Faster Hashing to $\mathbb{G}_2$

Laura Fuentes-Castañeda[1], Edward Knapp[2], and Francisco Rodríguez-Henríquez[1]

[1] CINVESTAV-IPN, Computer Science Department
lfuentes@computacion.cs.cinvestav.mx, francisco@cs.cinvestav.mx
[2] University of Waterloo, Dept. Combinatorics & Optimization
edward.m.knapp@gmail.com

Abstract. An asymmetric pairing $e\colon \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{G}_T$ is considered such that $\mathbb{G}_1 = E(\mathbb{F}_p)[r]$ and $\mathbb{G}_2 = \tilde{E}(\mathbb{F}_{p^{k/d}})[r]$, where k is the embedding degree of the elliptic curve $E/\mathbb{F}_p$, r is a large prime divisor of $\#E(\mathbb{F}_p)$, and $\tilde{E}$ is the degree-d twist of E over $\mathbb{F}_{p^{k/d}}$ with $r \mid \tilde{E}(\mathbb{F}_{p^{k/d}})$. Hashing to $\mathbb{G}_1$ is considered easy, while hashing to $\mathbb{G}_2$ is done by selecting a random point Q in $\tilde{E}(\mathbb{F}_{p^{k/d}})$ and computing the hash value cQ, where $c \cdot r$ is the order of $\tilde{E}(\mathbb{F}_{p^{k/d}})$. We show that for a large class of curves, one can hash to $\mathbb{G}_2$ in $\mathrm{O}(1/\varphi(k) \log c)$ time, as compared with the previously fastest-known $\mathrm{O}(\log p)$. In the case of BN curves, we are able to double the speed of hashing to $\mathbb{G}_2$. For higher-embedding-degree curves, the results can be more dramatic. We also show how to reduce the cost of the final-exponentiation step in a pairing calculation by a fixed number of field multiplications.

Keywords: Pairing-based cryptography, fast hashing, final exponentiation

1 Introduction

Let E be an elliptic curve defined over $\mathbb{F}_p$ and let r be a large prime divisor of $\#E(\mathbb{F}_p)$. The embedding degree of E (with respect to r, p) is the smallest positive integer k such that $r \mid p^k - 1$. The Tate pairing on ordinary elliptic curves maps two linearly independent rational points defined over the order-r groups $\mathbb{G}_1$, $\mathbb{G}_2 \subseteq E(\mathbb{F}_{p^k})$ to the group of r-th roots of unity of the finite field $\mathbb{F}_{p^k}$. In practice, the Tate pairing is computed using variations of an iterative algorithm that was proposed by Victor Miller in 1986 [21]. The result is in the quotient group $\mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r$ and is followed by a final exponentiation in order to obtain a unique representative.

Efficient realizations of the Tate pairing have been intensively pursued in recent years. Using different strategies, that research effort has produced several remarkable algorithm improvements that include: construction of pairing-friendly elliptic curves with prescribed embedding degree [4, 8, 23], decreases of the Miller loop length [3, 13, 14, 29], and reductions in the associated towering field arithmetic costs [6, 11, 15, 17].

With the increase in efficiency of the Miller loop calculation, the final exponentiation step has become more of a computational bottleneck. Several research works have reported more refined methods for computing the final exponentiation on pairings defined over ordinary elliptic curves [6, 12, 26]. In particular, the results by Scott *et al.* [26] represent the current state-of-the-art in this topic, as can be verified from the fact that most recent implementations of pairings (see for example [1, 5]) have obtained significant accelerations by computing the final exponentiation according to the vectorial addition chain based method described in that work.

Another important task related to pairing computation that has been less studied is the problem of generating random points in $\mathbb{G}_1$ and $\mathbb{G}_2$, known in the literature as *hashing to* $\mathbb{G}_1$ and *hashing to* $\mathbb{G}_2$, respectively. The group $\mathbb{G}_1$ is defined as $E(\mathbb{F}_p)[r]$. Hashing to $\mathbb{G}_1$ is normally seen as a straightforward task, whereas hashing to $\mathbb{G}_2$ is considered more challenging.

The customary method for representing $\mathbb{G}_2$ is as the order-r subgroup of $\tilde{E}(\mathbb{F}_{p^{k/d}})$, where $\tilde{E}$ is the degree-d twist of E over $\mathbb{F}_{p^{k/d}}$ with $r \mid \#\tilde{E}(\mathbb{F}_{p^{k/d}})$; here $\#S$ denotes the cardinality of S. Hashing to $\mathbb{G}_2$ can be accomplished by finding a random point $Q \in \tilde{E}(\mathbb{F}_{p^{k/d}})$ followed by a multiplication by $c = \#\tilde{E}(\mathbb{F}_{p^{k/d}})/r$. The main difficulty of this hashing is that c is normally a relatively large scalar (for example, larger than p). Galbraith and Scott [10] reduce the computational cost of this task by means of an endomorphism of $\tilde{E}$. This idea was further exploited by Scott *et al.* [27], where explicit formulae for hashing to $\mathbb{G}_2$ were given for several pairing-friendly curves.

In this work, we offer improvements in both the final exponentiation and hashing to $\mathbb{G}_2$. We draw on the methods that Vercauteren [29] employed to reduce the cost of the Miller function. Our results for the final exponentiation reduce the cost by a fixed number of operations in several curves, a modest but measurable improvement. Nonetheless, the techniques we use can be applied to increase the speed of hashing as well, saving a fixed number of point additions and doublings. Our framework for fast hashing produces more dramatic results. For example, we estimate that for BN curves [4] at the 128-bit security level, our results yield a hashing algorithm that is at least two times faster than the previous fastest-known algorithm. For higher-embedding-degree curves, the results can be more dramatic.

The rest of this paper is organized as follows. In Section 2 we review Vercauteren's "optimal" pairings. Sections 3 and 4 present our lattice-based method for computing the final exponentiation and exponentiation examples for several emblematic pairing-friendly elliptic curves, respectively. Sections 5 and 6 give our lattice-based approach for hashing to $\mathbb{G}_2$ and hashing for several families of elliptic curves.

2 Background

The Tate pairing is computed in two steps. First, the Miller function value $f = f_{r,P}(Q) \in \mathbb{F}_{p^k}^*$ is computed. This gives a value in the quotient group $\mathbb{F}_{p^k}^*/(\mathbb{F}_{p^k}^*)^r$.

Second, to obtain a unique representative in this quotient group, the value f is raised to the power $(p^k-1)/r$.

The Miller function is computed using a square-and-multiply algorithm via the following relation

$$f_{a+b,P} = f_{a,P} f_{b,P} \frac{\ell_{aP,bP}}{v_{(a+b)P}}.$$

Using this method, the function $f_{r,P}$ can be computed in $\log r$ steps.

