A Family of Implementation-Friendly BN Elliptic Curves

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

[1] Departamento de Engenharia de Computação e Sistemas Digitais,
Escola Politécnica, Universidade de São Paulo, Brazil.
{pbarreto,geovandro,mjunior}@larc.usp.br
[2] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA.
mnaehrig@microsoft.com

Abstract. We describe a class of Barreto-Naehrig (BN) curves that are not only computationally very simple to generate, but also specially suitable for efficient implementation on the broadest possible range of platforms.

1 Introduction

Barreto-Naehrig (BN) curves arguably constitute one of the most versatile classes of pairing-friendly elliptic curves. Among other things, they are known [5] to (this list may not be complete):

- facilitate the deployment of bilinear pairings at the 128-bit level of security [12];
- enable all kinds of pairing-based cryptographic schemes and protocols (including short signatures) [16];
- be plentiful and easily found [29, Section 2.1.1];
- support a sextic twist [21], so the pairing parameters can be defined over relatively small finite fields $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ respectively;
- be amenable to twofold or threefold pairing compression [30];
- attain high efficiency for all pairing computation algorithms known, including the Tate [34], ate [21], eil [20], R-ate [25], Xate [32] and optimal [38] pairings;
- admit optimizations based on endomorphisms and homomorphisms for all groups involved [15, 17], thereby enabling fast non-pairing operations as well;
- be suitable for software and hardware implementations on a wide range of platforms [13, 18].

Recent research has focused on certain individual curves to attain exceptional performance gains [6, 31]. This is essential since pairings are usually the

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

most computationally expensive operation in any pairing-based cryptographic scheme. On the other hand, one may argue that targeting fast pairings alone is insufficient, and can 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. server or clusters) while constrained parties only need to perform non-pairing operations [3, 7, 26, 39]. In such scenarios, 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 parametrized curves with certain prescribed properties, so as to avoid computationally expensive tests during curve generation or, more importantly, curve parameter testing, as required e.g. to check that the purported BN curve contained in a given digital certificate does indeed exhibit the expected properties before using that certificate, so as to avoid attacks. 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, and could 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 to 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 [33]. In contrast, the related tasks of choosing suitable representations for all extension fields involved (which are usually chosen *a priori*, based on features of supporting libraries and oblivious to the peculiar nature of BN curves) and selecting the correct twist $E'(\mathbb{F}_{p^2})$ have received limited attention in the literature and would seem to still need quadratic/cubic character tests in extension fields and full group arithmetic in that twist.

Our contribution in this paper is the definition of a (rather large) subclass of BN curves that is particularly suitable for efficient construction/checking and implementation, while retaining a very simple description. The proposed subclass favours efficiency of all typical arithmetic operations needed to instantiate cryptographic protocols on the broadest possible landscape (in the sense of targeting the widest possible range of platforms and applications). In particular, our construction automatically yields the right twist $E'(\mathbb{F}_{p^2})$ (entirely avoiding curve arithmetic for that purpose) and gives to the 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. [32, 37]), offering additional benefits in those cases.

We stress that it is not our purpose 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. Rather, our goal is to explore a simple yet comprehensive theoretical setting that avoids most if not all general drawbacks and implementation hindrances, while offering and favoring as many optimization opportunities as possible for complete pairing-based cryptosystems, regardless of particular platform idiosyncrasies.

We also point out that the proposed techniques may be useful to obtain optimized settings for other classes of pairing-friendly curves, particularly in the choice of extension field representations. Further exploring these possibilities lies beyond the scope of this work, however.

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. We conclude in Section 5.

