A Family of Implementation-Friendly BN Elliptic Curves

Geovandro C. C. F. Pereira[1], Marcos A. Simplício Jr.[1], Michael Naehrig[2], and Paulo S. L. M. Barreto[1] *

[1] Departamento de Engenharia de Computação e Sistemas Digitais (PCS), Escola Politécnica, Universidade de São Paulo. Av. Prof. Luciano Gualberto, trav. 3, № 158 05508-900 São Paulo (SP), Brazil. {geovandro,mjunior,pbarreto}@larc.usp.br

[2] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA mnaehrig@microsoft.com

Abstract. For the last decade, elliptic curve cryptography has gained increasing interest in industry and in the academic community. This is especially due to the high level of security it provides with relatively small keys and to its ability to create very efficient and multifunctional cryptographic schemes by means of bilinear pairings. Pairings require pairing-friendly elliptic curves and among the possible choices, Barreto-Naehrig (BN) curves arguably constitute one of the most versatile families.
In this paper, we further expand the potential of the BN curve family. We describe BN curves that are not only computationally very simple to generate, but also specially suitable for efficient implementation on a very broad range of scenarios. We also present implementation results of the optimal ate pairing using such a curve defined over a 254-bit prime field.

1 Introduction

Since the introduction of elliptic curves to cryptography [27, 32], they have raised increasing interest as a useful tool for creating efficient asymmetric schemes (e.g., digital signatures) with a high security level per bit of the used keys. Hence, they are able to cope with important (albeit challenging) requirements of modern systems such as reductions on the usage of processing power, storage, bandwidth and energy. Furthermore, elliptic curves are the foundation of pairing-based cryptography, which has been responsible for the development of state-of-the-art applications such as identity-based encryption [9] and certificateless public key cryptography [1]. Inherently, feasible pairing-based cryptography requires pairing-friendly curves.

Barreto-Naehrig (BN) curves arguably constitute one of the most versatile classes of pairing-friendly elliptic curves. Among other things, they are known [6] to (this list may not be complete):

* Supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under research productivity grant 303163/2009-7.

- facilitate the deployment of bilinear pairings at the 128-bit security level [15, 37, 8, 2];
- enable all kinds of pairing-based cryptographic schemes and protocols (including short signatures) [19];
- be plentiful and easily found [35, Section 2.1.1];
- support a sextic twist, so pairing arguments can be defined over relatively small finite fields $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ respectively [24];
- be amenable to twofold or threefold pairing compression [36];
- attain high efficiency for all pairing computation algorithms known, including the Tate [40], ate [24], eil [23], R-ate [30], Xate [38] and optimal [45] pairings;
- admit optimizations based on endomorphisms and homomorphisms for all groups involved [18, 20], thereby enabling fast non-pairing operations as well;
- be suitable for software and hardware implementations on a wide range of platforms [16, 21].

Recent research has focused on certain individual curves to attain exceptional performance gains [37, 8, 2]. This is essential since pairings are usually the most computationally expensive operation in any pairing-based cryptographic scheme. On the other hand, one may argue that only targeting fast pairings is insufficient, and may lead to annoying or unacceptable inefficiencies on certain highly constrained platforms like smart cards or wireless sensor networks. Indeed, because of the intrinsic high cost of pairings, many protocols are already designed to rely on them only when the corresponding protocol parties are assumed to have plentiful computational resources (e.g. servers or clusters) while constrained parties only need to perform non-pairing operations [4, 10, 31, 46]. In such scenarios, curve parameters leading to fast (but still proportionally slow) pairings at the price of deteriorating performance elsewhere would be harmful rather than helpful.

A different line of research is that of obtaining parameterized curves with prescribed properties to avoid computationally expensive tests during curve generation. Simplification of curve parameter testing is even more important. Certain attacks can be prevented by checking that the purported BN curve contained in a given digital certificate does indeed exhibit the expected properties before using that certificate. This procedure is commonplace for non-pairing-friendly curves, but the special-purpose nature of BN curves exacerbates the amount of necessary computations. By adopting a curve where certain properties are guaranteed to hold, the testing overhead would be greatly reduced. It could then be carried out on much simpler platforms; e.g. a lightweight certificate server would only need plain integer arithmetic up to primality checking (and no elliptic curve arithmetic support) to attest the well-formedness of the curves.

Constructing the right twist of the curve over the base field without resorting to any elliptic curve arithmetic has been carried out successfully [39]. In contrast, the related tasks of choosing suitable representations for all extension fields involved and selecting the correct sextic twist over the quadratic extension field have received limited attention in the literature. They would still seem to need

square and cube detection in extension fields and full group arithmetic in that twist. We note that extension fields are usually chosen *a priori*, based on features of supporting libraries and oblivious to the peculiar nature of BN curves.

Our contribution in this paper is the definition of a (rather large) subclass of BN curves that is particularly suitable for efficient construction, parameter checking and implementation, while retaining a very simple description. Specifically, we propose to use curves of form $E : y^2 = x^3 + b$ over $\mathbb{F}_p$ where $p \equiv 3 \pmod 4$ and $b = c^4 + d^6$ or $b = c^6 + 4d^4$ for $c, d \in \mathbb{F}_p^*$, particularly for certain choices of c and d. We discuss in detail the rationale of this proposal and how it favors the efficiency of all typical arithmetic operations needed to instantiate cryptographic protocols on the broadest possible landscape, i.e. in the sense of targeting the widest possible range of platforms and applications. In particular, our construction automatically yields the right sextic twist entirely avoiding curve arithmetic for that purpose. It gives to field representations an overall unity that provides several optimization opportunities. Our proposal has intersections with other interesting curve families that occur in the literature (e.g. [38, 44]), offering additional benefits in those cases.

We stress that it is not the purpose of this paper to evaluate optimization techniques that are exclusive to a particular platform, nor to focus on the operation of pairing computation itself or on techniques that are only available on a narrow set of circumstances. Our goal is rather to explore a simple yet comprehensive theoretical setting that avoids most if not all general drawbacks and implementation hindrances, while favoring complete pairing-based cryptosystems.