The eta and ate pairings reduces the length of the Miller loop from $\log r$ to $\log |t| \leq \frac{1}{2} \log p$, where t is the trace of the p-power Frobenius acting on E [2, 14]. The R-ate pairing [18] provided further improvement, reducing the Miller loop to length $(1/\varphi(k)) \log r$ in some cases. This idea was further generalized by Vercauteren [29] to reduce the Miller loop length to $(1/\varphi(k)) \log r$ for all curves. The idea behind Vercauteren's result lies in the fact that for $h(p) = \sum_{i=0}^{s} h_i p^i$ divisible by r, we have

$$f_{r,P}^{h(p)/r} = g_P \prod_{i=0}^{s} f_{h_i,p^iP} f_{p^i,P}^{h_i},$$

where g_P is the product of s lines. By observing that $f_{r,P}^{(p^k-1)/r}$, $f_{p,P}^{(p^k-1)/r}$, $\ldots$, $f_{p^s,P}^{(p^k-1)/r}$ are pairings, it follows that

$$\left(g_P \prod_{i=0}^{s} f_{h_i,p^iP} \right)^{(p^k-1)/r} \tag{1}$$

is a pairing. By choosing a polynomial h with small coefficients, Vercauteren showed that the loop length for each Miller function in (1) can be reduced to at most $(1/\varphi(k)) \log r$.

3 A Lattice-based Method for Efficient Final Exponentiation

The exponent $e = (p^k-1)/r$ in the final exponentiation can be broken into two parts by

$$(p^k-1)/r = [(p^k-1)/\Phi_k(p)] \cdot [\Phi_k(p)/r],$$

where $\Phi_k(x)$ denotes the k-th cyclotomic polynomial [17]. Computing the map $f \mapsto f^{(p^k-1)/\Phi_k(p)}$ is relatively inexpensive, costing only a few multiplications, inversions, and very cheap p-th powerings in $\mathbb{F}_{p^k}$. Raising to the power $d = \Phi_k(p)/r$ is considered more difficult.

Observing that p-th powering is much less expensive than multiplication, Scott *et al.* [26] give a systematic method for reducing the expense of exponentiating by d. They showed that by writing $d = \Phi_k(p)/r$ in base p as $d = d_0 + d_1 p + \cdots + d_{\varphi(k)-1} p^{\varphi(k)-1}$, one can find short vectorial addition chains

to compute $f \mapsto f^d$ much more efficiently than the naive method. For parameterized curves, more concrete results can be given. For instance, Barreto-Naehrig curves [4] are constructed over a prime field $\mathbb{F}_p$, where p is a large prime number that can be parameterized as a fourth-degree polynomial $p = p(x)$, $x \in \mathbb{Z}$. The result of Scott *et al.* gives an algorithm to compute $f \mapsto f^d$, by calculating three intermediate exponentiations, namely, f^x, $(f^x)^x$, $(f^{x^2})^x$, along with a short sequence of products. By choosing the parameter $x \in \mathbb{Z}$ to have low hamming weight, the total cost of computing $f \mapsto f^d$ is $\frac{3}{4} \log p$ field squarings plus a small fixed-number of field multiplications and squarings.

Using the fact that a fixed power of a pairing is also a pairing, it suffices to raise to the power of any multiple d' of d, with r not dividing d'. Based on this observation, we present a lattice-based method for determining d' such that $f \mapsto f^{d'}$ can be computed at least as efficiently as $f \mapsto f^d$. For Barreto-Naehrig and several other curves, explicit d' polynomials yielding more-efficient final exponentiation computations are reported. However, it is noted that the main bottleneck remains, namely the exponentiation by powers of x.

In the case of parameterized curves, the key to finding suitable polynomials d' is to consider $\mathbb{Q}[x]$-linear combinations of $d(x)$. Specifically, we consider $\mathbb{Q}$-linear combinations of $d(x)$, $xd(x)$, $\ldots$, $x^{\deg r-1}d(x)$. To see why this set of multiples of $d(x)$ suffices, consider $f \in \mathbb{F}_{p^k}$ with order dividing $\Phi_k(p)$. Since $r(x)d(x) = \Phi_k(p)$, it follows that $f^{r(x)d(x)} = 1$ and so $f^{x^{\deg r}d(x)}$ is the product of $\mathbb{Q}$-powers of f, $f^{xd(x)}$, $\ldots$, $f^{x^{\deg r-1}d(x)}$.

Now, consider an arbitrary $\mathbb{Q}$-linear combination $d'(x)$ of the elements $d(x)$, $xd(x)$, $\ldots$, $x^{\deg r-1}d(x)$. Following the method of Scott *et al.* [26], $d'(x)$ can be written in base $p(x)$ as $d'(x) = d'_0(x) + d'_1(x)p(x) + \cdots + d'_{\deg r-1}(x)p(x)^{\deg r-1}$, where each d'_i has degree less than the degree of p. Set $d'_i = d_{i,0} + xd_{i,1} + \cdots + x^{\deg p-1}d_{i,\deg p-1}$ and assume that $d_{i,j} \in \mathbb{Z}$ for all i, j. Then $f^{d'(x)}$ can be computed in two steps. First, the exponentiations f^x, $\ldots$, $f^{x^{\deg p-1}}$ are performed. From these intermediate exponentiations, terms of the form $f^{x^i p^j}$ can be easily calculated. Second, a vectorial addition chain containing the $d_{i,j}$-s is found. This allows to compute $f^{d'(x)}$ from terms of the form $f^{x^i p^j}$ using the work of Olivos [24]. The advantage of allowing multiples of $d(x)$ for this computation is to provide more flexibility in the choices of the exponents $d'(x) = \sum d_{i,j}x^i p^j$ with $d_{i,j} \in \mathbb{Z}$, that can potentially yield shorter addition chains, which in turn means a more-efficient final exponentiation calculation. However the savings are necessarily modest, since as in the method of Scott *et al.* [26], the main expense in this exponentiation process comes from computing the terms f^x, $\ldots$, $f^{x^{\deg p-1}}$.

In order to find efficient polynomials $d'(x)$, let us construct a rational matrix M' with dimensions $\deg p \times \varphi(k) \deg p$ such that

$$\begin{bmatrix} d(x) \\ xd(x) \\ \vdots \\ x^{\deg p-1}d(x) \end{bmatrix} = M' \left(\begin{bmatrix} 1 \\ p(x) \\ \vdots \\ p(x)^{\varphi(k)-1} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{\deg p-1} \end{bmatrix} \right).$$

Here $\otimes$ denotes the Kronecker product and the $(i, u + v \deg p)$-th entry of M' is $d_{u,v}$, where $x^{i-1}d(x) = \sum d_{u,v} x^{u-1} p^{v-1}$.

