On the final exponentiation for calculating pairings on ordinary elliptic 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 to compute the Tate and related pairings, the computation consists of two main components, the Miller loop and the so-called final exponentiation. As a result of good progress being made to reduce the Miller loop component of the algorithm (particularly with the discovery of "truncated loop" pairings like the R-ate pairing [15]), the final exponentiation has become a more significant component of the overall calculation. Here we exploit the structure of pairing friendly elliptic curves to reduce to a minimum the computation required for the final exponentiation.
Keywords: Tate pairing, addition chains

1 Introduction

The most significant parameter of a pairing-friendly elliptic curve is its embedding degree. Using fields of characteristic $q = p^m$ (where p is prime) then there must exist a large group of points on the curve of prime order r, such that k is the smallest integer for which $r|q^k - 1$. This integer k is then the embedding degree with respect to r, and it must be in the range 2-50 to be considered useful [10]. In fact this condition can be simplified to k being the smallest integer such that $r|\Phi_k(q)$ [2], where $\Phi_k(.)$ is the k-th cyclotomic polynomial. Here we will restrict our attention to the case of even embedding degrees, which are more useful and practical, as they support the important"denominator elimination" optimization [2].

The Tate pairing $e(P, Q)$ (and its derivatives) takes as parameters two linearly independent points P and Q, at least one of which must be of order r, on $E(\mathbb{F}_{q^k})$, and the pairing $e(P, Q)$ evaluates as an element of order r over the extension field $\mathbb{F}_{q^k}$. In many cases the points P and Q can be over smaller extension fields, and at least one of them can be on $\mathbb{F}_q$ [4], [5].

Pairing based cryptography on elliptic curves depends on the existence of pairing friendly curves. Two basic choices are available, the supersingular curves

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

over fields of any characteristic, and ordinary pairing-friendly elliptic curves over $\mathbb{F}_p$. In the former case we are strictly limited in terms of the available embedding degree; a maximum of $k = 6$ is possible, but only on curves of characteristic 3.

Note that the embedding degree relates the two types of "hard problem" which supports the security of pairing based cryptography. We need both the "ECC" levels of security as determined by the group size r, and the "RSA" levels of security necessary to defend against index calculus attacks on the discrete logarithm problem over the extension field $\mathbb{F}_{q^k}$. For optimal efficiency at the minimal 160-bit ECC and 1024-bit RSA level, we would want k to be about 1024/160, which is approximately 6. However this level of security is already being questioned. At higher levels of security, a larger value of k would be desirable. Indeed at the standard AES-128 bit level of security it has been suggested that pairing friendly curves with an embedding degree of $k = 12$ would be ideal [8], [12].

Fortunately ordinary pairing friendly elliptic curves are also possible, and here, unlike the supersingular case, we have an unlimited choice of k. Therefore it seems that the long term viability of pairing-friendly cryptosystems is largely dependent on the efficient use of these curves.

2 Ordinary Pairing friendly elliptic curves

One of the first suggested method for the construction of non-supersingular pairing friendly elliptic curves $E(\mathbb{F}_p)$ was by Cocks and Pinch [6]. Their method easily generates curves of any embedding degree k, but with one major disadvantage – the ratio $\rho = \lg(p)/\lg(r)$ is approximately 2. This ρ value is a useful figure-of-merit for pairing friendly curves, and we would prefer it to be closer to 1, as this results in faster implementations. It is normal to choose one of the parameters to the pairing to be a point on the base field $E(\mathbb{F}_p)$, and we would therefore like p to be as small as possible. However with a Cocks-Pinch curve p will have twice as many bits as necessary to support a pairing-friendly group of order r.

If we exclude the Cocks-Pinch curves we are left with numerous "families" of pairing friendly curves which have been discovered, each of which has a ρ value usually much closer to one than to two. Many such familes of ordinary pairing friendly elliptic curves have been suggested – see the Freeman, Scott and Teske taxonomy for details [10]. These families have one striking feature in common – the prime modulus p and the group r are described as rather simple polynomials with relatively small integer coefficients. It is our aim to exploit this simple form in a systematic way to speed up the final exponentiation for all families of non-supersingular pairing-friendly elliptic curves.

