarings in $\mathbb{F}_{q^k}$ and the doubling steps in $\mathbb{G}_1$ (Tate), resp. $\mathbb{G}_2$ (ate). With all parameterized pairing-friendly families, the scalar decomposition gives all but one trivial Miller function, and the ate, or twisted-ate

Algorithm 2: Miller loop for optimal pairing with endomorphism ϕ on $\mathbb{G}_1$ (Tate), resp. $\mathbb{G}_2$ (ate) of eigenvalue λ and degree 2.

Input: $P \in \mathbb{G}_i$, $Q \in \mathbb{G}_{1-i}$, end. ϕ on $\mathbb{G}_i$ of eigenvalue λ, scalars a_0, a_1 s. t. $a_0 + a_1\lambda = 0 \bmod r$

Output: $f_{a_0+a_1\lambda,P}(Q)$

$P_0 \leftarrow P$; $P_1 \leftarrow \phi(P)$
if $a_0 < 0$ **then** $a_0 \leftarrow -a_0$; $P_0 \leftarrow -P_0$
if $a_1 < 0$ **then** $a_1 \leftarrow -a_1$; $P_1 \leftarrow -P_1$
$P_{1+\lambda} \leftarrow P_0 + P_1$; $\ell_{1,\lambda} \leftarrow \ell_{P_0,P_1}(Q)$
$l_0 \leftarrow \text{bits}(a_0)$; $l_1 \leftarrow \text{bits}(a_1)$
if $\#l_0 = \#l_1$ **then** $S \leftarrow P_{1+\lambda}$; $f \leftarrow \ell_{1,\lambda}$; $n \leftarrow \#l_0$
else if $\#l_0 < \#l_1$ **then** $S \leftarrow P_1$; $f \leftarrow 1$; $n \leftarrow \#l_1$; pad l_0 with 0 s.t. $\#l_0 = n$
else $S \leftarrow P_0$; $f \leftarrow 1$; $n \leftarrow \#l_0$; pad l_1 with 0 s.t. $\#l_1 = n$
for $i = n - 2$ *downto* 0 **do**
    $f \leftarrow f^2$; $\ell_t \leftarrow \ell_{S,S}(Q)$; $S \leftarrow [2]S$
    if $l_0[i] = 0$ *and* $l_1[i] = 0$ **then** $f \leftarrow f \cdot \ell_t$ // Eq. (12), $m_{\texttt{full-sparse}}$
    else if $l_0[i] = 1$ *and* $l_1[i] = 1$ **then** // Eq. (15)
        $S \leftarrow S + P_{1+\lambda}$; $\ell \leftarrow \ell_{S,P_{1+\lambda}}(Q)$
        $f \leftarrow (f \cdot \ell_t) \cdot (\ell \cdot \ell_{1,\lambda})$ // $m_k + m_{\texttt{full-sparse}} + m_{\texttt{sparse-sparse}}$
    else if $l_0[i] = 1$ **then** // Eq. (13)
        $S \leftarrow S + P_0$; $\ell \leftarrow \ell_{S,P_0}(Q)$
        $f \leftarrow f \cdot (\ell_t \cdot \ell)$ // $m_k + m_{\texttt{sparse-sparse}}$
    else $(l_1[i] = 1)$ // Eq. (14)
        $S \leftarrow S + P_1$; $\ell \leftarrow \ell_{S,P_1}(Q)$
        $f \leftarrow f \cdot (\ell_t \cdot \ell)$ // $m_k + m_{\texttt{sparse-sparse}}$
return f

pairing boils down to one Miller loop computation of optimal length, and a few line additions [51]. In our case, while being short, none of the scalars a_0, a_1 is trivial. It is possible to derive a 2-NAF variant of Alg. 2. It requires the additional precomputations of $P - \phi(P)$ and $\ell_{P,-\lambda P}(Q)$. From the estimate in Table 8, our Miller loop variant in Alg. 2 would give up to a 7% speed-up compared to [18, Alg. 5], for BLS24-BW6 curves. Our Alg. 2 works for Tate and ate pairing. If there is an endomorphism of higher degree on $\mathbb{G}_2$ (or two independent endomorphisms), use Alg. 4 instead.

4.2 BW6 with BLS-12

Table 3 gives the parameters of the BW6-BLS12 curves in terms of the seed x, and the two lifting cofactors h_t, h_y.

Optimal Ate Pairing Computation. We investigate two pairings on our BW6 curves: optimal ate and optimal Tate. In [18], the authors presented an optimal ate Miller loop formula, for any BW6 curve with $t_{\text{bw},3}$:

$$m_{\text{opt. ate}} = f_{u+1,Q}(P) f^q_{u^3-u^2-u,Q}(P) \quad \text{and } e_{\text{opt. ate}} = m_{\text{opt. ate}}^{(q_{\text{bw}}^6-1)/r_{\text{bw}}} \tag{16}$$

Table 3. Parameters of a BW6 outer curve with a BLS12 inner curve, with $x \equiv 1 \bmod 3$.

parameter	value	property
r_{bw}	$q_{\mathrm{bls}} = (x-1)^2/3(x^4-x^2+1)+x$	generates prime
$\overline{\zeta_6}$	$-x^5+3x^4-3x^3+x-1$	
ζ_6	$x^5-3x^4+3x^3-x+2$	
$1/\sqrt{-3}$	$-(2x^5-6x^4+6x^3-2x+3)/3$	
$t_{\mathrm{bw},0}$	$-x^5+3x^4-3x^3+x$	$6 \mid t_{\mathrm{bw},0}$
$t_{\mathrm{bw},3}$	$x^5-3x^4+3x^3-x+3$	$3 \mid t_{\mathrm{bw},3}, 2 \nmid t_{\mathrm{bw},3}$
$y_{\mathrm{bw},0}$	$(x^5-3x^4+3x^3-x)/3 = -t_{\mathrm{bw},0}/3$	$2 \mid y_{\mathrm{bw},0}$
$y_{\mathrm{bw},3}$	$(x^5-3x^4+3x^3-x+3)/3 = t_{\mathrm{bw},3}/3$	$2 \nmid y_{\mathrm{bw},3}$
$q_{\mathrm{bw},0}$	$((t_{\mathrm{bw},0}+h_t r_{\mathrm{bw}})^2+3(y_{\mathrm{bw},0}+h_y r_{\mathrm{bw}})^2)/4$	generates prime
$q_{\mathrm{bw},3}$	$((t_{\mathrm{bw},3}+h_t r_{\mathrm{bw}})^2+3(y_{\mathrm{bw},3}+h_y r_{\mathrm{bw}})^2)/4$	generates prime
$\Phi_6(t_{\mathrm{bw},i}-1)$	$3r_{\mathrm{bw}}(x^4-4x^3+7x^2-6x+3)$	
$c_{\mathrm{bw},0}$	$(h_t^2+3h_y^2)/4r_{\mathrm{bw}}+(h_t-h_y)/2t_{\mathrm{bw},0}+x^4-4x^3+7x^2-6x+3-h_t$	
$c_{\mathrm{bw},3}$	$(h_t^2+3h_y^2)/4r_{\mathrm{bw}}+(h_t+h_y)/2t_{\mathrm{bw},3}+x^4-4x^3+7x^2-6x+3-h_t$	
$c'_{\mathrm{bw},0}$ ($\mathbb{G}_2$)	$c_{\mathrm{bw},0}+(h_t+3h_y)/2$	
$c'_{\mathrm{bw},3}$ ($\mathbb{G}_2$)	$c_{\mathrm{bw},3}+(h_t-3h_y)/2$	

with optimized computation in [18, Alg. 5]:

$$f_u = f_{u,Q}(P);\ m_{\text{opt. ate}} = f_u \cdot (f_u)^q_{u^2-u-1,[u]Q}(P)\ell_{[u]Q,Q}(P)\ , \tag{17}$$

where $[u]Q$ is precomputed together with $f_{u,Q}(P)$. The equivalent formula for a trace $t_{\mathrm{bw},0}$ is

$$f_{u(u^2-u-1),Q}(P)f^q_{u+1,Q}(P) \tag{18}$$

whose optimized version is

$$f_u = f_{u,Q}(P);\ m_{\text{opt. ate}} = (f_u \cdot \ell_{[u]Q,Q}(P))^q (f_u)_{u^2-u-1,[u]Q}(P)\ . \tag{19}$$

In the two cases $t_{\mathrm{bw},0}$ and $t_{\mathrm{bw},3}$, the cost in terms of multiplications in the base field are the same.

Optimal Pairing Computation with Alg. 2. $\mathbb{G}_1$ and $\mathbb{G}_2$ have an endomorphism ϕ_1, ϕ_2 of eigenvalue $\lambda_{\mathrm{bw},i} = t_{\mathrm{bw},i} - 1 \bmod r_{\mathrm{bw}}$. Low degree polynomials (short scalars once evaluated at a seed u) a_0, a_1 s.t. $a_0 + a_1\lambda_{\mathrm{bw},i} = 0 \bmod r_{\mathrm{bw}}$ are

$$(x^3-x^2-x)+(x+1)(t_{\mathrm{bw},0}-1) = -3r_{\mathrm{bw}} \tag{20}$$
$$-x-1+(x^3-x^2+1)(t_{\mathrm{bw},0}-1) = -3(x^2-2x+2)r_{\mathrm{bw}} \tag{21}$$
$$(x+1)+(x^3-x^2-x)(t_{\mathrm{bw},3}-1) = 3(x-1)^2 r_{\mathrm{bw}} \tag{22}$$
$$(x^3-x^2+1)-(x+1)(t_{\mathrm{bw},3}-1) = -3r_{\mathrm{bw}} \tag{23}$$

The optimal Tate or ate Miller loop with e.g. (21), (23) are:

$$m_{\mathrm{Tate}} = f_{-(u+1)+(u^3-u^2+1)\lambda_{\mathrm{bw},0},P}(Q)\ ,\ m_{\mathrm{ate}} = f_{-(u+1)+(u^3-u^2+1)q_{\mathrm{bw},0},Q}(P) \tag{24}$$
$$m_{\mathrm{Tate}} = f_{u^3-u^2+1-(u+1)\lambda_{\mathrm{bw},3},P}(Q)\ ,\ m_{\mathrm{ate}} = f_{u^3-u^2+1-(u+1)q_{\mathrm{bw},3},Q}(P)\ . \tag{25}$$

$\mathbb{G}_1$ and $\mathbb{G}_2$ membership testing. For $\mathbb{G}_1$ membership testing, one uses one of Eqs. (20), (21), resp. (22), (23), with $x = u$. However, these formulas (e.g. $[u^3 - u^2 - u]P + [u + 1]\phi(P)$) will output $\mathcal{O}$ for any point in the subgroup of order $3r_{\mathrm{bw}}$. For $\mathbb{G}_2$ membership testing, the same equations can be re-used: we showed in Sec. 4.1 that the twisted curve E' of $\mathbb{G}_2$ has the same trace as E modulo r_{bw}, either $(t_{\mathrm{bw},0} - 3y_{\mathrm{bw},0})/2 = t_{\mathrm{bw},0}$, or $(t_{\mathrm{bw},3} + 3y_{\mathrm{bw},3})/2 = t_{\mathrm{bw},3}$.

