Efficient hash maps to $\mathbb{G}_2$ on BLS curves

Alessandro Budroni[1] and Federico Pintore[2]

[1] MIRACL Labs, London, England - alessandro.budroni@miracl.com
[2] Department of Mathematics, University of Trento, Italy - federico.pintore@unitn.it

15 May 2017

Abstract

When a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, on an elliptic curve E defined over $\mathbb{F}_q$, is exploited for an identity-based protocol, there is often the need to hash binary strings into $\mathbb{G}_1$ and $\mathbb{G}_2$. Traditionally, if E admits a twist $\tilde{E}$ of order d, then $\mathbb{G}_1 = E(\mathbb{F}_q) \cap E[r]$, where r is a prime integer, and $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{q^{k/d}}) \cap \tilde{E}[r]$, where k is the embedding degree of E w.r.t. r. The standard approach for hashing into $\mathbb{G}_2$ is to map to a general point $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$ and then multiply it by the cofactor $c = \#\tilde{E}(\mathbb{F}_{q^{k/d}})/r$. Usually, the multiplication by c is computationally expensive. In order to speed up such a computation, two different methods (by Scott *et al.* and by Fuentes *et al.*) have been proposed. In this paper we consider these two methods for BLS pairing-friendly curves having $k \in \{12, 24, 30, 42, 48\}$, providing efficiency comparisons. When $k = 30, 42, 48$, the Fuentes *et al.* method requires an expensive one-off pre-computation which was infeasible for the computational power at our disposal. In these cases, we theoretically obtain hashing maps that follow Fuentes *et al.* idea.

Keywords: pairing-based cryptography, pairing-friendly elliptic curves, fast hashing.

1 Introduction

Pairings on elliptic curves have been first used in cryptography to transport elliptic curve discrete logarithms into finite field discrete logarithms ([26], [13]), for which there are index-calculus algorithms running in subexponential time. In recent years, several protocols have been proposed with pairings on elliptic curves as building blocks. Among them, it is possible to enumerate Joux's three party key agreement protocol [19], identity-based encryption [7], non-interactive key-exchange [28] and short signatures schemes [8].

Traditionally, pairings that have been considered for applications are the Tate and Weil pairings on elliptic curves over finite fields, and other related pairings, for example the Eta pairing [4], the Ate pairing [18] and their generalisations [17]. For a given finite field $\mathbb{F}_q$ and an elliptic curve E defined over it, all these pairings take as inputs points on $E(\mathbb{F}_q)$ or on $E(\mathbb{F}_{q^k})$ - where $\mathbb{F}_{q^k}$ is an extension field of the base field $\mathbb{F}_q$ - and return as outputs elements of $(\mathbb{F}_{q^k})^*$. In this paper

we will only consider asymmetric pairings e. In particular, given a prime r such that $r||\#E(\mathbb{F}_q)$, then e will be of the form:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

where $\mathbb{G}_1$ and $\mathbb{G}_2$ are elliptic curve subgroups of order r defined as:

- $\mathbb{G}_1 = E(\mathbb{F}_q) \cap E[r]$,
- $\mathbb{G}_2 = E[r] \cap \{(x, y) \in E(\mathbb{F}_{q^k})|(x^q, y^q) = [q](x, y)\}$,

while $\mathbb{G}_T$ is a subgroup of order r of $(\mathbb{F}_{q^k})^*$. With k is denoted the embedding degree of E with respect to r, i.e. the smallest positive integer such that $r \mid q^k - 1$.

For pairing-based schemes to be secure, the discrete logarithm problems on both $E(\mathbb{F}_q)$ and $(\mathbb{F}_{q^k})^*$ must be computationally infeasible. Those elliptic curves providing a fixed level of security along with efficiency of computations are called *pairing-friendly elliptic curves*. For that class of elliptic curves the first formal definition has been formulated in the comprehensive paper of Freeman *et al.* [12]. The works of Balasubramanian and Koblitz [1] and Luca *et al.* [24] showed that pairing-friendly elliptic curves are rare and then they require dedicated constructions. In recent years, a number of methods for constructing such curves have been proposed ([27], [5], [6], [11], [20]). The general pattern is the same for all of them: given an embedding degree k, three integers n, r, q for which there exists an elliptic curve E defined over $\mathbb{F}_q$ and such that

- $\#E(\mathbb{F}_q) = n$,
- $r||n$,
- k is the embedding degree of E w.r.t. r

are computed. Then the complex multiplication (CM) method is used to determine the equation of the above elliptic curve E.

However, instead of producing single pairing-friendly elliptic curves by means of specific integers k, n, r, q, all the cited methods produce *families* of pairing-friendly elliptic curves. In fact, such methods replace the integers n, r, q with suitable polynomials $n(x), r(x), q(x) \in \mathbb{Q}[x]$. For some appropriate $x_0 \in \mathbb{Z}$, three integers $n(x_0), r(x_0), q(x_0)$ are obtained such that there exists an elliptic curve E defined over $\mathbb{F}_{q(x_0)}$ having $\#E(\mathbb{F}_{q(x_0)}) = n(x_0)$, $r(x_0)||n(x_0)$ and k as embedding degree w.r.t. $r(x_0)$. The triple $\{n(x), r(x), q(x)\}$ defines a *family* of pairing-friendly elliptic curves, each of them parametrised by the integers $n(x_0), r(x_0), q(x_0)$ for some $x_0 \in \mathbb{Z}$. If for every $x_0 \in \mathbb{Z}$ there exists an elliptic curve with $n(x_0), r(x_0), q(x_0)$ as parameters, the family defined by $\{n(x), r(x), q(x)\}$ is said *complete*, otherwise it is called *sparse*.

Among the paring-friendly families (sparse or complete) of curves obtained with the methods enumerated above, we have MNT curves [27], BLS curves [5], BN curves [6], Freeman curves [11] and KSS curves [20].

When pairings on elliptic curves are exploited for identity-based protocols, there is often the need to *hash* binary strings into $\mathbb{G}_1$ and $\mathbb{G}_2$. Hashing to $\mathbb{G}_1$ is relatively easy. In fact, since $\mathbb{G}_1$ is the unique subgroup of order r of $E(\mathbb{F}_q)$, the standard approach is to hash to a general point $P \in E(\mathbb{F}_q)$ and then multiply it by the cofactor $c = \#E(\mathbb{F}_q)/r$. If E admits a twist of degree d that divides k, then $\mathbb{G}_2$ is isomorphic to $\tilde{E}(\mathbb{F}_{q^{k/d}}) \cap \tilde{E}[r]$, where $\tilde{E}$ is a degree d twist of $E/\mathbb{F}_{q^{k/d}}$ [18], and consequently the same approach can be used for hashing into $\mathbb{G}_2$. Nevertheless, the latter requires a multiplication by a large cofactor and hence expensive computations.