3 The final exponentiation

After the Miller loop the Tate pairing (and its derivatives) must all carry out an extra step to ensure a unique result of the pairing. To this end the output of

the Miller loop m must be raised to be power of $(p^k - 1)/r$ to create a result of order r. Note that this exponent is determined by fixed system parameters, and therefore methods of exponentiation optimised for fixed exponents are applicable here.

However this final exponent can be broken down into three components. Let $d = k/2$. Then

$$(p^k - 1)/r = (p^d - 1).[(p^d + 1)/\Phi_k(p)].[\Phi_k(p)/r]$$

For example if $k = 12$ the final exponent becomes

$$(p^{12} - 1)/r = (p^6 - 1).(p^2 + 1).[(p^4 - p^2 + 1)/r]$$

The first two parts of the exponentiation are "easy" as raising to the power of p is an almost free application of the Frobenius operator, as p is the characteristic of the extension field. However the first part of the exponentiation is not only cheap (although it does require an extension field division), it also simplifies the rest of the final exponentiation. After raising to the power of $p^d - 1$ the field element becomes "unitary" [20]. This has important implications, as squaring of unitary elements is significantly cheaper than squaring of non-unitary elements, and any future inversions can be implemented by simple conjugation [21], [20], [12].

This brings us to the "hard part" of the final exponentiation, raising to the power of $\Phi_k(p)/r$. The usual continuation is to express this exponent to the base p as $\lambda_{n-1}.p^{n-1} + ... + \lambda_1.p + \lambda_0$, where $n = \phi(k)$, and $\phi(.)$ is the Euler Totient function. If the value to be exponentiated is m, then we need to calculate

$$m^{\lambda_{n-1}.p^{n-1}}....m^{\lambda_1.p}.m^{\lambda_0}$$

which is the same as

$$(m^{p^{n-1}})^{\lambda_n - 1}.....(m^p)^{\lambda_1}.m^{\lambda_0}$$

The m^{p^i} can be calculated using the Frobenius, and the hard part of the final exponentiation can be calculated using a fast multi-exponentiation algorithm [13], [11], [16].

However this method does not exploit the polynomial description of p and r. It is our intention to do so, and hence obtain a faster hard-part of the final exponentiation. Each family is different in detail, so we will proceed on a case-by-case basis.

4 The MNT curves

The MNT pairing friendly elliptic curves were reported by Miyaji et al. [17]. For the $k = 6$ case the prime p and the group order r parameter are expressed as

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

In this case the hard part of the final exponentiation is to the power of $(p^2-p+1)/r$. Substituting from the above this becomes $(x^4+x^2+1)/(x^2-x+1) = x^2 + x + 1$. Expressing this to the base p, it becomes simply $p + x$. So the hard part of the final exponentiation is $m^p.m^x$ – an application of the Frobenius and a simple exponentiation to the power of x. The advantage of deriving the hard part of the exponentiation in terms of the family parameter x is clearly illustrated.

5 The BN curves

The BN family of pairing friendly curves [5] 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 hard part of the final exponentiation is to the power of $(p^4 - p^2 + 1)/r$. After some work this can be expressed to the base p as

$$\lambda_3.p^3 + \lambda_2.p^2 + \lambda_1.p + \lambda_0$$

where

$$\lambda_3(x) = 1$$
$$\lambda_2(x) = 6x^2 + 1$$
$$\lambda_1(x) = -36x^3 - 18x^2 - 12x + 1$$
$$\lambda_0(x) = -36x^3 - 30x^2 - 18x - 2$$

Now we take a new approach. BN curves are very plentiful, and it already helps the Miller loop if we choose x to have a low Hamming weight. In fact Nogami et al. [18] have suggested the nice choice of $x = -4080000000000001_{16}$ for a curve appropriate for the AES-128 level of security. Next we compute m^x, $m^{x^2} = (m^x)^x$, and $m^{x^3} = (m^{x^2})^x$. These are simple exponentiations, and the low Hamming weight of x ensures that each requires a minimum of multiplications when using a simple square-and-multiply algorithm. We next calculate m^p, m^{p^2}, m^{p^3}, $(m^x)^p$, $(m^{x^2})^p$, $(m^{x^3})^p$ and $(m^{x^2})^{p^2}$ using the Frobenius.

