Faster Pairings on Special Weierstrass Curves

Craig Costello

craig.costello@qut.edu.au
Queensland University of Technology

Pairing 2009

Joint work with Huseyin Hisil, Colin Boyd, Juanma Gonzalez-Nieto, Kenneth Koon-Ho Wong

Table of contents

The evolution of faster pairings: 3 bags of tricks

1. **Tricks "inside" the Miller iterations**
   - optimal group choices $\longrightarrow$ avoiding irrelevant operations - denominator elimination
   - avoiding costly inversions - homogenization
   - minimize additions - low Hamming weight loop parameter
   - operations over smaller fields - employ twisted curve

Goal 1

Minimize the number (cost) of field operations throughout each Miller iteration

The evolution of faster pairings: 3 bags of tricks

2. Pairing-friendly curves

- An array of constructions (FST - taxonomy)
- For a 'small' k, we want group size r, field size q, trace t, $(n = \#E = q + 1 - t)$
- Not-in-family, 'individual' curve constructions (Cocks-Pinch, DEM, supersingular curves, etc)
- Families of curves (MNT, GMV, Freeman, cyclotomic families, Scott-Barreto families, KSS curves, BN curves, etc)
- Pairing-friendly fields

Goal 2

$$\rho = \log q / \log r \text{ close to } 1$$

The evolution of faster pairings: 3 bags of tricks

3. Loop shortening techniques

- Exploiting efficiently computable endomorphisms on CM (complex multiplication) curves e.g. Scott's NSS curves
- η_T-pairing
- ate pairing
- ate pairing variants (optimized ate pairing, ate_i pairings, R-ate pairing)

Goal 3

Minimize the loop length
(Vercauteren's conjecture $\approx \log_2(r)/\varphi(k)$)

Where does this work fit in?

We work on a special j-invariant zero (CM discriminant $D = 3$) curve

1: Minimize the number of field operations throughout each Miller iteration

This curve allows new faster formulas in the Miller loop that reduce the operation count throughout each iteration

2: Low embedding degree k and $\rho = \log q / \log r$ close to 1

For the majority of embedding degrees $k \leq 50$, this curve can be constructed with the best (currently known) ρ-value

3: Minimize the loop length ($\approx \log_2(r)/\varphi(k)$)

... more on this later

Computations in a Miller iteration

- Doubling stage
  i. Double: $R \leftarrow [2]R$
  ii. Compute lines l and v for doubling $R = (x_R, y_R)$
  iii. $f \leftarrow f^2 \cdot l(Q)/v(Q)$

- Addition stage (if necessary)
  i. Add: $R \leftarrow R + P$
  ii. Compute lines l and v for adding $R = (x_R, y_R)$ and $P = (x_P, y_P)$
  iii. $f \leftarrow f \cdot l(Q)/v(Q)$

Attractive doublings: a good place to start

- Standard doubling of $[2](x_1, y_1) = (x_3, y_3)$ on $y^2 = x^3 + ax + b$

$$x_3 = \lambda^2 - 2x_1, \qquad y_3 = \lambda(x_1 - x_3) - y_1$$

  with $\lambda = (3x_1^2 + a)/(2y_1)$.

- Let a function $f = g/h$. Define $deg_{TOTAL}(f) = deg(g) + deg(h)$.
- **Key observation:** curve constant b is a square in $\mathbb{F}_q$, ($b = c^2$, $c \in \mathbb{F}_q$), we can write

$$x_3 = x_1(\mu - \mu^2) + a\sigma, \qquad y_3 = (y_1 - c)\mu^3 + a\delta - c$$

  with

$$\mu = (y_1 + 3c)/(2y_1), \qquad \sigma = (a - 3x_1^2)/(2y_1)^2$$
$$\delta = (3x_1(y_1 - 3c)(y_1 + 3c) - a(9x_1^2 + a))/(2y_1)^3$$

- At first glance latter formulas look worse... but total degrees less (Monaghan/Pearce simplification algorithm)