To underline the flexible nature of our proposal and to reproduce the operation counts, we have implemented in Java the optimal ate pairing on one of the proposed curves (254 bits). We have focused on employing the best known techniques and algorithms for each operation as listed in Section 5. The implementation is available online[3].

The remainder of this paper is organized as follows. We introduce theoretical concepts related to bilinear maps and BN curves in Section 2. We describe the proposed implementation-friendly family of BN curves and discuss its features in Section 3. Concrete examples tailored for practical deployment are suggested in Section 4. In Section 5, we compare the efficiency of our pairing implementation on one of the proposed curves to previous work, and we conclude in Section 6.

2 Preliminaries

Let p be a prime and let $e > 0$. The conjugates of $a \in \mathbb{F}_{p^e}$ are the elements a^{p^i}, $0 \leqslant i < e$. The norm $N(a)$ of $a \in \mathbb{F}_{p^e}$ is defined to be the product of all its conjugates, $N(a) := \prod_i a^{p^i}$. Whenever $p \equiv 3 \pmod 4$, the finite field $\mathbb{F}_{p^2}$ can be represented as $\mathbb{F}_p[i]/(i^2+1)$. In analogy to the complex numbers, the non-trivial conjugate of the field element $\gamma = \alpha + i\beta \in \mathbb{F}_{p^2}$ is $\bar{\gamma} := \gamma^p = \alpha - i\beta$.

Given three groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ of the same prime order n, a pairing is a feasibly computable, non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Usually $\mathbb{G}_1$

[3] `http://code.google.com/p/bnpairings/`

and $\mathbb{G}_2$ are written additively, while $\mathbb{G}_T$ is written multiplicatively. In practice, the pairing groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are most commonly determined by the eigenspaces of the Frobenius endomorphism ϕ_p on some elliptic curve $E/\mathbb{F}_p$ of embedding degree $k > 1$. Specifically, $\mathbb{G}_1$ is taken to be the 1-eigenspace $E[n] \cap \ker(\phi_p - [1]) = E(\mathbb{F}_p)[n]$. The group $\mathbb{G}_2$ is taken to be the preimage $E'(\mathbb{F}_{p^e})[n]$ of the p-eigenspace $E[n] \cap \ker(\phi_p - [p]) \subseteq E(\mathbb{F}_{p^k})[n]$ under a twisting isomorphism $\psi : E' \to E$, $(x, y) \mapsto (\mu^2 x, \mu^3 y)$ for some $\mu \in \mathbb{F}_{p^k}^*$. In particular, $e = k/d$ where the curve $E'/\mathbb{F}_{p^e}$ is the unique twist of largest possible twist degree $d \mid k$ for which n divides $\#E'(\mathbb{F}_{p^e})$ (see [24] for details). This means that e is as small as possible. Typical pairing algorithms are based on Miller's algorithm [33] with a number of optimizations [3, 24, 30, 38, 45], most notably optimal pairings [45] which have loop order of length $\lceil \lg n \rceil / \varphi(k)$ in general, where φ is Euler's totient function. Note that this compares well with the original Tate pairing which has loop order of length $\lceil \lg n \rceil$.

A Barreto-Naehrig (BN) curve [5] is an elliptic curve $E_b : y^2 = x^3 + b$ defined over a finite prime field $\mathbb{F}_p$ with the following properties. It has prime order $n = \#E(\mathbb{F}_p)$, and the primes p and n are given by $p = p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ and $n = n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1$ for some $u \in \mathbb{Z}$. We drop the subscript and simply write E when the specific equation coefficient b is irrelevant to the discussion or clear from context. For primes p of the above polynomial form the condition $p \equiv 3 \pmod 4$ holds if and only if u is odd. The BN field $\mathbb{F}_p$ contains a primitive cube root of unity $\zeta(u) = 18u^3 + 18u^2 + 9u + 1$ as one can check by straightforward inspection. BN curves have embedding degree $k = 12$ and admit a sextic twist ($d = 6$), so that one can set $\mathbb{G}_2 = E'(\mathbb{F}_{p^2})[n]$, and there exists an optimal ate pairing with loop order $\omega = |6u + 2|$.

Since BN curves have j-invariant 0, it is relatively easy to find them when compared to pairing-friendly curves from other families (see [17] for an extensive survey). In particular, there is no need to resort to the CM method explicitly. To generate a BN curve, one chooses an integer u until p and n as given by the above polynomials are prime. The size of u is selected such that it yields a desired size for p and n. To find a corresponding curve, one chooses $b \in \mathbb{F}_p$ so that the curve $E : y^2 = x^3 + b$ has order n. For these steps, we need primality tests, possibly square detection and square root computations in $\mathbb{F}_p$ to obtain a point in $E(\mathbb{F}_p)$, and finally a scalar multiplication to check for order n.

The corresponding twist $E'/\mathbb{F}_{p^2}$ is usually selected by finding a non-square and non-cube $\xi \in \mathbb{F}_{p^2}$ and then checking via scalar multiplication whether the curve $E' : y^2 = x^3 + b'$ given by $b' = b/\xi$ or by $b' = b/\xi^5$ has order divisible by n. The element ξ can be used to represent the field extensions of $\mathbb{F}_{p^2}$ contained in $\mathbb{F}_{p^{12}}$ since the polynomial $z^r - \xi$ is irreducible over $\mathbb{F}_{p^{2h}}$ for $r \in \{2, 3, 6\}$ and $h \in \{1, 2, 3\}$ whenever $\gcd(h, r) = 1$ [35, Lemma 2.14].

Example 1. Let $p^e \equiv 1 \pmod 6$. For each $\xi \in \mathbb{F}_{p^e}$ that is neither a square nor a cube, one can represent $\mathbb{F}_{p^{6e}}$ as a tower extension of $\mathbb{F}_{p^e}$ in the following three different ways:

- $\mathbb{F}_{p^{6e}} = \mathbb{F}_{p^e}[u]/(u^6 - \xi)$;

– $\mathbb{F}_{p^{6e}} = \mathbb{F}_{p^{2e}}[v]/(v^3 - \xi)$ with $\mathbb{F}_{p^{2e}} = \mathbb{F}_{p^e}[s]/(s^2 - \xi)$;
– $\mathbb{F}_{p^{6e}} = \mathbb{F}_{p^{3e}}[w]/(w^2 - \xi)$ with $\mathbb{F}_{p^{3e}} = \mathbb{F}_{p^e}[t]/(t^3 - \xi)$.

