Simplified pairing computation and security implications

Steven D. Galbraith[1], Colm Ó hÉigeartaigh[2], and Caroline Sheedy[2]

[1] Mathematics Department, Royal Holloway University of London,
Egham, Surrey, TW20 0EX, U.K.
steven.galbraith@rhul.ac.uk
[2] School of Computing, Dublin City University.
Ballymun, Dublin 9, Ireland.
{coheigeartaigh,csheedy}@computing.dcu.ie

Abstract. Recent progress on pairing implementation has made certain pairings extremely simple and fast to compute. Hence, it is natural to examine if there are consequences for the security of pairing-based cryptography.
This paper gives a method to compute eta pairings in a way which avoids the requirement for a final exponentiation. The method does not lead to any improvement in the speed of pairing implementation. However, it seems appropriate to re-evaluate the security of pairing based cryptography in light of these new ideas. A multivariate attack on the pairing inversion problem is proposed and analysed. Our findings support the belief that pairing inversion is a hard computational problem.

1 Introduction

The use of pairings as a component of protocols is a major topic in public key cryptography [2]. The security of cryptosystems based on pairings depends on the difficulty of various new computational problems. However, in comparison with many other problems in cryptography, there has been little scrutiny of these computational problems.

One of the most fundamental computational problems in this area is the pairing inversion problem (see Section 2.1 for a statement of this problem). Results of Verheul [11, 12] and Satoh [10] provide evidence that this problem is hard.

Recent progress on pairing implementation has made some pairings extremely simple and fast to compute. Hence, it seems appropriate to re-evaluate the security of pairing based cryptography in light of such progress.

We show that, for some very specific curves, the final exponentiation in the pairing computation is not required. This makes pairing computation very simple and it is natural to wonder if the difficulty of the pairing inversion problem is affected for such curves. The fact that the final exponentiation can sometimes be avoided may also have implications for side-channel analysis of pairing implementations (see [9]).

We propose and analyse a multivariate attack on the pairing inversion problem. Our findings further support the belief that pairing inversion is a hard computational problem.

The paper is outlined as follows. In Section 2 we recall some well-known facts about pairings. Section 3 details how to compute Tate pairings in a way which does not require a final exponentiation. The remaining sections discuss and analyse various possible attacks on the pairing inversion problem.

2 Pairings

Let E be an elliptic curve over a finite field $\mathbb{F}_q$ (we will only consider supersingular elliptic curves in this article). We denote the point at infinity by 0. For any $n \in \mathbb{N}$ we denote by $E(\mathbb{F}_{q^n})$ the group of points on E defined over $\mathbb{F}_{q^n}$. Suppose r is a (large) prime, coprime to q, which divides $\#E(\mathbb{F}_q)$. Let k be the smallest positive integer such that $r \mid (q^k - 1)$. We define $\mu_r = \{z \in \mathbb{F}_{q^k}^* : z^r = 1\}$ and define $E(\mathbb{F}_{q^n})[r] = \{P \in E(\mathbb{F}_{q^n}) : [r]P = 0\}$.

We are interested in non-degenerate bilinear pairings of the form

$$e : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_q)[r] \to \mu_r \subseteq \mathbb{F}_{q^k}^*$$

For supersingular elliptic curves of cryptographic interest, we may obtain such a pairing from the Tate pairing twisted by an endomorphism ψ called a distortion map [11, 12].

For example, if the Tate pairing is used then we define

$$e(P, Q) = \langle P, \psi(Q) \rangle_r^{(q^k - 1)/r}.$$

2.1 Inverting pairings

A natural way to attack some pairing-based cryptosystems is to solve the following computational problem:

Pairing inversion problem: Suppose

$$e : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_q)[r] \to \mu_r \subseteq \mathbb{F}_{q^k}^*$$

is as above and suppose $P \in E(\mathbb{F}_q)[r]$ and $z \in \mu_r$ are given. Find a point $Q \in E(\mathbb{F}_q)[r]$ such that

$$e(P, Q) = z.$$

The pairing inversion problem is essentially to compute a specific group homomorphism from μ_r to $E(\mathbb{F}_q)[r]$. Hence it is also natural to consider the more general problem of computing homomorphisms between finite fields and elliptic curves.

