IMPROVED WEIL AND TATE PAIRINGS FOR ELLIPTIC AND HYPERELLIPTIC CURVES

KIRSTEN EISENTRÄGER, KRISTIN LAUTER, AND PETER L. MONTGOMERY

ABSTRACT. We present algorithms for computing the *squared* Weil and Tate pairings on an elliptic curve and the *squared* Tate pairing for hyperelliptic curves. The squared pairings introduced in this paper have the advantage that our algorithms for evaluating them are deterministic and do not depend on a random choice of points. Our pairings save about 20-30% over the usual pairings.

1. INTRODUCTION

The Weil and Tate pairings have been proposed for use in many aspects of cryptography, including one-round 3-way key establishment, identity-based encryption, and short signatures [7]. For a fixed positive integer m, the Weil pairing e_m is a bilinear map that takes as input two m-torsion points on an elliptic curve, and outputs an mth root of unity in the field. For elliptic curves, the Tate pairing is related to the Weil pairing by the fact that the Weil pairing is a quotient of the output of two applications of the Tate pairing, except that the Tate pairing needs an exponentiation which the Weil pairing omits.

For cryptographic applications, the objective is to have a bilinear map with a specific recipe for efficient evaluation, and no clear way to invert. The Weil and Tate pairings provide such tools, since each has a practical definition which involves finding functions with prescribed zeros and poles on the curve, and evaluating those functions at pairs of points.

For elliptic curves, Miller [8] gave an algorithm for the Weil pairing. (See also the Appendix B to [2], for a probabilistic implementation of Miller's algorithm which recursively generates and evaluates the required functions based on a random choice of points.) For Jacobians of hyperelliptic curves, Frey and Rück [5] gave a recursive algorithm to generate the required functions, assuming the knowledge of intermediate functions having prescribed zeros and poles.

For elliptic curves, we present an improved algorithm for computing the *squared* Weil pairing, $e_m(P,Q)^2$. Our deterministic algorithm does not depend on a random choice of points for evaluation of the pairing. Our algorithm saves about 20% over the standard implementation of the Weil pairing

The research for this paper was done while the first author was visiting Microsoft Research.

[2]. We use the same idea to obtain an improved algorithm for computing the *squared* Tate pairing for elliptic and hyperelliptic curves. The Tate pairing is already more efficient to implement than the Weil pairing. Our new squared Tate pairing is more efficient than the standard implementation of the Tate pairing for elliptic curves, again saving roughly 20%. For the pairings on special families of elliptic curves in characteristics 2 and 3, some implementation improvements were given in [6] and [1]. A different deterministic algorithm was also given in [1].

For hyperelliptic curves, we use Cantor's algorithm to produce the intermediate functions assumed by Frey and Rück. We define a squared Tate pairing for hyperelliptic curves, and use the knowledge of these intermediate functions to implement the pairing and give examples. Our analysis shows that using the squared Tate pairing saves roughly 30% over the standard Tate pairing for genus 2 curves. The squared Weil pairing or the squared Tate pairing can be substituted for the Weil or Tate pairing in many of the above cryptographic applications.

The paper is organized as follows. Section 2 provides background on the Weil pairing for elliptic curves and gives the algorithm for computing the squared Weil pairing. Section 3 does the same for the squared Tate pairing for elliptic curves. Section 4 presents the squared Tate pairing for hyperelliptic curves and shows how to implement it. Section 5 gives an example of the hyperelliptic pairing.

2. Weil pairings for elliptic curves

Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over a finite field $\mathbb{F}_q$, not of characteristic 2 or 3. In the following $\mathbf{O}$ denotes the point at infinity on E. If P is a point on E, then $x(P)$ and $y(P)$ denote the rational functions mapping P to its affine x and y-coordinates.

2.1. **Definition of the Weil pairing.** Let m be a positive integer such that E has two independent points of order m over some field of definition $\mathbb{F}_q$. The definition of the Weil pairing which is used in practice is given in [9, p. 107]. To compute $e_m(P, Q)$, given two distinct m-torsion points P and Q on E, pick two divisors $\mathcal{A}_P$ and $\mathcal{A}_Q$ which are equivalent to $(P) - (\mathbf{O})$ and $(Q) - (\mathbf{O})$, respectively, and such that $\mathcal{A}_P$ and $\mathcal{A}_Q$ have disjoint support. Let $f_{\mathcal{A}_P}$ be a function on E whose divisor of zeros and poles is $(f_{\mathcal{A}_P}) = m \cdot \mathcal{A}_P$. Similarly, let $f_{\mathcal{A}_Q}$ be a function on E whose divisor of zeros and poles is $(f_{\mathcal{A}_Q}) = m \cdot \mathcal{A}_Q$. Then

$$e_m(P, Q) = \frac{f_{\mathcal{A}_P}(\mathcal{A}_Q)}{f_{\mathcal{A}_Q}(\mathcal{A}_P)}.$$

2.2. **Rational functions needed in the evaluation of the pairing.** Fix an integer $m > 0$ and an m-torsion point P on an elliptic curve E. Let $\mathcal{A}_P$ be a divisor equivalent to $(P) - (\mathbf{O})$. For a positive integer j, let $f_{j,\mathcal{A}_P}$ be a

rational function on E with divisor

$$(f_{j,\mathcal{A}_P}) = j\mathcal{A}_P - (jP) + (\mathbf{O})$$

This means that $f_{j,\mathcal{A}_P}$ has j-fold zeros and poles at the points in $\mathcal{A}_P$, as well as a simple pole at jP and a simple zero at $\mathbf{O}$, and no other zeros or poles. Since $mP = \mathbf{O}$, it follows that $f_{m,\mathcal{A}_P}$ has divisor $m\mathcal{A}_P$, so in fact $f_{\mathcal{A}_P} = f_{m,\mathcal{A}_P}$. Throughout the paper the notation $f_{j,P}$ will be used to denote the function $f_{j,\mathcal{A}_P}$ with $\mathcal{A}_P = (P) - (\mathbf{O})$.