2 Preliminaries

Let p be a prime and let $m > 0$. The conjugates of $a \in \mathbb{F}_{p^m}$ are the elements a^{p^i}, $0 \leqslant i < m$. The norm of $a \in \mathbb{F}_{p^m}$ is the product of all its conjugates, $|a| := \prod_i a^{p^i}$. Whenever $p \equiv 3 \pmod 4$ the finite field $\mathbb{F}_{p^2}$ can be represented as $\mathbb{F}_p[i]/(i^2+1)$, mimicking complex numbers. In this analogy, 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 $\mathbb{G}_1$ 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 $E[n] \cap \ker(\phi_p - [1]) = E(\mathbb{F}_p)[n]$, and $\mathbb{G}_2$ 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 $\mu \in \mathbb{F}_{p^k}$, where E' is defined over $\mathbb{F}_{p^e}$ and $e \mid k$ is as small as possible (or, equivalently, where the twist E' has the largest degree $d = k/e$). Typical pairing algorithms are based on Miller's algorithm [27] with a number of optimizations [2, 21, 25, 32, 38], most notably optimal pairings [38] which have loop order of length $\lceil \lg n \rceil / \varphi(k)$ in general (where φ is Euler's totient function), comparing well with the original Tate pairing which has loop order of length $\lceil \lg n \rceil$.

A Barreto-Naehrig (BN) curve [4] is an elliptic curve $E_b : y^2 = x^3 + b$ of prime order $n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1$ over a finite field $\mathbb{F}_p$ where $p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ is prime for some $u \in \mathbb{Z}$. We drop the b subscript and write simply E when the specific equation coefficient b is irrelevant to the discussion or clear from context. 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 $\mathbb{G}_2 = E'(\mathbb{F}_{p^2})[n]$. For BN curves the condition $p \equiv 3$

(mod 4) holds if and only if u is odd, and the loop order of optimal ate pairing is $\omega = |6u + 2|$.

Since BN curves have j-invariant 0, it is relatively easy to find them when compared to pairing-friendly curves from other families (see [14] 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 choses $b \in \mathbb{F}_p$ so that the curve $E : y^2 = x^3 + b$ has 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 $\gcd(h, r) = 1$ [29, 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 these 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 [19], with their associated efficiency gains. □

We will propose a subfamily of BN curves that does away with the quadratic and/or cubic character tests usually needed when deciding how to represent the finite fields 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. ([29, Lemma 2.7]) *Any b that defines a curve $E : y^2 = x^3 + b$ over $\mathbb{F}_p$ of order n such that $2 \nmid n$ and $3 \nmid n$ is neither a square nor a cube in $\mathbb{F}_p$.*

Proof. For any $\gamma, \delta \in \mathbb{F}_p$, the point $(0, \gamma) \in E : y^2 = x^3 + \gamma^2$ has order 3 and hence $3 \mid n$, while the point $(-\delta, 0) \in E : y^2 = x^3 + \delta^3$ 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 this useful result:

Theorem 1. *Let $\xi \in \mathbb{F}_{p^e}$ and let $b = |\xi|$. If $E : y^2 = x^3 + b$ over $\mathbb{F}_p$ has order n such that $2 \nmid n$ and $3 \nmid n$, then ξ is neither a square nor a cube in $\mathbb{F}_{p^e}$.*

Proof. If ξ were 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 $b = \prod_j \xi^{p^j} = \prod_j (\gamma^r)^{p^j} = (\prod_j \gamma^{p^j})^r = |\gamma|^r$, i.e. b would be a square or a cube in $\mathbb{F}_p$, contradicting Lemma 1. □

This means that testing for quadratic or cubic character is not necessary in either $\mathbb{F}_p$ or $\mathbb{F}_{p^e}$. In particular, the element ξ specified in Theorem 1 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 [14]). Pursuing this possibility, however, transcends the scope of this paper.

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

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

Proof. We first notice that the element $\eta := b/\xi^2$ is unitary, i.e. $|\eta| = \eta^{1+p} = 1 = \zeta^3$. Since $1 + p \equiv 2 \pmod 3$, η itself must be a cube, and hence $b/\xi^5 = \eta/\xi^3$ is a cube, as claimed. □

We are finally in a position to state the following theorem:

Theorem 2. *Given a BN curve of form $E : y^2 = x^3 + b$ with $b = |\xi|$ 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. The sextic twist E' of E has one of only two possible orders [21, Proposition 2]. It is known [4, Section 3], on the one hand, that the correct order is $n' := (p - 1 + t)(p + 1 - t)$, which is a multiple of the order $n = p + 1 - t$ of $E(\mathbb{F}_p)$, and on the other hand, that for any $\xi \in \mathbb{F}_{p^2}$ that is neither a square nor a cube, the correct twist is either $E' : y'^2 = x'^3 + b/\xi$ or $E' : y'^2 = x'^3 + b/\xi^5$. Substituting the BN parameters $p(u)$ and $t(u)$ yields $n'(u) \equiv 1 \pmod 2$, i.e. n' must be odd. But since $p \equiv 1 \pmod 3$ and $b = |\xi|$ the value b/ξ^5 is a cube by Lemma 2. This means that $E' : y'^2 = x'^3 + b/\xi^5$ has a point of order 2, hence the order of this particular twist is even. By elimination, $E' : y'^2 = x'^3 + b/\xi$ is the twist one seeks. Notice that $b/\xi = \bar{\xi}$. □

3 The proposed family of curves

We propose to use BN curves of form $E_{b,\ell} : y^2 = x^3 + b$ where either $b = c^4 + d^6$ or $b = c^6 + 4d^4$ for some $c, d \in \mathbb{N} \setminus \{0\}$, the BN prime p satisfies $p \equiv 3 \pmod 4$ (and possibly also $p \equiv 4 \pmod 9$), the Hamming weights of the (signed) binary representations of either the loop order ω of optimal pairings or the BN parameter u (or both) are minimal for each bit length $\ell := \lceil \lg p \rceil$, and b is as small as possible (preferably with low Hamming weight). We slightly abuse notation and identify the integer b with a field element in $\mathbb{F}_p$. The corresponding

$\xi \in \mathbb{F}_{p^2}$ such that $b = |\xi|$ is then $\xi = c^2 + d^3 i$ or $\xi = c^3 + 2d^2 i$, respectively. Notice that the choice of b is consistent with both Theorem 1 and Theorem 2, and it is also compatible with [33, Algorithm 3.5].

The rationale for our proposal is summarized as follows.

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 [36]. Small values of b favor faster pairing computation [10], 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, arise when $b = 2$ and $\xi = 1 + i$, since 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) and 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 [6] 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 Cippolla-Lehmer method simplifies to one quadratic character test and one exponentiation for square roots in $\mathbb{F}_p$, namely, $\sqrt{a} = a^{(p+1)/4}$, and the KCYL [23] method applies to the computation of square roots in $\mathbb{F}_{p^2}$, taking one quadratic character test and 1.5 exponentiations. The case $p^2 \equiv 17 \pmod{32}$ is almost as efficient, taking one quadratic character test and 2 exponentiations to extract roots in $\mathbb{F}_{p^2}$ with the method of [28]. 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 [4, 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 $\mathbb{F}_{p^2}$, and further processing (e.g. explicit or implicit exponentiation) of pairing values need efficient algorithms for $\mathbb{F}_{p^{12}}$ itself, or in some cases for the subfield $\mathbb{F}_{p^6}$ or $\mathbb{F}_{p^4}$, 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 $G = (-c^2, 2d^2)$. Besides, by Theorem 2 the sextic twist of form respectively $E' : y'^2 = x'^3 + (c^2 - d^3 i)$ or $E' : y'^2 = x'^3 + (c^3 - 2d^2 i)$ always contains a subgroup of the same order n as E, and the curve equation for E' admits the obvious solution $G' = (-di, c)$ or $G' = (-c, d(1 - i))$ respectively, so that the point $h \cdot G'$, where $h = p - 1 + t$, only fails to be a generator of $E'(\mathbb{F}_{p^e})[n]$ with negligibly low probability $O(1/h)$. The cofactor multiplication can be carried out very efficiently [35, Section 6].

Suitable field sizes

An obvious bottleneck is $\mathbb{F}_{p^2}$ arithmetic, since it is at the bottom 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 [6, Section 5.2].

4 Sample curves

We provide on Table 1 practical curves of the proposed family for fields of bit length $\ell := 32m - 2$ where $5 \leqslant m \leqslant 20$, thus ranging between 80-bit and 192-bit security levels. All of them have the form $E_{c^4+1,\ell} : y^2 = x^3 + (c^4 + 1)$ over $\mathbb{F}_{p(u)}$, prime order $n(u)$, 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.

