

Implementation of the η_T pairing

Colm Ó hÉigearthaigh

<http://www.computing.dcu.ie/~coheigearthaigh>

`coheigearthaigh@computing.dcu.ie`

School of Computing,
Dublin City University

Table of Contents

Background:

- Introduction
- The Curve/Arithmetic
- Octupling Operation
- Functions/Implementation

Pairings:

- BKLS using Octupling
- The η and η_T pairings
- Results
- Conclusion

Introduction

- In [1], we introduce a new and efficient bilinear pairing for supersingular abelian varieties called the η_T pairing.
- In this talk, we focus on the efficient implementation of the genus 2 case.
- Key ingredients for a fast pairing are using degenerate divisors and taking advantage of a speedy octupling operation.
- This work has been done with Paulo S.L.M. Barreto (USP), Steven Galbraith (RHUL) and Michael Scott (DCU).

The Curve

- We consider the computation of the tate pairing over the supersingular genus 2 curve
 $C_d : y^2 + y = x^5 + x^3 + d, d \in \mathbb{F}_2$, over the field \mathbb{F}_{2^m} , with m coprime to 6.
- This curve has embedding degree $k = 12$.
- The explicit formulae for the group law given by Lange are not optimal for this curve.
- Here we give the order of the Jacobian over \mathbb{F}_{2^m} :

$\#\text{Jac}(C_d)(\mathbb{F}_{2^m})$	condition
$2^{2m} + (-1)^d 2^{(3m+1)/2} + 2^m + (-1)^d 2^{(m+1)/2} + 1$	$m \equiv 1, 7, 17, 23 \pmod{24}$
$2^{2m} - (-1)^d 2^{(3m+1)/2} + 2^m - (-1)^d 2^{(m+1)/2} + 1$	$m \equiv 5, 11, 13, 19 \pmod{24}$

Arithmetic on the curves

- We need a distortion map ψ to map elements from \mathbb{F}_{2^m} to the extension field $\mathbb{F}_{2^{12m}}$.
- Choose $w \in \mathbb{F}_{2^6}$ to be a root of the polynomial $x^6 + x^5 + x^3 + x^3 + 1$
- Define $s_1 = w^2 + w^4$, $s_2 = w^4 + 1$ and let $s_0 \in \mathbb{F}_{2^{12}}$ be a solution of $s_0^2 + s_0 = w^5 + w^3$.
- $\mathbb{F}_{2^{12m}}$ representation:
 $\{1, w, w^2, w^3, w^4, w^5, s_0, ws_0, w^2s_0, w^3s_0, w^4s_0, w^5s_0\}$.
- Distortion map: $\psi(x, y) = (x + w, y + s_2x^2 + s_1x + s_0)$.
- If $(x, y) \in E(\mathbb{F}_{2^6})$ then the x-coordinate of $\psi(x, y)$ is also in \mathbb{F}_{2^6} .

Octupling operation

- If $D = (P) - (\infty)$, then jD is generally not equivalent to a divisor of the form $(Q) - (\infty)$.
- However, we give an octupling formula $8D = (P') - (\infty)$, where
$$P' = \phi\pi^6(P) = (x^{2^6} + 1, (y + x^2 + 1)^{2^6}).$$
- Therefore, we can *octuple* a general divisor extremely quickly, using at worst 24 field squarings.
- We will consider the η_T pairing to the base 8 rather than a binary basis (by working with $q = 2^{3m}$).

Functions

- We show in [1] that for $\psi(Q) = (x, y)$;

$$f_{8,P}(\psi(Q)) = \frac{(y + b_4(x))^2(y + b_8''(x))}{a_4'(x)^2 a_8'(x)}$$

where

$$b_4(x) = x^3 + (x_P^8 + x_P^4)x^2 + (x_P^4)x + y_P^4$$

$$b_8''(x) = (x_P^{32} + 1)x^2 + (x_P^{32} + x_P^{16})x + (y_P^{16} + x_P^{16} + x_P^{48} + 1).$$

- The denominator $a_4'(x)^2 a_8'(x)$ can be ignored due to the form of our distortion map.
- We can write $f_{8,P} = \alpha\beta$ where $\alpha = (y + b_4(x))^2 \circ \psi$ and $\beta = (y + b_8'') \circ \psi$.

Implementation

- We need to calculate $f_{8,P}$ each iteration of Miller's algorithm. Some obvious optimisations are available;
 1. Precompute powers of P that are needed, and then just use array indexing in the actual for loop. This involves computing $\pi^i(x_P) = x_P^{2^i}$ and $\pi^i(y_P) = y_P^{2^i}$ for $i = 0, 1, \dots, m - 1$.
 2. Absorb powers of 8 into formulae. This involves precomputing powers of Q as well.
 3. Unroll multiplication using Karatsuba multiplication.