The components of an element from $\mathbb{F}_{p^{6e}}$ in any of these can be extracted directly without the need to perform expensive computations. Thus: $a_0 + a_1u + a_2u^2 + a_3u^3 + a_4u^4 + a_5u^5 \leftrightarrow (a_0 + a_3s) + (a_1 + a_4s)v + (a_2 + a_5s)v^2 \leftrightarrow (a_0 + a_2t + a_4t^2) + (a_1 + a_3t + a_5t^2)w$, for $a_i \in \mathbb{F}_{p^e}$. This shows that the suggested setting automatically yields so-called "compositum" or tower-friendly fields [7, 22], with their associated efficiency gains. □

We will propose a subfamily of BN curves that does away with the square and cube detection tests usually needed when deciding how to represent the finite field extensions that occur in a typical implementation of pairing-based protocols. The following lemma captures an important property of the class of elliptic curves to which BN curves belong:

Lemma 1. ([35, Lemma 2.7]) *Let $E : y^2 = x^3 + b$ be an elliptic curve over $\mathbb{F}_p$ of order $n = \#E(\mathbb{F}_p)$ such that $2 \nmid n$ and $3 \nmid n$. Then b is neither a square nor a cube in $\mathbb{F}_p$.*

Proof. For any $\gamma, \delta \in \mathbb{F}_p$, the point $(0, \gamma)$ on $E : y^2 = x^3 + \gamma^2$ has order 3 and hence $3 \mid n$, while the point $(-\delta, 0)$ on $E : y^2 = x^3 + \delta^3$ has order 2 and hence $2 \mid n$, either way contradicting the assumption that $2 \nmid n$ and $3 \nmid n$. □

As a consequence, we arrive at the following useful lemma:

Lemma 2. *Let $\xi \in \mathbb{F}_{p^e}^*$ and let $b = N(\xi) \in \mathbb{F}_p$. If $E : y^2 = x^3 + b$ has order $n = \#E(\mathbb{F}_p)$ with $2 \nmid n$ and $3 \nmid n$, then ξ is neither a square nor a cube in $\mathbb{F}_{p^e}$.*

Proof. Assume that ξ is a square or a cube in $\mathbb{F}_{p^e}$, i.e. $\xi = \gamma^r$ for some $\gamma \in \mathbb{F}_{p^e}$ and $r \in \{2, 3\}$. Then $b = N(\xi) = N(\gamma^r) = N(\gamma)^r$, i.e. b is a square or a cube in $\mathbb{F}_p$, contradicting Lemma 1. □

This means that square or cube detection is not necessary in either $\mathbb{F}_p$ or $\mathbb{F}_{p^e}$. In particular, the element ξ specified in Lemma 2 can be used to define all extensions of $\mathbb{F}_{p^e}$ that are of interest to pairing implementation and to facilitate changes of representations in field towers, as shown in Example 1. We remark that this choice of representation for finite field extensions may favor the implementation of other families of pairing-friendly elliptic curves (see [17]). Pursuing this possibility, however, transcends the scope of this paper.

The next result addresses the matter of avoiding order computation and arithmetic on the sextic twist $E'(\mathbb{F}_{p^2})$ by revealing immediately which twist has the correct order. To that end we need one more property:

Lemma 3. *Let $p \equiv 1 \pmod 3$ be a prime. For any $\xi \in \mathbb{F}_{p^2}$, let $b = N(\xi) = \xi\bar{\xi}$. Then b/ξ^5 is a cube.*

Proof. We first notice that $b = \xi\bar{\xi} = \xi\xi^p = \xi^{p+1}$ and thus $b/\xi^5 = \xi^{p-4}$. Since $p - 4$ is divisible by 3, we see that b/ξ^5 is a cube. □

We are finally in a position to state the following theorem which allows capturing the right twist:

Theorem 1. *Given a BN curve of form $E : y^2 = x^3 + b$ with $b = N(\xi)$ for some $\xi \in \mathbb{F}_{p^2}$, the particular sextic twist $E' : y^2 = x^3 + \bar{\xi}$ satisfies $\#E(\mathbb{F}_p) \mid \#E'(\mathbb{F}_{p^2})$.*

Proof. Since E is assumed to be a BN curve, the parameter b cannot be the norm of an element in $\mathbb{F}_p$, because such a norm is a square in $\mathbb{F}_p$, contradicting Lemma 1. Therefore, the assumptions imply $\xi \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$. Remark 2.13 in [35] shows that the order of the desired twist E' over $\mathbb{F}_{p^2}$ is $n' = \#E'(\mathbb{F}_{p^2}) = n(2p - n)$. Since n is odd, also n' is odd. If $\xi \in \mathbb{F}_{p^2}$ is neither a square nor a cube, the correct twist is either given by $y^2 = x^3 + b/\xi$ or $y^2 = x^3 + b/\xi^5$. Since $p \equiv 1 \pmod 3$ and $b = N(\xi)$ the value b/ξ^5 is a cube by Lemma 3. This means that the curve given by $y^2 = x^3 + b/\xi^5$ has a point of order 2, hence the order of this particular twist is even. Therefore, $E' : y^2 = x^3 + b/\xi$ is the twist one seeks. Notice that $b/\xi = \bar{\xi}$. □

3 The proposed family of curves

As our main contribution in this work, we propose to use curves of a certain form, belonging to a subfamily of BN curves defined as follows:

Definition 1. *A BN curve $E_b : y^2 = x^3 + b$ over $\mathbb{F}_p$ is called* friendly *if $p \equiv 3 \pmod 4$ and if there exist $c, d \in \mathbb{F}_p^*$ such that either $b = c^4 + d^6$ or $b = c^6 + 4d^4$.*

A friendly BN curve $E = E_b$ with corresponding parameters c and d as defined above has the following properties. Note that since $p \equiv 3 \pmod 4$, we represent $\mathbb{F}_{p^2}$ by $\mathbb{F}_p[i]/(i^2 + 1)$.