In 2009, Scott *et al.* [29] reduced the computational cost of this cofactor multiplication exploiting an efficiently-computable endomorphism $\psi : \tilde{E} \to \tilde{E}$. An improvement of this method was then obtained by Fuentes *et al.* in 2011 [14]. Since pairing-friendly families vary significantly, in order to highlight the benefits of the two methods, families of curves were considered case-by-case in [29] and in [14]. In particular, both papers focus on BN curves ($k = 12$), Freeman curves ($k = 10$) and KSS curves ($k = 8, 18$). However, new advances on the Number Field Sieve ([3], [21]) for computing discrete logarithms in $\mathbb{G}_T$ decrease the security of some asymmetric parings, including those build on BN curves [25], [2].

In the light of these results, BLS curves are attracting more interest, also for efficiency reasons. Despite this, we do not know of any publication or source where both Scott *et al.* and Fuentes *et al.* methods have been explicitly applied to BLS curves with $k \in \{12, 24, 30, 42, 48\}$.
In this paper that gap is filled for BLS curves having $k = 12, 24$, and efficiency comparisons between the two methods are provided. Such a comparison contrasts with that of a recently-published book [10, 8-21] where it is stated that for BLS curves with $k = 12, 24$ the most efficient method for mapping into $\mathbb{G}_2$ is the one proposed by Scott *et al.*. Both methods require a pre-computation to obtain formulas for a specific family of pairing-friendly curves. Scott *et al.* method needs only polynomial modular arithmetic, while Fuentes *et al.* method goes through the application of the LLL algorithm to a polynomial matrix, in order to obtain a lattice's polynomial $h(z)$ having *small* coefficients. We executed the former computation also for BLS curves having $k = 30, 42, 48$, while the latter computation is prohibitive as the embedding degree k grows. Nevertheless, without LLL algorithm, here we supply a suitable polynomial $h(z)$ that allows to speed up cofactor multiplications. Our efficiency conclusions are that hashing following Fuentes *et al.* method is faster than applying Scott *et al.* method, for every $k \in \{12, 24, 30, 42, 48\}$.

The remainder of this paper is organized as follows. In Section 2 we recall Scott *et al.* and Fuentes *et al.* methods. For the sake of easy reference, in Subsection 2.1 we summarise the parameters of BLS curves. In Section 3 and 4, Scott *et al.* and Fuentes *et al.* methods are applied to BLS curves with embedding degree $k \in \{12, 24, 30, 42, 48\}$. Finally, in Section 5 an efficiency comparison between

the two methods is provided.

2 Known methods for efficiently mapping into $\mathbb{G}_2$

The problem of generating random points in $\mathbb{G}_2$, known as *hashing to* $\mathbb{G}_2$, is usually solved selecting a random point $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$ and then computing cP, where c is the cofactor defined as $c = \#\tilde{E}(\mathbb{F}_{q^{k/d}})/r$. Due to the size of c, this scalar multiplication is generally expensive and consequently a bottleneck in hashing to $\mathbb{G}_2$. In [16], Gallant, Lambert and Vanstone give a method to speed up point multiplications wP in $E(\mathbb{F}_q)[r]$. This method is based on the knowledge of a non-trivial multiple of the point P, that is deduced from an efficiently computable endomorphism ψ of E such that $\psi(P)$ is a multiple of P. Starting from this idea, Galbraith and Scott [15] reduce the computational cost of multiplying by the cofactor c introducing a suitable group endomorphism $\psi : \tilde{E} \to \tilde{E}$. Such an endomorphism is defined as $\psi = \phi^{-1} \circ \pi \circ \phi$, where π is the q-power Frobenius on E and ϕ is an isomorphism from the twist curve $\tilde{E}$ to E. The endomorphism ψ satisfies

$$\psi^2(P) - t\psi(P) + qP \tag{1}$$

for all $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$. In the above relation t is the trace of Frobenius, i.e. $\#E(\mathbb{F}_q) = q + 1 - t$. Galbraith and Scott propose to first express the cofactor c to the base q as

$$c = c_0 + c_1 q + \cdots + c_\ell q^\ell \tag{2}$$

and then use (1) to simplify the multiplication cP to

$$[g_0]P + [g_1]\psi(P) + \cdots + [g_{2\ell}]\psi^{2\ell}(P) \tag{3}$$

where $\mid g_i \mid < q$ for every i. This approach is further exploited by Scott *et al.* in [29], where it is applied to several families of pairing-friendly curves. In particular, the curves take into account in [29] are: the MNT curves for the case $k = 6$, the BN curves with $k = 12$, the Freeman curves with $k = 10$ and the KSS curves for the cases $k = 8$ and $k = 18$. It is important to highlight that all these families of curves are defined over a prime field $\mathbb{F}_p$ and they have p, the order r and the trace t expressed as polynomials. Consequently, also the cofactor c can be described as a polynomial of $\mathbb{Q}[x]$. Thanks to that parameterisation Scott *et al.* reduce the cofactor multiplication cP to the evaluation of a polynomial of the powers $\psi^i(P)$, with coefficients that are polynomials in x having degree smaller than $deg(p(x))$. In this way a further speed up in the cofactor multiplication is reached.