Elements in the rational lattice formed by the matrix M' correspond to $\mathbb{Q}$-linear combinations $d'(x)$ of $d(x)$, $xd(x)$, ..., $x^{\deg r - 1}d(x)$. Short vectorial addition chains can be produced from the elements of M' with small integer entries. The *LLL algorithm* of Lenstra, Lenstra, and Lovasz [19] produces an integer basis of an integer matrix with small coefficients. Let us consider the integer matrix M constructed from M' as the unique matrix whose rows are multiples of the rows of M' such that the entries of M are integers, and the greatest common divisor of the set of entries is 1. Next, the LLL algorithm is applied to M to obtain an integer basis for M with small enties. Finally, small integer linear combinations of these basis elements are examined with the hope of finding short addition chains. It is worth mentioning that even if these results do not yield an advantage over the results of Scott *et al.* [26], since the lattice contains an element corresponding to $d(x)$, the method described in this section includes the results of that work.

In the next section, the main mechanics of our method are explained by applying it to the computation of the final exponentiation step of several pairing-friendly families of curves.

4 Exponentiation Examples

4.1 BN curves

BN curves [4] have embedding degree 12 and are parameterized by x such that

$$\begin{aligned} r = r(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1 \\ p = p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \end{aligned}$$

are both prime.

The value $d = \Phi_k(p)/r = (p^4 - p^2 + 1)/r$ can be expressed as the polynomial

$$\begin{aligned} d = d(x) = {} & 46656x^{12} + 139968x^{11} + 241056x^{10} + 272160x^9 \\ & + 225504x^8 + 138672x^7 + 65448x^6 + 23112x^5 \\ & + 6264x^4 + 1188x^3 + 174x^2 + 6x + 1. \end{aligned}$$

At first glance, it appears that exponentiations by multiples of large powers of x are required. However, following the work of Scott *et al.* [26], d can be written in base p such that the degree of the coefficients is at most 3. In particular,

$$\begin{aligned} d(x) = {} & -36x^3 - 30x^2 - 18x - 2 \\ & + p(x)(-36x^3 - 18x^2 - 12x + 1) \\ & + p(x)^2(6x^2 + 1) \\ & + p(x)^3. \end{aligned}$$

Scott *et al.* [26] applied the work of Olivos [24] to compute the map $f \mapsto f^d$ using vectorial addition chains. From the above representation for d, vectorial addition chains can be used to compute $f \mapsto f^d$ using 3 exponentiations by x, 13 multiplications, and 4 squarings.

For the method described in Section 3, consider multiples of d represented in the base p with coefficients in $\mathbb{Q}[x]/(p(x))$.

A 4×16 integer matrix M is found such that

$$\begin{bmatrix} d(x) \\ xd(x) \\ 6x^2d(x) \\ 6x^3d(x) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(x) \\ p(x)^2 \\ p(x)^3 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \right).$$

The first row in M corresponds to the final exponentiation given by Scott *et al.* [26]. Any non-trivial integer linear combination of the rows corresponds to an exponentiation. For computational efficiency, a linear combination with coefficients as small as possible is desired.

None of the basis vectors returned by the LLL algorithm has an advantage over [26]. However, if small integer linear combinations of the short vectors returned by the LLL algorithm are considered, a multiple of d which corresponds to a shorter addition chain could potentially be found. A brute force search of linear combinations of the LLL basis yields 18 non-zero vectors with maximal entry 12. Among these vectors we consider the vector

$$(1, 6, 12, 12, 0, 4, 6, 12, 0, 6, 6, 12, -1, 4, 6, 12),$$

which corresponds to the multiple $d'(x) = \lambda_0 + \lambda_1 p + \lambda_2 p^2 + \lambda_3 p^3 = 2x(6x^2 + 3x + 1)d(x)$, where

$$\begin{aligned} \lambda_0(x) &= 1 + 6x + 12x^2 + 12x^3 \\ \lambda_1(x) &= 4x + 6x^2 + 12x^3 \\ \lambda_2(x) &= 6x + 6x^2 + 12x^3 \\ \lambda_3(x) &= -1 + 4x + 6x^2 + 12x^3. \end{aligned}$$

The final exponentiation which results can be computed more efficiently without using addition chains.

First, the following exponentiations are computed

$$f \mapsto f^x \mapsto f^{2x} \mapsto f^{4x} \mapsto f^{6x} \mapsto f^{6x^2} \mapsto f^{12x^2} \mapsto f^{12x^3}$$

which requires 3 exponentiations by x, 3 squarings, and 1 multiplication. The terms $a = f^{12x^3} \cdot f^{6x^2} \cdot f^{6x}$ and $b = a \cdot (f^{2x})^{-1}$ can be computed using 3 multiplications. Finally, the result $f^{d'}$ is obtained as

$$[a \cdot f^{6x^2} \cdot f] \cdot [b]^p \cdot [a]^{p^2} \cdot [b \cdot f^{-1}]^{p^3}$$

which requires 6 multiplications.

In total, our method requires 3 exponentiations by x, 3 squarings, and 10 multiplications. [3]

4.2 Freeman curves

Freeman curves [7] have embedding degree $k = 10$ and are parameterized by x as follows

$$r = r(x) = 25x^4 + 25x^3 + 15x^2 + 5x + 1$$
$$p = p(x) = 25x^4 + 25x^3 + 25x^2 + 10x + 3.$$

For $d = \Phi_{10}(p)/r = (p^4 - p^3 + p^2 - p + 1)/r$, let us consider a 4×16 integer matrix M such that

$$\begin{bmatrix} d(x) \\ xd(x) \\ 5x^2d(x) \\ 5x^3d(x) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(x) \\ p(x)^2 \\ p(x)^3 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \right).$$

In the lattice spanned by M, there exist two short vectors,

$$\pm(1, -2, 0, -5, -1, -4, -5, -5, 1, 3, 5, 5, 2, 5, 5, 5).$$

Both of these vectors have maximal coefficient 5. Consider the vector corresponding to the multiple

$$d'(x) = (5x^3 + 5x^2 + 3x + 1)d(x) = \lambda_0 + \lambda_1 p + \lambda_2 p^2 + \lambda_3 p^3,$$

where

$$\lambda_0(x) = 1 - 2x - 5x^3$$
$$\lambda_1(x) = -1 - 4x - 5x^2 - 5x^3$$
$$\lambda_2(x) = 1 + 3x + 5x^2 + 5x^3$$
$$\lambda_3(x) = 2 + 5x + 5x^2 + 5x^3,$$

Now, the map $f \mapsto f^{d'}$ can be computed as

$$f \mapsto f^x \mapsto f^{2x} \mapsto f^{4x} \mapsto f^{5x} \mapsto f^{5x^2} \mapsto f^{5x^3},$$

followed by

$$A = f^{5x^3} \cdot f^{2x}, \qquad B = A \cdot f^{5x^2},$$
$$C = f^{2x} \cdot f, \qquad D = B \cdot f^x \cdot f,$$

and finally

$$f^{d'} = [A^{-1} \cdot f] \cdot [B^{-1} \cdot C^{-1}]^p \cdot [D]^{p^2} \cdot [C \cdot D]^{p^3},$$

requiring a total of 12 multiplications, 2 squarings, and 3 exponentiations by x.

[3] We ignore the relatively inexpensive p-power Frobenius maps. Since the embedding degree k is even, we have that $f^{-1} = f^{p^{k/2}}$ for all f in the cyclotomic subgroup of $\mathbb{F}_{p^k}$. That is, inversion can be done using a p-power Frobenius. Hence, we ignore inversions as well.