- The parameters c and d automatically provide $\xi \in \mathbb{F}_{p^2}$ with $b = N(\xi)$ according to Lemma 2 to represent the required extensions of $\mathbb{F}_{p^2}$. The element ξ is $\xi = c^2 + d^3 i$ or $\xi = c^3 + 2d^2 i$, respectively.
- Since $b = N(\xi)$, the parameters c and d determine the sextic twist of correct order according to Theorem 1 as given by the equation $y^2 = x^3 + \bar{\xi}$.
- Generators of $E(\mathbb{F}_p)$ are given by obvious solutions to the curve equation as $G = (-d^2, c^2)$ or $G = (-c^2, 2d^2)$, respectively.
- Generators for $E'(\mathbb{F}_{p^2})[n]$ can be found as $[h]G'$, where $h = 2p - n$ and $G' = (-di, c)$ or $G' = (-c, d(1 - i))$, respectively.

Remark 1. In order to achieve highly efficient implementation results in practice, we propose to use friendly BN curves with the following additional properties. We assume p to have a fixed bit length $\ell := \lceil \lg p \rceil$.

- The BN prime p satisfies $p^2 \equiv 9 \pmod{16}$ or $p^2 \equiv 17 \pmod{32}$ (and possibly also $p \equiv 4 \pmod 9$));
- The Hamming weights of the (signed) binary representations of either the BN parameter u or the loop order $\omega = 6u + 2$ of the optimal ate pairing (or both) are minimal for the given bit length $\ell = \lceil \lg p \rceil$;

– The integer in $[0, p-1]$ representing b is as small as possible;
– A careful choice of c and d, specifically taking c and d to be small powers of 2 so that b has low Hamming weight, enables further efficiency advantages in field and elliptic curve arithmetic, since multiplication by b (and by ξ) consists only of shifts and additions.

Notice that the choice of b in Definition 1 is compatible with [39, Algorithm 3.5] in the sense that one can certainly look for c and d such that the resulting b has the form prescribed by that algorithm.

3.1 Rationale

We now examine the rationale for our proposal from the points of view of pairing efficiency, overall efficiency, uniform finite field arithmetic, generator simplicity, and choice of suitable field sizes:

Pairing efficiency

First and foremost, pairing computation must be as efficient as possible, since this is the most expensive operation in any pairing-based protocol. Low-weight ω minimizes the cost of the Miller loop in optimal pairings, while low-weight u minimizes the cost of the final exponentiation [42]. Small values of b favor faster pairing computation [14], especially if b has low Hamming weight, which is clearly possible with the prescribed form we suggest (e.g. if c and d are small powers of 2). One of the best situations, though not the only one, arises when $b = 2$ and $\xi = 1 + i$. In this case, multiplications by b are most efficient on all platforms (not only on those where a dedicated multiplication by a small constant is readily available, but also those where it has to be emulated with simpler operations like shifts or additions). Furthermore, the computation of conjugates, which involves multiplications by ξ, incurs the least overhead.

Overall efficiency

All operations involved in pairing-based protocols must be as efficient as possible. Works like [37] and [8] only consider pairing computation speed as a metric, disregarding operations like generating random points or hashing to the pairing groups $\mathbb{G}_1$ and $\mathbb{G}_2$ which are essential to most cryptographic schemes based on pairings. For BN curves, this means there must be a very efficient method to compute square roots in $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$. This is least expensive when $p \equiv 3 \pmod 4$ and $p^2 \equiv 9 \pmod{16}$, since the Cipolla-Lehmer method simplifies to one square detection and one exponentiation for square roots in $\mathbb{F}_p$, namely, $\sqrt{a} = a^{(p+1)/4}$, and the KCYL [28] method applies to the computation of square roots in $\mathbb{F}_{p^2}$, taking one square detection and 1.5 exponentiations. The case $p^2 \equiv 17 \pmod{32}$ is almost as efficient, taking one square detection and 2 exponentiations to compute roots in $\mathbb{F}_{p^2}$ with the method of [34]. In certain scenarios (e.g. when threefold pairing compression is desired) one might wish to require $p \equiv 4 \pmod 9$ as well, since this facilitates the computation of cube roots with methods similar to those for computing square roots [5, Section 3.1].

Uniform finite field arithmetic

Arithmetic in all finite fields involved must be efficient. Operations in $\mathbb{G}_1$ and $\mathbb{G}_2$ already need efficient arithmetic in $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$. Further processing of pairing values as for example explicit or implicit exponentiation needs efficient algorithms for $\mathbb{F}_{p^{12}}$ itself, or in some cases for the subfields $\mathbb{F}_{p^6}$ or $\mathbb{F}_{p^4}$. Efficient subfield arithmetic is needed if pairing compression techniques are adopted (by factors of 2 and 3, respectively). Also, potential support for efficient conversions between different representations has to be planned for the sake of interoperability.

Generator simplicity

Obvious generators that do not involve any extra processing or storage are clearly desirable. A curve equation of form $E : y^2 = x^3 + (c^4 + d^6)$ admits the obvious solution $G = (-d^2, c^2)$, while one of form $E : y^2 = x^3 + (c^6 + 4d^4)$ admits the solution $G = (-c^2, 2d^2)$. By Theorem 1 the sextic twists of form $E' : y^2 = x^3 + (c^2 - d^3 i)$ and $E' : y^2 = x^3 + (c^3 - 2d^2 i)$, respectively, always contain a subgroup of the same order n as E, and the curve equations for E' admit the obvious solution $G' = (-di, c)$ and $G' = (-c, d(1 - i))$, respectively. Then the point $[h]G'$, where $h = 2p - n$, only fails to be a generator of $E'(\mathbb{F}_{p^2})[n]$ with negligibly low probability $O(1/h)$. The cofactor multiplication can be carried out very efficiently [41, Section 6].

Choice of suitable field sizes

An obvious bottleneck is $\mathbb{F}_{p^2}$ arithmetic, since it is the foundation of all operations in $\mathbb{G}_2$, $\mathbb{G}_T$, and pairing computation. Choosing p slightly smaller than a multiple of the platform word size (say, more than two bits but less than three bits) is interesting because it enables not only postponing modular reductions in critical operations like $\mathbb{F}_{p^2}$ multiplication or squaring, but also simplifying the actual reduction when it is finally applied, as pointed out in [8, Section 5.2].