Fuentes *et al.* [14] improve Scott *et al.* method observing that in order to compute cP it is sufficient to multiply P by c', a multiple of c such that $c' \not\equiv 0 \pmod{r}$. They show that if $\tilde{E}(\mathbb{F}_{q^{k/d}})$ is cyclic and $q \equiv 1 \pmod{d}$, then there exists a polynomial

$$h(z) = h_0 + h_1 z + \cdots + h_{\varphi(k)-1} z^{\varphi(k)-1} \in \mathbb{Z}[x] \tag{4}$$

such that $h(\psi)P$ is a multiple of cP for all $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$. Furthermore, the coefficients of $h(z)$ are such that $\mid h_i \mid^{\varphi(k)} \leq c$ for all i. The proof of this result is by construction and, exploiting the *LLL algorithm* of Lenstra, Lenstra and Lovasz [23], it leads to a procedure to explicitly compute $h(z)$. For the sake of easy reference we briefly recall the steps of the proof. With $\tilde{n}$ we denote the cardinality $\#\tilde{E}(\mathbb{F}_{q^{k/d}}) = q^{k/d} + 1 - \tilde{t}$, with $\tilde{f}$ the integer such that $\tilde{t}^2 - 4q^{k/d} = D\tilde{f}^2$ (where D is square-free) and, analougously, with f the integer for which $t^2 - 4q = Df^2$. First of all it is observed that, for every point $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$, it holds $\psi(P) = aP$ with:

$$a = \frac{t}{2} \pm \frac{f(\tilde{t} - 2)}{2\tilde{f}} \tag{5}$$

and therefore $[h(\psi)]P = [h(a)]P$. Secondly, it is necessary to note that

$$(\psi_{|\tilde{E}(\mathbb{F}_{q^{k/d}})})^k = id_{\tilde{E}(\mathbb{F}_{q^{k/d}})}.$$

Hence $\Phi_k(a) \equiv 0 \pmod{\tilde{n}}$, where Φ_k is the k-th cyclotomic polynomial (it has degree equal to $\varphi(k)$). This allows us to restrict the search of $h(z)$ into the set of all polynomials of $\mathbb{Z}[z]$ having degree less than $\varphi(k)$. Considering the vectors of the integer lattice generated by the matrix

$$M = \left[\begin{array}{c|c} c & \mathbf{0} \\ \hline \mathbf{a} & I_{\varphi(k)-1} \end{array}\right]$$

as coefficients of $1, z, z^2, \ldots, z^{\varphi(k)-1}$ respectively, we obtain polynomials $h(z) \in \mathbb{Z}[z]$ such that $h(a) \equiv 0 \pmod{c}$. Finally, it is observed that the considered lattice and the set of all vectors $(\pm \mid c \mid^{1/\varphi(k)}, \ldots, \pm \mid c \mid^{1/\varphi(k)})$ have non-empty intersection. A lattice element lying in this intersection could be obtained using the *LLL algorithm* [23]; such an element determines the coefficients of apolynomial $h(z) \in \mathbb{Z}[z]$ with the desired properties.

In [14], such a polynomial is obtained for the BN curves with $k = 12$, the Freeman curves with $k = 10$, the KSS curves for the cases $k = 8$ and $k = 18$. We underline that the computed $h(z)$ have coefficients that are polynomials in x having degrees smaller than $deg(c(x))/\varphi(k)$. Consequently, they compute a formula for hashing into $\mathbb{G}_2$ for each of these curves and compare their computational results with those of Scott *et al.* method for the same curves, showing that their method is faster for all the considered curves.

Families of pairing-friendly curves vary significantly and hence it is not possible to a priori determine if one of the two above hashing methods is more efficient than the other for a given family. BLS curves [5] are recently gaining increasing interest. Thus it is of great concern to determine also for these curves which is, among the Scott *et al.* and the Fuentes *et al.* methods, the more efficient one. In [10, Sec. 8.5], Scott *et al.* method is explicitly applied to BLS curves having $k \in \{12, 24\}$ and authors state that in these cases the most efficient method for

hashing into $\mathbb{G}_2$ is the one proposed by Scott *et al.*.

Concerning BLS curves having $k = \{12, 24\}$, in this paper we deduce the formulas derived from the application of both methods and we provide evidences that, on the contrary, the most efficient method is the one of Fuentes *et al.*. Furthermore, we present analogous formulas for BLS curves with $k \in \{30, 42, 48\}$, together with an efficiency comparison. Focusing on Fuentes *et al.* method, for the the cases $k = 12, 24$ we determine $h(z)$ performing computations with MAGMA [9] on a CPU with an Intel Xeon Process 5460 at 3.16 GHz with a cache of 6 MB. In particular, we exploit MAGMA to apply the implemented *LLL algorithm* to the polynomial matrix M above. However, the same MAGMA computations for the cases $k = 30, 42, 48$ are infeasible with the mentioned computational power. Then we deduce $h(z)$ theoretically. We verify the correctness of the proposed $h(z)$'s by checking that $h(a(x))$ is equivalent to a multiple of $c(x)$ modulo $\tilde{n}(x)$. In particular, we obtain that

$$h(a(x)) \equiv 3(x^{k/6} - 1)c(x) \mod \tilde{n}(x)$$

for $k \in \{30, 42, 48\}$. Furthermore, for $k = 48$ the proposed polynomial is such that $deg(h_i(x)) \leq deg(c(x))/\varphi(k)$ for every i, while for $k = 30, 42$ this condition holds for every $h_i(x)$ except $h_0(x)$, that has degree equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1$.

We conclude this Section briefly recalling BLS curves parameters.

2.1 BLS curves

In 2003 Barreto, Lynn and Scott [5] proposed a polynomial parameterisation for complete pairing-friendly families of curves having fixed embedding degrees, CM discriminant D equal to 3 and short Weierstrass equation $E : y^2 = x^3 + b$. All the curves of these families are defined over prime fields $\mathbb{F}_p$.
For efficiency reasons, in the following we consider only those BLS curves with embedding degree $k \equiv 0 \pmod 6$ and $k \nmid 18$. They admit a twist of degree $d = 6$ [18] and this allow to consider $\mathbb{G}_2$ as a subgroup of $\tilde{E}(\mathbb{F}_{p^{k/6}})$. In this case BLS curves are parameterised by the following polynomials [12]:

$$\begin{aligned} r(x) &= \Phi_k(x) \\ t(x) &= x + 1 \\ p(x) &= \frac{1}{3}(x-1)^2(x^{k/3} - x^{k/6} + 1) + x \end{aligned}$$

where Φ_k is the cyclotomic polynomial of order k.

3 Scott *et al.* method on BLS curves

In this section the Scott *et al.* hashing method is applied to BLS curves having embedding degree k equal to $12, 24, 30, 42$ and 48 respectively. Such an application requires first to determine the cardinality $\tilde{n}(x) \in \mathbb{Q}[x]$ of $\tilde{E}(\mathbb{F}_{p(x)^{k/d}})$ -

where d, in what follows, is always equal to 6 - and then to execute polynomial modular arithmetic as specified in Algorithm 2 of [29].

