Applicable Algebra in Engineering, Communication and Computing (2022) 33:261–281
https://doi.org/10.1007/s00200-020-00453-9

ORIGINAL PAPER

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

Alessandro Budroni[1] · **Federico Pintore**[2]

Received: 4 September 2017 / Revised: 28 January 2019 / Accepted: 6 July 2020 /
Published online: 14 July 2020
© The Author(s) 2020

Abstract

When a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, on an elliptic curve E defined over a finite field $\mathbb{F}_q$, 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. (International conference on pairing-based cryptography. Springer, Berlin, pp 102–113, 2009) and by Fuentes-Castaneda et al. (International workshop on selected areas in cryptography)—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 $k = 42, 48$, the application of Fuentes et al. method requires expensive computations which were infeasible for the computational power at our disposal. For these cases, we propose hashing maps that we obtained following Fuentes et al. idea.

Keywords Pairing-based cryptography · Pairing-friendly elliptic curves · Fast hashing

Mathematics Subject Classification 14G50 · 94A60

A. Budroni: Large part of this work was done when employed at MIRACL Labs, London, England.
F. Pintore: This work was done when at the Department of Mathematics, University of Trento, Italy, and it was supported by CARITRO Foundation.

✉ Federico Pintore
federico.pintore@maths.ox.ac.uk

Alessandro Budroni
alessandro.budroni@uib.no

[1] Department of Informatics, University of Bergen, Bergen, Norway

[2] Mathematical Institute, University of Oxford, Oxford, UK

1 Introduction

1.1 Pairings in cryptography

Pairings on elliptic curves have been first used in cryptography to transport elliptic curve discrete logarithms into finite field discrete logarithms[15, 28], 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[21], a non-interactive key-exchange[32], an identity-based encryption[8], and a short signatures scheme[9].

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[5], the Ate pairing[20], and their generalisations[19]. 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 $r||\#E(\mathbb{F}_q)$ (i.e. $r|\#E(\mathbb{F}_q)$ but $r^2 \nmid \#E(\mathbb{F}_q)$), 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:

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

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 cryptosystems 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*.

1.2 Families of pairing-friendly elliptic curves

The first formal definition of pairing-friendly elliptic curves has been formulated by Freeman et al. in their comprehensive paper[14]. The works of Balasubramanian and Koblitz[2] and Luca et al.[26] show that pairing-friendly elliptic curves are rare, and hence they require dedicated constructions. In recent years a number of methods for constructing such curves have been proposed[6, 7, 10, 13, 22, 29]. 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[31] 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 particular, the integers n, r, q are replaced by suitable polynomials $n(x)$, $r(x)$, $q(x) \in \mathbb{Q}[x]$. For some appropriate $x_0 \in \mathbb{Z}$, $n(x_0), r(x_0), q(x_0)$ are integers such that there exists an elliptic curve E defined over $\mathbb{F}_{q(x_0)}$, having $n(x_0)$ rational points, with $r(x_0)||n(x_0)$, 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*.

The paring-friendly (sparse or complete) families of curves obtained with the methods enumerated above are known as MNT curves[29], BLS curves[6, 10], BN curves[7], Freeman curves[13] and KSS curves[22], respectively.

1.3 Hashing to $\mathbb{G}_2$

When pairings on elliptic curves are exploited for identity-based protocols, there is often the need to map binary strings into $\mathbb{G}_1$ or $\mathbb{G}_2$ in a seemingly random fashion. These problems are known as *hashing to* $\mathbb{G}_1$ and *hashing to* $\mathbb{G}_2$ respectively.

Hashing to $\mathbb{G}_1$ is relatively easy. In fact, since $\mathbb{G}_1$ is the unique subgroup of order r in $E(\mathbb{F}_q)$ (thanks to the assumption $r||\#E(\mathbb{F}_q)$), the standard approach is to hash to a general point $P \in E(\mathbb{F}_q)$ and then multiply it by the cofactor $c = \#E(\mathbb{F}_q)/r$. On the other hand, if E admits a twist of degree d that divides k, then $\mathbb{G}_2$ is isomorphic to $\tilde{E}(\mathbb{F}_{q^{k/d}}) \cap \tilde{E}[r]$ for a unique degree d twist $\tilde{E}$ of E[20]. 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.

We note that the intermediate step of hashing into a general rational point should be handled carefully for efficiency and security reasons. In particular, some cryptosystems are proved to be secure when such an intermediate hash function is modelled as a random oracle into the curve. In order to guarantee its secure replacement with the random oracle, the concept of *indifferentiable* hash function has been introduced[11].