Final Exponentiation. Writing the hard part of the final exponentiation $z^{\Phi_6(q_{\mathrm{bw},i})/r_{\mathrm{bw}}}$ in terms of x, h_t, h_y, Magma runs LLL on multivariate polynomials and provides the result. With $t_{\mathrm{bw},i}$, LLL gives short vectors for the exponent:

$$e_{\mathrm{bw},i} = 3(x+1)\Phi_k(q_{\mathrm{bw},i})/r_{\mathrm{bw}}(x) \tag{26}$$

and the formulas for $e_{\mathrm{bw},i}$ are

$$e_{\mathrm{bw},0} = 3(c_{\mathrm{bw},0} + h_t)(x^3 - x^2 + 1 - (x+1)q_{\mathrm{bw},0}) - 9(x^2 - 2x + 2 - q_{\mathrm{bw},0}) \tag{27}$$
$$e_{\mathrm{bw},3} = 3(c_{\mathrm{bw},3} + h_t)(x^3 - x^2 - x + (x+1)q_{\mathrm{bw},3}) + 9(x^2 - 2x + 1 + q_{\mathrm{bw},3}) \tag{28}$$

We explicit in Sec. A.3 the link with Hayashida, Hayasaka and Teruya's formulas [34] and show that our formulas are the most efficient.

Cofactor clearing on $\mathbb{G}_1$ with one endomorphism. The cofactors are $c_{\mathrm{bw},0}$ given in Eq. (5), $c_{\mathrm{bw},3}$ in Eq. (6). The curve has an endomorphism defined over $\mathbb{F}_q$, of characteristic polynomial $x^2 + x + 1$ and eigenvalue λ such that $\lambda^2 + \lambda + 1 = 0$ modulo the curve order. There are two formulas, one for each choice of eigenvalue modulo the curve order, and $l_0 + l_1\lambda = 0 \bmod c_{\mathrm{bw},i}$. With $c_{\mathrm{bw},0}$ we have

$$l_0 = (h_t^2 + 3h_y^2)/4 \cdot (x^3 - x^2 + 1) - h_t(x^2 - 2x + 1) - (h_t - 3h_y)/2$$
$$l_1 = (h_t^2 + 3h_y^2)/4 \cdot (x + 1) - (h_t + 3h_y)/2 \cdot (x^2 - 2x + 1) - h_t$$

The alternative formulas for the other choice of eigenvalue $\overline{\lambda}$ are

$$l_0 = (h_t^2 + 3h_y^2)/4 \cdot (x + 1) - (h_t + 3h_y)/2 \cdot (x^2 - 2x + 1) - h_t$$
$$l_1 = (h_t^2 + 3h_y^2)/4 \cdot (x^3 - x^2 + 1) - h_t(x^2 - 2x + 1) - (h_t - 3h_y)/2$$

With $c_{\mathrm{bw},3}$ we have

$$l_0 = (h_t^2 + 3h_y^2)/4 \cdot (x^3 - x^2 + 1) + h_t + (h_t + 3h_y)/2 \cdot (x - 1)^2$$
$$l_1 = (h_t^2 + 3h_y^2)/4 \cdot (x + 1) - (h_t - 3h_y)/2 \cdot (x^2 - 2x + 2) + h_t$$

The alternative formulas for the other choice of eigenvalue $\overline{\lambda}$ are

$$l_0 = (h_t^2 + 3h_y^2)/4 \cdot (x + 1) - (h_t - 3h_y)/2 \cdot (x^2 - 2x + 2) + h_t$$
$$l_1 = (h_t^2 + 3h_y^2)/4 \cdot (x^3 - x^2 + 1) + (h_t + 3h_y)/2 \cdot (x^2 - 2x + 1) + h_t$$

Cofactor clearing on $\mathbb{G}_2$ with one endomorphism. The cofactors $c'_{\mathrm{bw},i}$ are given by Eqs. (7), (8), and in Table 3. We decompose $c'_{\mathrm{bw},i}$ with the eigenvalue of the endomorphism. There are two possible eigenvalues, λ and $\overline{\lambda} = -\lambda - 1$. The formulas of l_0, l_1 satisfy $l_0 + l_1\lambda = 0 \bmod c'_{\mathrm{bw},i}$. With $c'_{\mathrm{bw},0}$ we have

$$
\begin{aligned}
l_0 &= (h_t^2 + 3h_y^2)/4(x+1) + (h_t + 3h_y)/2(x^2 - 2x + 2) - h_t \\
l_1 &= (h_t^2 + 3h_y^2)/4(x^3 - x^2 + 1) - (h_t - 3h_y)/2(x^2 - 2x + 1) - h_t
\end{aligned}
$$

The alternative formulas for the other choice of eigenvalue $\overline{\lambda}$ are

$$
\begin{aligned}
l_0 &= -(h_t^2 + 3h_y^2)/4(x+1) - (h_t + 3h_y)/2(x^2 - 2x + 2) + h_t \\
l_1 &= (h_t^2 + 3h_y^2)/4(x^3 - x^2 - x) - h_t(x^2 - 2x + 1) - (h_t + 3h_y)/2
\end{aligned}
$$

With $c'_{\mathrm{bw},3}$ we have

$$
\begin{aligned}
l_0 &= -(h_t^2 + 3h_y^2)/4(x+1) - (h_t - 3h_y)/2(x^2 - 2x + 1) - h_t \\
l_1 &= (h_t^2 + 3h_y^2)/4(x^3 - x^2 - x) + (h_t + 3h_y)/2(x^2 - 2x + 2) - h_t
\end{aligned}
$$

The alternative formulas for the other choice of eigenvalue $\overline{\lambda}$ are

$$
\begin{aligned}
l_0 &= (h_t^2 + 3h_y^2)/4(x+1) + (h_t - 3h_y)/2(x^2 - 2x + 1) + h_t \\
l_1 &= (h_t^2 + 3h_y^2)/4(x^3 - x^2 + 1) + h_t(x^2 - 2x + 1) + (h_t + 3h_y)/2
\end{aligned}
$$

4.3 BW6 with BLS-24

We follow the same process as for BW6-BLS12 and report the parameters in Table 4.

Pairing computation: Miller Loop. Assuming an endomorphism of eigenvalue $\lambda_{\mathrm{bw},i} = t_{\mathrm{bw},i} - 1$, the formulas are

$$
\begin{aligned}
-x - 1 + (x^5 - x^4 + 1)(t_{\mathrm{bw},0} - 1) &= -3r_{\mathrm{bw}}((x-1)^2(x^2+1) + 1) && (29)\\
x^5 - x^4 - x + (x+1)(t_{\mathrm{bw},0} - 1) &= -3r_{\mathrm{bw}} && (30)\\
x + 1 + (x^5 - x^4 - x)(t_{\mathrm{bw},3} - 1) &= 3r_{\mathrm{bw}}(x-1)^2(x^2+1) && (31)\\
x^5 - x^4 + 1 - (x+1)(t_{\mathrm{bw},3} - 1) &= -3r_{\mathrm{bw}} && (32)
\end{aligned}
$$

and one obtains optimal ate and Tate (a.k.a. twisted ate) pairings from (29), (32)

$$
\begin{array}{ll}
m_{\mathrm{Tate}} = f_{-(u+1)+(u^5-u^4+1)\lambda_{\mathrm{bw},0},P}(Q), & m_{\mathrm{Tate}} = f_{u^5-u^4+1-(u+1)\lambda_{\mathrm{bw},3},P}(Q), \\
m_{\mathrm{ate}} = f_{-(u+1)+(u^5-u^4+1)q_{\mathrm{bw},0},Q}(P), & m_{\mathrm{ate}} = f_{u^5-u^4+1-(u+1)q_{\mathrm{bw},3},Q}(P).
\end{array} \quad (33)
$$

Table 4. Parameters of a BW6 outer curve with a BLS24 inner curve, with $x \equiv 1 \bmod 3$.

r_{bw}	$q_{\mathrm{bls}} = (x-1)^2/3(x^8-x^4+1)+x$ $(x^{10}-2x^9+x^8-x^6+2x^5-x^4+x^2+x+1)/3$	prime
$\overline{\zeta_6}$	$-x^9+3x^8-4x^7+4x^6-3x^5+2x^3-2x^2+x-1$	
ζ_6	$x^9-3x^8+4x^7-4x^6+3x^5-2x^3+2x^2-x+2$	
$1/\sqrt{-3}$	$(2x^9-6x^8+8x^7-8x^6+6x^5-4x^3+4x^2-2x+3)/3$	
$t_{\mathrm{bw},0}$	$-x^9+3x^8-4x^7+4x^6-3x^5+2x^3-2x^2+x$	$6 \mid t_{\mathrm{bw},0}$
$t_{\mathrm{bw},3}$	$x^9-3x^8+4x^7-4x^6+3x^5-2x^3+2x^2-x+3$	$3 \mid t_{\mathrm{bw},3}, 2 \nmid t_{\mathrm{bw},3}$
$y_{\mathrm{bw},0}$	$(x^9-3x^8+4x^7-4x^6+3x^5-2x^3+2x^2-x)/3$	
$y_{\mathrm{bw},0}$	$-t_{\mathrm{bw},0}/3$	$2 \mid y_{\mathrm{bw},0}$
$y_{\mathrm{bw},3}$	$(x^9-3x^8+4x^7-4x^6+3x^5-2x^3+2x^2-x+3)/3$	
$y_{\mathrm{bw},3}$	$t_{\mathrm{bw},3}/3$	$2 \nmid y_{\mathrm{bw},3}$
$q_{\mathrm{bw},0}$	$((t_{\mathrm{bw},0}+h_t r_{\mathrm{bw}})^2+3(y_{\mathrm{bw},0}+h_y r_{\mathrm{bw}})^2)/4$	prime
$q_{\mathrm{bw},3}$	$((t_{\mathrm{bw},3}+h_t r_{\mathrm{bw}})^2+3(y_{\mathrm{bw},3}+h_y r_{\mathrm{bw}})^2)/4$	prime
$\Phi_6(t_{\mathrm{bw},i}-1)$	$(x^8-4x^7+8x^6-12x^5+15x^4-14x^3+10x^2-6x+3)\cdot 3\cdot r_{\mathrm{bw}}$	
$c_{\mathrm{bw},0}$	$(h_t^2+3h_y^2)/4r_{\mathrm{bw}}+(h_t-h_y)/2t_{\mathrm{bw},0}+\Phi_6(t_{\mathrm{bw},0}-1)/(3r_{\mathrm{bw}})-h_t$	
$c_{\mathrm{bw},3}$	$(h_t^2+3h_y^2)/4r_{\mathrm{bw}}+(h_t+h_y)/2t_{\mathrm{bw},3}+\Phi_6(t_{\mathrm{bw},i}-1)/(3r_{\mathrm{bw}})-h_t$	
$c'_{\mathrm{bw},0}$ $(\mathbb{G}_2)$	$c_{\mathrm{bw},0}+(h_t+3h_y)/2$	
$c'_{\mathrm{bw},3}$ $(\mathbb{G}_2)$	$c_{\mathrm{bw},3}+(h_t-3h_y)/2$	

Pairing computation: Final Exponentiation. Like for BLS12-BW6, the hard part can be expressed in terms of $q_{\mathrm{bw},i}, h_t, h_y$. One obtains two cases. Note that according to Table 2, $(h_t^2+3_y^2)/4$ and $(h_t-h_y)/2$ are integers. With the parameters of Table 4, the exponent $(q_{\mathrm{bw},i}^2-q_{\mathrm{bw},i}+1)/r_{\mathrm{bw}}$ multiplied by $3(x+1)$ has coefficients of low degree in x in basis $q_{\mathrm{bw},i}$. The highest power to compute is u^{15} due to $c_{\mathrm{bw},i}$ of degree 10 in u. The two cases have very similar formulas.

$$(-x^5+x^4-1+(x+1)q_{\mathrm{bw},0})3(c_{\mathrm{bw},0}+h_t)+9(x^4+2(-x^3+x^2-x+1)-q_{\mathrm{bw},0}),$$
$$(x(x^4-x^3-1)+(x+1)q_{\mathrm{bw},3})3(c_{\mathrm{bw},3}+h_t)+9(x^4+2(-x^3+x^2-x)+1+q_{\mathrm{bw},3})\ .$$

4.4 Two-chains with inner BLS and outer Cocks-Pinch

Section 4.2 showed that a Brezing-Weng outer curve of embedding degree $k=6$ is optimal with a BLS-12 curve whose prime-order subgroup is about 256 bits long. However BW6 is no longer optimal with BLS24 over a prime field of about 320 bits: we measure the security in the finite field $\mathbb{F}_{q^6}$ whose q is roughly 640 bits long to be about 124 bits in Sec. 5.3. To increase the security in the finite field $\mathbb{F}_{q^k}$, we can increase the size of the prime q thanks to the choice of lifting co-factors h_t, h_y, and obtain a q of 672 bits, or we can increase the embedding degree k, but then the BW construction is no longer available: we move to the Cocks-Pinch construction. To allow twist optimisation, we focus on $k=8$ with $D=1$ (quartic twist) and $k=12$ with $D=3$ (sextic twist). Our Cocks-Pinch curves are similar to the curves of Guillevic, Masson and Thomé [32] (see also [43, Chapter 5]). The lifting cofactor idea appeared before in Fotiadis and Konstantinou paper [20].