Corollary 3.5 on page 67 of [9] asserts that these functions exist. Each $f_{i,\mathcal{A}_P}$ is unique up to a nonzero multiplicative scalar. Miller's algorithm gives an iterative construction of these functions (see for example [1]). The construction of $f_{1,\mathcal{A}_P}$ depends on $\mathcal{A}_P$. Given $f_{i,\mathcal{A}_P}$ and $f_{j,\mathcal{A}_P}$, one constructs $f_{i+j,\mathcal{A}_P}$ as the product

$$f_{i+j,\mathcal{A}_P} = f_{i,\mathcal{A}_P} \cdot f_{j,\mathcal{A}_P} \cdot \frac{g_{iP,jP}}{g_{(i+j)P}}. \tag{1}$$

Here the notation $g_{U,V}$ (two subscripts) denotes the line passing through the points U and V on E. The notation g_U (one subscript) denotes the vertical line through U and $-U$. For more details on efficiently computing $f_{m,\mathcal{A}_P}$, see [4].

2.3. **Squared Weil pairing for elliptic curves.** The purpose of this section is to construct a new pairing, which we call the 'squared Weil pairing', and which has the advantage of being more efficient to compute than Miller's algorithm for the original Weil pairing. Our algorithm also has the advantage that it is guaranteed to output the correct answer and does not depend on inputting a randomly chosen point. In contrast Miller's algorithm may restart, since the randomly chosen point can cause the algorithm to fail.

2.4. **Algorithm for $e_m(P,Q)^2$.** Fix a positive integer m and the curve E. Given two m-torsion points P and Q on E, we want to compute $e_m(P,Q)^2$.

Start with an addition-subtraction chain for m. That is, after an initial 1, every element in the chain is a sum or difference of two earlier elements in the chain, until an m appears. Well-known techniques give a chain of length $O(\log(m))$.

For each j in the addition-subtraction chain, form a tuple

$$t_j = [jP,\, jQ,\, n_j,\, d_j]$$

such that

$$\frac{n_j}{d_j} = \frac{f_{j,P}(Q)\ f_{j,Q}(-P)}{f_{j,P}(-Q)\ f_{j,Q}(P)}. \tag{2}$$

Start with $t_1 = [P,\, Q,\, 1,\, 1]$. Given t_j and t_k, this procedure gets t_{j+k}:

(1) Form the elliptic curve sums $jP + kP = (j+k)P$ and $jQ + kQ = (j+k)Q$.
(2) Find coefficients of the line $g_{jP,kP}(X) = c_0 + c_1 x(X) + c_2 y(X)$.

(3) Find coefficients of the line $g_{jQ,kQ}(X) = c'_0 + c'_1 x(X) + c'_2 y(X)$.
(4) Set

$$n_{j+k} = n_j n_k (c_0 + c_1 x(Q) + c_2 y(Q))(c'_0 + c'_1 x(P) - c'_2 y(P))$$

$$d_{j+k} = d_j d_k (c_0 + c_1 x(Q) - c_2 y(Q))(c'_0 + c'_1 x(P) + c'_2 y(P)).$$

A similar construction gives t_{j-k} from t_j and t_k. The vertical lines through $(j+k)P$ and $(j+k)Q$ do not appear in the formulae for n_{j+k} and d_{j+k}, because the contributions from Q and $-Q$ (or from P and $-P$) are equal.

When $j + k = m$, one can further simplify this to $n_{j+k} = n_j n_k$ and $d_{j+k} = d_j d_k$, since c_2 and c'_2 will be zero.

```
procedure Squared_Weil_Pairing(m, P, Q)
Issue an error if m is not a positive integer.
if (P = O or Q = O or P = ±Q) then
    return 1;
else
    t_1 = [P, Q, 1, 1];
    Use an addition-subtraction chain to get
        t_m = [mP, mQ, n_m, d_m];
    Issue an error if mP or mQ is not O.
    if (n_m = 0 or d_m = 0) then
        return 1;
    else
        return -n_m/d_m;
    end if;
end if;
```

When n_m and d_m are nonzero, then the computation

$$\frac{n_m}{d_m} = \frac{f_{m,P}(Q)\ f_{m,Q}(-P)}{f_{m,P}(-Q)\ f_{m,Q}(P)}.$$

has been successful, and we have the correct output. If, however, n_m or d_m is zero, then some factor such as $c_0 + c_1 x(Q) + c_2 y(Q)$ must have vanished. That line was chosen to pass through jP, kP, and $(-j-k)P$, for some j and k. It does not vanish at any other point on the elliptic curve. Therefore this factor can vanish only if $Q = jP$ or $Q = kP$ or $Q = (-j-k)P$. In all of these cases Q will be a multiple of P, ensuring $e_m(P, Q) = 1$.

2.5. **Correctness proof.**

Theorem 1 (Squared Weil Pairing Formula)**.** *Let m be a positive integer. Suppose P and Q are m-torsion points on E, with neither being the identity and P not equal to $\pm Q$. Then*

$$\frac{f_{m,P}(Q) \cdot f_{m,Q}(-P)}{f_{m,P}(-Q) \cdot f_{m,Q}(P)} = (-1)^m e_m(P, Q)^2,$$

where e_m denotes the Weil-pairing.

Proof. Let R_1, R_2 be two points on E such that the divisors

$$\mathcal{A}_P := (P + R_1) - (R_1) \quad \text{and} \quad \mathcal{A}_Q := (Q + R_2) - (R_2)$$

have disjoint support. Let $\mathcal{A}_{-Q} := (-Q + R_2) - (R_2)$. Let $f_{\mathcal{A}_P}$ and $f_{\mathcal{A}_Q}$ be as above. Then

$$e_m(P, Q) = \frac{f_{\mathcal{A}_P}((Q + R_2) - (R_2))}{f_{\mathcal{A}_Q}((P + R_1) - (R_1))} = \frac{f_{\mathcal{A}_P}(Q + R_2)}{f_{\mathcal{A}_P}(R_2)} \cdot \frac{f_{\mathcal{A}_Q}(R_1)}{f_{\mathcal{A}_Q}(P + R_1)}.$$

If we denote $g(X) = f_{m,P}(X - R_1)$, then

$$(g) = m(P + R_1) - m(R_1) = m\mathcal{A}_P = (f_{\mathcal{A}_P}),$$

This implies $g(X)/f_{\mathcal{A}_P}(X)$ is constant and

$$\frac{f_{\mathcal{A}_P}(Q + R_2)}{f_{\mathcal{A}_P}(R_2)} = \frac{g(Q + R_2)}{g(R_2)} = \frac{f_{m,P}(Q + R_2 - R_1)}{f_{m,P}(R_2 - R_1)}.$$