4 Sample curves

In Table 1, we provide practical friendly BN curves for fields of bit length $\ell := 32m - 2$ where $5 \leqslant m \leqslant 20$, thus ranging between 158-bit and 638-bit prime fields, covering security levels roughly between 80 and 192 bits. We denote them by $E_{b,\ell}$ where we include the bit length ℓ in the index. All suggested curves have the form $E_{c^4+1,\ell} : y^2 = x^3 + (c^4 + 1)$ over $\mathbb{F}_p$ with $p = p(u)$, prime order $n = n(u)$, and admit a twist of correct order given by $E' : y^2 = x^3 + (c^2 - i)$ over $\mathbb{F}_{p^2}$. Also, c is always a power of 2. These parameters were obtained from a script in Magma [11] that searches primes with prescribed properties.

We note that the values for u and c uniquely determine all needed parameters, i.e. the primes p and n, the curve equations for E and its twist as well as the generator points. Namely, field extensions $\mathbb{F}_{p^{2r}}$ can be represented directly as

$\mathbb{F}_{p^2}[z]/(z^r - c^2 - i)$ for $r = 2, 3, 6$, or via towers as indicated in Example 1. The pairing groups are $\mathbb{G}_1 = \langle G \rangle$ for $G = (-1, c^2)$, and $\mathbb{G}_2 = \langle H' \rangle$ for $H' = [h](-i, c)$ with $h = 2p - n$, respectively. The low weight of u enables very efficient multiplication by the cofactor h [41, Section 6].

The peculiar choice $\ell = 32m - 2$ deserves some attention, since it is smaller (albeit not by much) than a multiple of typical word sizes (more precisely, a multiple of 8 bits) and hence leads to security levels that are very slightly lower than usual. This was done so that, adopting Montgomery arithmetic in the base field, all values listed here enable all modular reductions involved in an $\mathbb{F}_{p^2}$ multiplication or squaring to be postponed and carried out only once at the very end of that operation, in a very simple and efficient manner as suggested by [8, Section 5.2]. The value $\lfloor 2^{32m}/p \rfloor$ indicates how many modular reductions can be postponed if $\mathbb{F}_p$ elements are held in $32m$-bit variables. With the suggested choice of $\ell = 32m - 2$, $\lfloor 2^{32m}/p \rfloor = 7$ for all examples on Table 1 except for the entry at $\ell = 254$, where it is 6 ($\mathbb{F}_{p^2}$ multiplication and squaring do not need this value to be larger than 5).

Square roots in $\mathbb{F}_{p^2}$ can be efficiently computed with the suggested method, either KCYL [28] or Müller [34].

Example 2. The 254-bit curve corresponding to the parameter $u = -(2^{62} + 2^{55} + 1)$ is given by $E_{2,254} : y^2 = x^3 + 2$ with $G = (-1, 1)$, $E' : y^2 = x^3 + (1 - i)$, $G' = [h](-i, 1)$.

Example 3. All the examples on Table 1 are of the first form of friendly BN curves we suggest. As an example of the second form, to be used in scenarios where efficient cube root computation is desired, one could adopt the 254-bit curve (not listed on Table 1) with $u = -(2^{62} - 2^{49} - 2^2 + 1)$ and $E_{5,254} : y^2 = x^3 + 5$, $G = (-1, 2)$, $E' : y^2 = x^3 + (1 - 2i)$, $G' = [h](-1, 1 - i)$. One can check by direct inspection that $p \equiv 4 \pmod 9$ for this curve.

The particular curve of Example 2 has been apparently first suggested in [38, Section 4.2], and curves with $c = 1$ (and hence $b = 2$), which make up the majority of Table 1, have been singled out in [44], albeit without the benefit of a unified view of the curve equation, its correct twist, and the finite fields involved as pointed out in Section 3. For a detailed discussion see B.

5 Efficiency

It is instructive to compare the relative efficiency of the proposed family with available results in the literature. Curves at roughly the same security level as $E_{2,254}$ of Example 2 appeared in [37, 8, 2]. The results are summarized on Table 2. Following [2, 8], we denote by $\tilde{m}_u$ the number of $\mathbb{F}_{p^2}$ multiplications performed without modular reductions, by $\tilde{s}_u$ the number of $\mathbb{F}_{p^2}$ squarings performed without modular reductions, by $\tilde{r}$ the number of $\mathbb{F}_{p^2}$ modular reductions (counting one half when only a simple $\mathbb{F}_p$ modular reduction is needed), and by $\tilde{a}$ the number of $\mathbb{F}_{p^2}$ additions/subtractions involved in computing one (optimal ate)

Table 1. Sample curves $E_{b,\ell} : y^2 = x^3 + b$ with $b = c^4 + 1$ over $\mathbb{F}_p$, where $p = p(u)$, bitsize of p is $l = 32m - 2$, and suggested method for square root computation in $\mathbb{F}_{p^2}$

m	ℓ	u	$\mathsf{wt}(6u+2)$	c	b	$\sqrt{\mathbb{F}_{p^2}}$
5	158	$-(2^{38} + 2^{28} + 1)$	5	2	17	KCYL
6	190	$-(2^{46} + 2^{23} + 2^{22} + 1)$	5	8	4097	KCYL
7	222	$2^{54} - 2^{44} + 1$	5	4	257	Müller
8	254	$-(2^{62} + 2^{55} + 1)$	5	1	2	KCYL
9	286	$-(2^{70} + 2^{58} + 2^{38} + 1)$	7	1	2	KCYL
10	318	$2^{78} + 2^{62} + 2^1 + 1$	6	1	2	KCYL
11	350	$-(2^{86} - 2^{69} + 2^{28} + 1)$	7	1	2	KCYL
12	382	$-(2^{94} + 2^{76} + 2^{72} + 1)$	7	1	2	KCYL
13	414	$-(2^{102} + 2^{84} - 2^{55} + 1)$	7	1	2	KCYL
14	446	$2^{110} + 2^{36} + 1$	5	4	257	Müller
15	478	$-(2^{118} - 2^{55} - 2^{19} + 1)$	7	1	2	KCYL
16	510	$-(2^{126} + 2^{53} - 2^{50} + 1)$	6	4	257	KCYL
17	542	$-(2^{134} + 2^{114} + 2^{30} + 1)$	7	1	2	KCYL
18	574	$-(2^{142} + 2^{120} - 2^{99} + 1)$	7	1	2	KCYL
19	606	$-(2^{150} - 2^{95} + 2^8 + 1)$	7	1	2	KCYL
20	638	$2^{158} - 2^{128} - 2^{68} + 1$	7	4	257	Müller

pairing, within the Miller loop (ML), the final exponentiation (FE), and the total count (TC).