With the Cocks-Pinch construction of embedding degree not 6, the optimal ate pairing like for BW6 curves is no longer available because the eigenvalue

of the Frobenius endomorphism π_q on a CP curve $E(\mathbb{F}_{q^k})$ does not have a simple polynomial form modulo the subgroup order $r_{\rm cp} = q_{\rm bls}$. In other words, there is no k-th root of unity modulo $q_{\rm bls}(x)$ (as polynomials). However, π_q has an eigenvalue (as a scalar integer) modulo $r_{\rm cp}(u) \in \mathbb{Z}$, and one can use the LLL algorithm to obtain a decomposition with short scalars a_i, of size $r_{\rm cp}^{1/4}$: $a_0 + a_1 q_{\rm cp} + a_2 q_{\rm cp}^2 + a_3 q_{\rm cp}^3 = 0 \bmod r_{\rm cp}$. This 4-fold holds for CP8 and CP12 curves as $\varphi(8) = \varphi(12) = 4$. The optimal ate Miller loop would be

$$f_{a_0,Q}(P) f_{a_1,Q}^{q}(P) f_{a_2,Q}^{q^2}(P) f_{a_3,Q}^{q^3}(P) \ell_{a_0 Q, a_1 \pi_q(Q)}(P) \ell_{a_2 \pi_{q^2}(Q), a_3 \pi_{q^3}(Q)}$$

But the scalars a_i are not sparse and none of them is trivial, contrary to [51]. Instead, we generalize our Alg. 2 and obtain Alg. 4. Algorithm 3 precomputes the data and Alg. 4 computes the pairing, with the formulas (12)–(15) adapted to the ate pairing with swapped P and Q and $\lambda = q$, and

$$f_{2(\sum_i c_i q^i),Q}(P) = f_{\sum_i c_i q^i,Q}^2(P) \ell_{[\sum_i c_i q^i]Q,[\sum_i c_i q^i]Q}(P) \tag{34}$$

$$\begin{aligned} &f_{(\sum_i c_i q^i)+q^j+q^l+q^m,Q}(P) = f_{\sum_i c_i q^i,Q}(P) f_{q^j+q^l+q^m,Q}(P) \ell_{[\sum_i c_i q^i]Q,[q^j+q^l+q^m]Q}(P) \\ &= f_{\sum_i c_i q^i,Q}(P) \ell_{[\sum_i c_i q^i]Q,[q^j+q^l+q^m]Q}(P) \ell_{[q^j+q^l]Q,[q^m]Q}(P) \ell_{[q^j]Q,[q^l]Q}(P) \end{aligned} \tag{35}$$

$$\begin{aligned} &f_{(\sum_i c_i q^i)+1+q+q^2+q^3,Q}(P) = f_{\sum_i c_i q^i,Q}(P) f_{1+q+q^2+q^3,Q}(P) \ell_{[\sum_i c_i q^i]Q,[1+q+q^2+q^3]Q}(P) \\ &= f_{\sum_i c_i q^i,Q}(P) \ell_{[\sum_i c_i q^i]Q,[1+q+q^2+q^3]Q}(P) \\ &\cdot \ell_{[1+q]Q,[q^2+q^3]Q}(P) \ell_{Q,[q]Q}(P) \ell_{[q^2]Q,[q^3]Q}(P) \end{aligned} \tag{36}$$

The $f_{q^j,Q}(P)$ terms can be removed [35] and the point $[q^j]Q$, $[q^j+q^l]Q$, $[q^j+q^l+q^m]Q$, $[1+q+q^2+q^3]Q$, and lines $\ell_{[q^m]Q,[q^n]Q}(P)$, $\ell_{[q^j+q^l]Q,[q^m]Q}(P)$, $\ell_{[1+q]Q,[q^2+q^3]Q}(P)$, and their products, are precomputed.

On CP8 curves, $\mathbb{G}_1$ has an endomorphism ϕ: $(x, y) \mapsto (-x, \sqrt{-1}y)$ of eigenvalue $\lambda \equiv q^2 \bmod r$, $\lambda^2 \equiv -1 \bmod r$. On CP12 curves, $\mathbb{G}_1$ has the same endomorphism as BW6 curves, of eigenvalue $\lambda \equiv q^2 \bmod r$. The twisted ate pairing on our CP curves has Miller loop $f_{\lambda,P}(Q) = f_{q^2,P}(Q)$, and we derive our optimal Tate pairing like for BW6 curves, with short scalars $a_0 + a_1\lambda \equiv 0 \bmod r$.

4.5 Comparison of BW6, CP8 and CP12 outer curve performances

We reproduce the field arithmetic estimates from [32,18] in Table 5 and the pairing cost estimates in Table 6. Parameters of CP8 and CP12 curves are given in Table 7. Parameters of BW6 curves can be found in Table 11. We justify our choice of seeds and curve parameters in Sec. 5. We obtain Table 8 for ate and Tate pairing estimates for our BW6 and CP curves. We obtain a speed-up of the optimal ate pairing on BW6 curves compared to [18] with the formula (37) with $v = u^2 - 2u + 1$ for BLS12-BW6 and $v = u^4 - 2(u^3 - u^2 + u) + 1$ for BLS24-BW6 because the 2-NAF Hamming weight of the scalar v is lower:

$$f_{u+1} = f_{u+1,Q}(P); \quad m_{\text{opt. ate}} = (f_{u+1})_{v,[u+1]Q}^{q}(P) \ell_{[(u+1)v]Q,-Q}^{q}(P) \ . \tag{37}$$

BW6 curves as outer curves of BLS24 have a faster pairing than CP8 and CP12 curves: *a larger characteristic gives better performances than a larger embedding*

Algorithm 3: Precomputations of sums of points and lines

Input: $P \in E(\mathbb{F}_q)[r]$, $Q_0, Q_1, Q_2, Q_3 \in E'(\mathbb{F}_{q^{k/d}})[r]$

Output: array T of length 15, of precomputed points and lines

$T \leftarrow$ array of length 15
for $i = 0$ *to* 3 **do**
    $T[2^i - 1][0] \leftarrow Q_i$; $T[2^i - 1][1] \leftarrow 1$
for $0 \leq m < n \leq 3$ **do**
    $i \leftarrow 2^m + 2^n$
    $T[i-1][0] \leftarrow T[2^m - 1] + T[2^n - 1]$
    $T[i-1][1] \leftarrow \ell_{Q_m, Q_n}(P)$
for $0 \leq m < n < s \leq 3$ **do**
    $i \leftarrow 2^m + 2^n + 2^s$
    $T[i-1][0] \leftarrow T[2^m + 2^n - 1][0] + T[2^s - 1][0]$
    $T[i-1][1] \leftarrow T[2^m + 2^n - 1][1] \cdot \ell_{Q_m + Q_n, Q_s}(P)$
$T[15-1][0] \leftarrow T[7-1][0] + T[8-1]$
$T[15-1][1] \leftarrow T[7-1][1] \cdot \ell_{Q_0+Q_1+Q_2, Q_3}(P)$
return T

Algorithm 4: Miller loop for optimal ate pairing, Cocks-Pinch

Input: $P \in \mathbb{G}_1 = E(\mathbb{F}_q)[r]$, $Q \in \mathbb{G}_2 = \ker(\pi_q - [q]) \cap E(\mathbb{F}_{q^k})[r]$, scalars a_0, a_1, a_2, a_3 such that $a_0 + a_1 q + a_2 q^2 + a_3 q^3 = 0 \bmod r$

Output: $f_{a_0 + a_1 q + a_2 q^2 + a_3 q^3, Q}(P)$

$Q_0 \leftarrow Q$; $Q_1 \leftarrow \pi_q(Q)$; $Q_2 \leftarrow \pi_{q^2}(Q)$; $Q_3 \leftarrow \pi_{q^3}(Q)$
for $i = 0$ *to* 3 **do**
    if $a_i < 0$ **then** $a_i \leftarrow -a_i$; $Q_i \leftarrow -Q_i$
$T \leftarrow$ precomputations(Q_0, Q_1, Q_2, Q_3)
$l_i \leftarrow bits(a_i)$ for $0 \leq i \leq 3$
$i \leftarrow \max_{0 \leq j \leq 3}(\operatorname{len} l_j)$
$j \leftarrow l_{0,i} + 2l_{1,i} + 4l_{2,i} + 8l_{3,i}$
$f \leftarrow T[j-1][1]$
$S \leftarrow T[j-1][0]$
for $i = i - 1$ *downto* 0 **do**
    $f \leftarrow f^2$
    $\ell_t \leftarrow \ell_{S,S}(P)$; $S \leftarrow [2]S$
    $j \leftarrow l_{0,i} + 2l_{1,i} + 4l_{2,i} + 8l_{3,i}$
    if $j > 0$ **then**
        $Q_j \leftarrow T[j-1][0]$; $\ell \leftarrow \ell_{S,Q_j}(P)$; $S \leftarrow S + Q_j$
        $f \leftarrow f \cdot (\ell_t \cdot \ell)$
        if $T[j-1][1] \neq 1$ **then** $f \leftarrow f \cdot T[j-1][1]$
    else $f \leftarrow f \cdot \ell_t$
return f

degree. Assuming a ratio $\mathbf{m}_{704}/\mathbf{m}_{640} = 1.25$, an ate Miller loop on CP8-632 is 25% slower compared to BW6-672, but the final exp. is 15% faster. A full pairing on CP8 is about 7% slower, and 59% slower on CP12. BLS24-BW6 has a faster pairing than BLS12-BW6, but the 2-adicity of BLS24 curves is much smaller.