1.4 Related work

In 2009, Scott et al.[33] exploited an efficiently-computable endomorphism $\psi : \tilde{E} \to \tilde{E}$ to reduce the computational cost of the cofactor multiplication required for hashing to $\mathbb{G}_2$. An improvement of this method was then proposed by Fuentes et al.[16]. 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[33] and in[16]. In particular, both papers focus on BN curves with $k = 12$, Freeman curves with $k = 10$ and KSS curves with $k = 8, 18$. However, new advances on the Number Field Sieve ([4, 23, 24]) for computing discrete logarithms in multiplicative groups of finite fields, and hence in $\mathbb{G}_T$, have decreased the security of some asymmetric parings, including those built on BN curves[3, 27]. In the light of these results, BLS curves are attracting more interest for efficiency reasons, since their security has been only slightly reduced by recent NFS advances[3, 27].

A developer using pairings on BLS curves for cryptosystems needing to hash to $\mathbb{G}_2$ during their execution, has to tackle the expensive cofactor multiplications in $\mathbb{G}_2$. Scott et al. and Fuentes et al. methods are the only two proposed so far, that improve on standard point multiplication on elliptic curves. In the light of this, the developer has to choose one of these two methods in order to optimise their implementation. However, to the best of our knowledge, there are not published sources explicitly applying both Scott et al. and Fuentes et al. methods to BLS curves with $k \in \{12, 24, 30, 42, 48\}$, and providing efficiency comparison of the outcomes.

1.5 Contributions and outline

In this paper that gap is filled for BLS curves having $k = 12, 24, 30$, and efficiency comparisons of the results obtained with the two methods are presented. Such a comparison contrasts with a recently-published book[12], 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.

Scott et al. and Fuentes et al. methods both require a pre-computation to obtain parameterised hashing formulas valid for all the curves that belong to a specific family of pairing-friendly curves. In particular, Scott et al. method needs polynomial modular arithmetic, while Fuentes et al. method goes through the application of a generalisation 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 not just for BLS curves with $k \in \{12, 24, 30\}$, but also for BLS curves having $k = 42, 48$. On the other hand, the latter computation is prohibitive as the embedding degree k grows. Consequently, we were able to explicitly apply Scott et al. method also to BLS curves with $k = 42, 48$, but we were not able to accomplish the same for Fuentes et al. method. Nevertheless, for the cases $k = 42$ and $k = 48$ here we propose suitable polynomials $\mathfrak{h}(z)$ having bounded coefficients, which allow to speed up the execution of cofactor multiplications with respect to Scott et al. method.

Our efficiency conclusions are that hashing on BLS curves 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 Sect. 2 we recall Scott et al. and Fuentes et al. methods. For the sake of easy reference, in Sect. 2.1 we summarise BLS curves' parameters. In Sect. 3, Scott et al. method is applied to BLS curves with embedding degree $k \in \{12, 24, 30, 42, 48\}$. In Sect. 4, Fuentes et al. method is applied to BLS curves with $k \in \{12, 24, 30\}$. The proposed polynomials $\mathfrak{h}(z)$, for BLS curves with $k = 42, 48$, are the subject of Sect. 5. Finally, in Sect. 6 an efficiency comparison of the obtained results 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 $[c]P$, 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[18], Gallant, Lambert and Vanstone give a method to speed up scalar multiplications $[w]P$ in $E(\mathbb{F}_q)[r]$. This method is based on the knowledge of a non-trivial multiple of the point P, that is obtained from an efficiently computable endomorphism $\omega : E \to E$ such that $\omega(P)$ is a multiple of P. Building on this idea, Galbraith and Scott[17] reduced 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

$$\psi^2(P) - [t]\psi(P) + [q]P = \infty \tag{1}$$

for all $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$. In the above relation t is the trace of Frobenius $q + 1 - \#E(\mathbb{F}_q)$. Galbraith and Scott proposed 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 as

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

where $| g_i | < q$ for every i.

2.1 Scott et al. method

The above approach was further developed by Scott et al. in[33], where it is applied to several families of pairing-friendly curves. In particular, the curves taken into account in[33] 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 are composed by curves defined over a prime field $\mathbb{F}_p$, with p, the order r and the trace t expressed as polynomials having rational coefficients. Consequently, also the cofactor c can be described as a polynomial in $\mathbb{Q}[x]$. Thanks to such a parameterisation, Scott et al. speed up the cofactor multiplication $[c]P$ reducing it to the evaluation of a polynomial of the powers $\psi^i(P)$, with coefficients that are polynomials in x. Such coefficients are obtained by means of polynomial modular arithmetic. In particular, due to Euclidean Division, all these coefficients have degrees smaller than $deg(p(x))$ (for the same reason, numerical coefficients g_i are bounded by q).

2.2 Fuentes et al. method

Fuentes et al.[16] improved Scott et al. method observing that, in order to obtain a non-zero multiple of $P \in \tilde{E}(\mathbb{F}_q^{k/d})$ having order r, it is sufficient to multiply P by c', a multiple of c such that $c' \not\equiv 0 \pmod{r}$. In particular they proved the following result (see[16], page 11):

Theorem 1 *If $\tilde{E}(\mathbb{F}_{q^{k/d}})$ is cyclic and $q \equiv 1 \pmod{d}$, then there exists a polynomial*

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

such that:

- *$h(\psi)P$ is a multiple of $[c]P$ for all $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$;*
- *the coefficients of $h(z)$ satisfy $\mid h_i \mid^{\varphi(k)} \leq c$ for all i.*

We note that here φ stands for the Euler's totient function, while ψ is the efficiently computable endomorphism satisfying (1).