Similarly

$$\frac{f_{\mathcal{A}_Q}(R_1)}{f_{\mathcal{A}_Q}(P + R_1)} = \frac{f_{m,Q}(R_1 - R_2)}{f_{m,Q}(P + R_1 - R_2)}.$$

Plugging these into Miller's formula gives

$$e_m(P, Q) = \frac{f_{m,P}(Q + R_2 - R_1)}{f_{m,P}(R_2 - R_1)} \; \frac{f_{m,Q}(R_1 - R_2)}{f_{m,Q}(P + R_1 - R_2)}.$$

Using the same argument for $e_m(P, -Q)$ we obtain

$$\begin{aligned} e_m(P, -Q) &= \frac{f_{m,P}(-Q + R_2 - R_1)}{f_{m,P}(R_2 - R_1)} \; \frac{f_{m,-Q}(R_1 - R_2)}{f_{m,-Q}(P + R_1 - R_2)} \\ &= \frac{f_{m,P}(-Q + R_2 - R_1)}{f_{m,P}(R_2 - R_1)} \; \frac{f_{m,Q}(-R_1 + R_2)}{f_{m,Q}(-P - R_1 + R_2)} \end{aligned}$$

Hence we can simplify $e_m(P, Q)^2$ to

$$\begin{aligned} e_m(P, Q)^2 &= \frac{e_m(P, Q)}{e_m(P, -Q)} \\ &= \frac{f_{m,P}(Q + R_2 - R_1)\; f_{m,Q}(R_1 - R_2)\; f_{m,Q}(-P - R_1 + R_2)}{f_{m,P}(-Q + R_2 - R_1)\; f_{m,Q}(-(R_1 - R_2))\; f_{m,Q}(P + R_1 - R_2)}. \end{aligned} \tag{3}$$

Let $R := R_2 - R_1$. Equation (3) becomes

$$e_m(P, Q)^2 = \frac{f_{m,P}(Q + R)\; f_{m,Q}(-R)\; f_{m,Q}(-P + R)}{f_{m,P}(-Q + R)\; f_{m,Q}(R)\; f_{m,Q}(P - R)}.$$

Fix two linearly independent m-torsion points P and Q. Consider the right side as a rational function of R, and call it $\psi = \psi(R)$. Since $f_{m,P}$ only has zeros and poles at P and $\mathbf{O}$ and $f_{m,Q}$ only has zeros and poles at Q and $\mathbf{O}$, this function $\psi(R)$ can only have zeros or poles at $R = -Q, Q, P - Q, P + Q, P$, and $\mathbf{O}$. By looking at the factors of ψ we can check that at each of these points, the value of $\psi(R)$ is well-defined, because the zeros and poles cancel each other out. Since ψ is a rational function on an elliptic curve which does not have any zeros or poles, ψ must be constant. Since

for certain values of R, $\psi(R) = e_m(P,Q)^2$, this must be the case for all values of R. Hence we may in particular choose $R = \mathbf{O}$, or equivalently $R_1 = R_2$. So let $R_1 = R_2$. By Lemma 2 below, $\frac{f_{m,Q}(R_1-R_2)}{f_{m,Q}(-(R_1-R_2))} = (-1)^m$, and by assumption $f_{m,P}$ does not have a zero or pole at Q and $f_{m,Q}$ does not have a zero or pole at P. Hence expression (3) simplifies to

$$e_m(P,Q)^2 = (-1)^m \frac{f_{m,P}(Q)\ f_{m,Q}(-P)}{f_{m,P}(-Q)\ f_{m,Q}(P)}. \tag{4}$$

□

Lemma 2. *Let $f : E \to \mathbb{F}_q$ be a rational function on E with a zero of order m (or a pole of order $-m$) at $\mathbf{O}$. Define $g : E \to \mathbb{F}_q$ by $g(X) = f(X)/f(-X)$. Then $g(\mathbf{O})$ is finite and $g(\mathbf{O}) = (-1)^m$.*

Proof of lemma. The rational function $h(X) = x(X)/y(X)$ has a zero of order 1 at $X = \mathbf{O}$ and satisfies $h(-X) = -h(X)$. The function $f_1 = f/h^m$ has neither a pole nor a zero at $X = \mathbf{O}$, so $f_1(\mathbf{O})$ is finite and nonzero. We can easily check that the rational function $\phi(X) = h(X)/h(-X)$ has no zeros and poles on E. Hence ϕ is constant, and by computing $\phi(X)$ for a finite point $X = (x,y)$ on E with $x, y \neq 0$, we see that ϕ is equal to -1. Hence

$$g(X) = \frac{f(X)}{f(-X)} = \frac{h(X)^m f_1(X)}{h(-X)^m f_1(-X)} = (-1)^m \frac{f(X)}{f(-X)},$$

and $g(\mathbf{O}) = (-1)^m$. □

2.6. **Estimated Savings.** In this section we will compare our algorithm for the squared Weil pairing to Miller's algorithm for the Weil pairing. We count operations in the underlying finite field, counting field squarings as field multiplications throughout.

In practice, some of these arithmetic operations may be over a base field and others over an extension field. This issue is discussed in more detail in [6]. Without knowing the precise context of the application, we don't distinguish these, although individual costs may differ considerably.

Miller's algorithm. Miller's algorithm chooses two points R_1, R_2 on E, and lets $\mathcal{A}_P = (P + R_1) - (R_1)$ and $\mathcal{A}_Q = (P + R_2) - (R_2)$. Recall that in the notation of Section 2.1, $f_{\mathcal{A}_P}$ is a function whose divisor is $m\mathcal{A}_P$. As in Section 2.2, let $f_{j,\mathcal{A}_P}$ be a function whose divisor is

$$(f_{j,\mathcal{A}_P}) = j(P + R_1) - j(R_1) - (jP) + (\mathbf{O}).$$

This is the function f_j in the notation of [2, p. 611f.]. Then $f_{m,\mathcal{A}_P} = f_{\mathcal{A}_P}$. As pointed out in Equation (B.1) of [2, p. 612], (1) leads to the recurrence

$$f_{i+j,\mathcal{A}_P}(\mathcal{A}_Q) = f_{i,\mathcal{A}_P}(\mathcal{A}_Q) \cdot f_{j,\mathcal{A}_P}(\mathcal{A}_Q) \cdot \frac{g_{iP,jP}(\mathcal{A}_Q)}{g_{(i+j)P}(\mathcal{A}_Q)}. \tag{5}$$

During the computations, each $f_{j,\mathcal{A}_P}(\mathcal{A}_Q)$ is a known field element, unlike the unevaluated rational functions $f_{j,\mathcal{A}_P}$. Since $\mathcal{A}_Q$ has degree 0, the

value of $f_{j,\mathcal{A}_P}(\mathcal{A}_Q)$ is unambiguous, whereas $f_{j,\mathcal{A}_P}$ is defined only up to a multiplicative scale factor.