Table 5. Cost from [32, Table 6] of $\mathbf{m}_k$, $\mathbf{s}_k$ and $\mathbf{i}_k$ for finite field extensions $\mathbb{F}_{p^k}$. Inversions in $\mathbb{F}_{p^{ik}}$ come from $\mathbf{i}_{2k} = 2\mathbf{m}_k + 2\mathbf{s}_k + \mathbf{i}_k$ and $\mathbf{i}_{3k} = 9\mathbf{m}_k + 3\mathbf{s}_k + \mathbf{i}_k$. $\mathbb{F}_{p^{12}}$, resp. $\mathbb{F}_{p^{24}}$ always have a first quadratic, resp. quartic extension, $\mathbf{i}_{24} = 2\mathbf{m}_{12} + 2\mathbf{s}_{12} + \mathbf{i}_{12} = 293\mathbf{m} + \mathbf{i}$ with $\mathbf{i}_{12} = 9\mathbf{m}_4 + 3\mathbf{s}_4 + \mathbf{i}_4$, and for $\mathbb{F}_{p^{12}}$, $\mathbf{i}_{12} = 2\mathbf{m}_6 + 2\mathbf{s}_6 + \mathbf{i}_6 = 97\mathbf{m} + \mathbf{i}$ with $\mathbf{i}_6 = 9\mathbf{m}_2 + 3\mathbf{s}_2 + \mathbf{i}_2$.

k	1	2	3	4	6	8	12	24
$\mathbf{m}_k$	$\mathbf{m}$	$3\mathbf{m}$	$6\mathbf{m}$	$9\mathbf{m}$	$18\mathbf{m}$	$27\mathbf{m}$	$54\mathbf{m}$	$162\mathbf{m}$
$\mathbf{s}_k$	$\mathbf{m}$	$2\mathbf{m}$	$5\mathbf{m}$	$6\mathbf{m}$	$12\mathbf{m}$	$18\mathbf{m}$	$36\mathbf{m}$	$108\mathbf{m}$
$\mathbf{f}_k$	0	0	$2\mathbf{m}$	$2\mathbf{m}$	$4\mathbf{m}$	$6\mathbf{m}$	$10\mathbf{m}$	$22\mathbf{m}$
$\mathbf{s}_k^{\text{cyclo}}$	–	$2\mathbf{s}$	–	$4\mathbf{m}$	$6\mathbf{m}$	$12\mathbf{m}$	$18\mathbf{m}$	$54\mathbf{m}$
$\mathbf{i}_k - \mathbf{i}_1$	0	$2\mathbf{m} + 2\mathbf{s}$	$9\mathbf{m} + 3\mathbf{s}$	$14\mathbf{m}$	$34\mathbf{m}$	$44\mathbf{m}$	$97\mathbf{m}$	$293\mathbf{m}$
$\mathbf{i}_k$, with $\mathbf{i}_1 = 25\mathbf{m}$	$25\mathbf{m}$	$29\mathbf{m}$	$37\mathbf{m}$	$39\mathbf{m}$	$59\mathbf{m}$	$69\mathbf{m}$	$119\mathbf{m}$	$318\mathbf{m}$

Table 6. Miller loop cost in non-affine, Weierstrass model [16,3]. For $6 \mid k$, two sparse-dense multiplications cost $26\mathbf{m}_{k/6}$ whereas one sparse-sparse and one multiplication cost $6\mathbf{m}_{k/6} + \mathbf{m}_k = 24\mathbf{m}_{k/6}$. For $4 \mid k$, this is $16\mathbf{m}_{k/4}$ compared to $6\mathbf{m}_{k/4} + \mathbf{m}_k = 15\mathbf{m}_{k/4}$.

k	D	curve	DoubleLine and AddLine	ref	SparseM and SparseSparseM
$6 \mid k$	-3	$Y^2 = X^3 + b$ sextic twist	$3\mathbf{m}_{k/6} + 6\mathbf{s}_{k/6} + (k/3)\mathbf{m}$ $11\mathbf{m}_{k/6} + 2\mathbf{s}_{k/6} + (k/3)\mathbf{m}$	[3, §4]	$13\mathbf{m}_{k/6}$ $6\mathbf{m}_{k/6}$
$4 \mid k$	-1	$Y^2 = X^3 + ax$ quartic twist	$2\mathbf{m}_{k/4} + 8\mathbf{s}_{k/4} + (k/2)\mathbf{m}$ $9\mathbf{m}_{k/4} + 5\mathbf{s}_{k/4} + (k/2)\mathbf{m}$	[16, §4]	$8\mathbf{m}_{k/4}$ $6\mathbf{m}_{k/4}$

Table 7. CP8 and CP12 outer curve parameters on top of BLS24-315

outer curve	u	(h_t, h_y)	$(t-1)^2 + 1 \bmod r, u$	equation	$\mathbb{F}_{q^k}$ (bits)	est. DL in $\mathbb{F}_{q^k}$
BLS24-315-CP8-632	`-0xbfcfffff`	(6,2)	–	$y^2 = x^3 - x$	5056	140
BLS24-315-CP12-630	`-0xbfcfffff`	(1,2)	0	$y^2 = x^3 - 1$	7560	166

5 Implementation and benchmarking

In previous sections, we presented families of SNARK-friendly 2-chains that are suitable for Groth'16 and KZG-based universal SNARKs. These families

Table 8. Pairing cost estimates on BLS12-BW6, BLS24-BW6, BLS24-CP8, BLS24-CP12 curves. BLS12-BW6 curves use Eq. (22) with [18, Alg. 5], and $v = u^2 - 2u + 1$. BLS24-BW6 curves use Eq (30), (31) with $v = u^4 - 2(u^3 - u^2 + u) + 1$.

	BLS12-377-BW6-761	BLS12-379-BW6-764
ate $f_{u+1,Q}(f_u)^q_{u^2-u-1,[u]Q}$	$7863\mathbf{m}_{768}$	$7653\mathbf{m}_{768}$
ate $f_{u+1,Q}(f_{u+1})^q_{v,[u+1]Q}\ell^q_{(u+1)vQ,-Q}$	$7555\mathbf{m}_{768}$	$7389\mathbf{m}_{768}$
Tate $f_{u+1+(u^3-u^2-u)\lambda,P}$ Alg. 2	$7729\mathbf{m}_{768}$	$7540\mathbf{m}_{768}$
Final exp. [18, § 3.3, Tab. 7]	$5081\mathbf{m}_{768}$	–
Final exp. Eq. (28)	$5195\mathbf{m}_{768}$	$5033\mathbf{m}_{768}$
	BLS24-315-BW6-633	BLS24-315-BW6-672
ate $f_{u+1,Q}(f_{u+1})^q_{v,[u+1]Q}\ell^q_{(u+1)vQ,-Q}$	$7285\mathbf{m}_{640}$	$7285\mathbf{m}_{704}$
Tate $f_{u+1+(u^5-u^4-u)\lambda,P}$ Alg. 2	$6813\mathbf{m}_{640}$	$6813\mathbf{m}_{704}$
Final exp.	$5027\mathbf{m}_{640}$	$5501\mathbf{m}_{704}$
	BLS24-315-CP8-632	BLS24-315-CP12-630
ate $f_{a_0+a_1q+a_2q^2+a_3q^3,Q}$ Alg. 4	$10679\mathbf{m}_{640}$	$13805\mathbf{m}_{640}$
Tate $f_{a_0+a_1\lambda,P}$ Alg. 2	$12489\mathbf{m}_{640}$	$15780\mathbf{m}_{640}$
Final exp.	$5835\mathbf{m}_{640}$	$10312\mathbf{m}_{640}$

are composed of BLS12 and BLS24 inner curves and BW6, CP8 and CP12 outer curves. We demonstrated that the pair family BLS12/BW6 is suitable for recursive Groth'16 applications and meets the best security/performance tradeoff. Similarly, we showed that BLS24/BW6 is suitable for KZG-based universal SNARKs. We also investigated the family pairs BLS24/CP8 and BLS24/CP12 as more conservative choices and showed that CP8-632 is competitive with BLS24/BW6-672. BW6-633, CP8 and CP12 are defined over a base field of roughly the same bit length, and all have a GLV endomorphism, hence performances on $\mathbb{G}_1$ are expected to be the same. On $\mathbb{G}_2$, BW6 are always faster because they are defined over the same base field as $\mathbb{G}_1$, contrary to CP curves. For the pairing computation, as discussed in 4.5, CP8 and CP12 are slower than both choices of BW6. Therefore, we have chosen to focus our benchmarks on BLS12/BW6 and BLS24/BW6 families of curves.

In this section, we first present an open-sourced SageMath library to derive these curves and test our generic formulas. Then, based on additional practical criteria, we recommend a short list of SNARK-friendly 2-chains. Finally, we implement in the open-sourced `gnark` ecosystem [15] this short-list. We benchmark the relevant curve operations in $\mathbb{G}_1$ and $\mathbb{G}_2$, and the pairings, and compare efficiency of all choices in practical Groth'16 setting and PlonK setting, which is a popular KZG-based universal SNARK. Both schemes are implemented in `gnark` and maintained by ConsenSys zkTeam.

5.1 SageMath library: Derive the curves

In this Git repository [19], we present SageMath scripts to derive all the SNARK-friendly 2-chain families and verify the formulae presented in sections 3 and 4, and the pairing cost estimates of Table 8.

5.2 Our short-list of curves

For all curves, in addition to SNARK-friendliness and security level λ, we shall consider the following properties:

- A seed u with low Hamming weight $\mathrm{HW}(u)$, allowing fast Miller loops in pairings.
- Isogenies of low degree d from a curve with j-invariant different from 0 and 1728, allowing use of the "simplified SWU" method for hashing to the curve [52].
- Small integer α relatively prime to $r-1$, allowing the use of x^α as an S-box in the algebraic SNARK-hashes (e.g. Poseidon [28]).
- Small non-residues in $\mathbb{F}_q$, for an efficient tower arithmetic.
- "Spare" bits in $\mathbb{F}_q$, for carries, infinity point or compressed point flag.

For outer curves, an additional property is

- Smallest $h_t^2 + 3h_y^2$ with low Hamming weight, allowing fast final exponentiation.

BLS12/BW6. The security of BLS12-384 and BLS12-448 is explained in [33,31], BLS12-448 being presented as a more conservative choice: it offers about 132 bits of security in $\mathbb{F}_{q^{12}}$ instead of 126 bits. Because a BLS12-448 would imply a much larger BW6-896, we concentrate on the BLS12 curves of 377 to 383 bits of Table 9. Given the above requirements, we short-list BLS12-377 with $u = \texttt{0x8508c00000000001}$ and BLS12-379 with $u = \texttt{0x9b04000000000001}$. The former was proposed in [12] and used in [18] and the latter is new, of higher 2-adicity. Both have a $\mathrm{HW}(u) = 7$, $d = 2$, $\alpha \leq 7$ and tower fields can be constructed as $\mathbb{F}_q \xrightarrow{i^2+5} \mathbb{F}_{q^2} \xrightarrow{v^3-i} \mathbb{F}_{q^6} \xrightarrow{w^2-v} \mathbb{F}_{q^{12}}$.

Now, we construct outer BW6 curves to these inner BLS12 curves. For BLS12-377, we find BW6-761 to be optimal and refer the reader to [18] for a more detailed study. For BLS12-379, we restrict the search to curves up to 768 bits and suggest the corresponding BW6-764 with $h_t = -23$, $h_y = 3$ and equation $Y^2 = X^3 + 1$ (and M-twist $Y^2 = X^3 + 2$). Both BW6-761 and BW6-764 fall in the $t_{\mathrm{bw},3}$ case (Table 3).

Table 9. Seeds of SNARK-friendly inner BLS12 curves around 128 bits of security.

u	q (bits)	r (bits)	λ $E(\mathbb{F}_q)$	λ $\mathbb{F}_{q^{12}}$	2-adicity L	d	α
`0x8508c00000000001`	377	253	126	126	47	2	11
`-0x7fb80fffffffffff`	377	252	126	126	45	2	5
`0x9b04000000000001`	379	254	127	126	51	2	7
`-0xfffbc3ffffffffff`	383	256	128	126	43	2	7
`-0xfff7c1ffffffffff`	383	256	128	126	42	2	7
`-0xffc3bfffffffffff`	383	256	128	126	47	2	7
`0x105a8000000000001`	383	257	128	126	52	2	7

BLS24/BW6. A BLS24 curve defined over a 320-bit prime field offers 128 bits of security on the curve thanks to a subgroup of prime order r of 256 bits, and offers around 160 bits in $\mathbb{F}_{q^{24}}$. Accordingly, we find the following SNARK-friendly inner BLS24 curves (Table 10). Given all the requirements, we choose BLS24-

Table 10. Seeds of SNARK-friendly inner BLS24 curves around 128 bits of security.

u	q (bits)	r (bits)	λ $E(\mathbb{F}_q)$	λ $\mathbb{F}_{q^{24}}$	2-adicity L	d	α
`0x60300001`	305	245	122	158	22	2	7
`-0x950fffff`	311	250	125	159	22	2	7
`0x9f9c0001`	312	251	125	159	20	2	7
`-0xbfcfffff`	315	253	126	160	22	2	7
`-0xc90bffff`	315	254	126	160	20	2	13
`0xe19c0001`	317	255	127	160	20	2	17
`-0x10487ffff`	319	257	128	161	21	2	11

315 (u = `-0xbfcfffff`) over $\mathbb{F}_q$ of 315 bits and with $\mathbb{F}_r$ of 253 bits. It has 2-adicity 22 and security level almost 128. The tower fields can be constructed as $\mathbb{F}_q \xrightarrow{i^2-13} \mathbb{F}_{q^2} \xrightarrow{v^2-i} \mathbb{F}_{q^4} \xrightarrow{w^2-v} \mathbb{F}_{q^8} \xrightarrow{c^3-w} \mathbb{F}_{q^{24}}$.