BKLS using Octupling

- The group order $N \approx 2^{2m}$ requires about $2m/3$ octuplings.
- However, as the additions will destroy the special form of the divisor, the best approach is to postpone the additions until the end.
- So, compute the functions for $2^{2m}P$, $2^{(3m+1)/2}P$, 2^mP and $2^{(m+1)/2}P$ separately in different loops, and multiply them together at the end, along with the functions that arise from Cantor composition and reduction of the divisors at $2^{2m}P$, $2^{(3m+1)/2}P$, etc.

BKLS using Octupling (2)

- Absorbing powers of 8 as earlier is a bit more tricky...
- As we're raising to four different powers in the loops, we require four different sets of formulae.
- This leaves us with a lot more than $2m/3$ octuplings, however relations can be exploited between the functions to keep the number of octuplings at $2m/3$.
- The end result is an efficient if messy pairing computation.

The η pairing

- The η pairing allows us to generalise the Duursma-Lee idea.
- Taking advantage of the octupling operation, we define the η pairing in the genus 2 case we have been working with as:

$$\eta_T(P, Q) = \prod_{i=0}^{m-1} f_{8, [8^i]P}(\psi(Q))^{8^{m-1-i}}$$

- So we have a loop of m iterations, as opposed to $2m/3$ loop iterations for BKLS. However the extra work for BKLS makes both schemes roughly equivalent.

The η_T pairing

- For $N = 2^{2m} \pm 2^{(3m+1)/2} + 2^m \pm 2^{(m+1)/2} + 1$, notice that:

$$(2^m \mp 2^{(m+1)/2} + 1)N = 2^{3m} \pm 2^{(3m+1)/2} + 1$$

- Therefore if D is a divisor defined over \mathbb{F}_q , we take $T = \mp 2^{(3m+1)/2} - 1$:

$$[T]D = [q - (2^m \mp 2^{(m+1)/2} + 1)N]D = [q]D = \gamma(D)$$

- The genus 2 η_T pairing is defined as:

$$\left(\eta_T(D, D')^M\right)^{2T} = \left(\langle D, \psi(D') \rangle_N^M\right)^L.$$

where $M = (2^{12m} - 1)/N$ and $L = 2^{m+1} \mp 2^{(m+3)/2} + 2$.

The η_T pairing (2)

- Taking advantage of the fast octupling operation, we can write $T = \mp 2^{(3m+1)/2} - 1$ as $(m - 1)/2$ octuplings, two doublings and an addition.
- This is clearly superior to the BKLS method ($2m/3$ (+) octuplings) or the straightforward η pairing (m octuplings).

The Final Addition

- Recall that the order we're working with is:

$$T = \mp 2^{(3m+1)/2} - 1.$$

- If both divisors are of the form $D_i = (P_i) - (\infty)$, we can skip the final addition.
- $(2^{(3m+1)/2} + 1)D_1 = \phi(D_1)$ where $\phi(D_1)$ has only one point on its support.
- Therefore the points $(-P)$ and (P) cancel on the final addition in the composition stage of Cantor's algorithm, and so the final addition has no impact upon the function.
- As we can skip the final addition, we can also simplify the final two doublings.

The Final Exponentiation

- One can compute a bilinear pairing by computing the η_T pairing and raising to the power of M . However, it is actually more efficient to compute the full tate pairing with $\eta_T^{\frac{M2T}{L}}$.
- M factors as
$$M = (2^{6m} - 1)(2^m \mp 2^{(m+1)/2} + 1)(2^{3m} \mp 2^{(3m+1)/2} + 1)$$
- Note that L can be written as $L = 2(2^m \mp 2^{(m+1)/2} + 1)$, which cancels out with the middle factor of M and the squaring of the η_T function.
- When unrolled, the final exponentiation can be computed in $(m + 1)/2$ squarings, 15 Frobenius actions, 3 multiplications and a division.

Results

case	curve	optimisation	pairing time (ms)
1	$E(\mathbb{F}_{2^{307}})$	elliptic char 2 η_T	≈ 2.9
3	$C(\mathbb{F}_{2^{103}})$	genus 2 η_T	1.96
4	$C(\mathbb{F}_{2^{103}})$	genus 2 general η_T	6.52
5	$C(\mathbb{F}_{2^{103}})$	genus 2 BKLS	3.1

- 1230-bit finite field with (g=1) 2^{307} and (g=2) 2^{103} .
- Timings done on a 3ghz P4, C++ using miracl.
- We make use of 128-bit SSE2 registers for field arithmetic.

Conclusion

References

- [1] Paulo S. L. M. Barreto, Steven Galbraith, Colm O hEigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. Cryptology ePrint Archive, Report 2004/375, 2004. Available from <http://eprint.iacr.org/2004/375>.

- Thanks for listening!
- Questions?