4.3 KSS-8 curves

KSS-8 curves [16] have embedding degree $k = 8$ and are parameterized by x such that

$$r = r(x) = \frac{1}{450}(x^4 - 8x^2 + 25),$$
$$p = p(x) = \frac{1}{180}(x^6 + 2x^5 - 3x^4 + 8x^3 - 15x^2 - 82x + 125)$$

are both prime. For $d = \Phi_k(p)/r$, we compute an integer matrix M such that

$$\begin{bmatrix} 6d(x) \\ (6/5)xd(x) \\ (6/5)x^2d(x) \\ (6/5)x^3d(x) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(x) \\ p(x)^2 \\ p(x)^3 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \end{bmatrix} \right).$$

Note that since x needs to be chosen as a multiple of 5, the rows of M correspond to integer multiples of $d(x)$. We obtain a short vector corresponding to the multiple

$$d'(x) = \frac{6x}{5}d(x) = \lambda_0 + \lambda_1 p + \lambda_2 p^2 + \lambda_3 p^3$$

of $d(x)$, where

$$\lambda_0 = 2x^4 + 4x^3 + 5x^2 + 38x - 25$$
$$\lambda_1 = -x^5 - 2x^4 - x^3 - 16x^2 + 20x + 36$$
$$\lambda_2 = x^4 + 2x^3 - 5x^2 + 4x - 50$$
$$\lambda_4 = 3x^3 + 6x^2 + 15x + 72.$$

We use addition chains to compute $f^{d'}$. First, write $f^{d'}$ as

$$f^{d'} = y_0^1 y_1^2 y_2^3 y_3^4 y_4^5 y_5^6 y_6^{15} y_7^{16} y_8^{20} y_9^{25} y_{10}^{36} y_{11}^{38} y_{12}^{50} y_{13}^{72}$$

and compute the y_i's. The y_i's can be computed from f, f^x, ..., f^{x^5} using only multiplications and Frobenius maps.

Next, we find an addition chain containing all the powers of the y_i's. With the inclusion of the element 10, we obtain

$$\{1, 2, 3, 4, 5, 6, \underline{10}, 15, 16, 20, 25, 36, 38, 50, 72\}.$$

The work of Olivos gives an efficient method for producing a vectorial addition chain from an addition chain and states the computational expense of computing the final result $f^{d'}$ from the y_i's.

The computation of the y_i's requires 5 exponentiations by x, and 6 multiplications. The addition chain requires 7 multiplications and 7 squarings. The conversion to a vectorial addition chain requires 13 multiplications. In total, we require 5 exponentiations by x, 26 multiplications, and 7 squarings to compute the map $f \mapsto f^{d'}$.

4.4 KSS-18 curves

KSS-18 curves [16] have embedding degree $k = 18$ and a twist of order $d = 6$. These curves are parameterized by x such that

$$\begin{aligned} r = r(x) &= \frac{1}{343}(x^6 + 37x^3 + 343) \\ p = p(x) &= \frac{1}{21}(x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 \\ &\qquad + 259x^3 + 343x^2 + 1763x + 2401) \end{aligned}$$

are both prime. For $d = \Phi_k(p)/r$, we compute an integer matrix M such that

$$\begin{bmatrix} 3d(x) \\ (3/7)xd(x) \\ (3/49)x^2d(x) \\ (3/49)x^3d(x) \\ (3/49)x^4d(x) \\ (3/49)x^5d(x) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(x) \\ p(x)^2 \\ p(x)^3 \\ p(x)^4 \\ p(x)^5 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \\ x^7 \end{bmatrix} \right).$$

Since 7 divides x, the rows of M correspond to integer multiples of $d(x)$. We find a short vector corresponding to the multiple $d'(x) = \frac{3x^2}{49}d(x) = \lambda_0 + \lambda_1 p + \lambda_2 p^2 + \lambda_3 p^3 + \lambda_4 p^4 + \lambda_5 p^5$ of $d(x)$, where

$$\begin{aligned} \lambda_0 &= x^6 + 5x^5 + 7x^4 + 21x^3 + 108x^2 + 147x, \\ \lambda_1 &= -5x^5 - 25x^4 - 35x^3 - 98x^2 - 505x - 686, \\ \lambda_2 &= -x^7 - 5x^6 - 7x^5 - 19x^4 - 98x^3 - 133x^2 + 6, \\ \lambda_3 &= 2x^6 + 10x^5 + 14x^4 + 35x^3 + 181x^2 + 245x, \\ \lambda_4 &= -3x^5 - 15x^4 - 21x^3 - 49x^2 - 254x - 343, \\ \lambda_5 &= x^4 + 5x^3 + 7x^2 + 3. \end{aligned}$$

Proceeding as in the KSS-8 example, we construct an addition chain

$$\begin{aligned} \{&1, 2, 3, 5, 6, 7, 10, 14, 15, 19, 21, 25, 35, \underline{38}, 49, \underline{73}, \\ &98, 108, 133, 147, 181, 245, 254, 343, \underline{490}, 505, 686\}. \end{aligned}$$

Once again, applying Olivos' method for computing a short vectorial addition chain, we can compute the map $f \mapsto f^d$ using 7 exponentiations by x, 52 multiplications, and 8 squarings.

4.5 A comparison with Scott et al.

In Table 1, we compare our results against those given by Scott *et al.* [26]. Although operation counts are given for only the vectorial addition portion of

the exponentiation, the total cost can easily be computed from their work. The operation counts are given for field multiplications and squarings only, since the number of exponentiations by x is fixed for each curve and computing p-th powers maps is comparatively inexpensive.

Curve	Scott *et al.*	This work
BN	13M 4S	10M 3S
Freeman	14M 2S	12M 2S
KSS-8	31M 6S	26M 7S
KSS-18	62M 14S	52M 8S

Table 1. A comparison of our final exponentiation method with the method of Scott *et al.* [26]. 'M' denotes a multiplication and 'S' denotes a squaring. Both methods require the same number of exponentiations by x, determined by the curve.

For example, let us consider the case of BN curves parameterized with $x = -2^{62}-2^{54}+2^{44}$, which yields a 127-bit security level [5]. Further, assume that the relative cost of a field multiplication compared to a cyclotomic squaring on $\mathbb{F}_{p^k}$ is given as $M \approx 4.5S$ [1, 15]. Then, the total cost to perform the exponentiations $f^x, (f^x)^x, (f^{x^2})^x$, is of around $3 \cdot \lfloor \log_2 x \rfloor = 183$ cyclotomic squarings. Using the results reported in Table 1, this gives an approximate cost for the hard part of the final exponentiation of $187S + 13M \approx 245S$ for the method of Scott *et al.* and $186S + 10M \approx 231S$ using our method.