Now, we construct outer BW6 curves to BLS24-315. First, we search for less conservative curves over a field of up to 640 bits. We recommend the BW6-633 curve with $h_y = -7, h_y = -1$ and the equation $Y^2 = X^3 + 4$ (and M-twist $Y^2 = X^3 + 8$). For more conservative curves offering 128 bits of security, we search for q_{bw} of exactly 672 bits. We recommend the BW6-672 curve with $h_t = 5111800, h_y = 0$ ($\text{HW}_{\text{2-NAF}}(h_t^2 + 2h_y^2) = 8$) and equation $Y^2 = X^3 - 4$ (D-twist $Y^2 = X^3 - 4/3$). The former falls in the $t_{\text{bw},0}$ and the latter in the $t_{\text{bw},3}$ case.

Table 11. BW6 outer curve parameters, where $y^2 = x^3 + b$.

outer curve	u	(h_t, h_y)	$t \bmod r, u$	b	$\mathbb{F}_{q^k}$ (bits)	est. DL in $\mathbb{F}_{q^k}$
BLS12-377-BW6-761	`0x8508c00000000001`	(13, 9)	0	-1	4566	126
BLS12-379-BW6-764	`0x9b04000000000001`	(-25, 3)	0	1	4584	126
BLS24-315-BW6-633	`-0xbfcfffff`	(- 7,-1)	0	4	3798	124
BLS24-315-BW6-672	`-0xbfcfffff`	(`0x4dfff8`,0)	0	-4	4032	128

5.3 Estimated complexity of a DL computation in GF(q^k)

This section recalls the results from [5,33,30]. A BLS12 curve with r of about 256 bits has q of about 384 bits. In [33, Table 10] the estimated security in $\mathbb{F}_{q^{12}}$ for the BLS12-381 curve is 126 bits. Running the tool from [33], the paper [18] shows that BLS12-377 in $\mathbb{F}_{q^{12}}$ has 125 bits of security, and BW6-761 has 126 bits

of security in $\mathbb{F}_{q^6}$. With the same approach and the SageMath tool[6] from [33], our BLS12-379 curve has 125 bits in $\mathbb{F}_{q^{12}}$ and our BLS24-315 curve has 160 bits of security in $\mathbb{F}_{q^{24}}$.

We observe a notable difference between the BW6 outer curves of BLS12 and BLS24 because of the degree of the polynomial $q_{\text{bw}}(x)$. This polynomial is the key-ingredient of the Special (Tower) NFS [36,40]. However when its degree is too high, the general (Tower) NFS performs better, unless a tweak of $q_{\text{bw}}(x)$ is possible [30]. This tweak divides by n the degree of $q_{bw}(x)$ while increasing its coefficients by at most u^{n-1}. It works only if either q_{bw} has an automorphism of degree n, hence the new polynomial has coefficients as small as the initial one, or the seed u is small enough. Here q_{bw} has no automorphism, and u is 32 bits long. We obtain a new $\tilde{q}_{\text{bw}}(x)$ of degree 10 and coefficients of 40 bits. The lowest estimate of DL cost with STNFS is 2^{132} with h of degree 6 for the 633-bit curve. The general TNFS works slightly better: with h of degree 2, and the Conjugation method (Conj), we obtain a DL cost estimate of 2^{124}. This is coherent with MNT-6 curve security estimates, where the same choice of parameters for TNFS apply [32, Fig. 1]. To reach the 2^{128} cost, we increase q_{bw} up to 672 bits. We stress that the tool we use only gives an estimate, and recent progress are being made about TNFS [17]. In case of underestimate of the tool, one can consider a 704-bit BLS24-BW6 curve.

For the Cocks-Pinch construction, the parameters do not have a polynomial form. For the embedding degree 8 we consider the TNFS-Conj algorithm with h of degree 2 according to [32, Fig. 2]. We obtain 140 bits of security in $\mathbb{F}_{q^8}$ for the BLS24-315-CP8-632 curve. For the BLS24-315-CP12-630 curve we measure a DL cost of 166 bits in $\mathbb{F}_{q^{12}}$ with TNFS-Conj and h of degree 3 for the tower.

5.4 Golang library: our implementation of our short-list

In this Git repository [1], we present an optimized implementation, with `x86` assembly code for the finite fields, of the short-listed curves: BLS12-377, BW6-761, BLS12-379, BW6-764, BLS24-315, BW6-633 and BW6-672 (Table 12). All curve implementations are written in Golang (tested with 1.16 and 1.17 versions) and benefit from $\mathbb{F}_q$ and $\mathbb{F}_r$ `x86` assembly accelerated arithmetic. Also, they benefit from $D = -3$ endomorphism-based optimizations (GLV and 2-dimensional GLS scalar multiplication, fast subgroup checks and cofactor clearing). For the pairing, we follow optimizations from [2,47,27,34] and section 4.

5.5 Benchmarking

In this section, we benchmark our Golang implementation for all short-listed curves on two levels. First, independently from the context, we benchmark $\mathbb{G}_1$, $\mathbb{G}_2$ scalar multiplications (with GLV/GLS acceleration [25,24] and multi-scalar-multiplication (Bucket-list method [9, section 4]). Also, we benchmark the pairing computation (Miller loop, Final exponentiation and total pairing). Then, we

[6] SageMath code available at `https://gitlab.inria.fr/tnfs-alpha/alpha`

Table 12. Short-listed curves.

curve	equation	twist equation	tower fields
BLS12-377	$Y^2 = X^3 + 1$	$Y^2 = X^3 + 1/i$	$\mathbb{F}_q \xrightarrow{i^2+5} \mathbb{F}_{q^2} \xrightarrow{v^3-i} \mathbb{F}_{q^6} \xrightarrow{w^2-v} \mathbb{F}_{q^{12}}$
BLS12-379	$Y^2 = X^3 + 1$	$Y^2 = X^3 + 1/(5+i)$	$\mathbb{F}_q \xrightarrow{i^2+5} \mathbb{F}_{q^2} \xrightarrow{v^3-i} \mathbb{F}_{q^6} \xrightarrow{w^2-v} \mathbb{F}_{q^{12}}$
BLS24-315	$Y^2 = X^3 + 1$	$Y^2 = X^3 + 1/i$	$\mathbb{F}_q \xrightarrow{i^2-13} \mathbb{F}_{q^2} \xrightarrow{v^2-i} \mathbb{F}_{q^4} \xrightarrow{w^2-v} \mathbb{F}_{q^8} \xrightarrow{c^3-w} \mathbb{F}_{q^{24}}$
BLS12-377-BW6-761	$Y^2 = X^3 - 1$	$Y^2 = X^3 + 4$	$\mathbb{F}_q \xrightarrow{i^3+4} \mathbb{F}_{q^3} \xrightarrow{v^2-i} \mathbb{F}_{q^6}$
BLS12-379-BW6-764	$Y^2 = X^3 + 1$	$Y^2 = X^3 + 2$	$\mathbb{F}_q \xrightarrow{i^3-2} \mathbb{F}_{q^3} \xrightarrow{v^2-i} \mathbb{F}_{q^6}$
BLS24-315-BW6-633	$Y^2 = X^3 + 4$	$Y^2 = X^3 + 8$	$\mathbb{F}_q \xrightarrow{i^3-2} \mathbb{F}_{q^3} \xrightarrow{v^2-i} \mathbb{F}_{q^6}$
BLS24-315-BW6-672	$Y^2 = X^3 - 4$	$Y^2 = X^3 - 4/3$	$\mathbb{F}_q \xrightarrow{i^3-3} \mathbb{F}_{q^3} \xrightarrow{v^2-i} \mathbb{F}_{q^6}$

benchmark the time to setup, prove and verify Groth'16 and PlonK proofs of circuits with different number of constraints.

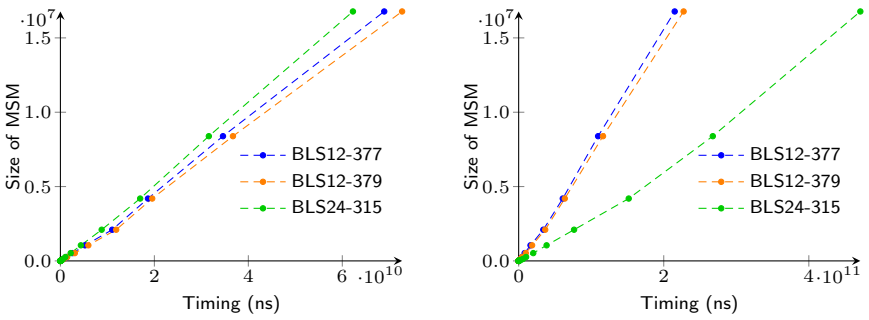
The first level benchmarks are run on a AWS z1d.large (3.4 GHz Intel Xeon) and the second level on a an AWS c5a.24xlarge (AMD EPYC 7R32). This allows to handle large proofs and to test different architectures. All with hyperthreading, turbo and frequency scaling disabled.

$\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ operations. $\mathbb{G}_1$ coordinates for all short-listed curves are over $\mathbb{F}_q$ and use $D = -3$ endomorphism to implement GLV [25]. For $\mathbb{G}_2$, BW6 coordinates are over $\mathbb{F}_q$ as well and implements GLV. For BLS12 and BLS24, the implementation uses 2-dim. GLS [24] over $\mathbb{F}_{q^2}$ and $\mathbb{F}_{q^4}$ respectively. Timings are reported in Tables 13 and 14. For multi-scalar-multiplication, we report timings in figures 1 and 2 for different sizes (2^5 to 2^{24} points).

Table 13. $\mathbb{G}_1$ and $\mathbb{G}_2$ scalar multiplication benchmarks.

curve	$\mathbb{G}_1$ scalar mul. (ns)	$\mathbb{G}_2$ scalar mul. (ns)
BLS12-377	77606	261607
BLS12-379	81090	272107
BLS24-315	65825	622044
BLS12-377-BW6-761	377360	377360
BLS12-379-BW6-764	390647	390647
BLS24-315-BW6-633	255600	255600
BLS24-315-BW6-672	300929	300929

On the one hand, we note that for inner curves BLS24-315 is the fastest on $\mathbb{G}_1$, the slowest on $\mathbb{G}_2$ while still competitive on $\mathbb{G}_T$ (especially for multi-pairings when the final exponentiation is factored out). Thus, it is suitable for KZG-based SNARKs where only $\mathbb{G}_1$ operations and pairings accounts for the `Setup`, `Prove` and `Verify` algorithms. On the other hand, BLS12-377 presents the best tradeoff on all operations making it suitable for Groth'16 SNARK. For



(a) $\mathbb{G}_1$-MSM (b) $\mathbb{G}_2$-MSM

Fig. 1. MSM on $\mathbb{G}_1$ 1(a) and $\mathbb{G}_2$ 1(b) for short-listed inner curves.