3.1 BLS-12

For BLS curves with $k = 12$, the prime p and the group order r are parameterised by the polynomials:

$$\begin{aligned} p(x) =& \frac{1}{3}(x-1)^2(x^4 - x^2 + 1) + x; \\ r(x) =& x^4 - x^2 + 1. \end{aligned}$$

Since $k/d = 2$, the group $\mathbb{G}_2$ is expressed as a subgroup of $\tilde{E}(\mathbb{F}_{p(x)^2})$ and the cofactor $c(x)$ is:

$$c(x) = \frac{1}{9}(x^8 - 4x^7 + 5x^6 - 4x^4 + 6x^3 - 4x^2 - 4x + 13) \tag{6}$$

Applying Scott *et al.* method, the scalar multiplication $[3c(x)]P$, for some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^2})$, is reduced to

$$[x^3 - x^2 - x + 4]P + [x^3 - x^2 - x + 1]\psi(P) + [-x^2 + 2x - 1]\psi^2(P) \tag{7}$$

We consider $[3c(x)]P$ instead of $[c(x)]P$ to ignore the common denominator of 3 that occurs writing $c(x)$ to the base $p(x)$. According to [10, sec. 8.5], this can be computed at the cost of 6 point additions, 2 point doublings, 3 scalar multiplications by the parameter x and 3 applications of ψ.

3.2 BLS-24

With the name BLS-24 we denote the BLS curves having embedding degree k equal to 24. Such curves are parameterised by the polynomials:

$$\begin{aligned} p(x) =& \frac{1}{3}(x-1)^2(x^8 - x^4 + 1) + x; \\ r(x) =& x^8 - x^4 + 1. \end{aligned}$$

As before, we consider $[3c(x)]$ instead of $[c(x)]P$ in order to ignore the common denominator of 3 that occurs writing $c(x)$ to the base $p(x)$. In this case $\mathbb{G}_2 \subset \tilde{E}(\mathbb{F}_{p(x)^4})$ and the cofactor is a polynomial $c(x)$ of degree 32. Applying Scott *et al.* method, the scalar multiplication $[3c(x)]P$ - where $P \in \tilde{E}(\mathbb{F}_{p(x)^4})$ - is reduced to

$$\lambda_0 P + \sum_{i=1}^{6} \lambda_i \psi^i(P) \tag{8}$$

where $\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ are polynomials of $\mathbb{Z}[x]$ of degrees less than or equal to 8. These polynomials are fully reported in Appendix A for the sake of readability.

According to [10, sec. 8.5], the multiplication $[3c(x)]P$ can be computed at the cost of 21 point additions, 4 point doublings, 8 scalar multiplications by the parameter x and 6 applications of ψ.

3.3 BLS-30

BLS curves having embedding degree $k = 30$ are parameterised by:

$$\begin{aligned} p(x) =& \frac{1}{3}(x-1)^2(x^{10}-x^5+1)+x; \\ r(x) =& x^8+x^7-x^5-x^4-x^3+x+1. \end{aligned}$$

In this case the cofactor is a polynomial $c(x)$ of degree 52 while $\mathbb{G}_2$ is subgroup of order $r(x)$ of $\tilde{E}(\mathbb{F}_{p(x)^5})$. The Scott *et al.* method leads to express the scalar multiplication $[3c(x)]P$, for some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^5})$, as:

$$\lambda_0 P + \sum_{i=1}^{8} \lambda_i \psi^i(P) \tag{9}$$

where $\{\lambda_j \mid j = 0, \ldots, 8\}$ are polynomials of $\mathbb{Z}[x]$ having degrees less than or equal to 11 (see Appendix A for their details).

Multiplication $[3c(x)]P$ can hence be computed at the cost of 82 point additions, 16 point doublings, 11 scalar multiplications by the parameter x and 67 applications of ψ.

3.4 BLS-42

In the case of BLS curves having $k = 42$, $\mathbb{G}_2$ is the subgroup $\tilde{E}(\mathbb{F}_{p(x)^7}) \cap \tilde{E}[r(x)]$, where:

$$\begin{aligned} p(x) =& \frac{1}{3}(x-1)^2(x^{14}-x^7+1)+x; \\ r(x) =& x^{12}+x^{11}-x^9-x^8+x^6-x^4-x^3+x+1. \end{aligned}$$

The cofactor is parameterised by a polynomial $c(x)$ of degree 100. Writing it to the base $p(x)$, the scalar multiplication $[3c(x)]P$, for some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^7})$, is reduced to

$$\lambda_0 P + \sum_{i=1}^{12} \lambda_i \psi^i(P) \tag{10}$$

where $\{\lambda_j \mid j = 0, \ldots, 12\}$ are polynomials in x with integral coefficients and having degrees less than or equal to 15 (see Appendix A for their complete form).

Then $[3c(x)]P$ can be computed at the cost of 151 point additions, 54 point doublings, 15 scalar multiplications by the parameter x and 125 applications of ψ.

3.5 BLS-48

For BLS curves having $k = 48$, the prime p and the group order r are parameterised by the polynomials:

$$\begin{aligned} p(x) =& \frac{1}{3}(x-1)^2(x^{16}-x^8+1)+x; \\ r(x) =& x^{16}-x^8+1. \end{aligned}$$

The cofactor $c(x)$ is a polynomial of degree 128 and $\mathbb{G}_2$ is a subgroup of $\tilde{E}(\mathbb{F}_{p(x)^8})$. Applying Scott *et al.* method, the scalar multiplication $[3c(x)]P$, for some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^8})$, is reduced to

$$\lambda_0 P + \sum_{i=1}^{14} \lambda_i \psi^i(P) \tag{11}$$

where $\{\lambda_j \mid j = 0, \ldots, 14\}$ are polynomials of $\mathbb{Z}[x]$ having degrees less than or equal to 16 (see Appendix A for details).

As in previous cases, we consider $[3c(x)]P$ instead of $[c(x)P]$ for the common denominator of 3 that occurs writing $c(x)$ to the base $p(x)$. This multiplication can be computed at the cost of 132 point additions, 120 point doublings, 16 scalar multiplications by the parameter x and 130 applications of ψ.