The special j-invariant zero curve

- Doubling of $[2](x_1, y_1) = (x_3, y_3)$ on $y^2 = x^3 + c^2$ simplifies to

$$\begin{aligned} \mu &= (y_1 + 3c)/(2y_1) \\ x_3 &= x_1(\mu - \mu^2) \\ y_3 &= (y_1 - c)\mu^3 - c \end{aligned}$$

- The curve $v^2 = u^3 + c^2$ is isomorphic over $\mathbb{F}_q$ to $y^2 = cx^3 + 1$ with the isomorphism $\sigma : (x, y) \mapsto (u, v) = (cx, cy)$ and $\sigma(\mathcal{O}) \mapsto \mathcal{O}$.
- Affine doubling on $y^2 = cx^3 + 1$

$$\begin{aligned} \mu &= (y_1 + 3)/(2y_1) \\ x_3 &= x_1(\mu - \mu^2) \\ y_3 &= (y_1 - 1)\mu^3 - 1 \end{aligned}$$

- Affine (almost schoolbook) addition on $y^2 = cx^3 + 1$

$$\begin{aligned} \mu &= (y_1 - y_2)/(x_1 - x_2) \\ x_3 &= c^{-1}\lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned}$$

The affine Miller lines

- Tate pairing $e(P, Q)$: $P \in E(\mathbb{F}_q)$, $x_Q \in \mathbb{F}_{q^d}$ (proper subfield)
- Only *factors* we need to carry through contain $y_Q \in \mathbb{F}_{q^k}$
- Addition line

$$g_{add} = \frac{l_{add}(Q)}{v_{add}(Q)} = c\,\frac{\lambda(x_2 - x_Q) - y_2 + y_Q}{c(x_1 + x_2 + x_Q) - \lambda^2}$$

becomes

$$g'_{add} = (y_1 - y_2)(x_2 - x_Q) - (x_1 - x_2)(y_2 - y_Q)$$

- Doubling line

$$g_{dbl} = \frac{l_{dbl}(Q)}{v_{dbl}(Q)} = \frac{2cy_1(x_1 - x_Q)^2}{x_1^2(3cx_Q) - y_1^2 + 3 + 2y_1 y_Q}$$

becomes

$$g'_{dbl} = x_1^2(3cx_Q) - y_1^2 + 3 - 2y_1 y_Q$$

Homogeneous projective coordinates

- Represent (x, y) on the curve $y^2 = cx^3 + 1$ as $(X : Y : Z)$ on $Y^2Z = cX^3 + Z^3$ where $(x, y) = (X/Z, Y/Z)$.
- Doubling $[2](X_1 : Y_1 : Z_1) = (X_3 : Y_3 : Z_3)$ gives

$$\begin{aligned} X_3 &= 2X_1Y_1(Y_1^2 - 9Z_1^2) \\ Y_3 &= (Y_1 - Z_1)(Y_1 + 3Z_1)^3 - 8Y_1^3Z_1 \\ Z_3 &= 8Y_1^3Z_1 \end{aligned}$$

with line equation

$$g''_{dbl} = X_1^2(3cx_Q) - Y_1^2 + 3Z_1^2 - 2Y_1Z_1y_Q$$

- Point doubling here costs 4**m**+3**s**
- Line computation only costs an extra k**m**+1**s** $(x_Q \in \mathbb{F}_{q^{k/2}}, y_Q \in \mathbb{F}_{q^k})$
- Total doubling stage cost = (k+3)**m**+5**s**

Results

- Comparison of doubling and addition stages in the Miller loop against best previous j-invariant zero (CM discriminant $D = 3$) formulas

Tate pairing	DBL	mADD	ADD
Arène *et al.*	$3\mathbf{m} + 8\mathbf{s}$	$6\mathbf{m} + 6\mathbf{s}$	$9\mathbf{m} + 6\mathbf{s}$
This work	$3\mathbf{m} + 5\mathbf{s}$	$10\mathbf{m} + 2\mathbf{s} + 1\mathbf{c}$	$13\mathbf{m} + 2\mathbf{s} + 1\mathbf{c}$

- $k\mathbf{m}$ (common for all) removed from above table
- These formulas offer a saving of $3\mathbf{s}$ at each doubling stage
- Addition stages slower by approximately 4 $\mathbf{m}/\mathbf{s}$ trade-offs
- The formulas in this work only apply to special j-invariant zero curves of the form $y^2 = cx^3 + 1$

