Faster Ate Pairing Computation on Selmer's Model of Elliptic Curve

Emmanuel Fouotsa
(joint work with Abdoul Aziz Ciss)

University of Bamenda
Cameroon

ECC 2016

Izmir, 05-07 Sept. 2016

Outline

1. Pairings
   1. Definition of Pairings on Elliptic Curves
   2. Computation of pairings on elliptic curves
   3. Some Optimisations
   4. History of the computation of pairings on Elliptic Curves

2. Ate Pairing on the Selmer model of Elliptic Curves
   1. The Selmer Model
   2. The Ate pairing on the Selmer model of Elliptic Curves
   3. Comparison

Pairing-Based Cryptography (PBC) enables many elegant solutions to cryptographic problems :

1. Identity-based encryption
2. Short signatures
3. Non-interactive authenticated key agreement

Pairing computation is the most expensive operation in PBC.

Important : Make it faster !

Pairings : General definition

$(\mathbb{G}_1, +)$ $(\mathbb{G}_2, +)$ and $(\mathbb{G}_3, \times)$ commutative groups of order n.
A pairing is a map

$$e_n : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$$

such that

- e_n is bilinear :
  - $e_n(S_1 + S_2, T) = e_n(S_1, T)e_n(S_2, T)$
  - $e_n(S, T_1 + T_2) = e_n(S, T_1)e_n(S, T_2)$
- e_n is non degenerate.
- e_n efficiently computable

Pairings : Realisation on elliptic curves

Context

- E, elliptic curve on $\mathbb{F}_q$, identity element $\mathcal{O}$.
- r, a large divisor (closed to) of $\sharp E(\mathbb{F}_q)$
- Two linearly independent points $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ of order r where
  - $\mathbb{G}_1 = E\left(\overline{\mathbb{F}_q}\right)[r] \cap \mathrm{Ker}(\pi_q - [1]) = E(\mathbb{F}_q)[r]$
  - $\mathbb{G}_2 = E\left(\overline{\mathbb{F}_q}\right)[r] \cap \mathrm{Ker}(\pi_q - [q]) = E(\mathbb{F}_{q^k})[r]$ (Balasubramanian and Koblitz)

  where k is called the embedding degree (smallest integer such that $r|(q^k - 1)$)

- Take two linearly independent points of order r : $P \in \mathbb{G}_1 = E(\mathbb{F}_q)[r]$ and $Q \in \mathbb{G}_2 = E(\mathbb{F}_{q^k})[r]$.
- Let $f_{m,R}$ be the function with divisor

$$\text{Div}\,(f_{m,R}) = m(R) - m(\mathcal{O}) \tag{1}$$

we have the pairings :

- The reduced Tate Pairing is the map

$$\begin{array}{rccc} e_r : & \mathbb{G}_1 \times \mathbb{G}_2 & \rightarrow & \mu_r \\ & (P, Q) & \mapsto & f_{r,P}(Q)^{\frac{q^k-1}{r}} \end{array} \tag{2}$$

Tate and Ate Pairings on elliptic curves

- Take two linearly independent points of order r : $P \in \mathbb{G}_1 = E(\mathbb{F}_q)[r]$ and $Q \in \mathbb{G}_2 = E(\mathbb{F}_{q^k})[r]$.
- Let $f_{m,R}$ be the function with divisor

$$\text{Div}\ (f_{m,R}) = m(R) - m(\mathcal{O}) \quad (1)$$

we have the pairings :

- The reduced Tate Pairing is the map

$$\begin{array}{rcl} e_r : \quad \mathbb{G}_1 \times \mathbb{G}_2 & \to & \mu_r \\ (P, Q) & \mapsto & f_{r,P}(Q)^{\frac{q^k-1}{r}} \end{array} \quad (2)$$

- The ate pairing is the map

$$\begin{array}{rcl} e_A : \quad \mathbb{G}_2 \times \mathbb{G}_1 & \to & \mu_r, \\ (Q, P) & \mapsto & f_{T,Q}(P)^{\frac{q^k-1}{r}}, \end{array} \quad (3)$$

where $T = t - 1; log(T) \approx log(r)/2$

Pairings : Tools for the computation

The computation of a pairing requires two main operations :

- The computation of the function $f_{m,R}$
- The final exponentiation $f_{m,R}^{\frac{q^k-1}{r}}$