4 Fuentes *et al.* method on BLS curves

In this section we apply the Fuentes *et al.* hashing method to BLS curves having embedding degree k equal to 12 or 24. We have already noticed that this method requires an expensive one-off pre-computation in order to obtain the polynomial $h(z)$. Such a computation was infeasible, for the computational power at our disposal, in the cases $k \in \{30, 42, 48\}$. However, we noticed a shared structure between the two polynomials returned for BLS-12 and BLS-24 curves. Then we exploit such a recursion to theoretically deduce suitable polynomials $h(z)$ also for BLS curves having $k \in \{30, 42, 48\}$. The one proposed for BLS-48 curves satisfies conditions of Theorem 1 in [14]. On the other hand, polynomials $h(z)$'s proposed for BLS-30 and BLS-42 are such that $h(a(x))$ is equivalent to a multiple of $c(x)$ modulo $\tilde{n}(x)$ and $deg(h_i(x)) \leq deg(c(x))/\varphi(k)$ for all i except $i = 0$. In fact, in both cases $h_0(x)$ satisfies the relation

$$\lfloor deg(c(x))/\varphi(k) \rfloor + 1.$$

Therefore, even if for $k \in \{30, 42\}$ we provide polynomials $h(z)$'s that do not fully satisfy conditions of Theorem 1 in [14], what we proposed is extremely tight to a polynomial $h(z)$ fully satisfying such conditions. As we will see, also polynomials $h(z)$'s that we proposed for BLS-30 and BLS-42 curves lead to a speed-up in the cofactor multiplication compared to the Scott *et al.* method.

4.1 BLS-12

For BLS curves with $k = 12$, the parameter a, deduced from (5), is the following polynomial in x:

$$a(x) = \frac{1}{2}\left(t(x) + f(x)\frac{\tilde{t}(x)-2}{\tilde{f}(x)}\right) \pmod{\tilde{n}(x)} = \frac{25}{299}x^{11} - \frac{25}{69}x^{10} + \frac{508}{897}x^9 -$$

$$-\frac{268}{897}x^8 - \frac{112}{897}x^7 + \frac{586}{897}x^6 - \frac{518}{897}x^5 - \frac{126}{299}x^4 + \frac{367}{299}x^3 - \frac{215}{897}x^2 + \frac{64}{299}x + \frac{41}{69}.$$

Reducing the matrix

$$M = \left[\begin{array}{c|c} c(x) & 0\ 0\ 0 \\ -a(x) \mod c(x) & 1\ 0\ 0 \\ -a(x)^2 \mod c(x) & 0\ 1\ 0 \\ -a(x)^3 \pmod{c(x)} & 0\ 0\ 1 \end{array}\right]$$

using the LLL algorithm [23], we obtain

$$M' = \begin{bmatrix} -x+1 & -2 & x-1 & x^2-x+1 \\ -2 & 0 & x^2-x+1 & x-1 \\ 0 & x^2-x-1 & x-1 & 2 \\ x^2-x-1 & x-1 & 2 & 0 \end{bmatrix}.$$

Considering the 4-th row of M', the polynomial $h(z)$ can be defined as

$$h(z) = \sum_{i=1}^{4} M'(4,i)z^{i-1} = (x^2-x-1) + (x-1)z + 2z^2$$

and so

$$h(a) = (x^2-x-1) + (x-1)a + 2a^2 \equiv (3x^2-3)c(x) \pmod{\tilde{n}(x)}$$

with $\gcd(3x^2-3, r(x)) = 1$. Hence, if $P \in \tilde{E}(\mathbb{F}_{p(x)^2})$, then $[h(a)]P$ is a multiple of $[c]P$. In particular:

$$[h(a)]P = [h(\psi)]P = [x^2-x-1]P + [x-1]\psi(P) + \psi^2(2P) \tag{12}$$

that can be computed at the cost of 5 point additions, 1 point doubling, 2 scalar multiplications by the parameter x and 3 applications of ψ.

4.2 BLS-24

Proceeding as in the previous case also for the BLS curves having $k = 24$, we obtain:

$$h(z) = (x^4-x^3-1) + (x^3-x^2)z + (x^2-x)z^2 + (x-1)z^3 + 2z^4$$

with $h(a)$ equivalent to $(3x^4-3)c(x)$ modulo $\tilde{n}(x)$. Since $\gcd(3x^4-3, r(x)) = 1$, the following map sends a point $P \in \tilde{E}(\mathbb{F}_{p(x)^4})$ to a point of $\mathbb{G}_2$:

$$P \to [x^4-x^3-1]P+[x^3-x^2]\psi(P)+[x^2-x]\psi^2(P)+[x-1]\psi^3(P)+2\psi^4(P) \quad (13)$$

To compute the image through such a map, the cost is of 9 point additions, 1 point doubling, 4 scalar multiplications by x and 10 applications of the endomorphism ψ.

4.3 BLS-30

Concerning the cofactor multiplication on BLS curves having embedding degree $k = 30$, we introduce the polynomial

$$h(z) = (x^7-2x^6+2x^5-x^4-x^2+x-1)+\sum_{i=1}^{4}(x^{7-i}-2x^{6-i}+2x^{5-i}-x^{4-i})z^i+(2x^2-2x+2)z^5$$

which leads to the following result.

Proposition 1 *Given a BLS curve with $k = 30$, the map*

$$\begin{aligned} P \to [h(\psi)]P &= \left[x^7 - 2x^6 + 2x^5 - x^4 - x^2 + x - 1\right] P \\ &+ \sum_{i=1}^{4} \left[x^{7-i} - 2x^{6-i} + 2x^{5-i} - x^{4-i}\right] \psi^i(P) + \left[2x^2 - 2x + 2\right] \psi^5(P) \end{aligned} \quad (14)$$

returns a point of $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{p(x)^5}) \cap \tilde{E}[r]$ for every $P \in \tilde{E}(\mathbb{F}_{p(x)^5})$.

Proof. Deducing $a(x)$ from relation (5), it is a straightforward computation to verify that:

$$h(a(x)) \equiv 3(x^5-1)c(x) \mod \tilde{n}(x)$$

with $\gcd(3x^5-3, r(x)) = 1$.

Denoting with $h_0(x), \ldots, h_5(x)$ the coefficients of $h(z)$, we underline that

$$deg(h_i(x)) \leq deg(c(x))/\varphi(k)$$

for all $i \in \{1, \ldots, 5\}$, since $c(x)$ has degree 52 and $\varphi(30) = 8$. On the other hand, $deg(h_0(x))$ is equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1$.

The image (14) can be computed at the cost of 27 point additions, 1 point doubling, 7 scalar multiplications by the parameter x and 25 applications of ψ.