Generating the curve $y^2 = cx^3 + 1$

- Construction 6.6 in FST - "A taxonomy of pairing-friendly elliptic curves" will always generate **families** of j-invariant zero curves for arbitrary embedding degrees $k \nmid 18$.
- Most embedding degrees give optimal ρ-value construction on a j-invariant zero (D=3) curves (all $k \leq 50$, except $k = 6, 16, 22, 28, 40, 46$)
- We want to generate $y^2 = cx^3 + 1$ which always has the point $(x, y) = (0, 1)$ of order 3
- If construction 6.6 (or any j-invariant construction) gives a curve with order divisible by 3, faster formulas apply. Most of the embedding degrees facilitate this...

Example curves

- k= 12, $\rho \approx 3/2$, c = 1
  q = 58894904074963910778639935239236932377543210263895/
  10984131168447713879913 (239 bits)
  r = 1461501669025015507443564621194276547766154173393 (161 bits)
  t = 1099511633738 (41 bits)
- ρ-value is much worse than what is achieved with BN curves ($\rho = 1$).

- k=24, $\rho \approx 5/4$, c = 3
  q = 548939984083804061129329064391756261063892295499/
  022387041217 (199 bits)
  r = 1490450500267642163962910277522470312138493750001 (161 bits)
  t = 1051151 (21 bits)
- ρ-value is current record for families of this embedding degree

- $k = 8$, no curve (at least not with construction 6.6)

Tying up a couple of loose ends

1. Scalar multiplication in Jacobian coordinates
   - The EFD reports $2\mathbf{m}+5\mathbf{s}$ for point doubling in Jacobian coordinates for j-invariant zero curves.
   - Protocols should only switch to homogeneous projective coordinates for the pairing.
   - Mapping $(X : Y : Z) \in \mathcal{J}$ to $(XZ : Y : cZ^3) \in \mathcal{P}$ costs $2\mathbf{m}+1\mathbf{s}+1\mathbf{c}$.
2. Supersingular scenario
   - Can't just use the distortion map ϕ to define $\hat{e}(P, Q) = e(P, \phi(Q))$
   - Define $\tilde{e}(P, Q) = e(P, \theta(Q))$ where $\theta(Q) = \phi(Q) - \pi_p(\phi(Q))$ so that $\theta(Q)$ is in the trace-zero subgroup
3. Many methods of curve construction
   - KSS curves, Brezing and Weng curves, etc

Summary (so far)

- So long as a *j*-invariant zero curve has a point of order 3, the formulas presented are applicable will give a solidly faster Tate pairing

3: Minimize the loop length ($\approx \log_2(r)/\varphi(k)$)

... more on this ~~later~~ NOW!

- Can we apply this work to the Ate pairing?

The ate pairing on $y^2 = cx^3 + 1$... or not?

- Raw ate pairing $a_T(Q, P)$ on curves not facilitating twists will always work
- When quadratic and sextic twists are applied to compute $a_T(Q', P)$, the original curve $E : v^2 = u^3 + B$ and the twisted curve $E' : v^2 = u^3 + \beta B$ $(\beta \neq z^2)$ can't both be written in the form $y^2 = cx^3 + 1$
- The formulas in this paper won't work since they assume that both points are on a curve of the form $y^2 = cx^3 + 1$
- For degree three twists, the formulas will work
- e.g. The (quadratic, sextic) twist of a BN curve (k=12) has order divisible by 3, but we can only twist Q onto this curve

Current/near future work

- Faster formulas that work for all j-invariant zero curves
- Ate-like pairing (quadratic and sextic twists) with both points on the curve $y^2 = cx^3 + 1$
- Speeding up ate pairings on BN curves, KSS curves, etc

Conclusion

- Tate pairing on j-invariant zero curves can save approximately 3**s** in each Miller iteration if the curve has order divisible by 3
- In the Tate pairing, the relative speed-up becomes less at larger embedding degrees
- Ate pairing will soon enjoy similar savings on these curves...

Thanks to Professor Tanja Lange and the anonymous referees for their valuable guidance