For the computation of the function $f_{m,R}$, let $f_{i,R}$ be the function such that $\mathrm{Div}(f_{i,R}) = i(R) - ([i]R) - (i-1)(\mathcal{O})$, then

- For $i = r$ we have $\mathrm{Div}\ (f_{r,P}) = r(P) - r(\mathcal{O})$
- $$f_{i+j,P} = f_{i,P} \cdot f_{j,P} \cdot h_{[i]P,[j]P} \qquad (4)$$

where $h_{R,S}$ is the function that define the group law on the elliptic curve $\mathrm{Div}(h_{R,S}) = (R) + (S) - (S+R) - (\mathcal{O})$

Examples

- For Weiertrass curves, $h_{R,S} = \frac{\ell_{R,S}}{v_{R+S}}$ quotient of line functions (Huff, Hessian,..)
- For Edward curves, $h_{R,S}$ is the quotient of quadratic functions !

We always have $H_{R,S} = \frac{u}{v}$

Miller's algorithm and Tate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, ...r_1, r_0)_2$.
Output : *The Tate pairing of* P *and* Q : $e_m(P, Q)$

1. *do* $f \leftarrow 1$ *and* $R \leftarrow P$
2. *for* $i = m - 1$ *à* 0

2.1 *do* $f \leftarrow f^2 \cdot H_{R,R}(Q)$ *and* $R \leftarrow 2R$

2.2 *if* $r_i = 1$ *then* $f \leftarrow f \cdot H_{R,P}(Q)$ *and* $R \leftarrow R + P$

3. $e_m(P, Q) \leftarrow f^{\frac{q^k-1}{r}}$

Pairings : the main computation tool

Miller's algorithm and ate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$T = (1, T_{m-1}, ... T_1, T_0)_2$.
Output : The ate pairing of P and Q : $e_m(Q, P)$

1. do $f \leftarrow 1$ and $R \leftarrow Q$
2. for $i = m - 1$ à 0
   2.1 do $f \leftarrow f^2 \cdot H_{R,R}(P)$ and $R \leftarrow 2R$
   2.2 if $T_i = 1$ then $f \leftarrow f \cdot H_{R,Q}(P)$ and $R \leftarrow R + Q$
3. $e_m(Q, P) \leftarrow f^{\frac{q^k-1}{r}}$

Miller's algorithm and Tate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, \ldots. r_1, r_0)_2$.
Output : *The Tate pairing of* P *and* Q : $e_m(P, Q)$

1. *do* $f \leftarrow 1$ *and* $R \leftarrow P$
2. *for* $i = m - 1$ *to* 0

2.1 *do* $f \leftarrow f^2 \cdot H_{R,R}(Q) = u(Q)$ *and* $R \leftarrow 2R$

2.2 *if* $r_i = 1$ *then* $f \leftarrow f \cdot u(Q)$ *and* $R \leftarrow R + P$

3.$e_m(P, Q) \leftarrow f^{\frac{q^k - 1}{r}}$

One can avoid the denominator of $H_{R,S} = \frac{u}{v}$

Pairings : Optimising computations

Miller's algorithm and Tate pairing computation, Mil'86

***Input** : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,*
$r = (1, r_{m-1}, \ldots .r_1, r_0)_2$.
***Output** : The Tate pairing of P and Q : $e_m(P, Q)$*

1. *do* $f \leftarrow 1$ *and* $R \leftarrow P$
2. *for* $i = m - 1$ *to* 0

2.1 *do* $f \leftarrow f^2 \cdot u(Q)$ *(projective)and* $R \leftarrow 2R$

2.2 *if* $r_i = 1$ *then* $f \leftarrow f \cdot u(Q)$*(projective)and* $R \leftarrow R + P$

3.$e_m(P, Q) \leftarrow f^{\frac{q^k - 1}{r}}$

Avoid inversions turning to projective coordinates

Miller's algorithm and Tate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, \ldots. r_1, r_0)_2$.
Output : The Tate pairing of P and Q : $e_m(P, Q)$

1. do $f \leftarrow 1$ and $R \leftarrow P$
2. for $i = m - 1$ to 0

2.1 do $f \leftarrow f^2 \cdot u(Q)$ and $R \leftarrow 2R$

2.2 if $r_i = 1$ then $f \leftarrow f \cdot u(Q)$ and $R \leftarrow R + P$

3. $e_m(P, Q) \leftarrow f^{\frac{q^k - 1}{r}}$

Improve the arithmetic in the extension $\mathbb{F}_{q^k}$:
$k = 2^i 3^j$ is nice since ...