Verheul [11, 12] considered the problem of computing a group homomorphism from μ_r to $E(\mathbb{F}_q)[r]$. He showed a number of striking consequences of being able to compute such a homomorphism, for example the fact that the Diffie-Hellman problem would become easy for a number of finite fields. These results can be interpreted as evidence that inverting pairings is a hard problem.

Satoh [10] has given further evidence that computing such a homomorphism is hard. He showed that if the group homomorphism is represented as a polynomial then, in many cases, the polynomial has large degree and all coefficients non-zero.

2.2 The final exponentiation as a security feature

When computing the Tate pairing it is necessary to compute the value of a Miller function and then to perform a final exponentiation to the power of $(q^k - 1)/r$ to obtain a unique and well-defined value. This final exponentiation seems to have a positive contribution to the security of the resulting system as it destroys information. More precisely, given a pairing value $z \in \mu_r$ it is not at all clear what the actual output value of Miller's algorithm is. This issue was also noted in [9].

3 Computing eta pairings without a final exponentiation

The eta pairing [1] is a generalisation of the Duursma-Lee [4] method for computing pairings. It greatly simplifies pairing computation for supersingular curves over fields of small characteristic with even embedding degree. There are two variants of the eta pairing, namely the basic version which is equivalent to the Duursma-Lee method and the truncated version which has better performance. Our methods seem to only apply in the case of the basic version.

We give the full details for the case of elliptic curves of embedding degree 4 in characteristic 2. Later we give a brief discussion of the characteristic 3 case.

Our results seem to rely on the coincidence between the characteristic and the base used for representing the multiplier in Miller's algorithm. Hence these results do not seem to immediately generalise to any other cases (e.g., [6]), though this is an interesting question for future research.

3.1 The characteristic two case

We recall the case of elliptic curves in characteristic 2. The curve $E : y^2+y = x^3+x+b$ (where $b = 0$ or 1) over $\mathbb{F}_{2^m}$, where m is odd, has embedding degree 4. Let $P, Q \in E(\mathbb{F}_{2^m})$ be points of order r and let ψ be the usual distortion map $\psi(x, y) = (x+s^2, y+sx + t)$ where $s \in \mathbb{F}_{2^2}$ satisfies $s^2 + s + 1 = 0$ and $t \in \mathbb{F}_{2^4}$ satisfies $t^2 + t + s = 0$. Denote by $f_{n,P}$ a function with divisor $(f_{n,P}) = n(P) - ([n]P) - (n - 1)(0)$. The eta pairing is defined to be the value

$$f_{T,P}(\psi(Q)) \in \mathbb{F}^*_{2^{4m}}$$

where $T = 2^m$. To get a uniquely defined pairing one should exponentiate to the power $2^{2m}-1$, which can be easily done by applying a linear map (2^{2m}-power Frobenius) and a division. This exponentiation ostensibly transforms the output of Miller's algorithm to an element of order dividing $2^{2m}+1$, equivalently an element of norm 1 with respect to the Galois field extension $\mathbb{F}_{2^{4m}}/\mathbb{F}_{2^{2m}}$. In fact, the pairing value is then an element of $\mathbb{F}^*_{2^{4m}}$ of order equal to the order r of the points P and Q.

It is shown in [1] how to compute the eta pairing efficiently, for example using denominator elimination. Furthermore, a loop shortening method is given which reduces the computation to just $(m + 1)/2$ iterations.

We now show how the final exponentiation can be avoided. The key idea is to not use the denominator elimination technique.

We use the standard eta pairing notation from [1]. Recall that if $P = (x_P, y_P)$, then $[2^i]P = (x_P^{(2i)} + i, y_P^{(2i)} + ix_P^{(2i)} + \tau(i))$ where $\tau(i) = 1$ if $i \equiv 2, 3 \pmod 4$ and zero otherwise (keeping in mind that we are working in characteristic 2).

