Fast hashing to G2 on pairing friendly curves

Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa

School of Computing
Dublin City University
Ballymun, Dublin 9, Ireland.
mike@computing.dcu.ie *

Abstract. When using pairing-friendly ordinary elliptic curves over prime fields to implement identity-based protocols, there is often a need to hash identities to points on one or both of the two elliptic curve groups of prime order r involved in the pairing. Of these G_1 is a group of points on the base field $E(\mathbb{F}_p)$ and G_2 is instantiated as a group of points with coordinates on some extension field, over a twisted curve $E'(\mathbb{F}_{p^d})$, where d divides the embedding degree k. While hashing to G_1 is relatively easy, hashing to G_2 has been less considered, and is regarded as likely to be more expensive as it appears to require a multiplication by a large cofactor. In this paper we introduce a fast method for this cofactor multiplication on G_2 which exploits an efficiently computable homomorphism.
Keywords: Tate pairing, addition chains

1 Introduction

The Tate pairing (and its derivatives) on ordinary elliptic curves $e(P, Q)$ takes as parameters two linearly independent points P and Q. For maximum efficiency P and Q are drawn from the groups G_1 of points on $E(\mathbb{F}_p)$ and G_2, a group of points on the twisted curve $E'(\mathbb{F}_{p^d})$ where d divides the embedding degree k. For the Tate pairing the first parameter P is chosen from G_1 and the second Q from G_2. However recent discoveries of the faster ate [9] and R-ate [11] pairings require P to be chosen from G_2 and Q from G_1. In either case P must be of prime order r, where k, the embedding degree, is the smallest integer for which $r|\Phi_k(t-1)$ [2], where $\Phi_k(.)$ is the k-th cyclotomic polynomial and t is the trace of the Frobenius of the elliptic curve. The second parameter Q need not strictly be of order r, as for these pairings it is sufficient for Q to be a coset representative.

The degree of the extension field d is a divisor of k, and can always be $k/2$ if k is even. In fact we prefer k to be even as it enables the important denominator elimination optimization in the pairing calculation [2]. Furthermore if the elliptic curve has a Complex Multiplication (CM) discriminant of -3, and $6|k$, then we can choose $d = k/6$. Similarly if the curve has a CM discrimant of -4, and $4|k$,

* Research supported by the Claude Shannon Institute, Science Foundation Ireland Grant 06/MI/006

then we can choose $d = k/4$. Clearly the smaller the degree of the extension, the easier it will be to manipulate points on G_2.

Pairing friendly ordinary elliptic curves can be constructed to have any arbitrary embedding degree. This compares favourably with the case of supersingular curves, for which elliptic curves have a maximum embedding degree of 6. However on a supersingular curve we have a distortion map, which in effect means that both arguments to a modified pairing can use linearly dependant points from G_1, where here G_1 represents a group of points over the base field. In contrast on ordinary elliptic curves we must be prepared to handle points over the potentially more awkward G_2 group. However in a recent paper Galbraith and Scott [7] observe that point multiplication of points of order r on G_2 is not as difficult as might be thought, as a useful homomorphism can be exploited.

Here we extend the ideas of [7] to the related problem of cofactor multiplication on G_2, as required to hash an identity to a point of order r on G_2.

2 Elliptic curves over extension fields

Consider an elliptic curve defined over $\mathbb{F}_p$. As is well known the number of points on the curve is defined as $\#E(\mathbb{F}_p) = p+1-t$, where t is the trace of the Frobenius, which obeys the Hasse bound $t \leq 2\sqrt{p}$. Consider now points whose coordinates are defined over an extension field $\mathbb{F}_{p^m}$, and the number of such points on the same elliptic curve [12]. It is well known for example, that

$$\#E(\mathbb{F}_{p^2}) = p^2 + 1 - (t^2 - 2p)$$
$$\#E(\mathbb{F}_{p^3}) = p^3 + 1 - (t^3 - 3tp)$$

In the general case the number of points can be calculated by this simple algorithm [12]

Algorithm 1 Returns $\#E(\mathbb{F}_{p^m})$

INPUT: m, p, t
1: $\tau_0 \leftarrow 2$
2: $\tau_1 \leftarrow t$
3: **for** $i \leftarrow 1$ **to** $m - 1$ **do**
4: $\quad \tau_{i+1} \leftarrow t.\tau_i - p.\tau_{i-1}$
5: **end for**
6: $q \leftarrow p^m$
7: $\tau \leftarrow \tau_m$
8: **return** $q + 1 - \tau$

To represent the group G_2 we like to use an isomorphic group on a twisted curve over the smallest possible extension field. The number of points on the twisted curve can also easily be determined from the output of this algorithm.