Miller's algorithm and Tate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, ...r_1, r_0)_2$.
Output : *The Tate pairing of* P *and* Q : $e_m(P, Q)$

1. *do* $f \leftarrow 1$ *and* $R \leftarrow P$
2. *for* $i = m - 1$ *to* 0

2.1 *do* $f \leftarrow f^2 \cdot u(Q)$ *and* $R \leftarrow 2R$

2.2 *if* $r_i = 1$ *(Unlikely) then* $f \leftarrow f \cdot u(Q)$ *and* $R \leftarrow R + P$

3.$e_m(P, Q) \leftarrow f^{\frac{q^k-1}{r}}$

Choose a lower Hamming weight r

Miller's algorithm and Tate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, ...r_1, r_0)_2$.
Output : *The Tate pairing of* P *and* Q : $e_m(P, Q)$

1. do $f \leftarrow 1$ and $R \leftarrow P$
2. for $i = m - 1$ to 0

   2.1 do $f \leftarrow f^2 \cdot u(Q)$ and $R \leftarrow 2R$

   2.2 if $r_i = 1$ then $f \leftarrow f \cdot u(Q)$ and $R \leftarrow R + P$
3. $e_m(P, Q) \leftarrow f^{\frac{q^k - 1}{r}}$

Split the final exponentiation : $\frac{p^k - 1}{r} = \left[\frac{p^k - 1}{\phi_k(p)}\right] \cdot \left[\frac{\phi_k(p)}{r}\right]$

Miller's algorithm and Tate pairing computation, Mil'86

Input *:* $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, ...r_1, r_0)_2$.
Output *: The Tate pairing of* P *and* Q *:* $e_m(P, Q)$

1. *do* $f \leftarrow 1$ *and* $R \leftarrow P$
2. *for* $i = m - 1$ *to* 0

2.1 *do*

2.2 *if* $r_i = 1$ *then* $f \leftarrow f \cdot u(Q)$ *and* $R \leftarrow R + P$

3. $e_m(P, Q) \leftarrow f^{\frac{q^k - 1}{r}}$

Split the final exponentiation : $\frac{p^k - 1}{r} = \left[\frac{p^k - 1}{\phi_k(p)}\right] \cdot \left[\frac{\phi_k(p)}{r}\right]$
Applied "vectorial addition chain method", Scott et al. Pairing 2009

Miller's algorithm and Tate pairing computation, Mil'86

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$,
$r = (1, r_{m-1}, ...r_1, r_0)_2$.
Output : *The Tate pairing of* P *and* Q : $e_m(P, Q)$

1. *do* $f \leftarrow 1$ *and* $R \leftarrow P$
2. *for* $i = m - 1$ *to* 0

2.1 *do*

2.2 *if* $r_i = 1$ *then* $f \leftarrow f \cdot u(Q)$ *and* $R \leftarrow R + P$

3. $e_m(P, Q) \leftarrow f^{\frac{q^k-1}{r}}$

Split the final exponentiation : $\frac{p^k-1}{r} = \left[\frac{p^k-1}{\phi_k(p)}\right] \cdot \left[\frac{\phi_k(p)}{r}\right]$

"Lattices-based method" by Fuentes et al. SAC 2011

Efficiency depends also on the shape of the curve and its arithmetic

1. Pairings on Weierstrass model $y^2 = x^3 + ax + b$
   - Costello, Hisil et al.(Pairing 2009)
   - Costello, Lange et al.(PKC 2010)
2. Pairings on Edwards curves $ax^2 + y^2 = 1 + dx^2y^2$
   - Sarkar,Laxman et al. (Pairing 2008)
   - Ionica and Joux (Indocrypt 2008)
   - Arène, Lange et al. (Journal of Number theory, 2011)

Efficiency depends also on the shape of the curve and its arithmetic

1. Pairings on the Huff model by Joye, Tibouchi et al.(2010) : $aX(Y^2 - Z^2) = bY(X^2 - Z^2)$
2. Pairings on the Selmer model by Zhang, Wang et al.(ISPEC 2011) :$ax^3 + by^3 = d$
3. Pairings on the Hessian model by Gu, Gu et al. (ICISC 2010) : $X^3 + Y^3 + Z^3 = 3dXYZ$
4. Pairings on the Jacobi model : $E_d : y^2 = dx^4 + 2\delta x^2 + 1$ by
   - Wang, Wang et al.(CJE 2011)
   - Fouotsa and Duquesne. Pairing 2012
   - Fouotsa, Duquesne, El Mrabet.(Journal of Mathematical Cryptology, 2014)