5 A Lattice-based Method for Hashing to $\mathbb{G}_2$

Let E be an elliptic curve over $\mathbb{F}_p$ with r, a large prime divisor of $n = \#E(\mathbb{F}_p)$, and let $k > 1$ be the embedding degree of E. Let q be an arbitrary power of p. An elliptic curve $\tilde{E}$ defined over $\mathbb{F}_q$ is said to be a *degree-d twist of E over $\mathbb{F}_q$*, if d is the smallest integer such that $\tilde{E}$ and E are isomorphic over $\mathbb{F}_{q^d}$. If $p \geq 5$, the only possible degrees of twists are those integers d which divide either 4 or 6. Since our examples deal only with curves where the degree of the twist divides the embedding degree k, we assume that d divides k and set $q = p^{k/d}$. However, with some modifications, the preliminary discussion and results apply to curves where d does not divide k.

Hess *et al.* [14] show that there exists a unique non-trivial twist $\tilde{E}$ of E over $\mathbb{F}_q$ such that r divides $\#\tilde{E}(\mathbb{F}_q)$. If $d = 2$, then $\#\tilde{E}(\mathbb{F}_q) = q + 1 + \hat{t}$, where $\hat{t}$ is the trace of the q-power Frobenius of E. In fact, the order of any twist can be found by first determining the trace $\hat{t}$ of the q-power Frobenius of E from the trace t of the p-power Frobenius of E via the Weil Theorem and then using a result given by Hess *et al.*[14].

The trace t_m of the p^m-power Frobenius of E for an arbitrary m can be determined using the recursion $t_0 = 2$, $t_1 = t$, and $t_{i+1} = t \cdot t_i - p \cdot t_{i-1}$ for

all $i > 1$ [20]. After computing the trace $\hat{t}$ of the q-power Frobenius of E, the possible values for the trace $\tilde{t}$ of the q-power Frobenius of $\tilde{E}$ over $\mathbb{F}_q$ can be determined using Table 2 [14], where D is the discriminant of E and $\hat{f}$ satisfies $\hat{t}^2 - 4q = D\hat{f}^2$.

d	2	3	4	6
$\tilde{t}$	$-\hat{t}$	$(\pm 3\hat{f} - \hat{t})/2$	$\pm\hat{f}$	$(\pm 3\hat{f} + \hat{t})/2$

Table 2. Possible values for the trace $\tilde{t}$ of the q-power Frobenius of a degree-d twist $\tilde{E}$ of E.

The group $\mathbb{G}_2$ can be represented as $\tilde{E}(\mathbb{F}_q)[r]$. In order to hash to $\mathbb{G}_2$, it suffices to hash to a random point $Q \in \tilde{E}(\mathbb{F}_q)$ followed by a multiplication by the cofactor $c = \#\tilde{E}(\mathbb{F}_q)/r$, to obtain the element $cQ \in \tilde{E}(\mathbb{F}_q)[r]$. Let $\phi\colon \tilde{E} \to E$ be an efficiently-computable isomorphism defined over $\mathbb{F}_{q^d}$ and let π be the p-th power Frobenius on E. Scott *et al.* [27] observed that the endomorphism $\psi = \phi^{-1} \circ \pi \circ \phi$ can be used to speed up the computation of $Q \mapsto cQ$. The endomorphism ψ satisfies

$$\psi^2 P - t\psi P + pP = \infty \tag{2}$$

for all $P \in \tilde{E}(\mathbb{F}_{p^k})$ [9, Theorem 1]. The cofactor c can be written as a polynomial in p with coefficients less than p. Scott *et al.* use this representation of c and reduce using (2) so that c is expressed as a polynomial in ψ with coefficients less than p. For parameterized curves, the speedup in the cost of computing $Q \mapsto cQ$ can become quite dramatic. For example, MNT curves have embedding degree $k = 6$ and are parameterized by x such that

$$\begin{aligned} p(x) &= x^2 + 1 \\ r(x) &= x^2 - x + 1 \end{aligned}$$

are both prime. It can be shown that

$$\begin{aligned} c(x)P &= (x^4 + x^3 + 3x^2)P = (p^2 + (x+1)p - x - 2)P \\ &= \psi(2xP) + \psi^2(2xP). \end{aligned}$$

It suffices to multiply by a multiple c' of c such that $c' \not\equiv 0 \pmod{r}$. Combining this observation with a new method of representing c in base ψ, we prove the following theorem.

Theorem 1. *Suppose that $\tilde{E}(\mathbb{F}_q)$ is cyclic and $p \equiv 1 \pmod{d}$. Then there exists a polynomial $h(z) = h_0 + h_1 z + \cdots + h_{\varphi(k)-1} z^{\varphi(k)-1} \in \mathbb{Z}[z]$ such that $h(\psi)P$ is a multiple of cP for all $P \in \tilde{E}(\mathbb{F}_q)$ and $|h_i|^{\varphi(k)} \leq \#\tilde{E}(\mathbb{F}_q)/r$ for all i.*

The proof of Theorem 1 is divided into two parts. We first prove a technical lemma and then show how the polynomial h can be obtained using an integer-lattice technique. Let f, $\tilde{f}$ be such that $t^2 - 4p = Df^2$ and $\tilde{t}^2 - 4q = D\tilde{f}^2$, where D is the discriminant. It also holds that $n + t = p + 1$ and $\tilde{n} + \tilde{t} = q + 1$, where $\tilde{n} = \#\tilde{E}(\mathbb{F}_q)$.

Recall that the endomorphism $\psi\colon \tilde{E} \to \tilde{E}$ is defined over $\mathbb{F}_{q^d}$. In the following lemma, it is proved that ψ fixes $\tilde{E}(\mathbb{F}_q)$ as a set.

Lemma 1. *If $p \equiv 1 \pmod{d}$, then $\psi P \in \tilde{E}(\mathbb{F}_q)$ for all $P \in \tilde{E}(\mathbb{F}_q)$.*

Proof. From the work of Hess *et al.* we have that the twist is defined by first selecting $\gamma \in \mathbb{F}_{q^d}$ such that $\gamma^d \in \mathbb{F}_q$. The map ϕ is then defined by $\phi(x, y) = (\gamma^2 x, \gamma^3 y)$ and hence ψ is defined by $\psi(x, y) = (\gamma^{2(p-1)} x^p, \gamma^{3(p-1)} y^p)$. Now, $\gamma^d \in \mathbb{F}_q$ and $p - 1 \equiv 0 \pmod{d}$ yield $\gamma^{p-1} \in \mathbb{F}_q$, which in turn implies that $\psi(x, y) \in \tilde{E}(\mathbb{F}_q)$ for $(x, y) \in \tilde{E}(\mathbb{F}_q)$. □

The following lemma illustrates the effect of ψ on elements in $\tilde{E}(\mathbb{F}_q)$.