For example if we use the quadratic twist, then the number of points $\#E'(\mathbb{F}_{p^{k/2}})$ is simply $q+1+\tau$. For formulae for the higher order twists we refer the reader to [9]. Where the quartic twist applies, $\#E'(\mathbb{F}_{p^{k/4}}) = q+1-f$, where $f = \sqrt{4q-\tau^2}$. Where the sextic twist applies $\#E'(\mathbb{F}_{p^{k/6}}) = q+1-(3f+\tau)/2$, where $f = \sqrt{(4q-\tau^2)/3}$.

To hash to a point of order r on G_2, the standard approach would be to first hash to a general point on G_2 and then multiply by the cofactor $c = \#E/r$. Consider now a pairing friendly curve with $k = 10$. In this case using the quadratic twist this cofactor c would be of a size in bits approximately the same as p^4. This would be prohibitively slow. However as we will see, the same outcome can be achieved in all cases with the equivalent work of a multiplication by a value less than p, and in some cases much less than p.

3 A fast cofactor multiplication algorithm for G_2

The issue of fast cofactor multiplication on the group G_2 was briefly consider for the case of BN curves by Galbraith and Scott [7], section 8. Here we generalise and extend that idea. In that paper the authors introduce the homomorphism $\psi^i = \phi^{-1}\pi_p^i\phi$, where $\phi : E' \rightarrow E$ is the isomorphism which takes us from the twisted curve $E'(\mathbb{F}_{p^d})$ to the isomorphic group on $E(\mathbb{F}_{p^k})$ as actually required by the pairing algorithm, and π_p is the p-power Frobenius map on E. Note that $\psi(P)$ can be calculated very quickly.

General points on G_2 obey this identity [7]

$$\psi^2(P) - [t]\psi(P) + [p]P = 0$$

Our main idea is to first express the cofactor c to the base p as

$$c = c_0 + c_1.p + c_2.p^2 ...$$

and then use the identity

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

repeatedly if necessary to reduce the co-factor multiplication to a form

$$[c_0 + c_1.p + c_2.p^2 + ...]P = [g_0]P + [g_1]\psi(P) + [g_2]\psi^2(P) + ...$$

where all of the g_i are less than p.

Observe that for example $[c_1.p]P = [c_1.t]\psi(P) - [c_1]\psi^2(P)$, and that $c_1.t$ may be of a size in bits 50% larger than p (recall that t is roughly half the size in bits as p as a consequence of the Hasse condition). Therefore further applications of the homomorphism may be necessary to effect a complete reduction.

In some circumstances we will also find the following identity to be useful

$$\psi^{k/2}(P) = -P \tag{2}$$

as it allows higher order terms to be removed from the calculation.

4 The application to ordinary pairing friendly elliptic curves

The most general method to create a pairing friendly elliptic curve is to use the method of Cocks-Pinch [4]. However these curves suffer from a ρ ratio that is close to 2, where $\rho = \lg(p)/\lg(r)$. Also they cannot exploit higher order twists on low CM discriminant elliptic curves. Therefore it is usually preferred to choose instead from one of the families of pairing-friendly curves identified by numerous authors, and collated together in the taxonomy paper of Freeman et al. [6]. These often have a ρ value closer to 1, and many are of the desirable low CM discriminant form. Also these families share another feature – the prime modulus p, the group r and the trace t are all described as rather simple polynomials. It is our aim to exploit this simple form in a systematic way to further speed up the cofactor multiplication required for hashing to G_2.

Before proceeding we need to formally describe the method of the previous section as an algorithm for reducing the co-factor multiplication to the evaluation of a polynomial of the powers $\psi^i(P)$, with coefficients less than p. When p is itself expressed as a polynomial $p(x)$, these coefficients can in turn be calculated as polynomials in x, and this we choose to do as it leads to further optimizations. Also in these cases the cofactor c itself can be calculated and presented as a polynomial in x. However we emphasise that the basic idea (with minor modifications) applies equally to non-parameterised Cocks-Pinch curves. See algorithm 2.