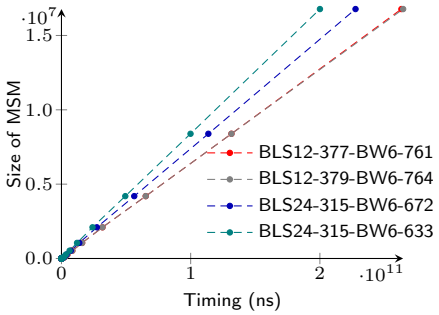


Fig. 2. $\mathbb{G}_1/\mathbb{G}_2$-MSM on short-listed outer curves.

Table 14. Pairing computation benchmarks.

curve	Miller Loop (ns)	Final Exp. (ns)	Pairing (ns)
BLS12-377 opt. ate	377191	422157	799348
BLS12-379 opt. ate	383753	453687	837440
BLS24-315 opt. ate	435958	993500	1429458
BLS12-377-BW6-761 opt. ate	1613306	1099533	2712839
BLS12-379-BW6-764 opt. ate	1548546	1057174	2605720
BLS24-315-BW6-633 opt. ate	918724	727918	1646642
BLS24-315-BW6-633 opt. Tate	809503	727918	1537421
BLS24-315-BW6-672 opt. ate	1073268	977436	2050704
BLS24-315-BW6-672 opt. Tate	973630	977436	1951066

the less conservative choice of outer curve to BLS24-315, namely BW6-633, a pairing computation is almost as fast as on BLS24-315 and MSMs are the fastest on all outer curves given the small field size. For the conservative choice, namely BW6-672, operations on all three groups are reasonably fast and notably faster than on outer curves to BLS12 (BW6-761 and BW6-764).

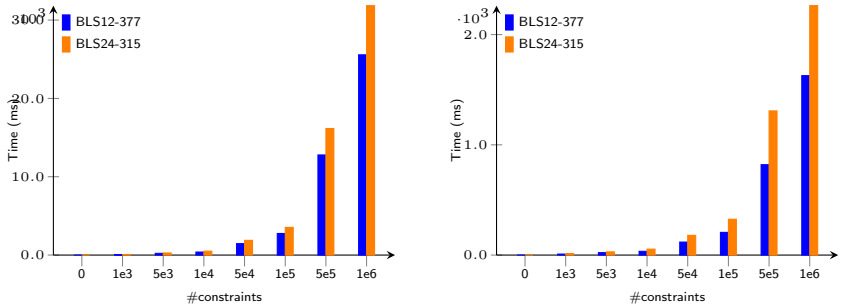
Groth' 16 and PlonK schemes Based on previous paragraph analysis, here, we discard BLS12-379/BW6-764 pair and choose to bench BLS12-377/BW6-761 and BLS24-315/BW6-633/BW6-762 pair of curves in the context of Groth'16 and PlonK SNARKs. We choose a simple circuit (proof of exponentiation: $a^w := b$ 2.2) to be able to control precisely the number of constraints. We bench the `Setup`, `Prove` and `Verify` algorithms for both Groth16 and PlonK schemes and report timings in figures 3, 4, 5, 6 and 7. The benchmark is run, this time, on an AWS c5a.24xlarge (AMD EPYC 7R32) to be able to test large circuits. In table 15 we recall the cost of SNARK algorithms in terms of preponderant groups operations.

Remark 4. The maximum number of constraints n_{max} a circuit can have is different per SNARK scheme and per curve. A PlonK prover runs operations on polynomials of degree $4n-1$, thus handles Fast Fourier Transforms (FFTs) of size $4n$ over a coset. The biggest root of unity we can have is a $8n$-th root. Hence, for PlonK, $n_{max} = 2^L/8 = 2^{L-3}$ where L is the curve subgroup 2-adicity. Similarly, a Groth16 prover runs operations over polynomials of degree $2n-1$. Hence, $n_{max} = 2^{L-2}$ for Groth16.

Table 15. Cost of `Setup`, `Prove` and `Verify` algorithms for Groth16 and PlonK. m =number of wires, n =number of multiplications gates, a =number of additions gates and ℓ =number of public inputs. $\mathtt{M}_{\mathbb{G}}$ =multiplication in $\mathbb{G}$ and P=pairing.

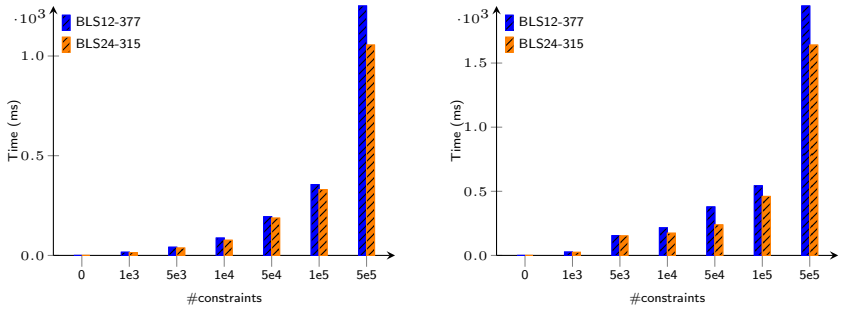
	`Setup`	`Prove`	`Verify`
Groth16	$3n$ $\mathtt{M}_{\mathbb{G}_1}$, m $\mathtt{M}_{\mathbb{G}_2}$	$(3n+m-\ell)$ $\mathtt{M}_{\mathbb{G}_1}$, n $\mathtt{M}_{\mathbb{G}_2}$	3 P, ℓ $\mathtt{M}_{\mathbb{G}_1}$
PlonK (KZG)	$d_{\geq n+a}$ $\mathtt{M}_{\mathbb{G}_1}$, 1 $\mathtt{M}_{\mathbb{G}_2}$	$9(n+a)$ $\mathtt{M}_{\mathbb{G}_1}$	2 P, 18 $\mathtt{M}_{\mathbb{G}_1}$

It is clear from figures 3, 4 and 7 that BLS12-377 is optimized to setup and prove Groth16 proofs while BLS24-315 is suitable to setup and prove PlonK proofs at the cost of acceptably slower verification time. For proof composition, we see from figures 5, 6 and 7 that the outer curves to BLS24-315, namely BW6-633 and BW6-672, are faster for all the SNARK algorithms for both Groth16 and PlonK. This confirms the recommendation of BLS24/BW6 pair of curves for KZG-based SNARK. We should also note that for applications where one would like to optimize the cost of generating and proving a proof of several proofs $\{\pi_i\}_{0\leq i\leq M}$ at the cost of slow generation of π_i (e.g. proof aggregation by light clients of off-chain generated proofs), one could use the BLS24/BW6 pair for Groth16.



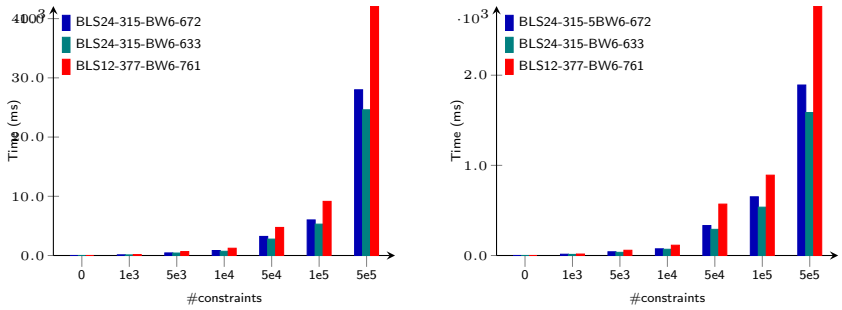
(a) Groth16 setup - inner curves (b) Groth16 prover - inner curves

Fig. 3. Groth16 `Setup` (a) and `Prove` (b) times per number of constraints for inner curves.



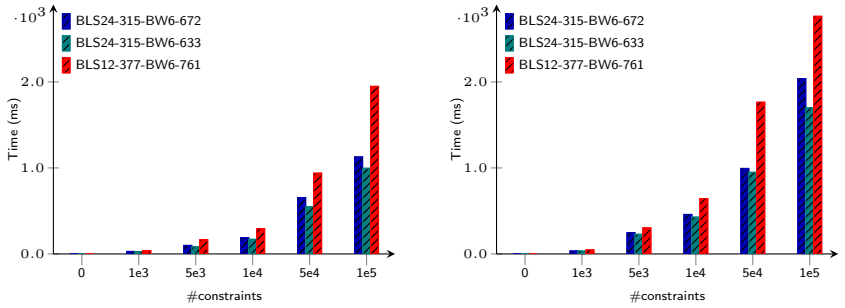
(a) PlonK setup - inner curves (b) PlonK prover - inner curves

Fig. 4. PlonK `Setup` (a) and `Prove` (b) times per number of constraints for inner curves.



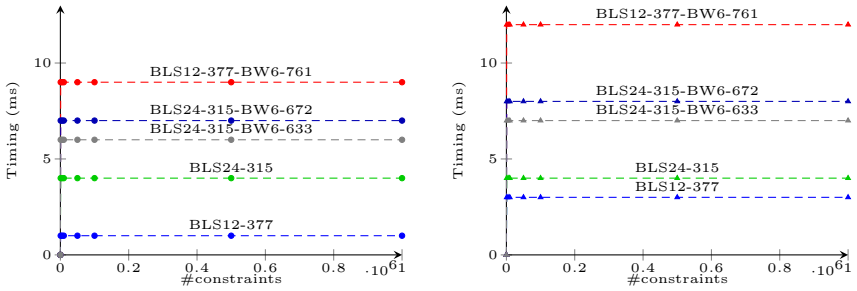
(a) Groth16 setup - outer curves (b) Groth16 prover - outer curves

Fig. 5. Groth16 `Setup` (a) and `Prove` (b) times per number of constraints for outer curves.



(a) PlonK setup - outer curves (b) PlonK prover - outer curves

Fig. 6. PlonK `Setup` (a) and `Prove` (b) times per number of constraints for outer curves.



(a) Groth16 verification (b) PlonK verification

Fig. 7. Groth16 (a) and PlonK (b) `Verify` times on short-listed curves.

6 Conclusion

We generalized the curve construction of [18] and proposed a family in which this curve falls. Precisely, a family of SNARK-friendly 2-chains built on top of BLS12 inner curves. We investigated another family composed of inner BLS24 curves and outer BW6, CP8 and CP12 curves. We first presented our results for a better arithmetic on all BLS curves and then derived generic formulas for group operations and pairings over our outer curves. Then, we analysed and compared the security and performance tradeoffs of all the constructions. In the context of SNARK applications, we short-listed several curves based on practical criteria. Finally, we presented a SageMath library to derive the curves and verify the formulas and an optimized Golang implementation of the short-listed curves along with benchmarks. We concluded that BLS12-377/BW6-761 is optimized in the Groth16 setting while BLS24-315/BW6-672 (or less conservative BW6-633) is optimized in the KZG-based SNARK setting.

As a future work, we would like to investigate optimized pairing algorithms for SNARK circuits (e.g. R1CS). In fact, while pairings are well studied in the

classical setting, there isn't much work in the SNARK setting. This would enable faster proof composition with SNARK-friendly 2-chains. Another avenue that is worth noting is combining our work with other techniques that allow proof composition, such as Plookup [22]. On a platform where only a target curve is available (e.g. Ethereum blockchain implements natively BN254), one can imagine composing efficiently multiple PlonK proofs with BLS24/BW6 pair of curves and then using the less efficient Plookup technique only once to prove the composed BW6-proof over the target curve.