4.4 BLS-42

For the case of BLS curves with embedding degree k equal to 42, we propose the polynomial

$$\begin{aligned} h(z) = &\left(x^9 - 2x^8 + 2x^7 - x^6 - x^2 + x - 1\right) + \\ &+ \sum_{i=1}^{6} \left(x^{9-i} - 2x^{8-i} + 2x^{7-i} - x^{6-i}\right) z^i + \left(2x^2 - 2x + 2\right) z^7 \end{aligned} \tag{15}$$

for which holds the following.

Proposition 2 *Given a BLS curve with $k = 42$ the map*

$$\begin{aligned} P \to &[h(\psi)]P = \left[x^9 - 2x^8 + 2x^7 - x^6 - x^2 + x - 1\right] P \\ &+ \sum_{i=1}^{6} \left[x^{9-i} - 2x^{8-i} + 2x^{7-i} - x^{6-i}\right] \psi^i(P) + \left[2x^2 - 2x + 2\right] \psi^7(P) \end{aligned} \tag{16}$$

returns a point of $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{p(x)^7}) \cap \tilde{E}[r]$ for every $P \in \tilde{E}(\mathbb{F}_{p(x)^7})$.

Proof. As before, once that $a(x)$ is deduced from relation (5), it could be easily verify that:

$$h(a(x)) \equiv 3(x^7 - 1)c(x) \mod \tilde{n}(x)$$

with $\gcd(3x^7 - 3, r(x)) = 1$.

Denoting with $h_0(x), \ldots, h_7(x)$ the coefficients of $h(z)$, it is easy to observe that $deg(h_i(x)) \leq deg(c(x))/\varphi(k)$ for all $i \in \{1, \ldots, 7\}$, since $c(x)$ has degree 100 and $\varphi(42) = 12$. The degree of $h_0(x)$ is equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1$.

The image (16) can be computed at the cost of 33 point additions, 1 point doubling, 9 scalar multiplications by the parameter x and 42 applications of ψ.

4.5 BLS-48

For BLS curves with embedding degree $k = 48$ we introduce the polynomial

$$h(z) = (x^8 - x^7 - 1) + \sum_{i=1}^{7} (x^{8-i} - x^{7-i}) z^i(P) + 2z^8$$

which leads to the following result.

Proposition 3 *Given a BLS curve with $k = 48$ the map*

$$P \to [x^8 - x^7 - 1]P + \sum_{i=1}^{7} [x^{8-i} - x^{7-i}]\psi^i(P) + 2\psi^8(P) \tag{17}$$

returns a point of $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{p(x)^8}) \cap \tilde{E}[r]$ for every $P \in \tilde{E}(\mathbb{F}_{p(x)^8})$. In particular, $h(z)$ satisfies the conditions of Theorem 1 in [14].

Proof. Deducing $a(x)$ from relation (5), it follows that:

$$h(a(x)) \equiv 3(x^8 - 1)c(x) \mod \tilde{n}(x)$$

with $\gcd(3x^8 - 3, r(x)) = 1$. Furthermore, denoting with $h_0(x), \ldots, h_8(x)$ the coefficients of $h(z)$, it is easy to observe that $deg(h_i(x)) \leq deg(c(x))/\varphi(k)$ for all $i \in \{0, \ldots, 8\}$, since $c(x)$ has degree 128 and $\varphi(48) = 16$.

The image (17) can be computed at the cost of 17 point additions, 1 point doubling, 8 scalar multiplications by the parameter x and 36 applications of ψ.

5 Comparisons and conclusions

Here we present an efficiency comparison between the hash maps into $\mathbb{G}_2$ found in the previous two sections. In the following table are reported computational costs for hashing into $\mathbb{G}_2$. The central column concerns results obtained applying Scott *et al.* method (see Section 3). The last column contains computational costs obtained following the *Fuentes et al.* method (see Section 4). With 'A' we denote a point addition, with 'D' a point doubling, with 'Z' a scalar multiplication by the parameter x and with 'ψ' an application of the endomorphism ψ.
We underline that, in each hashing map, the most significant component is the multiplication by x, since it computationally dominates other operations. In fact, the algorithms to compute large scalar multiplications require many point additions and doublings. Furthermore, the endomorphism ψ can be efficiently computed.

Curve	Scott et al.	Fuentes et al.
BLS-12	6A 2D 3Z 3ψ	5A 1D 2Z 3ψ
BLS-24	21A 4D 8Z 6ψ	9A 1D 4Z 10ψ
BLS-30	82A 16D 11Z 67ψ	27A 1D 7Z 25ψ
BLS-42	151A 54D 15Z 125ψ	33A 1D 9Z 42ψ
BLS-48	132A 120D 16Z 130ψ	17A 1D 8Z 36ψ

Table 1. Comparison between the computational cost of each hash map.

In all the cases we have examined the hash maps found following the Fuentes *et al.* method turned out to be more efficient than the ones found with the Scott *et al.* method. Among the hash maps of Section 4, only those for the cases $k = 12, 24$ are obtained applying rigorously Fuentes *et al.* method. For $k = 12$ we see a 3/2-fold improvement, while for $k = 24$ the hash map is twice as fast as that of Scott *et al.*. Concerning BLS curves with $k \in \{30, 42, 48\}$, we theoretically propose suitable polynomials $h(z)$ that satisfy conditions of Theorem 1 in [14] ($k = 48$) or that are extremely tight to a polynomial fully satisfying

such conditions ($k = 30, 42$). For the cases $k = 30$ and $k = 42$, our proposals lead, respectively, to a 11/7-fold improvement and an 11/9-fold improvement with respect to the method of Scott *et al.*. For $k = 48$, the introduced hash map is twice as fast as that of Scott *et al.*.

Using the *Apache Milagro Crypto Library* [22] we implemented the hash maps (7) and (12), obtained applying Scott *et al.* and Fuentes *et al.* methods on BLS curves with embedding degree $k = 12$. In Table 2 we summarise the timing results of a benchmark test on the two maps.

Processor	Scott et al.	Fuentes et al.
Intel(R) Core(TM) i5-5257U 64-bit - 2.7 GHz	2.83 ms	1.98 ms
Quad-core ARM Cortex A53 64-bit - 1.2 GHz	50.26 ms	35.88 ms

Table 2. *Each value corresponds to the average time (in milliseconds) considered for each hash from a sample of 1000 hashes.*

These experimental results show that the hashing map obtained with the Fuentes *et al.* method is approximately 30% faster than the map obtained with the Scott *et al.* method, as we expected from Table 1.