The first condition about $h(z)$ gives a tool for computing a multiple of $[c]P$ as the sum of some scalar multiplications. These multiplications are computationally light since their scalar factors are bounded thanks to the second condition satisfied by $h(z)$.

The proof of Thereom 1 is by construction and, exploiting the *LLL algorithm* of Lenstra, Lenstra and Lovasz[25], it leads to a procedure to explicitly compute $h(z)$. For the sake of easy reference we briefly sketch the proof's steps.

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 $t^2 - 4q = Df^2$ holds.

First of all it is observed that, for every point $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$, it holds $\psi(P) = [a]P$ with:

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

and therefore $h(\psi)P = [h(a)]P$. Then, the relation

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

is obtained. Hence $\boldsymbol{\Phi}_k(a) \equiv 0 \pmod{\tilde{n}}$, where $\boldsymbol{\Phi}_k$ is the k-th cyclotomic polynomial (which has degree equal to $\varphi(k)$). This allows to restrict the search of $h(z)$ into the set of all polynomials of $\mathbb{Z}[z]$ having degree less than $\varphi(k)$. Denoting with $\mathbf{a}$ the column vector with i-entry $-a^i$, if we consider 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, …, $z^{\varphi(k)-1}$ 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 convex set generated by all vectors of the form $(\pm \mid c \mid^{1/\varphi(k)}, \ldots, \pm \mid c \mid^{1/\varphi(k)})$ have non-empty intersection. A lattice element lying in this intersection could be obtained using the *LLL algorithm*[25]; such an element determines the coefficients of a polynomial $h(z) \in \mathbb{Z}[z]$ with the desired properties.

In[16], 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 $k = 18$. As already observed, these families are composed by curves defined over a prime field $\mathbb{F}_p$, with p, the order r and the trace t expressed as polynomials having rational coefficients. Consequently, also the cofactor c and the scalar a can be described as a polynomials in $\mathbb{Q}[x]$.

The matrix M obtained considering the parameterised forms of c and a is

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

where $\mathbf{a}$(x) is the column vector with i-entry $-a^i(x) \pmod{c(x)}$, and it generates a lattice in $\mathbb{Q}[x]^{\varphi(k)}$. Exploiting the algorithm in[30], the matrix M could be transformed into a new matrix M' having as rows the elements of a reduced basis for the lattice. Considering the polynomials composing a row of M' as coefficients of 1, z, z^2, …, $z^{\varphi(k)-1}$ respectively, Fuentes et al. were able to obtain a polynomial $h(z) = \sum_i h_i(x) z^i \in \mathbb{Z}[x][z]$ satisfying the following two conditions:

(CI) $h(a(x)) \equiv s(x)c(x) \pmod{\tilde{n}(x)}$, with $gcd(s(x), r(x)) = 1$, for some $s(x) \in \mathbb{Q}[x]$;

(CII) $deg(h_i(x)) \leq deg(c(x))/\varphi(k)$, where φ is the Euler's totient function.

The first condition assures that $[a(x_0)]P$ is a non-zero multiple of $[c(x_0)]P$ for every value $x_0 \in \mathbb{Z}$ of the parameter x, and that such a multiple can be computed as the sum of some scalar multiplications. These multiplications are computationally light thanks to the second condition in which scalar factors are bounded.

Consequently, for each of the curves in the above pairing-friendly families, Fuentes et al. compute a formula for hashing into $\mathbb{G}_2$ that is valid for every curve in the family itself. In particular, the cofactor multiplication $[c(x)]P$ is reduced to the evaluation of a polynomial of the powers $\psi^i(P)$, with coefficients that are polynomials in x. Comparing their computational results with those of Scott et al. method for the same families, Fuentes et al. provided evidence that their method is faster for all the considered curves.

2.3 BLS curves

Families of pairing-friendly curves vary significantly, 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 are recently gaining increasing interest[3, 27]. Thus it is of great concern to determine also for these curves which is, among Scott et al. and Fuentes et al. methods, the more efficient one. In[12, 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.

In this paper we deduce the formulas derived from the application of both methods to BLS curves having $k = \{12, 24, 30\}$, and we provide evidences that, on the contrary, the most efficient method is the one of Fuentes et al. Furthermore, we apply Scott et al. method also to BLS curves with $k \in \{42, 48\}$. On the other hand, the computations necessary, within Fuentes et al. method, to obtain the polynomial $h(z)$ for BLS curves having $k = 42, 48$ were infeasible for the computational power at our disposal. Nevertheless, in Sect. 5 we propose two polynomials $\mathfrak{h}(z)$, for the cases $k = 42$ and $k = 48$, that fully satisfies and *almost* fully satisfies conditions (CI), (CII), respectively. In particular, in both cases $\mathfrak{h}(a(x))$ is congruent to a multiple of $c(x)$ modulo $\tilde{n}(x)$, i.e. $\mathfrak{h}(\psi)P$ is a multiple of $[c(x)]P$ for all $P \in \tilde{E}(\mathbb{F}_{q^{k/d}})$. Furthermore, for $k = 48$ the proposed polynomial satisfies the relation $deg(\mathfrak{h}_i(x)) \leq deg(c(x))/\varphi(k)$ for every i, while for $k = 42$ this condition holds for every $\mathfrak{h}_i(x)$ except $\mathfrak{h}_0(x)$, that has degree equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1$.