Acknowledgements

We thank Thomas Piellard and Olivier Bégassat for valuable discussions and feedback. We thank Gautam Botrel for helping with the Go implementation. We thank Diego Aranha, Julien Doget and Mike Scott for stimulating discussions on $\mathbb{G}_T$ membership testing and subgroup security in $\mathbb{F}_{q^k}$.

References

1. A fork of gnark-crypto: Golang library for finite fields, fft, and elliptic curves. (2021), https://github.com/yelhousni/gnark-crypto
2. Aranha, D.F., Barreto, P.S.L.M., Longa, P., Ricardini, J.E.: The realm of the pairings. In: Lange, T., Lauter, K., Lisonek, P. (eds.) SAC 2013. LNCS, vol. 8282, pp. 3–25. Springer, Heidelberg (Aug 2014). doi:10.1007/978-3-662-43414-7_1
3. Aranha, D.F., Karabina, K., Longa, P., Gebotys, C.H., López-Hernández, J.C.: Faster explicit formulas for computing pairings over ordinary curves. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 48–68. Springer, Heidelberg (May 2011). doi:10.1007/978-3-642-20465-4_5
4. Aranha, D.F., Pagnin, E., Rodríguez-Henríquez, F.: LOVE a pairing. In: Longa, P., Ràfols, C. (eds.) LATINCRYPT 2021. LNCS, vol. 12912, pp. 320–340. Springer (2021). doi:10.1007/978-3-030-88238-9_16
5. Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. Journal of Cryptology **32**(4), 1298–1336 (Oct 2019). doi:10.1007/s00145-018-9280-5
6. Barreto, P.S.L.M., Costello, C., Misoczki, R., Naehrig, M., Pereira, G.C.C.F., Zanon, G.: Subgroup security in pairing-based cryptography. In: Lauter, K.E., Rodríguez-Henríquez, F. (eds.) LATINCRYPT 2015. LNCS, vol. 9230, pp. 245–265. Springer, Heidelberg (Aug 2015). doi:10.1007/978-3-319-22174-8_14
7. 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 02. LNCS, vol. 2576, pp. 257–267. Springer, Heidelberg (Sep 2003). doi:10.1007/3-540-36413-7_19
8. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Scalable zero knowledge via cycles of elliptic curves. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 276–294. Springer, Heidelberg (Aug 2014). doi:10.1007/978-3-662-44381-1_16
9. Bernstein, D.J., Doumen, J., Lange, T., Oosterwijk, J.J.: Faster batch forgery identification. In: Galbraith, S.D., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 454–473. Springer, Heidelberg (Dec 2012). doi:10.1007/978-3-642-34931-7_26

10. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Goldwasser, S. (ed.) ITCS 2012. pp. 326–349. ACM (Jan 2012). doi:10.1145/2090236.2090263
11. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: Recursive composition and bootstrapping for SNARKS and proof-carrying data. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 111–120. ACM Press (Jun 2013). doi:10.1145/2488608.2488623
12. Bowe, S., Chiesa, A., Green, M., Miers, I., Mishra, P., Wu, H.: ZEXE: Enabling decentralized private computation. In: 2020 IEEE Symposium on Security and Privacy. pp. 947–964. IEEE Computer Society Press (May 2020). doi:10.1109/SP40000.2020.00050
13. Charles, D.: On the existence of distortion maps on ordinary elliptic curves. ePrint 2006/128
14. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, N., Ward, N.P.: Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 738–768. Springer, Heidelberg (May 2020). doi:10.1007/978-3-030-45721-1_26
15. ConsenSys: gnark: a fast zk-snark library Golang that offers a high-level api to design circuits. (2021), https://github.com/ConsenSys/gnark
16. 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 (May 2010). doi:10.1007/978-3-642-13013-7_14
17. De Micheli, G., Gaudry, P., Pierrot, C.: Lattice enumeration for tower NFS: a 521-bit discrete logarithm computation. ePrint 2021/707
18. El Housni, Y., Guillevic, A.: Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition. In: Krenn, S., Shulman, H., Vaudenay, S. (eds.) CANS 20. LNCS, vol. 12579, pp. 259–279. Springer, Heidelberg (Dec 2020). doi:10.1007/978-3-030-65411-5_13
19. El Housni, Y., Guillevic, A.: Families of SNARK-friendly 2-chains of elliptic curves. https://gitlab.inria.fr/zk-curves/snark-2-chains (10 2021), MIT License
20. Fotiadis, G., Konstantinou, E.: TNFS resistant families of pairing-friendly elliptic curves. Theoretical Computer Science **800**, 73–89 (31 December 2019). doi:10.1016/j.tcs.2019.10.017
21. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. Journal of Cryptology **23**(2), 224–280 (Apr 2010). doi:10.1007/s00145-009-9048-z
22. Gabizon, A., Williamson, Z.J.: plookup: A simplified polynomial protocol for lookup tables. ePrint 2020/315
23. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. ePrint 2019/953
24. Galbraith, S.D., Lin, X., Scott, M.: Endomorphisms for faster elliptic curve cryptography on a large class of curves. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 518–535. Springer, Heidelberg (Apr 2009). doi:10.1007/978-3-642-01001-9_30
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 (Aug 2001). doi:10.1007/3-540-44647-8_11

26. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (Jun 2011). doi:10.1145/1993636.1993651
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 (May 2010). doi:10.1007/978-3-642-13013-7_13
28. Grassi, L., Khovratovich, D., Rechberger, C., Roy, A., Schofnegger, M.: Poseidon: A new hash function for zero-knowledge proof systems. In: USENIX Security Symposium (2021)
29. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 305–326. Springer, Heidelberg (May 2016). doi:10.1007/978-3-662-49896-5_11
30. Guillevic, A.: A short-list of pairing-friendly curves resistant to special TNFS at the 128-bit security level. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part II. LNCS, vol. 12111, pp. 535–564. Springer, Heidelberg (May 2020). doi:10.1007/978-3-030-45388-6_19
31. Guillevic, A.: Pairing-friendly curves. https://members.loria.fr/AGuillevic/pairing-friendly-curves/ (2021)
32. Guillevic, A., Masson, S., Thomé, E.: Cocks–Pinch curves of embedding degrees five to eight and optimal ate pairing computation. Des. Codes Cryptogr. **88**, 1047–1081 (March 2020). doi:10.1007/s10623-020-00727-w
33. Guillevic, A., Singh, S.: On the alpha value of polynomials in the tower number field sieve algorithm. Mathematical Cryptology **1**(1) (Feb 2021), https://journals.flvc.org/mathcryptology/article/view/125142
34. Hayashida, D., Hayasaka, K., Teruya, T.: Efficient final exponentiation via cyclotomic structure for pairings over families of elliptic curves. ePrint 2020/875
35. Hess, F., Smart, N.P., Vercauteren, F.: The eta pairing revisited. IEEE Trans. Inf. Theory **52**(10), 4595–4602 (2006). doi:10.1109/TIT.2006.881709
36. Joux, A., Pierrot, C.: The special number field sieve in $\mathbb{F}_{p^n}$ - application to pairing-friendly constructions. In: Cao, Z., Zhang, F. (eds.) PAIRING 2013. LNCS, vol. 8365, pp. 45–61. Springer, Heidelberg (Nov 2014). doi:10.1007/978-3-319-04873-4_3
37. Karabina, K.: Squaring in cyclotomic subgroups. Math. Comput. **82**(281), 555–579 (2013). doi:10.1090/S0025-5718-2012-02625-1
38. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 177–194. Springer, Heidelberg (Dec 2010). doi:10.1007/978-3-642-17373-8_11
39. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: 24th ACM STOC. pp. 723–732. ACM Press (May 1992). doi:10.1145/129712.129782
40. Kim, T., Barbulescu, R.: Extended tower number field sieve: A new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 543–571. Springer, Heidelberg (Aug 2016). doi:10.1007/978-3-662-53018-4_20
41. Lynn, B.: Pairing-based cryptography (PBC) library. https://crypto.stanford.edu/pbc/ (2013), v-0.5.14. C language, LGPL license
42. Lynn, B.: On the implementation of pairing-based cryptosystems. Phd thesis, Stanford University, department of computer science (2007), https://crypto.stanford.edu/pbc/thesis.html
43. Masson, S.: 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

44. Micali, S.: CS proofs (extended abstracts). In: 35th FOCS. pp. 436–453. IEEE Computer Society Press (Nov 1994). doi:10.1109/SFCS.1994.365746
45. Schoof, R.: 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
46. Scott, M.: A note on group membership tests for $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ on BLS pairing-friendly curves. ePrint 2021/1130
47. Scott, M.: Pairing implementation revisited. ePrint 2019/077
48. Scott, M.: Unbalancing pairing-based key exchange protocols. ePrint 2013/688
49. Silverman, J.H.: The Arithmetic of Elliptic Curves. Graduate texts in mathematics, Springer, Dordrecht (2009). doi:10.1007/978-0-387-09494-6
50. Valiant, P.: Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 1–18. Springer, Heidelberg (Mar 2008). doi:10.1007/978-3-540-78524-8_1
51. Vercauteren, F.: Optimal pairings. IEEE Transactions on Information Theory **56**(1), 455–461 (Jan 2010). doi:10.1109/TIT.2009.2034881
52. Wahby, R.S., Boneh, D.: Fast and simple constant-time hashing to the BLS12-381 elliptic curve. IACR TCHES **2019**(4), 154–179 (2019). doi:10.13154/tches.v2019.i4.154-179

A Complement on BLS Curves

A.1 BLS Curve Parameters, and Proofs of Lemmas

Table 16. Parameters of BLS curves, $6 \mid k$, $18 \nmid k$.

k	$6, 12, 24, 48, 96$	$30, 42, 66, 78$	$60, 84$
$t(x)$	$x+1$		
$y(x)$	$(x-1)(2x^{k/6}-1)/3$		
$r(x)$	$x^{k/3}-x^{k/6}+1$	$\Phi_k(x)$	
$q(x)$	$r(x)(x-1)^2/3+x$	$r(x)(x-1)^2/3c_2(x)+x$	
$c_2(x)$	1	x^2-x+1	x^4-x^2+1
ρ	$1+6/k$	$(k/3+2)/\varphi(k)$	

Table 17. Parameters of BLS curves, $k = 3 \bmod 6$.

k	$3, 9, 27, 81$	$15, 21, 33, 39,$ $51, 57, 69, 87, 93$	$45, 63, 99$	75
$t(x)$	$x+1$			
$y(x)$	$(x-1)(2x^{k/3}+1)/3$			
$r(x)$	$x^{2k/3}+x^{k/3}+1$	$\Phi_k(x)$		
$q(x)$	$r(x)/3(x-1)^2+x$	$r(x)(x-1)^2/3c_2(x)+x$		
$c_2(x)$	1	x^2+x+1	x^6+x^3+1	$x^{10}+x^5+1$
ρ	$1+3/k$	$(2k/3+2)/\varphi(k)$		

Proof (of Lemmas 1 and 3). Let us consider odd k. Observe that $x^{k/3}$ is a primitive third root of unity $(-1+\sqrt{-3})/2$ modulo $r(x) = \Phi_k(x)$, and $1/\sqrt{-3} = (2x^{k/3} + 1)/3$. A solution for $y(x) = (t(x)-2)/\sqrt{-3} \bmod r(x)$ is $y(x) = (x-1)(2x^{k/3}+1)/3$, an then $q(x) = (t^2(x) + 3y^2(x))/4$ is an irreducible polynomial which represents primes in the terms of [21, Definition 2.5]. The curve order is $q(x) + 1 - t(x) = ((t(x)-2)^2+3y^2(x))/4 = ((x-1)^2+(x-1)^2(2x^{k/3}+1)^2/3)/4 = (x-1)^2/3(x^{2k/3}+ x^{k/3} + 1) = c(x)r(x)$. Note that $x^k - 1 = (x^{2k/3} + x^{k/3} + 1)(x^{k/3} - 1)$, hence $\Phi_k(x)$ divides $x^{2k/3} + x^{k/3} + 1$ (as it does not divide $x^{k/3} - 1$), and the cofactor $c(x)$ has the form

$$c(x) = (x-1)^2/3 \cdot (x^{2k/3} + x^{k/3} + 1)/\Phi_k(x) \ .$$

In particular for $k = 3^j$, the k-th cyclotomic polynomial is $\Phi_{3^j}(x) = \Phi_3(x^{3^{j-1}}) = x^{2k/3} + x^{k/3} + 1$, in this case the cofactor $c(x)$ is exactly $(x-1)^2/3$.