We work in $\mathbb{F}_{q^2}$ where $q = 2^{2m}$ and m is odd. This field may be represented as $\{a + bt : a, b \in \mathbb{F}_q\}$ where $t \in \mathbb{F}_{2^4}$ satisfies $t^2 + t + s = 0$ and $s \in \mathbb{F}_{2^2}$ satisfies $s^2 + s + 1 = 0$. By the conjugate of $a + bt$ we mean $\overline{a + bt} = a + b(t + 1) = (a + b) + bt$. By the norm of $a + bt$ we mean

$$(a + bt)\overline{(a + bt)} = a^2 + ab + b^2 s.$$

If $Q = (x, y)$ we denote $\overline{Q} = (\overline{x}, \overline{y})$. We require the observation that if $Q \in E(\mathbb{F}_q)$ and $\psi(Q) = (x, y)$ then

$$\overline{\psi(Q)} = (x, y + 1) = -\psi(Q). \tag{1}$$

We denote by $l_P(x, y) = y - \lambda(x - x_P) - y_P$ the equation of the tangent to the curve at P (used in the addition formulae for doubling P) and $v_P(x) = x - x_P$ the vertical line through P.

Lemma 1. *We have* $l_P(\psi(Q))\overline{l_P(\psi(Q))} = v_P(\psi(Q))^2 v_{[2]P}(\psi(Q))$.
(More generally, if l_{P_1,P_2} *is the line between* P_1 *and* P_2 *then*
$l_{P_1,P_2}(\psi(Q))\overline{l_{P_1,P_2}(\psi(Q))} = v_{P_1}(\psi(Q))v_{P_2}(\psi(Q))v_{P_1+P_2}(\psi(Q))$*.)*

Proof. The divisor of the function $l_P(x, y)$ is $(l_P) = 2(P) + (-[2]P) - 3(0)$. As noted above, $\overline{l_P(\psi(Q))} = l_P(\psi(Q)) + 1$. The divisor of the function $l_P(x, y) + 1$ is $(l_P + 1) = 2(-P) + ([2]P) - 3(0)$. The function $v_P(x, y) = (x - x_P)$ has divisor $(v_P) = (P) + (-P) - 2(0)$ (and similarly for $v_{[2]P}(x, y)$). Hence,

$$(l_P(l_P + 1)) = 2(P) + 2(-P) + ([2]P) + (-[2]P) - 6(0) = (v_P^2 v_{[2]P})$$

As the functions $l_P(x, y)(l_P(x, y) + 1)$ and $v_P^2(x, y)v_{[2]P}(x, y)$ have the same divisor, they are equal up to a scalar multiple.

But $l_P(x, y)(l_P(x, y) + 1) = (y - \lambda x + c)(y - \lambda x + c + 1) = y^2 + y + \lambda^2 x^2 + \cdots = x^3 + x + b + \lambda^2 x^2 + \cdots$. Similarly, $v_P^2(x, y)v_{[2]P}(x, y) = (x - x_P)^2(x - x_{[2]P}) = x^3 + \cdots$. Since the coefficients of x^3 are 1 in both cases it follows that $l_P(x, y)(l_P(x, y) + 1) = v_P^2(x, y)v_{[2]P}(x, y)$ and hence the values when evaluated at $\psi(Q)$ are equal.

Now, consider computing the function value $f_{T,P}(\psi(Q))$ where $T = 2^m$ using Miller's algorithm and keeping the denominators.

Theorem 1. *Suppose we compute the function value* $z = f_{T,P}(\psi(Q))$ *where* $T = 2^m$ *using Miller's algorithm and keeping the denominators. Then* z *already has norm 1.*

Proof. We show that $z\overline{z} = 1$ where $\overline{z} = z^{q^2}$. We abuse notation by writing $l_{[2^i]P}$ for $l_{[2^i]P}(\psi(Q))$.