Pairing-Friendly Curves

Definition

An elliptic curve E is said pairing-friendly if :

1. k is small (less than 50)
2. $r > \sqrt{q}$

Pairing-friendly curves are rare ! ! ! but can be obtained by polynomial parameterisations

Pairing-Friendly Curves

Polynomial parameterisations

We are looking for a curve E such that :

1. $r \mid q^k - 1$

Pairing-Friendly Curves

Polynomial parameterisations

We are looking for a curve E such that :

1. $r \mid q^k - 1$
2. $r \nmid E$

Pairing-Friendly Curves

Polynomial parameterisations

We are looking for a curve E such that :

1. $r \mid q^k - 1$ implies (MNT) that $r \mid \varphi_k(q)$
2. $r \mid\neq E$

Polynomial parameterisations

We are looking for a curve E such that :

1. $r \mid q^k - 1$ implies (MNT) that $r \mid \varphi_k(q)$
2. $r \mid\neq E$ if furthermore $r \mid \varphi_k(q)$ then (BLS) $r \mid \varphi_k(t-1)$

Pairing-Friendly Curves

Polynomial parameterisations

We are looking for a curve E such that :

1. $r \mid q^k - 1$ implies (MNT) that $r \mid \varphi_k(q)$
2. $r \mid\neq E$ if furthermore $r \mid \varphi_k(q)$ then (BLS) $r \mid \varphi_k(t-1)$

So to find a pairing friendly curve, fix a small k and find $r(x), t(x)$ and $q(x)$ such that $r(x) \mid \varphi_k(t(x)-1)$ and $r(x) \mid q(x)^k - 1$

Pairing-Friendly Curves

Polynomial parameterisations of Barreto-Naehrig (BN) curves

$k = 12$
$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$
$r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$
$t(x) = 6x^2 + 1$

1. Ideal situation at the 128-bit security level with $\rho = \frac{log(p)}{log(r)} = 1$
2. curve of the form $y^2 = x^3 + b$

2- Faster ate pairing on Selmer Curves

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)
- Selmer curves are elliptic curves with discriminant $\Delta = -2^{12}3^9d^4$ and the j-invariant is 0.

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)
- Selmer curves are elliptic curves with discriminant $\Delta = -2^{12}3^9d^4$ and the j-invariant is 0.
- Can be regarded as a particular case of the generalized Hessian curves $x^3 + y^3 + e = fxy$ which also has good properties for cryptographic applications :

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)
- Selmer curves are elliptic curves with discriminant $\Delta = -2^{12}3^9d^4$ and the j-invariant is 0.
- Can be regarded as a particular case of the generalized Hessian curves $x^3 + y^3 + e = fxy$ which also has good properties for cryptographic applications :
- Resistance to side channel attacks (Unified formulas) : (Joye and Quisquater, CHES 2001)

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)
- Selmer curves are elliptic curves with discriminant $\Delta = -2^{12}3^9d^4$ and the j-invariant is 0.
- Can be regarded as a particular case of the generalized Hessian curves $x^3 + y^3 + e = fxy$ which also has good properties for cryptographic applications :
- Some standard curves can be transformed to Hessian curves : (Smart, CHES 2001)

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)
- Selmer curves are elliptic curves with discriminant $\Delta = -2^{12}3^9d^4$ and the j-invariant is 0.
- Can be regarded as a particular case of the generalized Hessian curves $x^3 + y^3 + e = fxy$ which also has good properties for cryptographic applications :
- Point operation can be implemented in a highly parallel way (40% performance improvement over Weiertrass curves) : (Smart, CHES 2001)