With even k, $x^{k/6}$ is a primitive 6-th root of unity $(1+\sqrt{-3})/2$ modulo $r(x) = \Phi_k(x)$, and $1/\sqrt{-3} = (2x^{k/6} - 1)/3$. Then $y(x) = (x-1)(2x^{k/6} - 1)/3$, and $q(x) = (t^2(x) + 3y^2(x))/4$ is an irreducible polynomial which represents primes in the terms of [21, Definition 2.5]. The curve order is $q(x) + 1 - t(x) = ((t(x)-2)^2+3y^2(x))/4 = ((x-1)^2+(x-1)^2(2x^{k/6}-1)^2/3)/4 = (x-1)^2/3(x^{k/3}- x^{k/6} + 1) = c(x)r(x)$. In the same way as for odd k, one observes that $x^k - 1 = (x^{k/3} - x^{k/6} + 1)(x^{k/3} + x^{k/6} + 1)(x^{k/3} - 1)$, hence $\Phi_k(x)$ divides $x^{k/3} - x^{k/6} + 1$, and the cofactor $c(x)$ has the form

$$c(x) = (x-1)^2/3 \cdot (x^{k/3} - x^{k/6} + 1)/\Phi_k(x) \ .$$

Proof (of Lemma 2). Observe that $q(x) - 1 = c(x)r(x) + t(x) - 2$, and $t(x) - 2 = x - 1$. For odd k, from Lemma 1 one has $q(x) - 1 = (x-1)^2/3 \cdot (x^{2k/3} + x^{k/3} + 1) + x - 1 = (x-1)/3 \cdot ((x-1)(x^{2k/3} + x^{k/3} + 1) + 1)$.

For even k, from Lemma 1 one has $q(x) - 1 = (x-1)^2/3 \cdot (x^{k/3} - x^{k/6} + 1) + x - 1 = (x-1)/3 \cdot ((x-1)(x^{k/3} - x^{k/6} + 1) + 1)$. In both cases, $(x-1)/3$ divides $q(x) - 1$.

Proof (of Lemma 4). Any BLS curve is an ordinary curve of j-invariant 0 and discriminant -3, of the form $y^2 = x^3 + b$, defined over a prime field $\mathbb{F}_q$ where $q = 1 \bmod 3$. In this case, it is well known that the (GLV) endomorphism is of the form $\psi\colon (x, y) \mapsto (\omega x, y)$, where $\omega \in \mathbb{F}_q$ is a primitive third root of unity. It has characteristic polynomial $\psi^2 + \psi + 1 = 0$ and is defined over $\mathbb{F}_q$. The endomorphism ring of the curve is $\mathbb{Z}[(1+\sqrt{-3})/2]$.

A.2 Subgroup Security, Distortion Map

The definition of subgroup security in [6] is the following.

Definition 3 (Subgroup Security, [6, Definition 1]). *Let $q(u), t(u), r(u) \in \mathbb{Q}[u]$ parameterize a family of ordinary pairing-friendly elliptic curves, and for any particular $u_0 \in \mathbb{Z}$ such that $q = q(u_0)$ and $r = r(u_0)$ are prime, let E be the resulting pairing-friendly elliptic curve over $\mathbb{F}_q$ of order divisible by r.*

Let $h_1 = \#E(\mathbb{F}_q)/r$, $h_2 = \#E'(\mathbb{F}_{q^{k/d}})/r$ and $h_T = \Phi_k(q)/r$. We say that E is subgroup-secure if all $\mathbb{Q}[u]$-irreducible factors of $h_1(u), h_2(u)$ and $h_T(u)$ that can represent primes and that have degree at least that of $r(u)$, contain no prime factors smaller than $r(u_0) \in \mathbb{Z}$ when evaluated at $u = u_0$.

If $c_0 = (x-1)/3$ is prime, since the structure of the subgroup of order c_0^2 is $\mathbb{Z}/c_0\mathbb{Z} \oplus \mathbb{Z}/c_0\mathbb{Z}$, and the subgroup is fully defined over the prime field $\mathbb{F}_q$ (Corollary 1), one can find a basis $\langle P_1, P_2\rangle$ so that P_1, P_2 are of order c_0 and linearly independent. Moreover there exists a distortion map ψ from the subgroup $\langle P_1\rangle$ to $\langle P_2\rangle$. The distortion map ψ is given by $(x, y) \mapsto (\omega x, y)$ where $\omega \in \mathbb{F}_q$ is such that $\omega^2 + \omega + 1 = 0$. (See [13] on distortion maps on embedding degree 1 curves). Because of this distortion map, one can transfer as in the MOV attack a discrete logarithm computation in the subgroup of order $(x-1)/3$ of $E(\mathbb{F}_q)$ to a discrete logarithm computation in the subgroup of order $(x-1)/3$ of $\mathbb{F}_q$ (note that this is the base field $\mathbb{F}_q$, not the extension field $\mathbb{F}_{q^k}$), where sub-exponential DL computation takes place. The DL computation in $\mathbb{F}_q$ has complexity $\exp((1+o(1))\sqrt{\ln q \ln\ln q})$ with the quadratic sieve, and $\exp((1.923+o(1))\sqrt[3]{\ln q(\ln\ln q)^2})$ with the number field sieve. Because the complexity is in q not c_0, the computation will be slower, nevertheless it exists. In practice, if an implementation of a generic DL computation algorithm like Pollard-ρ is faster in $\mathbb{F}_q$ than on $E(\mathbb{F}_q)$ for the subgroup of order $(x-1)/3$, it is possible to transfer the computation from the curve to the finite field thanks to the distortion map and a Weil pairing.

A.3 Hard part of the final exponentiation for BW6 curves

In [34], Hayashida, Hayasaka and Teruya develop formulas for the hard part of the final exponentiation. Applying [34, Theorem 4], we obtain

$$\begin{aligned}\Phi_6(q)/r &= c((t-1)+q-1) + \Phi_6(t-1)/r \\ &= c(q+t-2) + (t^2-3t+3)/r\ .\end{aligned} \tag{38}$$

Replacing $c = c_{\mathrm{bw},i}$, $q = q_{\mathrm{bw},i}$ and $t = t_{\mathrm{bw},i} + h_t r_{\mathrm{bw}}$, and simplifying, we obtain

$$\Phi_6(q_{\mathrm{bw},i})/r_{\mathrm{bw}} = (c_{\mathrm{bw},i} + h_t)(q_{\mathrm{bw},i} + t_{\mathrm{bw},i} - 2) + \Phi_6(t_{\mathrm{bw},i}-1)/r_{\mathrm{bw}}\ . \tag{39}$$

In (39), the factor of $(c_{\mathrm{bw},i} + h_t)$ as degree 5 in x (degree of $t_{bw,i} - 2$). Actually (27), (28) of degree 3 in x are related to (39). Observe that multiplying (39) by $(x+1)$ allows to substitute for (20): $-(x+1)(t_{\mathrm{bw},0}-2) = 3r_{\mathrm{bw}} + x^3 - x^2 + 1$; and (23): $(x+1)(t_{\mathrm{bw},3}-2) = 3r_{\mathrm{bw}} + x^3 - x^2 - 2x - 1$. Then again $c_{\mathrm{bw},i} r_{\mathrm{bw}} = q_{\mathrm{bw},i} + 1 - (t_{\mathrm{bw},i} + h_t r_{\mathrm{bw}})$. From (39) and (20) one gets

$$\begin{aligned}-(x+1)\Phi_6(q_{\mathrm{bw},0})/r_{\mathrm{bw}} &= (c_{\mathrm{bw},0}+h_t)(-(x+1)q_{\mathrm{bw},0} + x^3 - x^2 + 1) \\ &\quad +3(q_{\mathrm{bw},0} + 1 - t_{\mathrm{bw},0}) - (x+1)\Phi_6(t_{\mathrm{bw},0}-1)/r_{\mathrm{bw}} \\ &= (c_{\mathrm{bw},0}+h_t)(-(x+1)q_{\mathrm{bw},0} + x^3 - x^2 + 1) + 3(q_{\mathrm{bw},0} - x^2 + 2x - 2)\end{aligned}$$

and (27) is three times this formula. In the same way, we can obtain (28) from (39) and (23).

B STNFS-security of MNT6 curves

In [32], Guillevic, Massson and Thomé estimated the cost of the Special-Tower Number Field Sieve algorithm (STNFS) and its variants for MNT6 curves (MNT curves of embedding degree 6) for curve parameters obtained from PBC library developed by Ben Lynn [41,42]. In [33], Guillevic and Singh refined the cost model. We reproduce in Fig. 8 the estimated cost of computing a discrete logarithm in GF(p^6) with the Tower NFS algorithm. There is a cross-over point at p of about 1536 bits from the Conjugation method of polynomial selection, to the generalisation made by Sarkar–Singh, both with the TNFS algorithm. The crossover point from TNFS to NFS is at much larger p. In conclusion, to ensure a 128-bit security level in a field GF(p^6), the prime p should be at least 672-bit long. If moreover a Special variant of NFS or TNFS is available because the prime p has a *special form*, the size requirement will be larger, but this is not the case for MNT parameters.

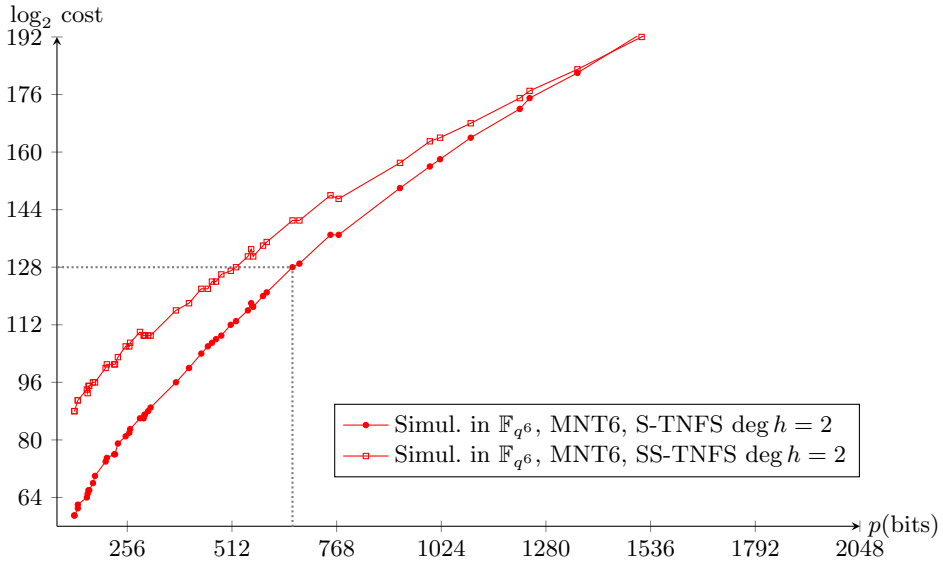


Fig. 8. Estimated cost of DL computation with TNFS in GF(p^6).