To compute the Weil pairing we need

$$\begin{aligned} e_m(P,Q) &= \frac{f_{\mathcal{A}_P}(Q+R_2)}{f_{\mathcal{A}_P}(R_2)} \frac{f_{\mathcal{A}_Q}(R_1)}{f_{\mathcal{A}_Q}(P+R_1)} \\ &= \frac{f_{m,\mathcal{A}_P}(Q+R_2)}{f_{m,\mathcal{A}_P}(R_2)} \frac{f_{m,\mathcal{A}_Q}(R_1)}{f_{m,\mathcal{A}_Q}(P+R_1)}. \end{aligned}$$

For integers j in an addition-subtraction chain for m, we construct $t_j = [jP, jQ, n_j, d_j]$ where n_j and d_j satisfy

$$\frac{n_j}{d_j} = \frac{f_{j,\mathcal{A}_P}(Q+R_2)}{f_{j,\mathcal{A}_P}(R_2)} \frac{f_{j,\mathcal{A}_Q}(R_1)}{f_{j,\mathcal{A}_Q}(P+R_1)}.$$

To compute t_{i+j} from t_i and t_j, one uses the above recurrence (5) to derive the following expression for n_{i+j}/d_{i+j}:

$$\begin{aligned} \frac{n_{i+j}}{d_{i+j}} = \frac{n_i}{d_i} \cdot \frac{n_j}{d_j} \cdot \frac{g_{iP,jP}(Q+R_2)}{g_{iP,jP}(R_2)} \cdot \frac{g_{(i+j)P}(R_2)}{g_{(i+j)P}(Q+R_2)} \\ \cdot \frac{g_{iQ,jQ}(R_1)}{g_{iQ,jQ}(P+R_1)} \cdot \frac{g_{(i+j)Q}(P+R_1)}{g_{(i+j)Q}(R_1)}. \end{aligned} \tag{6}$$

To evaluate, for example, $g_{iP,jP}(Q+R_2)/g_{iP,jP}(R_2)$, start with the elliptic curve addition $iP + jP = (i+j)P$. This costs one field division and two field multiplications in the generic case where iP and jP have distinct x-coordinates and neither is $\mathbf{O}$. Save the slope λ of the line

$$g_{iP,jP}(X) = y(X) - y(iP) - \lambda(x(X) - x(iP))$$

through iP and jP. Two field multiplications suffice to evaluate

$$g_{iP,jP}(Q+R_2) \text{ and } g_{iP,jP}(R_2).$$

Given $Q+R_2$ and R_2, no more field multiplications or divisions are needed to compute the numerator and denominator of

$$\frac{g_{(i+j)P}(R_2)}{g_{(i+j)P}(Q+R_2)} = \frac{x(R_2) - x((i+j)P)}{x(Q+R_2) - x((i+j)P)}.$$

Repeat this once more to evaluate the last two fractions in (6). Overall these evaluations cost 8 field multiplications and 2 field divisions. We need 10 multiplications to multiply the six fractions, for an overall cost of 18 multiplications and 2 divisions.

Squared pairing. The squared pairing needs n_m/d_m where n_j/d_j is given by (2). The recurrence formula is

$$\frac{n_{i+j}}{d_{i+j}} = \frac{n_i}{d_i} \frac{n_j}{d_j} \frac{g_{iP,jP}(Q)}{g_{iP,jP}(-Q)} \frac{g_{(i+j)P}(-Q)}{g_{(i+j)P}(Q)} \frac{g_{iQ,jQ}(-P)}{g_{iQ,jQ}(P)} \frac{g_{(i+j)Q}(P)}{g_{(i+j)Q}(-P)}. \tag{7}$$

This time the update from $t_i = [iP, iQ, n_i, d_i]$ and t_j to t_{i+j} needs two elliptic curve additions. Each elliptic curve addition needs two multiplications

and one division in the generic case. We can evaluate the numerator and denominator of

$$\frac{g_{iP,jP}(Q)}{g_{iP,jP}(-Q)} = \frac{y(Q) - y(iP) - \lambda(x(Q) - x(iP))}{y(-Q) - y(iP) - \lambda(x(-Q) - x(iP))}$$

with only one multiplication, since $x(Q) = x(-Q)$.

The fraction $g_{(i+j)P}(-Q)/g_{(i+j)P}(Q)$ simplifies to 1 since $g_{(i+j)P}(X)$ depends only on $x(X)$, not $y(X)$. Overall six multiplications and two divisions suffice to evaluate the numerators and denominators of the six fractions in (7). We multiply the four non-unit fractions with six field multiplications.

Overall, the squared Weil pairing advances from t_i and t_j to $i + j$ with 12 field multiplications and 2 field divisions in the generic case, compared to 18 field multiplications and 2 field divisions for Miller's method. When $i = j$, each algorithm needs two additional field multiplications due to the elliptic curve doublings. Estimating one division as five multiplications, this is roughly a 20% savings.

3. Squared Tate pairing for elliptic curves

3.1. **Squared Tate pairing formula.** Let m be a positive integer. Assume that E is defined over $\mathbb{F}_q$, where $q = p^n$ and m divides $q - 1$. Suppose P is an m-torsion point on E over $\mathbb{F}_q$ (notation: $P \in E(\mathbb{F}_q)[m]$), and Q is a point on the curve over $\mathbb{F}_q$, with neither being the identity and P not equal to a multiple of Q. The Tate pairing $\phi_m(P, Q)$ on $E(\mathbb{F}_q)[m] \times E(\mathbb{F}_q)/mE(\mathbb{F}_q)$ is defined as

$$\phi_m(P, Q) := (f_{\mathcal{A}_P}(\mathcal{A}_Q))^{(q-1)/m},$$

with the notation and evaluation as for the Weil pairing above. Now we define

$$v_m(P, Q) := \left(\frac{f_{m,P}(Q)}{f_{m,P}(-Q)}\right)^{(q-1)/m},$$

where $f_{m,P}$ is as above, and call v_m the squared Tate-pairing. To justify this terminology, we will show below that

$$v_m(P, Q) = \phi_m(P, Q)^2.$$

3.2. **Algorithm for $v_m(P, Q)$.** Fix a positive integer m and the curve E. Given an m-torsion point P on E and a point Q on E, we want to compute $v_m(P, Q)$. As before, start with an addition-subtraction chain for m. For each j in the addition-subtraction chain, form a tuple $t_j = [jP, n_j, d_j]$ such that

$$\frac{n_j}{d_j} = \frac{f_{j,P}(Q)}{f_{j,P}(-Q)}.$$

Start with $t_1 = [P, 1, 1]$. Given t_j and t_k, this procedure gets t_{j+k}:

(1) Form the elliptic curve sum $jP + kP = (j + k)P$.
(2) Find the line $g_{jP,kP}(X) = c_0 + c_1x(X) + c_2y(X)$.

(3) Set

$$n_{j+k} = n_j \cdot n_k \cdot (c_0 + c_1 x(Q) + c_2 y(Q)),$$

$$d_{j+k} = d_j \cdot d_k \cdot (c_0 + c_1 x(Q) - c_2 y(Q))$$

A similar construction gives t_{j-k} from t_j and t_k. The vertical lines through $(j+k)P$ and $(j+k)Q$ do not appear in the formulae for n_{j+k} and d_{j+k}, because the contributions from Q and $-Q$ are equal. When $j+k=m$, one can further simplify this to $n_{j+k} = n_j \cdot n_k$ and $d_{j+k} = d_j \cdot d_k$, since c_2 will be zero. When n_m and d_m are nonzero, then the computation

$$\frac{n_m}{d_m} = \frac{f_{m,P}(Q)}{f_{m,P}(-Q)}$$

is successful, and after raising to the $(q-1)/m$ power, we have the correct output. If some n_m or d_m were zero, then some factor such as $c_0 + c_1 x(Q) + c_2 y(Q)$ must have vanished. That line was chosen to pass through jP, kP, and $(-j-k)P$, for some j and k. It does not vanish at any other point on the elliptic curve. Therefore this factor can vanish only if $Q = jP$ or $Q = kP$ or $Q = (-j-k)P$ for some j and k. In all of these cases Q would be a multiple of P, contrary to our assumption.

3.3. **Correctness proof.**

Theorem 3. *Let m be a positive integer. Suppose P is an m-torsion point on E, and $Q \in E$ with neither being the identity and P not equal to $\pm Q$. Then*

$$\left(\frac{f_{m,P}(Q)}{f_{m,P}(-Q)}\right)^{(q-1)/m} = \phi_m(P,Q)^2,$$

where ϕ_m denotes the Tate-pairing.

Proof. Let R_1, R_2 be two points on E such that the divisors

$$(\mathcal{A}_P) = (P + R_1) - (R_1) \quad \text{and} \quad (\mathcal{A}_Q) = (Q + R_2) - (R_2)$$

have disjoint support. By the definition of the Tate pairing and the discussion in the proof of the correctness of the squared Weil pairing, we have

$$\phi_m(P,Q) = \left(\frac{f_{\mathcal{A}_P}(Q+R_2)}{f_{\mathcal{A}_P}(R_2)}\right)^{(q-1)/m} = \left(\frac{f_{m,P}(Q+R_2-R_1)}{f_{m,P}(R_2-R_1)}\right)^{(q-1)/m},$$

and

$$\begin{aligned}\phi_m(P,-Q) &= \left(\frac{f_{\mathcal{A}_P}(-Q+R_2)}{f_{\mathcal{A}_P}(R_2)}\right)^{(q-1)/m} \\ &= \left(\frac{f_{m,P}(-Q+R_2-R_1)}{f_{m,P}(R_2-R_1)}\right)^{(q-1)/m}.\end{aligned}$$

Then

$$\begin{aligned}\phi_m(P,Q)^2 &= \frac{\phi_m(P,Q)}{\phi_m(P,-Q)} \\ &= \left(\frac{f_{m,P}(Q+R_2-R_1)f_{m,P}(R_2-R_1)}{f_{m,P}(R_2-R_1)f_{m,P}(-Q+R_2-R_1)}\right)^{(q-1)/m} \\ &= \left(\frac{f_{m,P}(Q+R_2-R_1)}{f_{m,P}(-Q+R_2-R_1)}\right)^{(q-1)/m}.\end{aligned}$$

By the same argument as in the proof for the Weil pairing we may choose $R_2 = R_1$. This gives us

$$\phi_m(P,Q)^2 = \left(\frac{f_{m,P}(Q)}{f_{m,P}(-Q)}\right)^{(q-1)/m}.$$

□

3.4. **Estimated Savings.** The analysis of the improvement we obtain from the squared Tate pairing is almost identical to the comparison for the Weil pairing given in Section 2.6 above.

When analyzing Miller's algorithm for the Tate pairing, the main difference from Section 2.6 is that the formula analogous to (6) has 2 fewer fractions to evaluate and combine. One elliptic curve addition costs 1 division and 2 multiplications, while 2 multiplications are needed to evaluate the numerators and denominators of the two fractions. Then 6 multiplications are needed to multiply together the numerators and denominators of the 4 fractions. The total cost for each iterative step of Miller's algorithm where an addition is performed is 1 division and 10 multiplications.

For the squared Tate pairing, the formula analogous to (7) also has 2 fewer fractions in it. One elliptic curve addition costs 1 division and 2 multiplications, while only 1 multiplication is needed to evaluate the numerators and denominators of the two fractions. Then 4 multiplications are needed to multiply together the numerators and denominators of the 3 non-unit fractions. The total cost for each iterative step of the squared Tate pairing where an addition is performed is 1 division and 7 multiplications.

Overall, the squared Tate pairing advances from t_i and t_j to $i+j$ with 7 field multiplications and 1 field division in the generic case, compared to 10 field multiplications and 1 field division for Miller's method applied to the usual Tate pairing. When $i = j$, each algorithm needs one additional field multiplication due to the elliptic curve doubling. Estimating one division as five multiplications, this is roughly a 20% savings.