We now proceed to use this algorithm to find the quickest way to perform the co-factor multiplication required to hash to a point of order r on G_2. We proceed on a case-by-case basis for certain selected popular families of pairing friendly elliptic curves.

5 The MNT curves

The MNT pairing friendly elliptic curves were first reported by Miyaji et al. [13]. For the $k = 6$ case the prime p and the group order r parameters are expressed as

$$p(x) = x^2 + 1$$
$$r(x) = x^2 - x + 1$$

In this case $\rho = 1$, but although $k = 6$, no solution exists with a CM discriminant of -3, and so the best that can be done for G_2 is to represent it as a group of points on $E'(\mathbb{F}_{p^3})$. The cofactor is $c(x) = (p(x)^3 + 1 + t(x)^3 - 3t(x)p(x))/r(x)$, which in this case works out as

$$c(x) = x^4 + x^3 + 3x^2$$

Algorithm 2 Reduction of the co-factor $c(x)$

INPUT: $d, k, p(x), t(x)$, and $c(x)$
OUTPUT: $g_0(x), g_1(x)$.....

for $i \leftarrow 0$ **to** $d-2$ **do**
    $c_i(x) \leftarrow c(x) \bmod p(x)$
    $c(x) \leftarrow c(x)/p(x)$
end for
$g_0(x) \leftarrow c(x)$
for $i \leftarrow 0$ **to** $d-2$ **do**
    for $j \leftarrow 2i$ **downto** 0 **do**
        $g_{j+2}(x) \leftarrow g_{j+2}(x) - g_j(x)$
        $g_{j+1}(x) \leftarrow t(x)g_j(x)$
        $g_j(x) \leftarrow 0$
    end for
    $g_0(x) \leftarrow c_{d-i-2}(x)$
end for
for $j \leftarrow 1$ **to** $2d-2$ **do**
    $w(x) \leftarrow g_j(x)/p(x)$
    $g_j(x) \leftarrow g_j(x) \bmod p(x)$
    $g_{j+1}(x) \leftarrow g_{j+1}(x) + t(x)w(x)$
    $g_{j+2}(x) \leftarrow g_{j+2}(x) - w(x)$
end for
for $j \leftarrow k/2$ **to** $2d-2$ **do**
    $g_{j-k/2}(x) \leftarrow g_{j-k/2}(x) - g_j(x)$
    $g_j(x) \leftarrow 0$
end for

Applying algorithm 2 we first represent $c(x)$ to the base $p(x)$

$$c(x) = p^2(x) + (x+1).p(x) + (-x-2)$$

Now apply equation (1) to each term involving a power of $p(x)$, and use it to calculate $[c(x)].P$

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

As can be seen some of the coefficients are still of the same degree as $p(x)$, so apply equation (1) again to get

$$[-x-2]P + [2x]\psi(P) + [2x]\psi^2(P) + [-x-2]\psi^3(P)$$

Finally applying equation (2) we find that multiplication of a general point P by $c(x)$ can be expressed as

$$[2x]\psi(P) + [2x]\psi^2(P) = \psi(2xP) + \psi^2(2xP)$$

which can be calculated by only one multiplication by x, a point doubling, two applications of the homomorphism and a further point addition. The savings compared with a direct multiplication of P by $c(x)$ are obvious.

6 The BN curves

The BN family of pairing friendly curves [3] has an embedding degree of 12, and is parameterised as follows

$$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$$
$$r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$$

In this case the co-factor multiplication can be effected as [7]

$$\psi(6x^2P) + 6x^2P + \psi(P) - \psi^2(P)$$

The major work here is the point multiplication by $6x^2$. Since BN curves are plentiful it is not hard to find a value of x with a very low Hamming weight, and this will further speed the calculation, as the point multiplication will consist largely of point doublings, which are significantly faster than point additions in most curve and point representations.

7 Freeman Curves

In [5] a construction is suggested for pairing friendly elliptic curves of embedding degree 10.