We conclude this section briefly recalling BLS curves' parameters. Barreto, Lynn and Scott[6], and Brezing and Weng[10] proposed a polynomial parameterisation for complete families of pairing-friendly curves having prime fields $\mathbb{F}_p$ as basefields, fixed embedding degrees, and short Weierstrass equations of the form $y^2 = x^3 + b$.

In the following, we consider only those BLS curves with embedding degree $k \equiv 0 \pmod 6$, and such that $18 \nmid k$. This choice is due to efficiency reasons, since each of such curves admits a twist having the highest possible degree $d = 6$[20], allowing 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[14]:

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

where Φ_k is the cyclotomic polynomial of order k.

3 Scott et al. method on BLS curves

In this section 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 briefly described in the previous section (for further details the reader could refer to Algorithm 2 in[33]).

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:

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

Given some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^2})$, Scott et al. method reduces the scalar multiplication $[3c(x)]P$ 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[12, sec. 8.5], scalar multiplication (7) 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 family of elliptic 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 $[3c(x)]P$ 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. For the sake of readability, these polynomials are fully reported in "Appendix". According to[12, 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 a subgroup of order $r(x)$ of $\tilde{E}(\mathbb{F}_{p(x)^5})$. Given some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^5})$, Scott et al. method leads to express the scalar multiplication $[3c(x)]P$ 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" for their details). The multiplication $[3c(x)]P$ can 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 $[3c(x)]P$, with $P \in \tilde{E}(\mathbb{F}_{p(x)^7})$, 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 degrees less than or equal to 15 (see "Appendix" 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:

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

The cofactor $c(x)$ is a polynomial of degree 128 and $\mathbb{G}_2$ is a subgroup of $\tilde{E}(\mathbb{F}_{p(x)^8})$. Given some rational point $P \in \tilde{E}(\mathbb{F}_{p(x)^8})$, Scott et al. method reduces the scalar multiplication $[3c(x)]P$ 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" 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 scalar 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 with $k = 12, 24, 30$

In this section we apply Fuentes et al. hashing method to BLS curves having embedding degree k equal to 12, 24 or 30. 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, when $k \in \{42, 48\}$. This two cases will be considered in the next section.

4.1 BLS-12

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

$$a(x) = \frac{1}{2}\left(t(x) + f(x)\frac{\tilde{t}(x) - 2}{\tilde{f}(x)}\right) \equiv \frac{25}{299}x^{11} - \frac{25}{69}x^{10} + \frac{508}{897}x^9 - \frac{268}{897}x^8 - \frac{112}{897}x^7 +$$
$$+ \frac{586}{897}x^6 - \frac{518}{897}x^5 - \frac{126}{299}x^4 + \frac{367}{299}x^3 - \frac{215}{897}x^2 + \frac{64}{299}x + \frac{41}{69} \pmod{\tilde{n}(x)}.$$

Reducing the matrix

$$M = \begin{bmatrix} c(x) & 0 & 0 & 0 \\ -a(x) \mod c(x) & 1 & 0 & 0 \\ -a(x)^2 \mod c(x) & 0 & 1 & 0 \\ -a(x)^3 \mod c(x) & 0 & 0 & 1 \end{bmatrix}$$

by means of the algorithm[30], we obtain

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

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

$$h(z) = \sum_{i=1}^{4} M'(4,i) z^{i-1} = (x^2 - x - 1) + (x-1)z + 2z^2 \tag{12}$$

and so

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

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

$$[h(a(x))]P = h(\psi)P = [x^2 - x - 1]P + [x-1]\psi(P) + [2]\psi^2(P), \tag{13}$$

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 a polynomial $a(x)$ of degree 39, and $h(z)$ defined as:

$$h(z) = (x^4 - x^3 - 1) + (x^3 - x^2)z + (x^2 - x)z^2 + (x-1)z^3 + 2z^4 \tag{14}$$

with $h(a(x))$ congruent 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 $\mathbb{G}_2$:

$$P \mapsto [x^4 - x^3 - 1]P + [x^3 - x^2]\psi(P) + [x^2 - x]\psi^2(P) + [x-1]\psi^3(P) + [2]\psi^4(P). \tag{15}$$