4. Squared Tate pairing for hyperelliptic curves

Let $C : y^2 = f(x)$ be a hyperelliptic curve of genus g over a finite field $\mathbb{F}_q$ not of characteristic 2. For simplicity we assume that the degree of f is odd

so that C has one point P_∞ at infinity. The case where f has even degree can be handled similarly.

4.1. **Notation.** Let $J = J(C)$ be the Jacobian of C, and let $\mathrm{Div}^0(C)$ denote the group of divisors of degree 0 on C. If $P = (x, y)$ is a point on C, then P' will denote the point $P' := (x, -y)$. We will denote the identity element of J by **id**.

The theorem of Riemann and Roch assures that each element D of J contains a representative of the form $A - gP_\infty$, where A is an effective divisor of degree g. In the following we will always work with representatives of this form with the additional property that if a point $P = (x, y)$ occurs in A, then $P' := (x, -y)$ does not occur in A. The effective divisor in the representative for the identity **id** will be gP_∞. For an element D of J, a representative for iD will be $A_i - gP_\infty$, where A_i is effective of degree g with the additional property above.

To a representative $A_i - gP_\infty$ we can associate two polynomials (a_i, b_i) which represent the divisor. The first polynomial, $a_i(x)$, is monic and has as its zeros the x-coordinates of the points in the support of the divisor A_i. The second polynomial, $b_i(x)$, has degree less than the degree of $a_i(x)$, and has the property that its graph passes through the finite points in the support of the divisor A_i.

4.2. **Definition of the Tate Pairing.** Fix a positive integer m and assume that $\mathbb{F}_q$ contains an mth root of unity ζ_m. There exists a pairing ϕ_m:

$$J(\mathbb{F}_q)[m] \times J(\mathbb{F}_q)/mJ(\mathbb{F}_q) \to \mathbb{F}_q^*/\mathbb{F}_q^{*m} \cong < \zeta_m >$$

with the following explicit description: Let $D \in J(\mathbb{F}_q)[m]$, and $E \in J(\mathbb{F}_q)$. Now E is a divisor on the curve C, not an elliptic curve, and we assume that the support of E does not contain P_∞ and that E is prime to the A_i's which were defined above. Actually E only needs to be prime to those representatives which will be used in the addition-subtraction chain for m, so to about $\log m$ divisors. Let $h_{m,D}$ be a function on C such that the divisor of $h_{m,D}$ satisfies $(h_{m,D}) = mD$. Then

$$\phi_m(D, E) := h_{m,D}(E)^{\frac{q-1}{m}} \in < \zeta_m >$$

Remark: The value $h_{m,D}(E)$ is defined only up to mth powers. Hence we raise the result to the power $\frac{q-1}{m}$ to eliminate all mth powers.

This pairing is called the Tate pairing, and it is known to be well-defined, bilinear, and non-degenerate [5, p. 871].

4.3. **Evaluation of the Tate Pairing.** In [5, pp. 872–873] it is shown how to evaluate the Tate pairing on the Jacobian of a curve assuming an explicit reduction algorithm for divisors on a curve. For hyperelliptic curves, such an algorithm can be found in Cantor [3]. In Section 4.6 below, we will use Cantor's algorithm to explicitly compute the necessary intermediate

functions. They will be used to evaluate the *squared* Tate pairing, but they could just as well be used to evaluate the usual Tate pairing.

4.4. **Functions needed in the evaluation of the pairings.** Let D be an m-torsion element of J. For any integer j, let $h_{j,D}$ denote a rational function on C with divisor

$$(h_{j,D}) = jA_1 - A_j - (j-1)gP_\infty.$$

Since D is an m-torsion element, we have that $A_m = gP_\infty$, so the divisor of $h_{m,D}$ is $(h_{m,D}) = mA_1 - m \cdot gP_\infty$. For each j, $h_{j,D}$ is well-defined up to a multiplicative constant.

Using Cantor's algorithm we can, given positive divisors of degree g, as well as A_i and A_j, find a positive divisor A_{i+j} of degree g and determine a function $u_{i,j}$ such that the divisor of $u_{i,j}$ is equal to

$$(u_{i,j}) = A_i + A_j - A_{i+j} - gP_\infty.$$

Now we can show how to iteratively construct $h_{j,D}(E)$. For $j = 1$, let $h_{1,D}$ be 1. Suppose we have A_i, A_j, $h_{i,D}(E)$ and $h_{j,D}(E)$. Let $u_{i,j}$ be the above function on C. Then

$$h_{i+j,D}(E) = h_{i,D}(E) \cdot h_{j,D}(E) \cdot u_{i,j}(E).$$

4.5. **Squared Tate pairing v_m for hyperelliptic curves.** Given an m-torsion element D of J and an element E of J, with representatives

$$D = (P_1) + (P_2) + \cdots + (P_g) - (gP_\infty)$$

and

$$E = (Q_1) + (Q_2) + \cdots + (Q_g) - (gP_\infty),$$

respectively, with P_i not equal to Q_j or Q'_j for all i, j define

$$v_m(D,E) := \left(h_{m,D}(Q_1 - Q'_1 + Q_2 - Q'_2 + \cdots + Q_g - Q'_g)\right)^{(q-1)/m}.$$

Then it will be shown below that

$$v_m(D,E) = \pm\phi_m(D,E)^2,$$

where $\phi_m(D,E)$ is the Tate pairing on $J[m] \times J/mJ$.

4.6. **Algorithm to compute $v_m(D,E)$.** Assume for simplicity that $g = 2$. Let D and E be as above. Form an addition-subtraction chain for m. For each j in the addition-subtraction chain we need to form a tuple $t_j = [A_j, n_j, d_j]$ such that jD has representative $A_j - 2P_\infty$ and

$$\frac{n_j}{d_j} = \frac{h_{j,D}(Q_1)\ h_{j,D}(Q_2)}{h_{j,D}(Q'_1)\ h_{j,D}(Q'_2)}.$$

Start with $t_1 = [A_1, 1, 1]$. Given t_i and t_j, let (a_i, b_i) and (a_j, b_j) be the polynomials corresponding to the divisors A_i and A_j. Do a composition step as, for example, in Cantor's algorithm to obtain $(a_{\text{new}}, b_{\text{new}})$ corresponding to $A_i + A_j$ without performing the reduction step. Let $d(x)$ be the greatest common divisor of the three polynomials $(a_i(x), a_j(x), b_i(x) + b_j(x))$ as in