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

These curves are much rarer than the BN curves, and unfortunately it is not feasible to choose x to have a particularly small Hamming weight. Furthermore since the embedding degree is 10, the best that can be done for G_2 is to represent it as a group of points on $E'(\mathbb{F}_{p^5})$. This is a particularly large and rather awkward extension, and the cofactor multiplication threatens to be a large one. In fact $c(x)$ in this case works out as the rather intimidating polynomial

$$390625x^{16} + 4062500x^{14} + 7421875x^{13} + 10750000x^{12} + 12593750x^{11}$$
$$+ 12356250x^{10} + 10203125x^9 + 7178125x^8 + 4284375x^7 + 2171000x^6 + 920250x^5$$
$$+ 322400x^4 + 89875x^3 + 19120x^2 + 2740x + 217$$

Nevertheless applying our algorithm we find that $c(x)$ can be expressed as

$$g_0(x).P + g_1(x).\psi(P) + g_2(x).\psi^2(P) + g_3(x).\psi^3(P) + g_4(x).\psi^4(P)$$

where

$$g_0(x) = -10x^2 - 20x - 4$$
$$g_1(x) = -50x^3 - 50x^2 - 40x - 12$$
$$g_2(x) = -50x^3 - 40x^2 - 20x - 2$$
$$g_3(x) = -50x^3 - 20x^2 - 10x + 6$$
$$g_4(x) = -50x^3 - 20x^2 - 10x$$

At this stage we could substitute for x and use a simultaneous multiple point multiplication algorithm [8]. However a better idea is to instead calculate xP, $x^2P = x.xP$, $x^3P = x.x^2P$, and then $\psi^i(P), \psi^i(xP), \psi^i(x^2P)$ and $\psi^i(x^3P)$ for i=1 to 4 (in general not all of these values are needed). Then the calculation becomes

$$50(-\psi^4(x^3P) - \psi^3(x^3P) - \psi^2(x^3P) - \psi(x^3P) - \psi(x^2P))$$
$$+40(-\psi(xP) - \psi^2(x^2P)) + 20(-\psi^4(x^2P) - \psi^3(x^2P) - \psi^2(xP) - xP)$$
$$+12(-\psi(P)) + 10(-\psi^4(xP) - \psi^3(xP) - x^2P) + 6\psi^3(P) + 4(-P) + 2(-\psi^2(P))$$

which can be considered as

$$50A + 40B + 20C + 12D + 10E + 6F + 4G + 2H$$

when A, B, C, D, E, F, G and H are calculated using a total of 17 point additions. The optimal way to proceed is to form the smallest addition chain which includes all of the small multipliers in the above.

$$\{\underline{1}, 2, 4, 6, 10, 12, 20, 40, 50\}$$

In this case it is easily done - only a 1 needs to be added to the start.

Now we apply the Olivos algorithm [14], (see also [1], section 9.2) to find the optimal sequence of point additions and doublings to finally effect the cofactor multiplication.

$$\begin{aligned}
T_0 &= A + B \\
T_1 &= A + E \\
T_0 &= 2.T_0 \\
T_0 &= T_0 + C \\
T_0 &= 2.T_0 \\
T_0 &= T_0 + T_1 \\
T_1 &= 2.D \\
T_1 &= T_1 + F \\
T_1 &= T_1 + T_0 \\
T_0 &= T_0 + G \\
T_0 &= T_0 + T_1 \\
T_1 &= T_1 + H \\
T_0 &= 2.T_0 \\
T_0 &= T_0 + T_1 \\
T_0 &= 2.T_0
\end{aligned}$$

The final result is in T_0. This part of the calculation requires only 10 extra point additions and 5 point doublings.

8 KSS Curves

Recently Kachisa et al. [10] described a new method for generating pairing-friendly elliptic curves.

8.1 The $k = 8$ family of curves

Here are the parameters for the family of $k = 8$ KSS curves.

$$
\begin{aligned}
p(x) &= (x^6 + 2x^5 - 3x^4 + 8x^3 - 15x^2 - 82x + 125)/180 \\
r(x) &= x^4 - 8x^2 + 25 \\
t(x) &= (2x^3 - 11x + 15)/15
\end{aligned}
$$

For this curve $\rho = 3/2$. Like the BN curve x can be chosen to have a low Hamming weight. Proceeding as above we find

$$
\begin{aligned}
g_0(x) &= (2x^5 + 4x^4 - x^3 + 50x^2 + 65x - 36)/6 \\
g_1(x) &= (2x^5 + 4x^4 - x^3 - 7x^2 - 25x + 75)/6 \\
g_2(x) &= (-15x^2 - 30x - 75)/6
\end{aligned}
$$

A minor difficulty arises due to the common denominator of 6 which occurs here. We suggest a simple solution – complete the hashing to G_2 with the point multiplication $[6.c(x)]P$. Now the denominator can be ignored. To complete the calculation we need an addition chain which includes all of the integer coefficients that arise here.

$$\{1, 2, 4, \underline{5}, \underline{6}, 7, \underline{10}, 15, 25, 30, 36, 50, 65, 75\}$$

Proceeding as for the Freeman curve case, the computation using this addition chain can be completed with 18 point additions and 5 point doublings.