We also denote by m_u the number of $\mathbb{F}_p$ multiplications performed without modular reductions, by r the number of $\mathbb{F}_p$ modular reductions, and by a the number of $\mathbb{F}_p$ additions/subtractions corresponding to the aforementioned number of operations in $\mathbb{F}_{p^2}$, with each $\mathbb{F}_{p^2}$ multiplication incurring $3m_u + 2r + 8a$ and each $\mathbb{F}_{p^2}$ squaring incurring $2m_u + 2r + 3a$. We do not further break down $\mathbb{F}_{p^2}$ arithmetic for [37], since it is rather based on digit-sliced simultaneous than on separate, single $\mathbb{F}_p$ operations. We only provide operation counts for on-the-fly arithmetic without precomputations.

The figures in Table 2 refer to the following implementation decisions:

- Joint point-and-line computations within the Miller loop as suggested by Costello *et al.* [14, Section 5] (see also [13, Section 4]);
- Tailored multiplication in $\mathbb{F}_{p^{12}}$ to accumulate line function values, which are known to be rather sparse for even embedding degrees [14, Section 3];
- Improved computation of the hard part of the final exponentiation as proposed by Scott *et al.* [42];
- Squaring in $\mathbb{F}_{p^{12}}$ via the Chung-Hasan SQR_3 algorithm [12], which positively affects both the joint point-and-line computations and the easy part of the final exponentiation after the Miller loop;
- Lazy modular reduction for $\mathbb{F}_{p^2}$ multiplications and squarings that occur inside $\mathbb{F}_{p^{12}}$ multiplications and squarings, as described in [2];
- Improved Granger-Scott squaring [22] and Karabina compressed squaring [26] in the cyclotomic subgroup of $G_{\phi_6(q)}$, which positively affects the

Table 2. Experimental comparison of optimal ate pairing performance for 254 bits

source	part	$\tilde{m}_u$	$\tilde{s}_u$	$\tilde{r}$	$\tilde{a}$	m_u	r	a
Naehrig	ML	2022	590	2612	7140	NA	NA	NA
et al. [37]	FE	673	1719	2392	7921	NA	NA	NA
	TC	2695	2309	5004	15061	NA	NA	NA
Beuchat	ML	1952	568	2520	6912	6992	5040	13824
et al. [8]	FE	403	1719	2122	7021	4647	4244	14042
	TC	2355	2287	4642	13933	11639	9284	27866
Aranha	ML	1857	392	1335	10047	6355	2670	20094
et al. [2]	FE	430	1179	963	8435	3648	1926	16870
	TC	2287	1571	2298	18482	10003	4596	36964
This work	ML	1256	1209	1169	10515	6186	2338	21030
	FE	1162	66	902	6795	3618	1804	13590
	TC	2418	1275	2071	17310	9804	4142	34620

hard part of the final exponentiation and also contributes for efficient post-processing (e.g. further exponentiation) of pairing values as needed in several protocols;

- Careful scheduling of the product of conjugates during the computation of the inverse in $\mathbb{F}_{p^{12}}$ (see A) to keep most of the operations in subfields;
- (Minor optimization) Simplification of the final line functions that occur in the optimal ate pairing [37, Section 3.2]. This includes omitting the third line function and multiplying together the sparse values of the two remaining ones.
- (Minor optimization) When the BN parameter u is negative, replacement of the extra inversion incurred by a conjugation after final exponentiation [2].

After a preliminary version of this paper was completed, [2] pointed out that [26] offers further optimization opportunities for exponentiating elements in the cyclotomic subgroup with sparse exponents when using the BN subclass proposed herein. However, due to distinct implementation choices motivated by a rather different research focus, overall [2] does not attain the full level of optimization achievable by following the above list of recommended techniques.

While we do not claim that these figures are optimal, on a high-end processor a C implementation of our proposal using the Miracl [43] library achieves about 9% better pairing computation speed than the best previous results [2] for this particular example, or about 38% better speed than the best results that do not take advantage of the proposed subfamily [8]. A Java implementation on the same platform achieves a pairing computation speed that is 7% better than the techniques in [2] and 70% better than the proposal in [8]. These results are complemented by the bonus of simpler, enhanced arithmetic (including faster square root extraction) in the finite fields $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ underlying groups $\mathbb{G}_1$ and $\mathbb{G}_2$ as needed by most pairing-based protocols. Automatically, this also holds for

arithmetic in $\mathbb{G}_T$ enabled by the tower-friendly nature of the suggested finite-field representation.

Most of the listed techniques are applicable to different parameter choices as well as to other pairing-friendly elliptic curve families. Our proposal was chosen so as to favor the best cases of each technique, and hence to maximize their potential for efficient processing.

6 Conclusion

We have presented a subclass of Barreto-Naehrig curves that generically favors efficient implementation while retaining a very simple description. Our proposal targets not only pairing computation speed but also the efficiency of all typical arithmetic operations needed to instantiate pairing-based cryptographic protocols, and focuses on offering optimization opportunities on the broadest possible landscape of platforms rather than narrowing down to any particular one.

Adopting a BN curve with one of the proposed forms in Definition 1 (regardless of the coefficients c and d) is a sufficient condition for

- doing away with usual square and cube detection tests in $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ when constructing the curve;
- automatically suggesting finite field representations amenable to efficient arithmetic;
- indicating the correct sextic twist directly;
- providing simple generators for both the curve and its twist;
- enabling virtually all optimizations proposed in the literature for all involved algebraic structures.