Lemma 2. *If $p \equiv 1 \pmod{d}$, $\gcd(\tilde{f}, \tilde{n}) = 1$, and $\tilde{E}(\mathbb{F}_q)$ is a cyclic group, then $\psi Q = aQ$ for all $Q \in \tilde{E}(\mathbb{F}_q)$, where a is one of $(t+f(\tilde{t}-2)/\tilde{f})/2$, $(t-f(\tilde{t}-2)/\tilde{f})/2$.*

Proof. Since $\tilde{E}(\mathbb{F}_q)$ is cyclic and ψ fixes $\tilde{E}(\mathbb{F}_q)$, there exists an integer a such that $\psi Q = aQ$ for all $Q \in \tilde{E}(\mathbb{F}_q)$. By solving for a in (2) and using the fact that $t^2 - 4p = Df^2$, we obtain

$$a \equiv \frac{1}{2}(t \pm \sqrt{t^2 - 4p}) \equiv \frac{1}{2}(t \pm \sqrt{Df^2}) \equiv \frac{1}{2}(t \pm f\sqrt{D}) \pmod{\tilde{n}}.$$

Working modulo $\tilde{n}$, we observe that $D\tilde{f}^2 = \tilde{t}^2 - 4q = \tilde{t}^2 - 4\tilde{t} + 4 = (\tilde{t} - 2)^2$ and so $\sqrt{D} \equiv \pm(\tilde{t} - 2)/\tilde{f} \pmod{\tilde{n}}$. Without loss of generality, let f, $\tilde{f}$ be such that $a = \frac{1}{2}(t + f\sqrt{D})$ and $\sqrt{D} \equiv (\tilde{t} - 2)/\tilde{f} \pmod{\tilde{n}}$. Then, since $Q \in \tilde{E}(\mathbb{F}_q)$ has order dividing $\tilde{n}$, it follows that

$$\psi Q = aQ = \left(\frac{1}{2}(t + f\sqrt{D})\right) Q = \left(\frac{1}{2}(t + f(\tilde{t} - 2)/\tilde{f})\right) Q.$$

□

In the space of polynomials $h \in \mathbb{Q}[z]$ such that $h(a) \equiv 0 \pmod{c}$, we wish to find an h with small integer coefficients. Ignoring the small coefficient requirement for the moment, $h(z) = c$ and $h(z) = z^i - a^i$ satisfy the required condition for all integers i. Furthermore, any linear combination of these polynomials satisfies this condition.

Since π acting on $E(\mathbb{F}_{p^k})$ has order k and ψ is an automorphism when restricted to the cyclic group $\tilde{E}(\mathbb{F}_q)$, the automorphism ψ acting on $\tilde{E}(\mathbb{F}_q)$ has order k. Hence, the integer a satisfies $\Phi_k(a) \equiv 0 \pmod{\tilde{n}}$. Therefore, the polynomial $h(z) = z^i - a^i$ with $i \geq \varphi(k)$ can be written as a linear combination

(modulo c) of $z - a, \ldots, z^{\varphi(k)-1} - a^{\varphi(k)-1}$. For this reason, the polynomials of higher degree are excluded in the following construction.

Notice that polynomials $h \in \mathbb{Z}[z]$ such that $h(a) \equiv 0 \pmod{c}$ correspond to points in the integer lattice generated by the matrix

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

where $\mathbf{a}$ is the column vector with i-th entry $-a^i$. Consider the convex set $C \subseteq \mathbb{R}^{\varphi(k)}$ generated by all vectors of the form $(\pm|c|^{1/\varphi(k)}, \ldots, \pm|c|^{1/\varphi(k)})$. The volume of C is $2^{\varphi(k)}|c|$ and the lattice above has volume $|c|$. By Minkowski's Theorem [22], the region C contains a lattice point. Hence, there exists a non-zero polynomial h with coefficients at most $|c|^{1/\varphi(k)}$ such that $h(a) \equiv 0 \pmod{c}$. This concludes the proof of Theorem 1. □

6 Hashing Examples

6.1 BN curves

BN curves are parameterized by

$$\begin{aligned} p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \\ r(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1 \\ t(x) &= 6x^2 + 1 \\ f(x) &= 6x^2 + 4x + 1 \end{aligned}$$

where

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

After computing the trace $\hat{t}$ of the q-power Frobenious of E, we compute $\hat{f}$ such that $4q - \hat{t} = -3\hat{f}^2$. Using $\hat{t}$ and $\hat{f}$, we find the twist $\tilde{E}(\mathbb{F}_q)$ is parameterized by

$$\begin{aligned} \tilde{n}(x) &= q(x) + 1 - (3\hat{f}(x) + \hat{t}(x))/2 \\ &= (36x^4 + 36x^3 + 18x^2 + 6x + 1)(36x^4 + 36x^3 + 30x^2 + 6x + 1) \\ &= 36x^4 + 1 \end{aligned}$$

We have that $c(x) = p(x) + t(x) - 1$ is such that $\tilde{n}(x) = r(x)c(x)$. Using Lemma 2, we obtain

$$\begin{aligned} a(x) &= \frac{1}{2}(t + f(\tilde{t} - 2)/\tilde{f}) \\ &= -\frac{1}{5}(3456x^7 + 6696x^6 + 7488x^5 + 4932x^4 + 2112x^3 + 588x^2 + 106x + 6). \end{aligned}$$

As a sobriety check, note that $a(x) \equiv p(x) \pmod{r}$ and thus $\psi Q = a(x)Q = p(x)Q$ for all $Q \in \tilde{E}(\mathbb{F}_q)[r]$.

We construct the following lattice and reduce the $-a(x)^i$ entries modulo $c(x)$:

$$\left[\begin{array}{c|c} c(x) & 0\ 0\ 0 \\ -a(x) & 1\ 0\ 0 \\ -a(x)^2 & 0\ 1\ 0 \\ -a(x)^3 & 0\ 0\ 1 \end{array}\right] \rightarrow \left[\begin{array}{c|c} 36x^4+36x^3+30x^2+6x+1 & 0\ 0\ 0 \\ 48/5x^3+6x^2+4x-2/5 & 1\ 0\ 0 \\ 36/5x^3+6x^2+6x+1/5 & 0\ 1\ 0 \\ 12x^3+12x^2+8x+1 & 0\ 0\ 1 \end{array}\right].$$

From this lattice, we find the polynomial $h(z) = x + 3xz + xz^2 + z^3$. Working modulo $\tilde{n}(x)$, we have that

$$h(a) = -(18x^3 + 12x^2 + 3x + 1)c(x)$$

and since $\gcd(18x^3 + 12x^2 + 3x + 1, r(x)) = 1$, the following map is a homomorphism of $\tilde{E}(\mathbb{F}_q)$ with image $\tilde{E}(\mathbb{F}_q)[r]$:

$$Q \mapsto xQ + \psi(3xQ) + \psi^2(xQ) + \psi^3(Q).$$

We can compute $Q \mapsto xQ \mapsto 2xQ \mapsto 3xQ$ using one doubling, one addition, and one multiply-by-x. Given Q, xQ, $3xQ$, we can compute $h(a)Q$ using three ψ-maps, and three additions. In total, we require one doubling, four additions, one multiply-by-x, and three ψ-maps. As seen in Table 3 on page 17, the previous fastest-known method of computing such a homomorphism costs two doublings, four additions, two multiply-by-x's, and three ψ-maps.