Acknowledgement The authors acknowledge Professor Massimiliano Sala for insightful discussions and for the support, and greatly thank Professor Michael Scott for his critical reading of the manuscript.

References

1. R. Balasubramanian and N. Koblitz. The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes—Okamoto—Vanstone Algorithm. *Journal of Cryptology*, 11(2):141–145, 1998.
2. R. Barbulescu and S. Duquesne. Updating key size estimations for pairings. Cryptology ePrint Archive, Report 2017/334, 2017. `http://eprint.iacr.org/2017/334`.
3. R. Barbulescu, P. Gaudry, and T. Kleinjung. The tower number field sieve. *Advances in Cryptology - ASIACRYPT 2015*, LCNS 9453 (2015):31–55.
4. P. S. L. M. Barreto, S. Galbraith, C. Ó hÉigeartaigh, and M. Scott. Efficient pairing computation on supersingular abelian varieties. *Designs, Codes and Cryptography*, 42(3):239–271, 2007.
5. P. S. L. M. Barreto, B. Lynn, and M. Scott. Constructing Elliptic Curves with Prescribed Embedding Degrees. *International Conference on Security in Communication Networks*. Springer Berlin Heidelberg, 2002.
6. P. S. L. M. Barreto and M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. *International Workshop on Selected Areas in Cryptography*, pages 319–331. Springer Berlin Heidelberg, 2005.

7. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *Advances in Cryptology - CRYPTO 2001*, pages 213–229. Springer Berlin Heidelberg, 2001.
8. D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, 2004.
9. W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *Journal of Symbolic Computation*, 24(3-4):235–265, 1997.
10. N. El Mrabet and M. Joye. *Guide to Pairing-Based Cryptography*. Cryptography and Network Security. Chapman and Hall/CRC, 1st edition, 2017.
11. D. Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. *Algorithmic Number Theory Symposium*, pages 452–465. Springer Berlin Heidelberg, 2006.
12. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, 2010.
13. G. Frey and H. G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation, 62 (206), pp. 865-874*, 1994.
14. L. Fuentes-Castaneda, E. Knapp, and F. Rodriguez-Henriquez. Faster hashing to $\mathbb{G}_2$. *International Workshop on Selected Areas in Cryptography*, pages 412–430. Springer Berlin Heidelberg, 2011.
15. S. D. Galbraith and M. Scott. Exponentiation in pairing-friendly groups using homomorphisms. *International Conference on Pairing-Based Cryptography*, pages 211–224. Springer Berlin Heidelberg, 2008.
16. R. P. Gallant, R. J. Lambert, and S. A. Vanstone. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. *Annual International Cryptology Conference*, pages 190–200. Springer Berlin Heidelberg, 2001.
17. F. Hess. Pairing lattices. *International Conference on Pairing-Based Cryptography*, pages 18–38. Springer Berlin Heidelberg, 2008.
18. F. Hess, N. Smart, and F. Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, Oct. 2006.
19. A. Joux. A One Round Protocol for Tripartite Diffie–Hellman. *International Algorithmic Number Theory Symposium*, pages 385–393. Springer Berlin Heidelberg, 2000.
20. E. J. Kachisa, E. F. Schaefer, and M. Scott. Constructing Brezing-Weng Pairing-Friendly Elliptic Curves Using Elements in the Cyclotomic Field. *International Conference on Pairing-Based Cryptography*, pages 126–135. Springer Berlin Heidelberg, 2008.
21. T. Kim and R. Barbulescu. Extended tower number field sieve: A new complexity for medium prime case. *Annual Cryptology Conference*, LCNS 9814 (2016):543–571. Springer Berlin Heidelberg, 2016.
22. M. Labs. Apache Milagro Crypto Library (AMCL). `https://github.com/miracl/milagro-crypto-c`.
23. A. K. Lenstra, H. W. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen 261.4 (1982): 515-534*, 1982.
24. F. Luca, D. J. Mireles, and I. E. Shparlinski. MOV attack in various subgroups on elliptic curves. *Illinois Journal of Mathematics*, 48(3):1041–1052, 2004.
25. A. Menezes, P. Sarkar, and S. Singh. Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-based Cryptography. *Proceedings of Mycrypt.*, 2016.

26. A. Menezes, S. Vanstone, and T. Okamoto. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *iEEE Transactions on information Theory 39.5 (1993): 1639-1646.*
27. A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals,*, E84 A, no. 5, 1234-1243, May 2001.
28. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. *Symposium on Cryptography and Information Security. SCIS 2000*, 2000. Okinawa, Japan.
29. M. Scott, N. Benger, M. Charlemagne, L. J. D. Perez, and E. J. Kachisa. Fast hashing to G2 on pairing friendly Curves. *International Conference on Pairing-Based Cryptography*, pages 102–113. Springer Berlin Heidelberg, 2009.

Appendix A

Here are listes the polynomials in x which are the coefficients of the hash maps obtained applying the Scott *et al* method to BLS curves having $k = 24, 30, 42, 48$.

BLS-24

Given a rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^4})$, the map (8) sends P into the element $\lambda_0 P + \sum_{i=1}^{6} \lambda_i \psi^i(P)$ of $\mathbb{G}_2$, where:

$$\begin{aligned}
\lambda_0 =& -2x^8 + 4x^7 - 3x^5 + 3x^4 - 2x^3 - 2x^2 + x + 4,\\
\lambda_1 =& x^5 - x^4 - 2x^3 + 2x^2 + x - 1,\\
\lambda_2 =& x^5 - x^4 - x + 1,\\
\lambda_3 =& x^5 - x^4 - x + 1,\\
\lambda_4 =& -3x^4 + x^3 + 4x^2 + x - 3,\\
\lambda_5 =& 3x^3 - 3x^2 - 3x + 3,\\
\lambda_6 =& -x^2 + 2x - 1.
\end{aligned}$$