Cantor. The polynomial $d(x)$ depends on i and j, but we will omit the subscripts here for ease of notation. If $d(x) = 1$, then $a_{\text{new}}(x)$ is just the product of $a_i(x)$ and $a_j(x)$, and $b_{\text{new}}(x)$ is the cubic polynomial passing through the four distinct finite points in the support of A_i and A_j. The output polynomials satisfy

$$b_{\text{new}}(x)^2 \equiv f(x) \pmod{a_{\text{new}}(x)}.$$

Case i. If the degree of a_{new} is greater than 2, Cantor's algorithm performs a reduction step. In this case, we can let

$$u_{i,j}(P) := \frac{a_{\text{new}}(x(P))}{b_{\text{new}}(x(P)) + y(P)} \cdot d(x(P)).$$

Then $(u_{i,j}) = A_i + A_j - A_{i+j} - 2P_\infty$, and

$$\begin{aligned} \frac{u_{i,j}(P)}{u_{i,j}(P')} &= \frac{a_{\text{new}}(x(P))}{a_{\text{new}}(x(P'))} \cdot \frac{b_{\text{new}}(x(P')) + y(P')}{b_{\text{new}}(x(P)) + y(P)} \cdot \frac{d(x(P))}{d(x(P'))} \\ &= \frac{b_{\text{new}}(x(P')) + y(P')}{b_{\text{new}}(x(P)) + y(P)}. \end{aligned}$$

Let

$$n_{i+j} := n_i \cdot n_j \cdot (b_{\text{new}} + y)(Q_1') \cdot (b_{\text{new}} + y)(Q_2'), \tag{8}$$

and

$$d_{i+j} := d_i \cdot d_j \cdot (b_{\text{new}} + y)(Q_1) \cdot (b_{\text{new}} + y)(Q_2). \tag{9}$$

There is no contribution from a_{new} in n_{i+j} and d_{i+j} because the contributions from Q_i and Q_i' are equal. This is an improvement over the algorithm for the Tate pairing from [5].

Case ii. If on the other hand the degree of a_{new} is less than or equal to 2, then one can let $u_{i,j}(P) = d(x(P))$.

Note that if we evaluate at intermediate steps then it is not enough to assume that the divisors D and E are coprime. Instead, E must also be coprime to A_i for all i which occur in the addition chain for m. One way to ensure this condition is to require that E and D be linearly independent and that the polynomial $a(x)$ in the pair $(a(x), b(x))$ representing E be irreducible. There are other ways possible to achieve this, like changing the addition chain for m.

4.7. **Correctness proof.** Assume for simplicity of notation now that the genus of C is 2. Let $D \in J(\mathbb{F}_q)[m]$ and $E \in J(\mathbb{F}_q)$, where

$$D = P_1 + P_2 - 2P_\infty$$

and

$$E = Q_1 + Q_2 - 2P_\infty.$$

Recall that if P_1 is a point on C, say $P_1 = (x, y)$, then P_1' is the point $(x, -y)$. Similarly, if $D = P_1 + P_2 - 2P_\infty$, let $D' = P_1' + P_2' - 2P_\infty$. For the proof, we will compute $\phi_m(2D, 2E)$.

First observe that

$$Q_1 - Q_1' + Q_2 - Q_2' \sim 2E$$

in the Jacobian of C. Let $h_{m,D}$ denote the rational function on C with divisor

$$(h_{m,D}) = mP_1 + mP_2 - 2mP_\infty$$

as above. Then the divisor of $h_{m,D}/h_{m,D'}$ is of the form

$$\left(\frac{h_{m,D}}{h_{m,D'}}\right) = mP_1 - mP_1' + mP_2 - mP_2',$$

so

$$(h_{m,D}/h_{m,D'}) \sim 2mD$$

in the Jacobian. That means we can use $h_{m,D}/h_{m,D'}$ to compute the pairing $\phi_m(2D, 2E)$. If Q is any point on C, then it is easy to see that $h_{m,D}(Q) = c \cdot h_{m,D'}(Q')$, where c is a constant which does not depend on Q. We can see that by comparing the divisors of the two functions.

Hence

$$\begin{aligned}
\phi_m(2D,\, 2E) &= \left(\frac{h_{m,D}(Q_1 - Q_1' + Q_2 - Q_2')}{h_{m,D'}(Q_1 - Q_1' + Q_2 - Q_2')}\right)^{(q-1)/m} \\
&= \left(\frac{h_{m,D}(Q_1 - Q_1' + Q_2 - Q_2')}{h_{m,D}(Q_1' - Q_1 + Q_2' - Q_2)}\right)^{(q-1)/m} \\
&= \left(h_{m,D}(Q_1 - Q_1' + Q_2 - Q_2')^2\right)^{(q-1)/m}
\end{aligned}$$

Since $\phi_m(2D,\, 2E) = \phi_m(D,\, E)^4$, it follows that

$$\phi_m(P,\, Q)^2 = \pm(h_{m,D}(Q_1 - Q_1' + Q_2 - Q_2'))^{(q-1)/m}.$$

4.8. **Estimated Savings.** According to a straightforward implementation of Cantor's algorithm, the total costs for doubling and addition on the Jacobian of a hyperelliptic curve of genus 2 in odd characteristic, $C : y^2 = f(x)$, where f has degree 5, are as follows: doubling an element costs 34 multiplications and 2 inversions; adding two distinct elements of J costs 26 multiplications and 2 inversions. More efficient implementations of the group law may alter the total impact of our algorithm. Different field multiplication/inversion ratios and field sizes, as well as differing costs in an extension field will also affect the analysis, but these costs are chosen as representative for the purpose of estimating the savings.