As a highlight for future research, we point out that one problem still open regarding efficient implementation of pairing-based cryptography is that of deterministic, highly efficient hashing onto the groups $\mathbb{G}_1$ and $\mathbb{G}_2$. Although our proposal partially addresses this problem by supporting the fastest known arithmetic algorithms for these groups (particularly square root extraction), more advanced hashing techniques like that of Icart [25] are currently not applicable to any BN curve. Finding a secure hashing method of that kind for those groups or describing a subclass of BN curves where such a method is available is of great importance for many pairing-based protocols.

Acknowledgments

We are grateful to Diego F. Aranha and Mike Scott for enlightening discussions during the preparation of this work, and to the anonymous reviewers for their valuable comments.

References

1. S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In *Advances in Cryptology – ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer, 2003.
2. D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. López. Faster explicit formulas for computing pairings over ordinary curves. In *Advances in Cryptology – EUROCRYPT 2011*, Lecture Notes in Computer Science, Tallinn, Estonia, 2011. Springer. To appear.
3. P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.
4. P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advances in Cryptology – ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2005.
5. P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2006.
6. P. S. L. M. Barreto, M. Naehrig, and M. Scott. Pairing-friendly curves of prime order with embedding degree 12. IEEE P1363.3 Standard Specifications For Public-Key Cryptography – Identity Based Public Key Cryptography using Pairings, 2007. Technique submitted to standardization body.
7. Naomi Benger and Michael Scott. Constructing tower extensions of finite fields for implementation of pairing-based cryptography. In M. Anwar Hasan and Tor Helleseth, editors, *Arithmetic of Finite Fields – WAIFI 2010, Istanbul, Turkey*, volume 6087 of *Lecture Notes in Computer Science*, pages 180–195. Springer, 2010.
8. J.-L. Beuchat, J. E. González Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. High-speed software implementation of the optimal ate pairing over Barreto-Naehrig curves. In *Pairing-Based Cryptography – Pairing 2010*, volume 6487 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2010.
9. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
10. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432, Warsaw, Poland, 2003. Springer.
11. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).
12. J. Chung and M. A. Hasan. Asymmetric squaring formulae. In *IEEE Symposium on Computer Arithmetic – ARITH 2007*, Proceedings, pages 113–122. IEEE Press, 2007.
13. C. Costello, H. Hisil, C. Boyd, Juan Gonzalez Nieto, and K. K.-H. Wong. Faster pairings on special Weierstrass curves. In *Pairing-Based Cryptography – Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*, pages 89–101. Springer, 2009.

14. C. Costello, T. Lange, and M. Naehrig. Faster pairing computations on curves with high-degree twists. In *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 224–242. Springer, 2010.
15. A. J. Devegili, M. Scott, and R. Dahab. Implementing cryptographic pairings over Barreto-Naehrig curves. In *Pairing-Based Cryptography – Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 197–207. Springer, 2007.
16. J. Fan, F. Vercauteren, and I. Verbauwhede. Faster arithmetic for cryptographic pairings on Barreto-Naehrig curves. In *Cryptographic Hardware and Embedded Systems – CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 2009.
17. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, 2010.
18. S. D. Galbraith, X. Lin, and M. Scott. Endomorphisms for faster elliptic curve cryptography on general curves. In *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 518–535. Springer, 2009.
19. S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
20. S. D. Galbraith and M. Scott. Exponentiation in pairing-friendly groups using homomorphisms. In *Pairing-Based Cryptography – Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 211–224. Springer, 2008.
21. C. P. L. Gouvêa and J. C. López. Software implementation of pairing-based cryptography on sensor networks using the MSP430 microcontroller. In *Progress in Cryptology – Indocrypt 2009*, volume 5922 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 2009.
22. R. Granger and M. Scott. Faster squaring in the cyclotomic subgroup of sixth degree extensions. In *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2010.
23. F. Hess. Pairing lattices. In *Pairing-Based Cryptography – Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 18–38. Springer, 2008.
24. F. Hess, N. P. Smart, and F. Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.
25. T. Icart. How to hash into elliptic curves. In *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 303–316. Springer, 2009.
26. Koray Karabina. Squaring in cyclotomic subgroups. Cryptology ePrint Archive, Report 2010/542, 2010. `http://eprint.iacr.org/2010/542`.
27. N. Koblitz. Elliptic curve cryptosystem. *Mathematics of Computation*, 48(177):203–209, 1987.
28. F. Kong, Z. Cai, J. Yu, and D. Li. Improved generalized Atkin algorithm for computing square roots in finite fields. *Information Processing Letters*, 98(1):1–5, 2006.
29. K. Lauter, P. L. Montgomery, and M. Naehrig. An analysis of affine coordinates for pairing computation. In *Pairing-Based Cryptography – Pairing 2010*, volume 6487 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2010.
30. E. Lee, H. S. Lee, and C.-M. Park. Efficient and generalized pairing computation on Abelian varieties. *IEEE Transactions on Information Theory*, 55(4):1793–1803, 2009.
31. B. Libert and J. J. Quisquater. Improved signcryption from q-Diffie-Hellman problems. In *Security in Communication Networks – SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 220–234. Springer, 2005.