6.2 Freeman curves

Freeman curves [7] have embedding degree $k = 10$ and are parameterized by x as follows

$$\begin{aligned} r = r(x) &= 25x^4 + 25x^3 + 15x^2 + 5x + 1 \\ p = p(x) &= 25x^4 + 25x^3 + 25x^2 + 10x + 3. \end{aligned}$$

Since Freeman curves do not have a fixed discriminant, the algorithm given in the proof of Lemma 2 does not directly apply. However, we are able to apply the techniques of Scott *et al.* on $c(x)$, $xc(x)$, $x^2c(x)$, $x^3c(x)$ and then use our method from Section 3.

We find a short vector corresponding to the multiple $h(a) = \lambda_0 + \lambda_1 a + \lambda_2 a^2 + \lambda_3 a^3$ of c, where $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4)$ is such that

$$\begin{aligned} \lambda_0(x) &= 10x^3 + 5x^2 + 4x + 1 \\ \lambda_1(x) &= -3x \\ \lambda_2(x) &= -10x^3 - 10x^2 - 8x - 3 \\ \lambda_3(x) &= -5x^3 - 5x^2 - x \\ \lambda_4(x) &= -5x^3 + 2. \end{aligned}$$

Using the addition chain $\{1, 2, 3, 4, 5, 8, 10\}$, we can compute $h(a)Q$ using fourteen additions, four doublings, three multiply-by-x's, and four ψ maps.

6.3 KSS-8

KSS-8 curves [16] have embedding degree $k = 8$ and are parameterized by x such that

$$r = r(x) = \frac{1}{450}(x^4 - 8x^2 + 25)$$
$$p = p(x) = \frac{1}{180}(x^6 + 2x^5 - 3x^4 + 8x^3 - 15x^2 - 82x + 125)$$

are both prime. Set $q = p^{k/d} = p^2$. There exists a degree-4 twist $\tilde{E}(\mathbb{F}_q)$ of order

$$\tilde{n}(x) = \frac{1}{72}(x^8 + 4x^7 + 6x^6 + 36x^5 + 34x^4 - 84x^3 + 486x^2 + 620x + 193)r(x).$$

Set $c(x) = \tilde{n}(x)/r(x)$. After some work, we discover that ψ is such that $\psi Q = aQ$ for all $Q \in \tilde{E}(\mathbb{F}_q)$ where

$$\begin{aligned} a = \frac{1}{184258800}\big(& -52523x^{11} - 174115x^{10} + 267585x^9 - 193271x^8 \\ & - 325290x^7 + 15093190x^6 - 29000446x^5 - 108207518x^4 \\ & + 235138881x^3 + 284917001x^2 - 811361295x - 362511175\big). \end{aligned}$$

As we've done previously, we find a short basis for the lattice generated by the matrix

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

and discover a short vector corresponding to the multiple

$$h(a) = \frac{1}{75}(x^2 - 25)c(x) = \lambda_0 + \lambda_1 a + \lambda_2 a^2 + \lambda_3 a^3$$

of c such that $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3) = (-x^2 - x, x - 3, 2x + 6, -2x - 4)$.

For an element $Q \in \tilde{E}(\mathbb{F}_q)$, we can compute $h(a)Q$ with the following sequence of calculations. We compute $Q \mapsto xQ \mapsto (x+1)Q \mapsto (x^2+x)Q$ and $Q \mapsto 2Q \mapsto 4Q$ which requires one addition, two doublings, and two multiply-by-x's. Then we compute

$$\begin{aligned} \lambda_0 Q &= -(x^2 + x)Q \\ \lambda_1 Q &= (x+1)Q - 4Q \\ \lambda_2 Q &= 2(x+1)Q + 4Q \\ \lambda_3 Q &= -2(x+1)Q - 2Q \end{aligned}$$

which requires three more additions and another doubling. Finally, we compute

$$h(a)Q = \lambda_0 Q + \psi(\lambda_1 Q) + \psi^2(\lambda_2 Q) + \psi^3(\lambda_3 Q)$$

which requires three more additions and three ψ maps.

In total, we require seven additions, three doublings, two multiply-by-x's, and three ψ maps to compute $Q \mapsto h(a)Q$.

6.4 KSS-18

KSS-18 curves [16] have embedding degree $k = 18$ and a twist of order $d = 6$. These curves are parameterized by x such that

$$\begin{aligned} r = r(x) &= \frac{1}{343}(x^6 + 37x^3 + 343) \\ p = p(x) &= \frac{1}{21}(x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 \\ &\qquad + 259x^3 + 343x^2 + 1763x + 2401) \end{aligned}$$

are both prime. We find that

$$\begin{aligned} c(x) = \frac{1}{27}\big(&x^{18} + 15x^{17} + 96x^{16} + 409x^{15} + 1791x^{14} + 7929x^{13} + 27539x^{12} \\ &+ 81660x^{11} + 256908x^{10} + 757927x^9 + 1803684x^8 \\ &+ 4055484x^7 + 9658007x^6 + 19465362x^5 + 30860595x^4 \\ &+ 50075833x^3 + 82554234x^2 + 88845918x + 40301641\big). \end{aligned}$$

Constructing our lattice, we obtain the vector corresponding to the multiple

$$h(a) = -\frac{3}{343}x(8x^3 + 147)c(x) = \lambda_0 + \lambda_1 a + \lambda_2 a^2 + \lambda_3 a^3 + \lambda_2 a^4 + \lambda_3 a^5$$

of $c(x)$, where

$$\begin{aligned} \lambda_0 &= 5x + 18 \\ \lambda_1 &= x^3 + 3x^2 + 1 \\ \lambda_2 &= -3x^2 - 8x \\ \lambda_3 &= 3x + 1 \\ \lambda_4 &= -x^2 - 2 \\ \lambda_5 &= x^2 + 5x. \end{aligned}$$

We construct the addition chain $\{1, 2, 3, 5, 8, \underline{10}, 18\}$, from which we can compute $Q \mapsto h(a)Q$ using sixteen additions, two doublings, three multiply-by-x's, and five ψ maps.

6.5 Comparison with previous work

In Table 3, we compare our results to the work of Scott *et al.* [27, 28]. In the proceedings version [27] of their work, the authors assume that the identity $\Phi_k(\psi)P = \infty$ holds for all points P in $\tilde{E}(\mathbb{F}_q)$. However, there exist concrete examples showing that this identity does *not* hold for some curves. In particular, MNT and Freeman curves do not satisfy this identity in general. On the other hand, the identity $\psi^{k/2}P = -P$ is critically used in the eprint version [28] of their work. Fortunately, all curves except the MNT curve can be explicitly shown

Curve	Scott *et al.*	This work
BN	4A 2D 2X 3ψ	4A 1D 1X 3ψ
Freeman	20A 5D 3X 4ψ	14A 4D 3X 4ψ
KSS-8	22A 5D 5X 2ψ	7A 3D 2X 3ψ
KSS-18	59A 5D 7X 4ψ	16A 2D 3X 5ψ

Table 3. A comparison of our hashing algorithm with the hashing algorithm of Scott *et al.* 'A' denotes a point addition, 'D' denotes a point doubling, 'X' denotes a multiplication by x, and 'ψ' denotes an application of the map ψ.

to satisfy the identity $\psi^{k/2}P = -P$. In practice, we've found that MNT curves also satisfy this property. More work needs to be done to determine the structure of the twist and the action of ψ on various subgroups of the twist.