4.8.1. *Analysis of standard algorithm.* The standard algorithm described by Frey and Rück [5] proceeds as follows: Let $D := P_1 + P_2 - 2P_\infty$, and let R_1, R_2, R_3, R_4 be four points on C such that

$$Q_1 + Q_2 - 2P_\infty \sim R_1 + R_2 - R_3 - R_4$$

in J. We keep the notation from before. The algorithm computes t_{i+j} from t_i and t_j, where $t_i = [A_i, n_j, d_j]$ and

$$\frac{n_j}{d_j} = \frac{h_{j,D}(R_1)\ h_{j,D}(R_2)}{h_{j,D}(R_3)\ h_{j,D}(R_4)}.$$

The expression for n_{i+j}/d_{i+j} becomes

$$\frac{n_{i+j}}{d_{i+j}} = \frac{n_i}{d_i}\ \frac{n_j}{d_j}\ \frac{u_{i,j}(R_1)\ u_{i,j}(R_2)}{u_{i,j}(R_3)\ u_{i,j}(R_4)}\ .$$

To form $u_{i,j}$, we have to perform an addition or doubling step to obtain A_{i+j} from A_i and A_j. This costs 34 multiplications and 2 inversions for a doubling, 26 multiplications and 2 inversions for an addition. Then

$$u_{i,j}(P) = \frac{a_{\text{new}}(x(P))}{b_{\text{new}}(x(P)) + y(P)},$$

and to compute (n_{i+j}, d_{i+j}), we need to evaluate $u_{i,j}$ at four different points. Each evaluation of $a_{\text{new}}(x(P))$ costs 2 multiplications in a doubling step, 3 multiplications in an addition step. Evaluation of $b_{\text{new}}(x(P))$ costs 3 multiplications. Finally we multiply the partial numerators and denominators out, using 5 multiplications each, including the multiplications with n_i, n_j, d_i, and d_j. So the total cost for an addition step is 60 multiplications and 2 inversions, and the total cost for a doubling is 64 multiplications and 2 inversions.

4.8.2. *Squared Tate Pairing.* The squared Tate pairing works with the divisor $Q_1 - Q_1' + Q_2 - Q_2' \sim 2Q_1 - 2Q_2 - 4P_\infty$. After adding A_i and A_j to obtain A_{i+j} as above, we need to form

$$\frac{n_{i+j}}{d_{i+j}} = \frac{n_i}{d_i}\ \frac{n_j}{d_j}\ \frac{u_{i,j}(Q_1)\ u_{i,j}(Q_1')}{u_{i,j}(Q_2)\ u_{i,j}(Q_2')}\ .$$

As can be seen from (8) and (9) above, no evaluations of $a_{\text{new}}(x(P))$ are needed. For $i = 1, 2$, we need to evaluate $b_{\text{new}}(x(Q_i))$ and $b_{\text{new}}(x(Q_i'))$. This costs only 3 multiplications for each i, since the x-coordinates of Q_i and Q_i' are the same. Finally, we have to multiply the partial numerators and denominators, for a total cost of 12 multiplications for either a doubling or an addition.

So the total cost for an addition step is 38 multiplications and 2 inversions, and the total cost for a doubling is 46 multiplications and 2 inversions. Estimating one inversion as 4 multiplications this is a 25% improvement in the doubling case and a 33% improvement in the addition case.

5. Examples

5.1. $g = 2$, $p = 31$, $m = 5$. In this section, we evaluate the squared Tate pairing for 5-torsion on the Jacobian of a genus 2 hyperelliptic curve over a field of 31 elements.

Take the hyperelliptic curve C defined by the affine model $y^2 = f(x)$ where

$$f(x) = x^5 + 13x^4 + 2x^3 + 4x^2 + 11x + 1.$$

The group of points on the Jacobian of C over $\mathbb{F}_{31}$ has order $N = 1040$. Let D be the 5-torsion element of the Jacobian of C given by the pair of polynomials

$$D = [x^2 + 23x + 15, 13x + 28].$$

Let E be the element of the Jacobian of C of order 260 given by the pair

$$E = [x^2 + 4x + 2, 29x + 20].$$

Then the squared Tate pairing evaluated at D and E is

$$v_5(D, E) = 4,$$

where

$$h_{5,D} = \frac{(x+26)^2(x^4 + 19x^3 + 23x^2 + 16x + 19)(x^2 + 23x + 15)}{x^3 + 6x^2 + 9x + 21 + y}.$$

To see the bilinearity of this pairing, look for example at

$$2D = [x^2 + 25x + 9, 10x + 6],$$

$$3D = [x^2 + 25x + 9, 21x + 25],$$

and

$$2E = [x^2 + x + 3, 26x + 3].$$

Then we compute that indeed

$$v_5(2D, E) = 16 = v_5(D, E)^2,$$

with

$$h_{5,2D} = \frac{(x+26)(x^4 + 19x^3 + 23x^2 + 16x + 19)^2(x^2 + 25x + 9)}{(x^3 + 6x^2 + 9x + 21 + y)^2},$$

and

$$v_5(D, 2E) = 16 = v_5(D, E)^2,$$

with $h_{5,D}$ as above. Also

$$v_5(3D, E) = 2 = v_5(D, E)^3 \pmod{31},$$

with

$$h_{5,3D} = \frac{(x+26)(x^4 + 19x^3 + 23x^2 + 16x + 19)^2(x^2 + 25x + 9)}{(30x^3 + 25x^2 + 22x + 10 + y)^2}.$$

References

[1] Paulo S.L.M. Barreto, Hae Y. Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In M. Yung, editor, *Advances in Cryptology – Crypto 2002*, pages 354–368. LNCS **2442**, Springer-Verlag, 2002.

[2] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615 (electronic), 2003.

[3] David G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177):95–101, 1987.

[4] Kirsten Eisenträger, Kristin Lauter, and Peter L. Montgomery. Fast elliptic curve arithmetic and improved Weil pairing evaluation. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, pages 343–354. LNCS **2612**, Springer-Verlag, 2003.

[5] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, 1994.

[6] Steven Galbraith, Keith Harrison, and David Soldera. Implementing the Tate Pairing. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory, 5th International Symposium ANTS-V, Sydney, Australia, July 7-12, 2002*, pages 324–337. LNCS **2369**, Springer-Verlag, 2002.

[7] Antoine Joux. The Weil and Tate pairings as building blocks for public key cryptosystems (survey). In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory, 5th International Symposium ANTS-V, Sydney, Australia, July 7-12, 2002*, pages 20–32. LNCS **2369**, Springer-Verlag, 2002.

[8] Victor S. Miller. Short programs for functions on curves. Unpublished manuscript, 1986.

[9] Joseph Silverman. *The Arithmetic of Elliptic Curves*. GTM **106**, Springer-Verlag, 1986.

School of Mathematics, Institute for Advanced Study, Einstein Drive, Princeton, NJ 08540
E-mail address: eisentra@ias.edu

Microsoft Research, One Microsoft Way, Redmond, WA 98052
E-mail address: klauter@microsoft.com

Microsoft Research, 780 Las Colindas Road, San Rafael, CA 94903–2346
E-mail address: petmon@microsoft.com