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

4.3 BLS-30

In the case of BLS curves having embedding degree $k = 30$, Fuentes et al. method leads to a polynomial $a(x)$ having degree equal to 59 and to a polynomial $h(z)$ defined as follows:

$$\begin{aligned} h(z) =&(x^5 - x^4 - 1) + (-x^5 + 2x^4 - x^3 + 1)z + (x^5 - 2x^4 + 2x^3 - x^2 - 1)z^2 \\ &+ (x^4 - 2x^3 + 2x^2 - x)z^3 + (x^3 - 2x^2 + 2x - 1)z^4 + (x^2 - 2x + 3)z^5 \\ &+ (x - 3)z^6 + 2z^7 \end{aligned} \tag{16}$$

with $h(a(x))$ congruent to $(3x^5 - 3)c(x)$ modulo $\tilde{n}(x)$. Hence the following map

$$\begin{aligned} P \mapsto &[x^5 - x^4 - 1]P + [-x^5 + 2x^4 - x^3 + 1]\psi(P) \\ &+ [x^5 - 2x^4 + 2x^3 - x^2 - 1]\psi^2(P) + [x^4 - 2x^3 + 2x^2 - x]\psi^3(P) \\ &+ [x^3 - 2x^2 + 2x - 1]\psi^4(P) + [x^2 - 2x + 3]\psi^5(P) + [x - 3]\psi^6(P) \\ &+ [2]\psi^7(P) \end{aligned} \tag{17}$$

returns a point of $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{p(x)^5}) \cap \tilde{E}[r(x)]$ when applied to $P \in \tilde{E}(\mathbb{F}_{p(x)^5})$, since $gcd(3x^5 - 3, r(x)) = 1$. The image (17) can be computed at the cost of 25 point additions, 2 point doubling, 5 scalar multiplications by the parameter x and 27 applications of ψ.

5 Fuentes et al. method for BLS curves with $k = 42, 48$

From previous section, we can observe that the degree of the polynomial $a(x)$ grows when k grows. The results provided in Sect. 3 show that the same holds also for $c(x)$. This affects the sizes of the polynomials composing the matrix M, and then the computational cost necessary to reduce it. The computational power at our disposal did not allow us to complete the application of Fuentes et al. method to BLS curves with $k = 42$ or $k = 48$. Nevertheless, the aim of this section is to provide two polynomials $\mathfrak{h}(z)$ that resemble the polynomial $h(z)$ of Theorem 1. We begin considering the case $k = 48$.

5.1 BLS-48

We note that two of the polynomials $h(z)$ obtained in the previous section, precisely (12) and (14), share some common features:

- they both have degree $k/6$;
- their leading coefficients are equal to 2;
- given z^i, its coefficient is $x^{deg(h(z))-i} - x^{deg(h(z))-i-1}$ whenever $0 < i < k/6$;
- the constant terms are equal to $x^{deg(h(z))} - x^{deg(h(z))-1} - 1$.

When $k = 48$, the polynomial $\mathfrak{h}(z)$

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

has the above features and, surprisingly, satisfies the two conditions (CI), (CII), as proved in the following.

Proposition 2 *Given a BLS curve E, defined over $\mathbb{F}_{p(x)}$ and having $k = 48$, the polynomial*

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

satisfies the two conditions:

- *$\mathfrak{h}(\psi)P$ is a multiple of $[c(x)]P$ for all $P \in \tilde{E}(\mathbb{F}_{p(x)^{k/d}})$;*
- *the coefficients $\mathfrak{h}_i$ of $\mathfrak{h}(z)$ satisfy $deg(\mathfrak{h}_i(x)) \leq deg(c(x))/\varphi(k)$ for all i.*

and so the map

$$P \mapsto [x^8 - x^7 - 1]P + \sum_{i=1}^{7}[x^{8-i} - x^{7-i}]\psi^i(P) + [2]\psi^8(P) \tag{18}$$

returns a point of $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{p(x)^8}) \cap \tilde{E}[r(x)]$ for every $P \in \tilde{E}(\mathbb{F}_{p(x)^8})$.

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

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

with $\gcd(3x^8 - 3, r(x)) = 1$. Furthermore, denoting with $\mathfrak{h}_0(x), \ldots, \mathfrak{h}_8(x)$ the coefficients of $\mathfrak{h}(z)$, it is easy to observe that $deg(\mathfrak{h}_i(x)) \leq deg(c(x))/\varphi(k)$ for all $i \in \{0, \ldots, 8\}$, since $c(x)$ has degree 128 and $\varphi(48) = 16$. □

The image (18) 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.

5.2 BLS-42

The same approach does not work for the case of BLS curves with embedding degree k equal to 42. Indeed, for $k = 42$, the polynomial

$$(x^7 - x^6 - 1) + \sum_{i=1}^{6}(x^{8-i} - x^{7-i})z^i + 2z^7$$

satisfies the above features but it is not a multiple of $c(x)$. However, we observed that the following relation holds:

$$((x^7 - x^6 - 1) + \sum_{i=1}^{6}(x^{8-i} - x^{7-i})z^i + 2z^7)/c(x) = 3(x^7 - 1)/(x^2 - x + 1).$$

Defining $\mathfrak{h}(z)$ as $(x^2 - x + 1)((x^7 - x^6 - 1) + \sum_{i=1}^{6}(x^{8-i} - x^{7-i})z^i + 2z^7)$, we were able to obtain a multiple of $c(x)$ that *almost* satisfies the two conditions (CI), (CII). This is specified in the following proposition.