Now group the elements of the exponentiation together, and the expression becomes

$$[m^p.m^{p^2}.m^{p^3}].[1/m]^2.[(m^{x^2})^{p^2}]^6.[1/((m^x)^p)]^{12}.[1/(m^x.(m^{x^2})^p)]^{18}.[1/m^{x^2}]^{30}.[1/(m^{x^3}.(m^{x^3})^p)]^{36}$$

The individual components between the square brackets are then calculated with just 4 multiplications (recalling that division costs the same as a multiplication, as inversion is just a conjugation), and we end up with a calculation of the form

$$y_0.{y_1}^2.{y_2}^6.{y_3}^{12}.{y_4}^{18}.{y_5}^{30}.{y_6}^{36}$$

Note that the exponents here are simply the coefficients that arise in the λ_i equations above. Now how best to evaluate this expression?

In fact there is a well known algorithm to evaluate expressions of this form, which minimizes the number of required multiplications. See Olivos [19], and also [1], section 9.2 for a nice worked example. The starting point is to find an addition chain which includes within it the elements of the addition sequence formed from the set of integers which occur as exponents. In this case it is not hard to see that an optimal addition chain is given by

$$\{1, 2, \underline{3}, 6, 12, 18, 30, 36\}$$

Note that 3 is the only member of the addition chain which is not a member of the addition sequence. This is certainly serendipitous, as it means less work to do the evaluation. Observe here that an addition-subtraction chain is also a possibility (as divisions are as cheap as multiplications as a consequence of the unitary property). But we don't require one here. Application of the Olivos algorithm results in the following vectorial addition chain

$$
\begin{array}{ccccccc}
(1 & 0 & 0 & 0 & 0 & 0 & 0) \\
(0 & 1 & 0 & 0 & 0 & 0 & 0) \\
(0 & 0 & 1 & 0 & 0 & 0 & 0) \\
(0 & 0 & 0 & 1 & 0 & 0 & 0) \\
(0 & 0 & 0 & 0 & 1 & 0 & 0) \\
(0 & 0 & 0 & 0 & 0 & 1 & 0) \\
(0 & 0 & 0 & 0 & 0 & 0 & 1) \\
(2 & 0 & 0 & 0 & 0 & 0 & 0) \\
(2 & 0 & 1 & 0 & 0 & 0 & 0) \\
(2 & 1 & 1 & 0 & 0 & 0 & 0) \\
(0 & 1 & 0 & 1 & 0 & 0 & 0) \\
(2 & 2 & 1 & 1 & 0 & 0 & 0) \\
(2 & 1 & 1 & 0 & 1 & 0 & 0) \\
(4 & 4 & 2 & 2 & 0 & 0 & 0) \\
(6 & 5 & 3 & 2 & 1 & 0 & 0) \\
(12 & 10 & 6 & 4 & 2 & 0 & 0) \\
(12 & 10 & 6 & 4 & 2 & 1 & 0) \\
(12 & 10 & 6 & 4 & 2 & 0 & 1) \\
(24 & 20 & 12 & 8 & 4 & 2 & 0) \\
(36 & 30 & 18 & 12 & 6 & 2 & 1)
\end{array}
$$

which in turn allows us to evaluate the expression as follows, using just two temporary variables.

$$
\begin{aligned}
t_0 &= (y_6)^2 \\
t_0 &= t_0.y_4 \\
t_0 &= t_0.y_5 \\
t_1 &= y_3.y_5 \\
t_1 &= t_1.t_0 \\
t_0 &= t_0.y_2 \\
t_1 &= (t_1)^2 \\
t_1 &= t_1.t_0 \\
t_1 &= (t_1)^2 \\
t_0 &= t_1.y_1 \\
t_1 &= t_1.y_0 \\
t_0 &= (t_0)^2 \\
t_0 &= t_0.t_1
\end{aligned}
$$

The final result is in t_0. This part of the calculation requires only 9 multiplications and 4 squarings. We find this approach to the hard part of the final exponentiation for the BN curves to be about 5% faster than the rather ad hoc method proposed by Devegili et al. [8]. Moreover our more general method is applicable to all families of pairing friendly curves.