Field extensions $\mathbb{F}_{p^{2r}}$ can be represented, if desired, 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 $\mathbb{G}_2 = \langle G' \rangle$ for $G' = h \cdot (-i, c)$ with $h = p - 1 + t$, respectively. The low weight of u enables very efficient multiplication by the cofactor h [35, 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 [6, 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 or squaring does 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 [23] or Müller [28].

Example 2. The parameters for the 254-bit curve defined by $u = -(2^{62}+2^{55}+1)$ are $E_{2,254} : y^2 = x^3 + 2$, $G = (-1, 1)$, $E' : y'^2 = x'^3 + (1 - i)$, $G' = h \cdot (-i, 1)$. □

Example 3. All the examples on Table 1 satisfy the first form of implementation-friendly curves we suggest. As an example of the second form, in scenarios where efficient cube root computation is desired one could adopt the 254-bit curve (not listed on Table 1) defined by $u = -(2^{62} - 2^{49} - 2^2 + 1)$ are $E_{5,254} : y^2 = x^3 + 5$, $G = (-1, 2)$, $E' : y'^2 = x'^3 + (1 - 2i)$, $G' = h \cdot (-1, 1 - i)$. 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 [32, 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 [37], 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.

Table 1. Sample curves $E_{b,\ell}$

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

4.1 Efficiency

It is instructive to compare the relative efficiency of the proposed family with available results in the literature. Curves at the same security level as $E_{2,254}$ of Example 2 appeared in [6] and [31]. The results are summarized on Table 2. Following [6], we denote by $\tilde{m}$ the number of $\mathbb{F}_{p^2}$ multiplications, and by $\tilde{s}$ the

corresponding number of squarings, needed to compute one (optimal ate) pairing, within the Miller loop (ML), the final exponentiation (FE), and the total count (TC). We also provide the number m of equivalent $\mathbb{F}_p$ multiplications incurred, with $\tilde{m} \approx 3m$ and $\tilde{s} \approx 2m$. We only provide operation counts for on-the-fly arithmetic. Precomputation techniques (see e.g. [11]) would yield even better gains where applicable.

Table 2. Experimental comparison of optimal ate pairing performance

source	$\tilde{m}$	$\tilde{s}$	m
Naehrig et al. [31]	2022 (ML)	590 (ML)	7246 (ML)
	673 (FE)	1719 (FE)	5457 (FE)
	2695 (TC)	2309 (TC)	**12703** (TC)
Beuchat et al. [6]	1954 (ML)	568 (ML)	6998 (ML)
	443 (FE)	1719 (FE)	4767 (FE)
	2397 (TC)	2287 (TC)	**11765** (TC)
This work	1256 (ML)	1209 (ML)	6186 (ML)
	400 (FE)	1722 (FE)	4644 (FE)
	1656 (TC)	2931 (TC)	**10830** (TC)

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.* [10, Section 5] (see also [9, 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 [10, Section 3];
- Improved computation of the hard part of the final exponentiation as proposed by Scott *et al.* [36].
- Squaring in $\mathbb{F}_{p^{12}}$ via the Chung-Hasan SQR_3 algorithm [8], which positively affects both the joint point-and-line computations and the easy part of the final exponentiation after the Miller loop;
- Improved Granger-Scott squaring in the cyclotomic subgroup of $\mathbb{F}_{p^{12}}$ [19], which positively affects the hard part of the final exponentiation after the Miller loop 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 Appendix A), so as to keep most of the operations in subfields.
- (Minor optimization) Simplification of the final line functions that occur in the optimal ate pairing [31, 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 [1].

While we do not claim that these figures are optimal, one can see in this example that our proposal achieves about 8% better pairing computation speed

than the best previously reported results [6], with the added 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-friendly protocols, and automatically efficient arithmetic also in $\mathbb{G}_T$ as enabled by the tower-friendly nature of the suggested representation of the involved finite fields.