32. V. Miller. Uses of elliptic curves in cryptography. In *Advances in Cryptology – CRYPTO 1985*, volume 218 of *LNCS*, pages 417–426. Springer, 1985.
33. V. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, 2004. See also "Short programs for functions on curves," 1986 unpublished manuscript, `http://crypto.stanford.edu/miller/miller.pdf`.
34. S. Müller. On the computation of square roots in finite fields. *Designs, Codes and Cryptography*, 31(3):301–312, 2004.
35. M. Naehrig. *Constructive and Computational Aspects of Cryptographic Pairings*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2009.
36. M. Naehrig, P. S. L. M. Barreto, and P. Schwabe. On compressible pairings and their computation. In *Progress in Cryptology – AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2008.
37. M. Naehrig, R. Niederhagen, and P. Schwabe. New software speed records for cryptographic pairings. In *Progress in Cryptology – LATINCRYPT 2010*, volume 6212 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2010.
38. Y. Nogami, M. Akane, Y. Sakemi, H. Kato, and Y. Morikawa. Integer variable χ-based ate pairing. In *Pairing-Based Cryptography – Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 178–191. Springer, 2008.
39. K. Rubin and A. Silverberg. Choosing the correct elliptic curve in the CM method. *Mathematics of Computation*, 79:545–561, 2010.
40. M. Scott. Computing the Tate pairing. In *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2005.
41. M. Scott, N. Benger, M. Charlemagne, L. J. Domínguez Pérez, and E. J. Kachisa. Fast hashing to $\mathbb{G}_2$ on pairing friendly curves. In *Pairing-Based Cryptography – Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*, pages 102–113. Springer, 2009.
42. M. Scott, N. Benger, M. Charlemagne, L. J. Domínguez Pérez, and E. J. Kachisa. On the final exponentiation for calculating pairings on ordinary elliptic curves. In *Pairing-Based Cryptography – Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*, pages 78–88. Springer, 2009.
43. Shamus Software. Multiprecision integer and rational arithmetic C/C++ library (MIRACL) v.5.4.4. `http://www.shamus.ie/`, 2010.
44. M. Shirase. Barreto-Naehrig curve with fixed coefficient. IACR ePrint Archive, report 2010/134, 2010. `http://eprint.iacr.org/2010/134`.
45. F. Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.
46. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Public Key Cryptography – PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 277–290. Springer, 2004.

A Scheduling the product of conjugates for $\mathbb{F}_{p^{12}}$ inversion

We now describe a simple technique to save operations when inverting an element $\gamma \in \mathbb{F}_{q^6}$ by keeping most of the involved operations in subfields. We follow the notation of Example 1 and Section 5.

One can invert $\gamma \in \mathbb{F}_{q^6} \setminus \{0\}$ by computing (see e.g. [29, Section 3.1])

$$\gamma^{-1} = \gamma^{v-1} \cdot \gamma^{-v},$$

where $v := 1+q+q^2+q^3+q^4+q^5$. Defining the quantities $\lambda := \gamma^{1+q^3} \in \mathbb{F}_{q^3}$, $\mu := \lambda^q \cdot \lambda^{q^2} \in \mathbb{F}_{q^3}$, $\varepsilon := \gamma^v \in \mathbb{F}_q$, and $\eta := \mu \cdot \varepsilon^{-1}$, we can write $\varepsilon = \gamma^{1+q+q^2+q^3+q^4+q^5} = \lambda \cdot \mu$ and $\gamma^{v-1} = \gamma^{q+q^2+q^3+q^4+q^5} = \gamma^{q^3} \cdot \gamma^{q(1+q^3)} \cdot \gamma^{q^2(1+q^3)} = \gamma^{q^3} \cdot \lambda^q \cdot \lambda^{q^2} = \gamma^{q^3} \cdot \mu$, whereby $\gamma^{-1} = \gamma^{q^3} \cdot (\mu \cdot \varepsilon^{-1}) = \gamma^{q^3} \cdot \eta$.

Writing $\gamma = \alpha + \beta w$ for $\alpha, \beta \in \mathbb{F}_{q^3}$, one sees that $\lambda = (\alpha + \beta w)(\alpha - \beta w) = \alpha^2 - \beta^2 \xi$, where the $\mathbb{F}_{q^3}$ squarings can be performed via the Chung-Hasan SQR_3 method for $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$, incurring an overall cost of $2\tilde{m}+8\tilde{s}$, while $\mu = (\lambda \cdot \lambda^q)^q$ can be computed at a cost $3\tilde{m} + 3\tilde{s}$, apart from conjugation. Performing the product $\varepsilon = \lambda \cdot \mu$ requires only $3\tilde{m}$, since this value is known to lie in $\mathbb{F}_q$. Computing $\eta = \mu \cdot \varepsilon^{-1}$ then incurs one $\mathbb{F}_q$ inversion and one multiplication between an element from $\mathbb{F}_{q^3}$ and another from $\mathbb{F}_q$ (which takes $3\tilde{m}$). Finally we are faced with the multiplication $\gamma^{-1} = \gamma^{q^3} \cdot \eta$ between an element from $\mathbb{F}_{q^6}$ and another from $\mathbb{F}_{q^3}$, which takes $2 \cdot 6\tilde{m}$, apart from conjugation.

Therefore, apart from conjugations and one $\mathbb{F}_q$ inversion, the overall cost is $23\tilde{m} + 11\tilde{s} \approx 91m$, which compares well against the corresponding $288m$ cost incurred by naively computing γ^{-1} as a product of conjugates in sequential order.

B Comparison with Shirase curves

Since many curves on Table 1 appear in [44], namely, curves with $b = 2$, it is instructive to compare that proposal with ours.

In a sense, we generalize the idea from [44] of eliminating twist tests for $b = 2$ to all b satisfying Definition 1 (actually [44] describes how to do the same for $b = \pm 16$ as well, but this choice cannot lead to a suitable BN curve, since b must be a non-square by Lemma 1). However, [44] does not provide a unified view of the curve equation, its correct twist (this is arguably as important as finding the curve itself), and the finite fields involved, nor does it automatically yield suitable generators for *both* E and its twist E', a task that would otherwise require computing square roots explicitly. Costello et al.'s improved arithmetic formulas [14] also remain entirely useful as long as b is small (say, fitting a single processor word) and has low Hamming weight, which is certainly a possibility as indicated by the examples on Table 1 (also in Example 3).

The ability to efficiently compute cube roots is important when threefold pairing compression is desired. For BN curves this is achieved (as pointed out in [5, Section 3.1]) by requiring $p \equiv 4 \pmod 9$. Unfortunately this is impossible for curves with $b = 2$, which need the parameter u that determines $p = p(u)$ to be $2, 11 \pmod{12}$ as indicated in [39, 44], and hence imply $p \equiv 1 \pmod 9$. On the other hand, there are curves in our proposal which do indeed provide for this case, as indicated in Example 3 where $b = 5$.

Finally, curves with $b = 2$ alone may not be enough for all desired field sizes; the examples on Table 1 where $b \neq 2$ illustrate this.