Proposition 3 *Given a BLS curve E, defined over $\mathbb{F}_{p(x)}$ and having $k = 42$, the polynomial*

$$\begin{aligned}\mathfrak{h}(z) =& \left(x^9 - 2x^8 + 2x^7 - x^6 - x^2 + x - 1\right) \\ &+ \sum_{i=1}^{6}\left(x^{9-i} - 2x^{8-i} + 2x^{7-i} - x^{6-i}\right)z^i + \left(2x^2 - 2x + 2\right)z^7\end{aligned} \tag{19}$$

is such that:

- *$\mathfrak{h}(\psi)P$ is a multiple of $[c(x)]P$ for all $P \in \tilde{E}(\mathbb{F}_{p(x)^{k/d}})$;*
- *the coefficients $\mathfrak{h}_i$ of $\mathfrak{h}(z)$ satisfy $deg(\mathfrak{h}_i(x)) \leq deg(c(x))/\varphi(k)$ for all $i \neq 0$;*
- *the constant term $\mathfrak{h}_0$ of $\mathfrak{h}(z)$ has degree equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1$.*

Hence the map

$$\begin{aligned}P \mapsto h(\psi)P =& \left[x^9 - 2x^8 + 2x^7 - x^6 - x^2 + x - 1\right]P \\ &+ \sum_{i=1}^{6}\left[x^{9-i} - 2x^{8-i} + 2x^{7-i} - x^{6-i}\right]\psi^i(P) + \left[2x^2 - 2x + 2\right]\psi^7(P)\end{aligned} \tag{20}$$

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

Proof Once that $a(x)$ is deduced from relation (5), it could be verified by computations that:

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

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

Denoting with $\mathfrak{h}_0(x), \ldots, \mathfrak{h}_7(x)$ the coefficients of $\mathfrak{h}(z)$, it could be observed that $deg(\mathfrak{h}_i(x)) \leq deg(c(x))/\varphi(k)$ for all $i \in \{1, \ldots, 7\}$, since $c(x)$ has degree 100 and $\varphi(42)$ is equal to 12. The degree of $\mathfrak{h}_0(x)$ is equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1$. □

The image (20) 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. We observe that $\mathfrak{h}(z)$ does not fully satisfies the condition (CII), since the degree

of $\mathfrak{h}_0(x) = x^9 - 2x^8 + 2x^7 - x^6 - x^2 + x - 1$ is equal to $\lfloor deg(c(x))/\varphi(k) \rfloor + 1 = 9$, instead of being of degree less than or equal to 8. A coefficient $\mathfrak{h}_0$ with the latter degree would have save one point multiplication by the parameter x and a point addition. This gives an idea of the reason why the degrees of coefficients $\mathfrak{h}_i$ are relevant in terms of efficiency.

6 Comparisons and conclusions

Here we present an efficiency comparison between the hash maps into $\mathbb{G}_2$ found in the previous three sections. In Table 1, computational costs for hashing into $\mathbb{G}_2$ are reported. The second column refers to the results obtained applying Scott et al. method (see Sect. 3). The third column contains computational costs obtained applying Fuentes et al. method (see Sect. 4). The last column reports efficiency data relative to the hash functions we proposed in Sect. 5, that resemble those one would obtain applying Fuentes et al. method. 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 the 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.

In all the cases we have examined, the hash map found following Fuentes et al. method turned out to be more efficient than the one found with Scott et al. method. The hash maps of Sect. 4 were obtained applying rigorously Fuentes et al. method. For $k = 12$ we see a 3/2-fold improvement, for $k = 24$ the hash map is twice as fast as that of Scott et al., while for $k = 30$ the hash map determines a 11/5-fold improvement.

Concerning BLS curves with $k \in \{42, 48\}$, we propose two suitable polynomials $\mathfrak{h}(z)$: one satisfies the two conditions (CI), (CII) deduced from Theorem 1 ($k = 48$); the other is extremely tight to a polynomial fully satisfying such conditions ($k = 42$). The hash function deduced for the case $k = 42$ leads to a 15/9-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* [1] we implemented the hash maps (7) and (13), obtained applying Scott et al. and Fuentes et al. methods on BLS curves

Table 1 Comparison between the computational cost of each hash map

Curve	Scott et al.	Fuentes et al.	Our proposals
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	25A 2D 5Z 27ψ	
BLS-42	151A 54D 15Z 125ψ		33A 1D 9Z 42ψ
BLS-48	132A 120D 16Z 130ψ		17A 1D 8Z 36ψ

with embedding degree $k = 12$. In Table 2 we summarise the timing results of a benchmark test on the two maps.