5 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 the efficiency of all typical arithmetic operations needed to instantiate typical cryptographic protocols, and focuses on offering optimization opportunities on the broadest possible landscape of platforms rather than narrowing down to any particular one.

As a highlight for future research, we point out that one problem still open regarding efficient implementation of pairing-friendly curves is that of deterministic, highly efficient hashing onto the $\mathbb{G}_1$ and $\mathbb{G}_2$ groups. 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 [22] 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.

References

1. D. F. Aranha. Private communication, 2010.
2. 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.
3. 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 *Advanced in Cryptology – Asiacrypt'2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2005.
4. P. S. L. M. Barreto and M. Naehrig. Pairing-friendly curves of prime order. In *Selected Areas in Cryptography – SAC'2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2006.
5. 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.

6. 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*, Lecture Notes in Computer Science. Springer, 2010. To appear.
7. 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.
8. J. Chung and M. A. Hasan. Asymmetric squaring formulae. In *IEEE Symposium on Computer Arithmetic – ARITH'2007*, Proceedings, pages 113–122. IEEE Press, 2007.
9. 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.
10. 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.
11. C. Costello and D. Stebila. Fixed argument pairings. In *Progress in Cryptology – Latincrypt'2010*, volume 6212 of *Lecture Notes in Computer Science*, pages 92–108. Springer, 2010.
12. 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.
13. 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.
14. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, 2010.
15. S. D. Galbraith, X. Lin, and M. Scott. Endomorphisms for faster elliptic curve cryptography on general curves. In *Advanced in Cryptology – Eurocrypt'2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 518–535. Springer, 2009.
16. S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
17. 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.
18. 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.
19. 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.
20. F. Hess. Pairing lattices. In *Pairing-Based Cryptography – Pairing'2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 18–38. Springer, 2008.
21. F. Hess, N. P. Smart, and F. Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.
22. T. Icart. How to hash into elliptic curves. In *Advanced in Cryptology – Crypto'2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 303–316. Springer, 2009.

23. 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.
24. K. Lauter, P. L. Montgomery, and M. Naehrig. An analysis of affine coordinates for pairing computation. In *Pairing-Based Cryptography – Pairing'2010*, Lecture Notes in Computer Science. Springer, 2010. To appear.
25. 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.
26. 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.
27. 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`.
28. S. Müller. On the computation of square roots in finite fields. *Designs, Codes and Cryptography*, 31(3):301–312, 2004.
29. M. Naehrig. *Constructive and Computational Aspects of Cryptographic Pairings*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2009.
30. 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.
31. 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*. Springer, 2010. 109–123.
32. 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.
33. K. Rubin and A. Silverberg. Choosing the correct elliptic curve in the CM method. *Mathematics of Computation*, 79:545–561, 2010.
34. 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.
35. 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.
36. 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.
37. M. Shirase. Barreto-Naehrig curve with fixed coefficient. IACR ePrint Archive, report 2010/134, 2010. `http://eprint.iacr.org/2010/134`.
38. F. Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.
39. 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 describe an efficient implementation of the norm-based inversion algorithm for $\mathbb{F}_{q^6}$ (particularly, but not exclusively, the case $q = p^2$). By carefully scheduling products of conjugates, most of the involved operations are carried out in subfields. We follow the notation of Section 4.1 and Example 1.

One can invert $\gamma \in \mathbb{F}_{q^6} \setminus \{0\}$ by computing (see e.g. [24, 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 $\lambda := \|\gamma\|_{\mathbb{F}_{q^3}} = \gamma^{1+q^3} \in \mathbb{F}_{q^3}$, $\mu := \lambda^q \cdot \lambda^{q^2} \in \mathbb{F}_{q^3}$, $\varepsilon := \|\gamma\|_{\mathbb{F}_q} = \gamma^v \in \mathbb{F}_q$, 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 $\mathbb{F}_q$ [8], incurring a cost $2\tilde{m} + 8\tilde{s}$, 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 involves 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 the overall cost is $23\tilde{m} + 11\tilde{s} \approx 91m$ for $q = p^2$, apart from conjugations and one $\mathbb{F}_q$ inversion.