By the eta pairing formulation in [1] (ignoring the denominator elimination) we can compute z as

$$\prod_{i=0}^{m-1} \left(\frac{l_{[2^i]P}}{v_{[2^{i+1}]P}} \right)^{2^{m-1-i}}.$$

Hence

$$z\overline{z} = \prod_{i=0}^{m-1} \left(\frac{l_{[2^i]P}\overline{l_{[2^i]P}}}{v_{[2^{i+1}]P}^2} \right)^{2^{m-1-i}} = \prod_{i=0}^{m-1} \left(\frac{v_{[2^i]P}^2 v_{[2^{i+1}]P}}{v_{[2^{i+1}]P}^2} \right)^{2^{m-1-i}} .$$

Cancelling the $v_{[2^{i+1}]P}$ and expanding the powers gives (using the notation $\alpha^{(i)} = \alpha^{2^i}$)

$$z\overline{z} = \prod_{i=0}^{m-1} \frac{v_{[2^i]P}^{(m-i)}}{v_{[2^{i+1}]P}^{(m-1-i)}}.$$

We can set $j = i + 1$ and note that the above product is

$$\left(\prod_{i=0}^{m-1} v_{[2^i]P}^{(m-i)} \right) / \left(\prod_{j=1}^{m} v_{[2^j]P}^{(m-j)} \right) = \frac{v_P(\psi(Q))^{2^m}}{v_{[2^m]P}(\psi(Q))}.$$

One can check that

$$v_P(\psi(Q))^{2^m} = (x_Q + s^2 + x_P)^{2^m} = x_Q + s^2 + x_P + 1 = v_{[2^m]P}(\psi(Q))$$

and so the norm of the pairing is 1.

Since the value of Miller's algorithm already has norm 1, there is no need to perform the final exponentiation. Hence we can compute the eta pairing by including the vertical lines and thus avoiding the final exponentiation. From a performance point of view there is no saving: the final exponentiation is just a division and we have replaced this by having denominators in the algorithm.

We note that we cannot seem to simultaneously utilise the loop shortening idea (truncated eta pairing) and avoid the final exponentiation. In this case, let $T = 2^{(m+1)/2} \pm 1$. Then the pairing is computed as

$$z = \prod_{i=0}^{(m-1)/2} \left(\frac{l_{[2^i]P}}{v_{[2^{i+1}]P}} \right)^{2^{(m-1)/2-i}} l/v_{[T]P}$$

where the final l and $v_{[T]P}$ come from adding $[2^{(m+1)/2}]P$ and $\pm P$. Again, one can consider $z\overline{z}$ and much of the preceding proof applies. The problem is that after canceling terms we get

$$\frac{v_P(\psi(Q))^{2^{(m-1)/2}} v_P(\psi(Q))}{v_{[T]P}(\psi(Q))}$$

and this is not equal to 1.

3.2 The characteristic three case

Similar arguments can be applied in this case. Let $E : y^2 = x^3 - x + b$ where $b = \pm 1$ over $\mathbb{F}_{3^m}$, where m is coprime to 6. This curve has embedding degree 6. This curve has a point tripling formula, such that given a point $P = (x, y)$, one computes $[3]P = (x_3, y_3)$ as

$$x_3 = (x^9 - b), y_3 = -(y^9)$$

The distortion map is

$$\psi(x, y) = (\rho - x, \sigma y)$$

where $\sigma \in \mathbb{F}_{3^2}$ satisfies $\sigma^2 = -1$ and $\rho \in \mathbb{F}_{3^3}$ satisfies $\rho^3 - \rho - b = 0$. It is easy to see that if $\psi(Q) = (x, y)$ then $\overline{\psi(Q)} = (x, -y) = -\psi(Q)$.

We consider the function

$$g_P(x, y) = y_P^3 y - (x_P^3 - x + b)^2$$

which has divisor $(g_P) = 3(P) + (-3P) - 4(0)$. The following lemma is proven in a similar manner to Lemma 1.

Lemma 2. *We have* $g_P(\psi(Q))\overline{g_P(\psi(Q))} = v_P(\psi(Q))^3 v_{3P}(\psi(Q))$.