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

Acknowledgements 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.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix

In the following we report the polynomials in x which are the coefficients of the hash maps obtained applying 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:

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

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

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

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:

$$\begin{aligned}
\lambda_0 =& -4x^{15} + 7x^{14} - x^{13} - 4x^{12} + 4x^{11} + 2x^{10} - 4x^9 + 5x^8 - 4x^7 - 2x^6 + 2x^5\\
& - 2x^4 - 4x^3 + 9x^2 + 5x + 9,\\
\lambda_1 =& 6x^{15} - 7x^{14} - 9x^{13} + 15x^{12} - 14x^{10} + 7x^9 - 2x^8 - 5x^7 + 13x^6 - 3x^5\\
& - 7x^4 + 11x^3 + 6x^2 - 22x - 19,\\
\lambda_2 =& -7x^{14} + 15x^{13} - 4x^{12} - 14x^{11} + 15x^{10} + 2x^9 - 13x^8 + 19x^7 - 9x^6 - 14x^5\\
& + 15x^4 - 16x^2 + 4x + 22,\\
\lambda_3 =& 2x^{13} - 6x^{12} + 6x^{11} + x^{10} - 8x^9 + 8x^8 - 3x^7 - 9x^6 + 12x^5 + 2x^4 - 13x^3\\
& + 10x^2 + 4x - 6,\\
\lambda_4 =& -x^{12} + 4x^{11} - 6x^{10} + 5x^9 - 2x^8 + 3x^5 - 7x^4 + 5x^3 + x^2 - 5x + 3,\\
\lambda_5 =& x^{10} - 2x^9 + 2x^8 - x^7 - x^4 + 2x^3 - 2x^2 + x,\\
\lambda_6 =& x^{10} - 2x^9 + 2x^8 - x^7 - x^3 + 2x^2 - 2x + 1,\\
\lambda_7 =& -6x^9 - 2x^8 + 2x^7 + 6x^6 + 6x^3 + 2x^2 - 2x - 6,\\
\lambda_8 =& 15x^8 + 5x^7 - 19x^6 - 8x^5 + 14x^4 - 8x^3 - 19x^2 + 5x + 15,\\
\lambda_9 =& -20x^7 + 5x^6 + 30x^5 - 15x^4 - 15x^3 + 30x^2 + 5x - 20,\\
\lambda_{10} =& 15x^6 - 16x^5 - 12x^4 + 26x^3 - 12x^2 - 16x + 15,\\
\lambda_{11} =& -6x^5 + 12x^4 - 6x^3 - 6x^2 + 12x - 6,\\
\lambda_{12} =& x^4 - 3x^3 + 4x^2 - 3x + 1.
\end{aligned}$$

BLS-48

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

$$\begin{aligned}
\lambda_0 =& -6x^{16} - 2x^{15} + 8x^{14} + 14x^{13} - 14x^{11} - 8x^{10} + 3x^9 + 11x^8 + 8x^7 - 14x^5 \\
& - 14x^4 + 8x^2 + 5x + 4, \\
\lambda_1 =& 10x^{15} + 6x^{14} - 26x^{13} - 22x^{12} + 22x^{11} + 26x^{10} - 5x^9 - 11x^8 - 16x^7 - 24x^6 \\
& + 10x^5 + 46x^4 + 24x^3 - 16x^2 - 19x - 5, \\
\lambda_2 =& -14x^{14} + 4x^{13} + 34x^{12} - 34x^{10} - 3x^9 + 13x^8 + 24x^6 + 26x^5 - 34x^4 - 56x^3 \\
& + 29x + 11, \\
\lambda_3 =& 8x^{13} - 8x^{12} - 16x^{11} + 16x^{10} + 9x^9 - 9x^8 - 22x^5 - 10x^4 + 40x^3 + 24x^2 \\
& - 19x - 13, \\
\lambda_4 =& -4x^{12} + 8x^{11} - 7x^9 + 3x^8 + 12x^4 - 4x^3 - 20x^2 + 3x + 9, \\
\lambda_5 =& x^9 - x^8 - 4x^3 + 4x^2 + 3x - 3, \\
\lambda_6 =& x^9 - x^8 - x + 1, \\
\lambda_7 =& x^9 - x^8 - x + 1, \\
\lambda_8 =& -7x^8 - 13x^7 - 8x^6 + 14x^5 + 28x^4 + 14x^3 - 8x^2 - 13x - 7, \\
\lambda_9 =& 21x^7 + 43x^6 + 6x^5 - 70x^4 - 70x^3 + 6x^2 + 43x + 21, \\
\lambda_{10} =& -35x^6 - 55x^5 + 34x^4 + 112x^3 + 34x^2 - 55x - 35, \\
\lambda_{11} =& 35x^5 + 29x^4 - 64x^3 - 64x^2 + 29x + 35, \\
\lambda_{12} =& -21x^4 + x^3 + 40x^2 + x - 21, \\
\lambda_{13} =& 7x^3 - 7x^2 - 7x + 7, \\
\lambda_{14} =& -x^2 + 2x - 1.
\end{aligned}$$