8.2 The $k = 18$ family of curves

Here are the parameters for the family of $k = 18$ KSS curves.

$$
\begin{aligned}
p(x) &= (x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 + 259x^3 + 343x^2 + 1763x + 2401)/21 \\
r(x) &= x^6 + 37x^3 + 343 \\
t(x) &= (x^4 + 16x + 7)/7
\end{aligned}
$$

In this case $\rho = 4/3$ and like the BN curves x can in practise be chosen with a low Hamming weight. Proceeding again as above we find

$$
\begin{aligned}
g_0(x) &= (-5x^7 - 26x^6 - 98x^5 - 381x^4 - 867x^3 - 1911x^2 - 5145x - 5774)/3 \\
g_1(x) &= (-5x^7 - 18x^6 - 38x^4 - 323x^3 - 28x^2 + 784x)/3 \\
g_2(x) &= (-5x^7 - 18x^6 - 38x^4 - 323x^3 + 1029x + 343)/3 \\
g_3(x) &= (-11x^6 - 70x^5 - 98x^4 - 176x^3 - 1218x^2 - 2058x - 686)/3 \\
g_4(x) &= (28x^2 + 245x + 343)/3
\end{aligned}
$$

As before we actually evaluate $[3.c(x)].P$ to remove the awkward denominator of 3. In this case the best addition chain we could find that includes all of the small multipliers was

{1,2,3,5,7,8,11,18,26,28,31, 38,45,69,70,78,98,176,245,253,323,343,381,389,686, 784,829,867,1029,1218,1658,1911,2058,4116,5145,5774}.

which can be used to complete the calculation in 51 point additions and 5 point doublings.

9 Discussion

Since in most cases (dependent on the curve representation and the projective coordinate method used) point doublings are significantly faster than point additions, it may be sometimes preferable to select a slightly longer addition chain which trades additions for doublings. However the situation is complex and requires further study. For example if multiplying a point on $E'(\mathbb{F}_{p^5})$ it is likely that affine coordinates will in fact be faster than any kind of projective coordinates, in which case, using the standard short Weierstrass representation, additions may actually be faster than doublings [8].

Addition-subtraction chains may also be an attractive alternative in other cases.

10 Conclusions

We have suggested a general method for deriving a point on G_2 of order r given an initial hashing to a general point on G_2, on an ordinary pairing-friendly elliptic curve. The proposed method is significantly faster than the naive approach which would require multiplication by a very large cofactor.

11 Acknowledgement

Thanks to Robert Granger for suggestions and comments.

References

1. R. Avanzi, H. Cohen, D. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography.* Chapman and Hall/CRC, 2006.
2. P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology – Crypto'2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer-Verlag, 2002.
3. P.S.L.M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography – SAC'2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer-Verlag, 2006.

4. I. F. Blake, G. Seroussi, and N. P. Smart, editors. *Advances in Elliptic Curve Cryptography, Volume 2*. Cambridge University Press, 2005.
5. D. Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. In *ANTS VII*, volume 4076 of *Lecture Notes in Computer Science*, pages 452–465. Springer-Verlag, 2006.
6. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing friendly elliptic curves. Cryptology ePrint Archive, Report 2006/372, 2006. `http://eprint.iacr.org/2006/372`.
7. S. Galbraith and M. Scott. Exponentiation in pairing-friendly groups using homomorphisms. In *Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 211–224. Springer-Verlag, 2008.
8. D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curves Cryptography*. Springer, 2004.
9. F. Hess, N. Smart, and F. Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.
10. E. Kachisa, E. Schaefer, and M. Scott. Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In *Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 126–135. Springer-Verlag, 2008.
11. E. Lee, H-S. Lee, and C-M. Park. Efficient and generalized pairing computation on abelian varieties. Cryptology ePrint Archive, Report 2008/040, 2008. `http://eprint.iacr.org/2008/040`.
12. A. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993.
13. A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001.
14. J. Olivos. On vectorial addition chains. *Journal of Algorithms*, 2:13–21, 1981.