Proof. The divisor of the function $\overline{g_P(x, y)}$ is $(\overline{g_P}) = 3(-P) + (3P) - 4(0)$. The divisor of the function $v_P^3(x, y)$ is $(v_P^3) = 3(P) + 3(-P) - 6(0)$ and the divisor of the function $v_{3P}(x, y)$ is $(v_{3P}) = (3P) + (-3P) - 2(0)$. Therefore,

$$(g_P\overline{g_P}) = 3(P) + 3(-P) + (3P) + (-3P) - 8(0) = (v_P^3 v_{3P})$$

We have $l_P(x, y)\overline{l_P}(x, y) = (ay - x^2 + c)(-ay - x^2 + c) = x^4 - a^2y^2 + \cdots = x^4 - a^2(x^3 - x + b) + \cdots$. Similarly, $v_P^3(x, y)v_{3P}(x, y) = (x - x_P)^3(x - x_{3P}) = x^4 + \cdots$. As the coefficients of x^4 are 1 in both cases it follows that $g_P(x, y)\overline{g_P(x, y)} = v_P^3 v_{3P}$.

Theorem 2. *Suppose we compute the function value* $z = f_{T,P}(\psi(Q))$ *where* $T = 3^m$ *using Miller's algorithm and keeping the denominators. Then* z *already has norm 1.*

Proof. The eta pairing method computes z as

$$\prod_{i=0}^{m-1} \left(\frac{l_{[3^i]}P(\psi(Q))}{v_{[3^{i+1}]P}(\psi(Q))} \right)^{3^{m-1-i}}.$$

As with the proof of Theorem 1, one considers $z\overline{z}$, expands the functions and simplifies to get

$$z\overline{z} = \frac{v_P(\psi(Q))^{3^m}}{v_{[3^m]P}(\psi(Q))}$$

One sees that

$$v_P(\psi(Q))^{3^m} = (\rho - x_Q - x_P)^{3^m} = -x_Q - x_P + \rho + b = v_{[3^m]P}(\psi(Q))$$

and therefore the norm of the pairing is 1.

As before, we cannot seem to avoid the final multiplication when using the truncated eta pairing.

4 Security concerns

In this section we look at two possible resulting attacks on the system.

4.1 Inverting pairings using SLP representation

Assume an attacker has access to the 'raw' value $f_{T,P}(\psi(Q))$ of some efficient version of the Tate pairing.

One natural attack is to represent the Tate pairing $e(P, \psi(\cdot))$ as a function of a generic curve point. That is, to run Miller's algorithm on a generic point (x, y) and to compute the corresponding function $f(x, y) = f_{T,P}(\psi(x, y))$.

This function has exponential degree if written in expanded form. However, by construction it has a polynomial sized representation as a straight line program (SLP).

Hence, given a target $z \in \mu_r$ the task is simply to solve the equation

$$f(x, y) = z$$

where $f(x, y)$ is represented as an SLP. Consultations with experts on SLPs [7] indicate that this is a known hard problem.

4.2 Multivariate attack on pairing inversion

Given the simple nature of Tate pairing computation on supersingular curves, and the fact that the final exponentiation can be avoided in some cases, it seems natural to attempt to express the pairing inversion problem as a problem of solving a system of multivariate equations.

We consider a simplified situation of the characteristic 2 case. More precisely we suppose that the eta pairing can be computed, without a final exponentiation, as

$$z = \prod_{i=0}^{m-1} \left(y_Q^{(i)} + A_i x_Q^{(i)} + B_i x_P^{(-i)} + C_i y_P^{(-i)} + D_i \right) \tag{2}$$

where the $A_i, \ldots, D_i$ are explicit constants and where $x^{(i)}$ means x^{2^i}.

This attack assumes that x_P and y_P are fixed and that a target value z is specified. The goal is to find values for x_Q and y_Q. We write the unknown x_Q over some basis as $x_0\theta_0 + x_1\theta_1 + \cdots + x_{m-1}\theta_{m-1}$ where all the x_i lie in $\mathbb{F}_2$ and similarly for y_Q. Since we are working in characteristic two, x_Q^2 can be expressed in terms of the variables x_i (and similarly for y_Q^2 etc) by applying a known linear transformation.