References

1. Apache Milagro Crypto Library (AMCL): MIRACL Labs. https://github.com/milagro-crypto/milagro-crypto-c
2. Balasubramanian, R., Koblitz, N.: The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm. J. Cryptol. **11**(2), 141–145 (1998)
3. Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. J. Cryptol. **32**(4), 1298–1336 (2019)
4. Barbulescu, R., Gaudry, P., Kleinjung, T.: The tower number field sieve. In: Advances in Cryptology—ASIACRYPT 2015, LCNS 9453, pp. 31–55 (2015)
5. Barreto, P.S.L.M., Galbraith, S., Ó'hÉigeartaigh, C., Scott, M.: Efficient pairing computation on supersingular abelian varieties. Des. Codes Crypt. **42**(3), 239–271 (2007)
6. Barreto, P.S.L.M., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: International Conference on Security in Communication Networks. Springer, Berlin (2002)

7. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: International Workshop on Selected Areas in Cryptography, pp. 319–331. Springer, Berlin (2005)
8. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Advances in Cryptology—CRYPTO 2001, pp. 213–229. Springer, Berlin (2001)
9. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. J. Cryptol. **17**(4), 297–319 (2004)
10. Brezing, F., Weng, A.: Elliptic curves suitable for pairing based cryptography. Des. Codes Crypt. **37**(1), 133–141 (2005)
11. Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M.: Efficient indifferentiable hashing into ordinary elliptic curves. In: Annual Cryptology Conference, pp. 237–254. Springer, Berlin (2010)
12. El Mrabet, N., Joye, M.: Guide to Pairing-Based Cryptography. Cryptography and Network Security, 1st edn. Chapman and Hall, Boca Raton (2017)
13. Freeman, D.: Constructing pairing-friendly elliptic curves with embedding degree 10. In: Algorithmic Number Theory Symposium, pp. 452–465. Springer, Berlin (2006)
14. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptol. **23**(2), 224–280 (2010)
15. Frey, G., Rück, H.G.: A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Math. Comput. **62**(206), 865–874 (1994)
16. Fuentes-Castaneda, L., Knapp, E., Rodriguez-Henriquez, F.: Faster hashing to $\mathbb{G}_2$. In: Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 2011, Revised Selected Papers, pp. 412–430 (2011)
17. Galbraith, S.D., Scott, M.: Exponentiation in pairing-friendly groups using homomorphisms. In: International Conference on Pairing-Based Cryptography, pp. 211–224. Springer, Berlin (2008)
18. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: Annual International Cryptology Conference, pp. 190–200. Springer, Berlin (2001)
19. Hess, F.: Pairing lattices. In: International Conference on Pairing-Based Cryptography, pp. 18–38. Springer, Berlin (2008)
20. Hess, F., Smart, N., Vercauteren, F.: The eta pairing revisited. IEEE Trans. Inf. Theory **52**(10), 4595–4602 (2006)
21. Joux, A.: A one round protocol for tripartite Diffie–Hellman. In: International Algorithmic Number Theory Symposium, pp. 385–393. Springer, Berlin (2000)
22. Kachisa, E.J., Schaefer, E.F., Scott, M.: Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In: International Conference on Pairing-Based Cryptography, pp. 126–135. Springer, Berlin (2008)
23. Kim, T., Barbulescu, R.: Extended tower number field sieve: a new complexity for medium prime case. In: Annual Cryptology Conference, LCNS 9814, pp. 543–571. Springer, Berlin (2016)
24. Kim, T., Jeong, J.: Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. In: IACR International Workshop on Public Key Cryptography, pp. 388–408. Springer, Berlin (2017)
25. Lenstra, A.K., Lenstra, H.W., Lovasz, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
26. Luca, F., Mireles, D.J., Shparlinski, I.E.: MOV attack in various subgroups on elliptic curves. Ill. J. Math. **48**(3), 1041–1052 (2004)
27. Menezes, A., Sarkar, P., Singh, S.: Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In: Proceedings of Mycrypt (2016)
28. Menezes, A., Vanstone, S., Okamoto, T.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Trans. Inf. Theory **39**(5), 1639–1646 (1993)
29. Miyaji, A., Nakabayashi, M., Takano, S.: New explicit conditions of elliptic curve traces for FR-reduction. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **84**(5), 1234–1243 (2001)
30. Paulus, S.: Lattice basis reduction in function fields. In: International Algorithmic Number Theory Symposium. Springer, Berlin (1998)
31. Rubin, K., Silverberg, A.: Choosing the correct elliptic curve in the CM method. Math. Comput. **79**(269), 545–561 (2010)
32. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: Symposium on Cryptography and Information Security. SCIS, Okinawa, Japan (2000)

33. Scott, M., Benger, N., Charlemagne, M., Perez, L.J.D., Kachisa, E.J.: Fast hashing to G2 on pairing friendly Curves. In: International Conference on Pairing-Based Cryptography, pp. 102–113. Springer, Berlin (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.