6 Freeman Curves

In [9] 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. Nevertheless proceeding as above we find

$$\lambda_3(x) = 1$$
$$\lambda_2(x) = 10x^2 + 5x + 5$$
$$\lambda_1(x) = -5x^2 - 5x - 3$$
$$\lambda_0(x) = -25x^3 - 15x^2 - 15x - 2$$

In this case the coefficients form a perfect addition chain

$$\{1, 2, 3, 5, 10, 15, 25\}$$

The optimal vectorial addition chain in this case requires 10 multiplications and 2 squarings.

7 KSS Curves

Recently Kachisa et al. [14] described a new method for generating pairing-friendly elliptic curves.

7.1 The $k = 8$ family of curves

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

$$p(x) = (x^6 + 2x^5 - 3x^4 + 8x^3 - 15x^2 - 82x + 125)/180$$
$$r(x) = x^4 - 8x^2 + 25$$
$$t(x) = (2x^3 - 11x + 15)/15$$

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

$$\lambda_3(x) = (15x^2 + 30x + 75)/6$$
$$\lambda_2(x) = (2x^5 + 4x^4 - x^3 + 26x^2 - 55x - 144)/6$$
$$\lambda_1(x) = (-5x^4 - 10x^3 - 5x^2 - 80x + 100)/6$$
$$\lambda_0(x) = (x^5 + 2x^4 + 7x^3 + 28x^2 + 10x + 108)/6$$

A minor difficulty arises due to the common denominator of 6 which occurs here. We suggest a simple solution – evaluate instead the sixth power of the pairing. This does not effect the important properties of the pairing and now we can simply ignore the denominator. We find by brute-force computer search that we can construct the following optimal addition chain which contains all the exponents in the above equations.

$$\{1, 2, 4, 5, 7, 10, 15, \underline{25}, 26, 28, 30, \underline{36}, \underline{50}, 55, 75, 80, 100, 108, 144\}$$

Proceeding as for the BN case we find that the vectorial addition chain derived from this addition chain requires just 27 multiplications and 6 squarings to complete the calculation of the hard part of the final exponentiation.

7.2 The $k = 18$ family of curves

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

$$p(x) = (x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 + 259x^3 + 343x^2 + 1763x + 2401)/21$$
$$r(x) = x^6 + 37x^3 + 343$$
$$t(x) = (x^4 + 16x + 7)/7$$

In this case $\rho = 4/3$ but nonetheless this curve might make a good choice for a pairing at the AES-192 bit level of security. Again like the BN curves, x can in practise be chosen with a low Hamming weight. Proceeding again as above we find

$$\lambda_5(x) = (49x^2 + 245x + 343)/3$$
$$\lambda_4(x) = (7x^6 + 35x^5 + 49x^4 + 112x^3 + 581x^2 + 784x)/3$$
$$\lambda_3(x) = (-5x^7 - 25x^6 - 35x^5 - 87x^4 - 450x^3 - 609x^2 + 54)/3$$
$$\lambda_2(x) = (-49x^5 - 245x^4 - 343x^3 - 931x^2 - 4802x - 6517)/3$$
$$\lambda_1(x) = (14x^6 + 70x^5 + 98x^4 + 273x^3 + 1407x^2 + 1911x)/3$$
$$\lambda_0(x) = (-3x^7 - 15x^6 - 21x^5 - 62x^4 - 319x^3 - 434x^2 + 3)/3$$

As before we evaluate the cube of the pairing to remove the awkward denominator of 3. In this case the coefficients again "nearly" form a natural addition chain. Our best attempt to find an addition chain containing all of the

exponents in the above, is

{1,2,3,4,5,7,8,14,15,16,21,25,28,35,42,49,54,62,70,87,98,112,147,245,273,294,
319,343,392,434,450,581,609,784,931,1162,1407,1862,1911,3724,4655,4802,6517}.

Proceeding as for the BN case we find that the vectorial chain derived from this addition chain requires just 56 multiplications and 14 squarings to complete the calculation of the hard part of the final exponentiation. In fact we did eventually find (by partial computer search) an addition chain one element shorter than the above, but as it required 61 multiplications and only 7 squarings, we prefer to use the solution above.