Equation (2) therefore becomes a product of linear equations over $\mathbb{F}_{2^{4m}}$ in the $2m$ variables x_j, y_j. Equating coefficients over $\mathbb{F}_2$ gives a system of $4m$ non-linear equations of degree m in $2m$ variables.

In fact, the methods of Section 3 show that the pairing can be computed as a product of ratios

$$\prod_i \frac{l_i}{v_i} = z.$$

We obtain a system of equations by means analogous to the above as

$$(\prod_i l_i) - z(\prod_i v_i) = 0.$$

The number of monomials in an equation of degree D in M variables is bounded by

$$\binom{M+D-1}{D}.$$

Hence, for the eta pairing we have a system of equations, each consisting of roughly $(2m)^m$ monomials. This is grows exponentially in m so it seems unlikely that linearisation or Gröbner basis methods can be successfully applied to solve this problem.

Nevertheless, we have implemented the method for extremely small parameters using Magma [3] and can invert the pairing using Gröbner basis reduction fairly easily. Hence, there is clearly no conceptual obstacle to the method.

If it were possible to consider the truncated eta pairing, then we would require only $2(m+1)/2 = m+1$ variables and would obtain a system of $4m$ equations of degree $(m+1)/2$. This is a considerably simpler system, but it still has exponential size.

We conclude that the multivariate attack on pairing inversion is not feasible. However, it should be noted that any further progress in loop shortening could potentially jeopardise security.

Some possible variants/improvements on the attack are briefly listed below.

- The attack can also be developed in the characteristic three case, exploiting the fact that cubing is linear. In this case the values of m are typically smaller than in the characteristic two case, but the attack still seems to be infeasible in practice.
- Instead of solving $\prod_{i=1}^{m} f_i = z$ it could be split as

  $$\prod_{i=1}^{m/2} f_i = z \prod_{i=m/2+1}^{m} f_i^{-1}.$$

  This does not seem to be a feasible option, since the operation f_i^{-1} is not well-behaved.
- If m is not prime, the variables x_i could take values in some extension field of $\mathbb{F}_2$. This would reduce the number of variables, but squaring would no longer be linear.

5 Conclusion

We have presented a method to compute the eta pairing which does not require a final exponentiation. It would be very interesting if a similar idea can be applied to ordinary curves.

We have suggested several methods to attack the pairing inversion problem, none of which appear to lead to any practical attack on the system. This adds weight to the belief that pairing inversion is a hard problem.

A more general pairing inversion problem is: given z find points P and Q such that $e(P, Q) = z$. Our results do not shed any light on this problem.

The central computational problem in pairing based cryptography is the bilinear Diffie-Hellman problem. Our results do not say anything about this problem, and further research on it is required.

References

1. P. S. L. M. Barreto, S. D. Galbraith, C. Ó hÉigeartaigh, M. Scott, Efficient pairing computation on supersingular abelian varieties, to appear in Designs, Codes and Cryptography.
2. I. Blake, G. Seroussi and N. P. Smart, Advances in elliptic curve cryptography, Cambridge (2005).
3. The Magma computer algebra system. University of Sydney.
4. I. Duursma and H.-S. Lee", Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$, in C.-S. Laih (ed.), ASIACRYPT 2003, Springer LNCS 2894 (2003) 111–123.
5. G. Frey, H.-G. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, *Math. Comp.*, **62**, No.206 (1994) 865–874.
6. F. Hess, N. P. Smart and F. Vercauteren, The eta pairing revisted, preprint 2006.
7. E. Kaltofen and P. Buergisser, Personal communications, September 5, 2002.
8. A. J. Menezes, T. Okamoto and S. A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Trans. Inf. Theory*, **39**, No. 5 (1993) 1639–1646.
9. D. Page and F. Vercauteren, A fault attack on pairing based cryptography, To appear in IEEE Transactions on Computers 2006.
10. T. Satoh, On polynomial interpolations of homomorphisms from finite fields to elliptic curves, to appear in LMS JCM.
11. E. Verheul, Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, in B. Pfitzmann (ed.), EUROCRYPT 2001, Springer LNCS 2045 (2001) 195–210.
12. E. Verheul, Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, *J. Crypt.*, 17, No. 4 (2004) 277–296.