BLS-30

The map (9) sends $P \in \tilde{E}(\mathbb{F}_{p(x)^5})$ into the element $\lambda_0 P + \sum_{i=1}^{8} \lambda_i \psi^i(P) \in \mathbb{G}_2$, with:

$$\begin{aligned}
\lambda_0 &= x^{11} - x^{10} - 2x^9 + 3x^8 + 2x^7 - 3x^6 - x^5 + 2x^4 - x^3 + 4x^2 + x + 7,\\
\lambda_1 &= x^{11} - 3x^{10} + 3x^9 + x^8 - 5x^7 + x^6 + 4x^5 - x^4 - 4x^3 + 4x^2 - 8x - 11,\\
\lambda_2 &= -x^{10} + 4x^9 - 6x^8 + 5x^7 - 2x^6 + 2x^5 - 5x^4 + 4x^3 - 3x + 11,\\
\lambda_3 &= x^8 - 2x^7 + 2x^6 - x^5 - x^4 + 2x^3 - 2x^2 + x,\\
\lambda_4 &= x^8 - 2x^7 + 2x^6 - x^5 - x^3 + 2x^2 - 2x + 1,\\
\lambda_5 &= -4x^7 + 3x^6 + 2x^5 - x^4 - x^3 + 2x^2 + 3x - 4,\\
\lambda_6 &= 6x^6 - 7x^5 - 3x^4 + 8x^3 - 3x^2 - 7x + 6,\\
\lambda_7 &= -4x^5 + 8x^4 - 4x^3 - 4x^2 + 8x - 4,\\
\lambda_8 &= x^4 - 3x^3 + 4x^2 - 3x + 1.
\end{aligned}$$

BLS-42

The map (10) sends $P \in \tilde{E}(\mathbb{F}_{p(x)^7})$ into the element $\lambda_0 P + \sum_{i=1}^{12} \lambda_i \psi^i(P) \in \mathbb{G}_2$, with:

$$\begin{aligned}
\lambda_0 = &-4x^{15} + 7x^{14} - x^{13} - 4x^{12} + 4x^{11} + 2x^{10} - 4x^9 + 5x^8 - 4x^7 - 2x^6 + 2x^5\\
&- 2x^4 - 4x^3 + 9x^2 + 5x + 9,
\end{aligned}$$

$$\lambda_1 = 6x^{15} - 7x^{14} - 9x^{13} + 15x^{12} - 14x^{10} + 7x^9 - 2x^8 - 5x^7 + 13x^6 - 3x^5 - 7x^4 + 11x^3 + 6x^2 - 22x - 19,$$
$$\lambda_2 = -7x^{14} + 15x^{13} - 4x^{12} - 14x^{11} + 15x^{10} + 2x^9 - 13x^8 + 19x^7 - 9x^6 - 14x^5 + 15x^4 - 16x^2 + 4x + 22,$$
$$\lambda_3 = 2x^{13} - 6x^{12} + 6x^{11} + x^{10} - 8x^9 + 8x^8 - 3x^7 - 9x^6 + 12x^5 + 2x^4 - 13x^3 + 10x^2 + 4x - 6,$$
$$\lambda_4 = -x^{12} + 4x^{11} - 6x^{10} + 5x^9 - 2x^8 + 3x^5 - 7x^4 + 5x^3 + x^2 - 5x + 3,$$
$$\lambda_5 = x^{10} - 2x^9 + 2x^8 - x^7 - x^4 + 2x^3 - 2x^2 + x,$$
$$\lambda_6 = x^{10} - 2x^9 + 2x^8 - x^7 - x^3 + 2x^2 - 2x + 1,$$
$$\lambda_7 = -6x^9 - 2x^8 + 2x^7 + 6x^6 + 6x^3 + 2x^2 - 2x - 6,$$
$$\lambda_8 = 15x^8 + 5x^7 - 19x^6 - 8x^5 + 14x^4 - 8x^3 - 19x^2 + 5x + 15,$$
$$\lambda_9 = -20x^7 + 5x^6 + 30x^5 - 15x^4 - 15x^3 + 30x^2 + 5x - 20,$$
$$\lambda_{10} = 15x^6 - 16x^5 - 12x^4 + 26x^3 - 12x^2 - 16x + 15,$$
$$\lambda_{11} = -6x^5 + 12x^4 - 6x^3 - 6x^2 + 12x - 6,$$
$$\lambda_{12} = x^4 - 3x^3 + 4x^2 - 3x + 1.$$

BLS-48

The map (11) $P \in \tilde{E}(\mathbb{F}_{p(x)^8})$ into the element $\lambda_0 P + \sum_{i=1}^{14} \lambda_i \psi^i(P)$ of $\mathbb{G}_2$, where:

$$\lambda_0 = -6x^{16} - 2x^{15} + 8x^{14} + 14x^{13} - 14x^{11} - 8x^{10} + 3x^9 + 11x^8 + 8x^7 - 14x^5 - 14x^4 + 8x^2 + 5x + 4,$$
$$\lambda_1 = 10x^{15} + 6x^{14} - 26x^{13} - 22x^{12} + 22x^{11} + 26x^{10} - 5x^9 - 11x^8 - 16x^7 - 24x^6 + 10x^5 + 46x^4 + 24x^3 - 16x^2 - 19x - 5,$$
$$\lambda_2 = -14x^{14} + 4x^{13} + 34x^{12} - 34x^{10} - 3x^9 + 13x^8 + 24x^6 + 26x^5 - 34x^4 - 56x^3 + 29x + 11,$$
$$\lambda_3 = 8x^{13} - 8x^{12} - 16x^{11} + 16x^{10} + 9x^9 - 9x^8 - 22x^5 - 10x^4 + 40x^3 + 24x^2 - 19x - 13,$$
$$\lambda_4 = -4x^{12} + 8x^{11} - 7x^9 + 3x^8 + 12x^4 - 4x^3 - 20x^2 + 3x + 9,$$
$$\lambda_5 = x^9 - x^8 - 4x^3 + 4x^2 + 3x - 3,$$
$$\lambda_6 = x^9 - x^8 - x + 1,$$
$$\lambda_7 = x^9 - x^8 - x + 1,$$
$$\lambda_8 = -7x^8 - 13x^7 - 8x^6 + 14x^5 + 28x^4 + 14x^3 - 8x^2 - 13x - 7,$$
$$\lambda_9 = 21x^7 + 43x^6 + 6x^5 - 70x^4 - 70x^3 + 6x^2 + 43x + 21,$$

$$\lambda_{10} = -35x^6 - 55x^5 + 34x^4 + 112x^3 + 34x^2 - 55x - 35,$$
$$\lambda_{11} = 35x^5 + 29x^4 - 64x^3 - 64x^2 + 29x + 35,$$
$$\lambda_{12} = -21x^4 + x^3 + 40x^2 + x - 21,$$
$$\lambda_{13} = 7x^3 - 7x^2 - 7x + 7,$$
$$\lambda_{14} = -x^2 + 2x - 1.$$