We use the eprint version [28] to represent Scott *et al.*'s operation counts on Freeman curves. We have verified that the identity $\Phi_k(\psi)P = \infty$ holds for BN, KSS-8, and KSS-18 curves and use the counts from the proceedings version [27] of their work for those curves in Table 3. Since the multiplications by x dominate the other operations, it can be seen that our hash algorithm is approximately twice as fast as that of Scott *et al.* for BN curves. For the KSS-8 curve we see a $\frac{5}{2}$-fold improvement, and for the KSS-18 curves, we see a $\frac{7}{3}$-fold improvement.

Acknowledgments

The authors would like to express their deepest thanks to Professor Alfred Menezes for valuable discussions and constructive criticism related to this work and for the careful proof-reading of the technical sections of this paper.

References

1. D.F. Aranha, K. Karabina, P. Longa, C.H. Gebotys, and J. López, *Faster Explicit Formulas for Computing Pairings over Ordinary Curves*, EUROCRYPT 2011, LNCS 6632 (2011) 48–68.
2. P.S.L.M. Barreto, S. Galbraith, C. Ó hÉigeartaigh, and M. Scott, *Efficient pairing computation on supersingular Abelian varieties. Designs, Codes and Cryptography,* Designs, Codes and Cryptography Volume 42, Number 3, (2007) 239–271.
3. P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott, *Efficient Algorithms for Pairing-Based Cryptosystems*, CRYPTO 2002, LNCS 2442 (2002) 354–368.
4. P. S. L. M. Barreto and M. Naehrig, *Pairing-Friendly Elliptic Curves of Prime Order,* SAC 2005, LNCS 3897 (2006) 319–331.
5. J.-L. Beuchat, J.E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T Teruya, *High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves*, Pairing 2010, LNCS 6487 (2010) 21–39.
6. A. J. Devegili, M. Scott, and R. Dahab, *Implementing Cryptographic Pairings over Barreto-Naehrig Curves,* Pairing 2007, LNCS 4575 (2007) 197–207.
7. D. Freeman, *Constructing Pairing-Friendly Elliptic Curves with Embedding Degree 10,* ANTS VII, LNCS 4076 (2006) 452–465.

8. D. Freeman, M. Scott, and E. Teske, *A Taxonomy of Pairing-Friendly Elliptic Curves*, Journal of Cryptology, Volume 23, Number 2 (2010) 224-280.
9. S.D. Galbraith, X. Lin, and M. Scott, *Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves*, EUROCRYPT 2009, LNCS 5479 (2009) 518–535.
10. S.D. Galbraith and M. Scott, *Exponentiation in pairing-friendly groups using homomorphisms*, Pairing 2008, LNCS 5671 (2008) 78–88.
11. R. Granger and M. Scott, *Faster Squaring in the Cyclotomic Subgroup of Sixth Degree Extensions*, PKC 2010, LNCS 6056 (2010) 209–223.
12. D. Hankerson, A. Menezes, and M. Scott, *Software Implementation of Pairings*, Chapter 12 of Identity-Based Cryptography (2009) 188–206.
13. F. Hess, *Pairing Lattices,* Pairing 2008, LNCS 5209 (2008) 18–38.
14. F. Hess and N. Smart, and F. Vercauteren, *The Eta Pairing Revisited,* IEEE Transactions on Information Theory, Volume 52, Issue 10 (2006) 4595–4602.
15. K. Karabina, *Squaring in Cyclotomic Subgroups*, manuscript (2010). `http://eprint.iacr.org/2010/542`
16. E. Kaschisa, E. Schaefer, and M. Scott, *Constructing Brezing-Weng Pairing-Friendly Elliptic Curves using Elements in the Cyclotomic Field,* Pairing 2008, LNCS 5209 (2008) 126–135.
17. N. Koblitz and A. Menezes, *Pairing-Based Cryptography at High Security Levels*, Cryptography and Coding, 10th IMA International Conference, LNCS 3796 (2005) 13–36.
18. E. Lee, H-S. Lee, and C-M. Park, *Efficient and Generalized Pairing Computation on Abelian Varieties,* IEEE Transactions on Information Theory, Volume 55, Issue 4 (2009) 1793–1803.
19. A. K. Lenstra, H. W. Lenstra, Jr, and L. Lovasz, *Factoring Polynomials with Rational Coefficients,* Mathematische Annalen, Volume 261, Number 4 (1982) 515–534.
20. A. Menezes, *Elliptic Curve Public Key Cryptosystems,* Kluwer Academic Publishers (1993).
21. V.S. Miller, *The Weil Pairing, and Its Efficient Calculation,* Journal of Cryptology, Volume 17, Number 4 (2004) 235–261.
22. H. Minkowski, *Geometrie der Zahlen,* Leipzig und Berlin, Druck ung Verlag von B.G. Teubner (1910).
23. A. Miyaji, M. Nakabayashi, and S. Takano, *New Explicit Conditions of Elliptic-Curve Traces for FR-reduction*, IEICE Trans. Fundamentals E84 (2001) 1234–1243.
24. J. Olivos, *On Vectorial Addition Chains,* Journal of Algorithms, Volume 2, Issue 1 (1981) 13–21.
25. G.C.C.F. Pereira, M. A. Simplcio Jr, M. Naehrig, and P. S. L. M. Barreto, *A Family of Implementation-Friendly BN Elliptic Curves,* Journal of Systems and Software (2011) to appear.
26. M. Scott, N. Benger, M. Charlemagne, L. J. Dominguez Perez, and E. J. Kachisa, *On the Final Exponentiation for Calculating Pairings on Ordinary Elliptic Curves,* Pairing 2009, LNCS 5671 (2009) 78–88.
27. M. Scott, N. Benger, M. Charlemagne, L. J. Dominguez Perez, and E. J. Kachisa, *Fast Hashing to $\mathbb{G}_2$ on Pairing-Friendly Curves,* Pairing 2009, LNCS 5671, (2009) 102–113.
28. M. Scott, N. Benger, M. Charlemagne, L. J. Dominguez P., and E. J. Kachisa, *Fast Hashing to $\mathbb{G}_2$ on Pairing-Friendly Curves,* `http://eprint.iacr.org/2008/530`
29. F. Vercauteren, *Optimal Pairings,* IEEE Transactions on Information Theory, Volume 56, Issue 1 (2010) 455–461.