8 Discussion

Here we make a few general observations. First it seems that the proposed method results in surprisingly compact addition chains. We note also that the coefficients in the λ_i tend to be "smooth" numbers, having only relatively small factors. This may facilitate the construction of addition chains. Other intriguing patterns emerge – observe for example that for the KSS $k = 18$ curves the three most significant coefficients of the λ_i are all in the same ratio 1:5:7. Coefficients also appear to follow the same kind of distribution as numbers in a typical addition chain.

We have also used the proposed method for other families of pairing-friendly curves, and have observed that for example for the $k = 8$, $\rho = 5/4$ curve proposed by Brezing and Weng [7], and the $k = 12$, $\rho = 3/2$ curve found by Barreto et al. [3], the resulting addition chain is often as easy as

$$\{1, 2, 3\}$$

Since squarings are significantly faster than multiplications it may, as we have seen, be sometimes preferable to select a slightly longer addition chain which trades additions for doublings. Addition-subtraction chains may also be an attractive alternative in other cases.

9 Conclusions

We have suggested a general method for the implementation of the hard part of the final exponentiation in the calculation of the Tate pairing and its derivatives, which is faster, is generally applicable, and which requires less memory than previously described methods. The most promising Tate pairing derivative is the R-ate pairing [15]. An intriguing possibility is that, given only the polynomial equations defining a pairing-friendly family of elliptic curves, it should now be possible, and indeed appropriate, to write a computer program which would automatically generate the optimal R-ate pairing code.

10 Acknowledgements

Thanks to Yu Chen and Fre Vercauteren for pointing out an error in the sign of x for the BN curves.

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, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In *Security in Communication Networks – SCN'2002*, volume 2576 of *Lecture Notes in Computer Science*, pages 263–273. Springer-Verlag, 2002.
4. P. S. L. M. Barreto, B. Lynn, and M. Scott. On the selection of pairing-friendly groups. In *Selected Areas in Cryptography – SAC'2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 17–25. Springer-Verlag, 2003.
5. 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.
6. I. F. Blake, G. Seroussi, and N. P. Smart, editors. *Advances in Elliptic Curve Cryptography, Volume 2*. Cambridge University Press, 2005.
7. F. Brezing and A. Weng. Elliptic curves suitable for pairing based cryptography. *Designs, Codes and Cryptology*, 37:133–141, 2005.
8. A. J. Devegili, M. Scott, and R. Dahab. Implementing cryptographic pairings over Barreto-Naehrig curves. In *Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 197–2007. Springer-Verlag, 2007.
9. 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.
10. 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`.
11. R. Granger, D. Page, and N. P. Smart. High security pairing-based cryptography revisited. In *Algorithmic Number Theory Symposium – ANTS VII*, volume 4076 of *Lecture Notes in Computer Science*, pages 480–494. Springer-Verlag, 2006.
12. D. Hankerson, A. Menezes, and M. Scott. Software implementation of pairings. CACR Technical Report, 2008. `http://www.cacr.math.uwaterloo.ca/`.
13. L. Hei, J. Dong, and D. Pei. Implementation of cryptosystems based on Tate pairing. *J. Comput. Sci & Technolgy*, 20(2):264–269, 2005.
14. 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.
15. 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`.

16. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press, Boca Raton, Florida, 1996. URL: `http://cacr.math.uwaterloo.ca/hac`.
17. 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.
18. Y. Nogami, M. Akane, Y. Sakemi, H. Kato, and Y. Morikawa. Integer variable X-based ate pairing. In *Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 178–191. Springer-Verlag, 2008.
19. J. Olivos. On vectorial addition chains. *Journal of Algorithms*, 2:13–21, 1981.
20. M. Scott and P. Barreto. Compressed pairings. In *Advances in Cryptology – Crypto' 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 140–156. Springer-Verlag, 2004. Also available from `http://eprint.iacr.org/2004/032/`.
21. M. Stam and A. K. Lenstra. Efficient subgroup exponentiation in quadratic and sixth degree extensions. In *CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 318–332. Springer-Verlag, 2002.