The Selmer Curves

- Given by the affine equation $ax^3 + by^3 = c$ with $abc \neq 0$
- Named by Ian Connell in Elliptic curve handbook, 1999
- Can be transformed to a simpler form $x^3 + y^3 = d$
- The Selmer curve $S_d : x^3 + y^3 = d$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_d : v^2 = u^3 - 432d^2$, (Ian 1999)
- Selmer curves are elliptic curves with discriminant $\Delta = -2^{12}3^9d^4$ and the j-invariant is 0.
- Can be regarded as a particular case of the generalized Hessian curves $x^3 + y^3 + e = fxy$ which also has good properties for cryptographic applications :
- Fast formulas for the computation of the Tate pairing on Selmer curves (Zhang, Wang, Wang, Ye, ISPEC 2011)

point operation on Selmer curves : Wang et al. 2011

$(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$

$$\begin{cases} X_3 &= X_1 Z_1 Y_2^2 - X_2 Z_2 Y_1^2 \\ Y_3 &= Y_1 Z_1 X_2^2 - Y_2 Z_2 X_1^2 \\ Z_3 &= X_1 Y_1 Z_2^2 - X_2 Y_2 Z_1^2 \end{cases}$$

Cost : 12M

$2(X_1 : Y_1 : Z_1) = (X_3 : Y_3 : Z_3)$

$$\begin{cases} X_3 &= -Y_1(2X_1^3 + Y_1^3) \\ Y_3 &= X_1(X_1^3 + 2Y_1^3) \\ Z_3 &= Z_1(X_1^3 - Y_1^3) \end{cases}$$

Cost : 5M+2S

Ate Pairing on Selmer curve : choice of Q

Let $E : y^2 = x^3 + b$ and its twist $E' : y'^2 = x'^3 + b/\omega^6$ with $b = -432d^2$. The maps

$$\begin{array}{ccccc} E' & \longrightarrow & E & \longrightarrow & S_d \\ (x', y') & \longmapsto & (x'\omega^2, y'\omega^3) & \longmapsto & \left(\dfrac{36d - y'\omega^3}{6x'\omega^2}, \dfrac{36d + y'\omega^3}{6x'\omega^2} \right) \end{array}$$

.

enable to consider points in $\mathbb{G}_2$ as $Q = (S - T\omega : S + T\omega : V)$ in projective coordinates where $\mathbb{F}_{q^k} = \mathbb{F}_{q^{k/2}}(\omega)$ with ω in $\mathbb{F}_{q^k}$, $S = 36d$, $T = y'\omega^2$, $V = 6x'\omega^2 \in \mathbb{F}_{q^{k/2}}$.

Ate pairing on Selmer Curves : Addition of points

$(S_1 - T_1\omega : S_1 + T_1\omega : V_1) + (S_2 - T_2\omega : S_2 + T_2\omega : V_2) = (S_3 - T_3\omega : S_3 + T_3\omega : V_3)$

$$\begin{cases} S_3 = & (V_1 S_2 - V_2 S_1)(S_1 S_2 - 2T_1 T_2\omega^2) + (V_1 S_1 T_2^2 - V_2 S_2 T_1^2)\omega^2 \\ T_3 = & (V_1 T_2 - V_2 T_1)(T_1 T_2\omega^2 - 2S_1 S_2) + V_1 S_2^2 T_1 - V_2 S_1^2 T_2 \\ V_3 = & S_1 V_2 - S_2 V_1)(S_1 V_2 + S_2 V_1) + (V_1 T_2 - V_2 T_1)(V_1 T_2 + V_2 T_1)\omega^2 \end{cases} \quad (5)$$

$2(S_1 - T_1\omega : S_1 + T_1\omega : V_1) = (S_3 - T_3\omega : S_3 + T_3\omega : V_3)$

$$\begin{cases} S_3 = & -8S_1 T_1^3\omega^2 \\ T_3 = & T_1^4\omega^2 - 6S_1^2 T_1^2 - 3\frac{S_1^4}{\omega^2} \\ V_3 = & (-6V_1 S_1^2 T_1 - 2V_1 T_1^3\omega^2) \end{cases} \quad (6)$$

Ate pairing on Selmer Curves : Miller function and denominator elimination

Addition step :

$$h_{R,Q}(P) \;=\; \frac{c_X x_P + c_Y y_P + c_Z}{Z_3(x_P + y_P) - (X_3 + Y_3)} \;=\; \frac{l_1(P)}{l_2(P)} \qquad (7)$$

The denominator reduces to $V_3(x_P + y_P) - 2S_3 \in \mathbb{F}_{q^{k/2}}$

The addition step then consists in computing :

1. $h_{R,Q}(P) = c_X x_P + c_Y y_P + c_Z$ with
$$c_X = Y_1 Z_2 - Z_1 Y_2$$
$$c_Y = Z_1 X_2 - X_1 Z_2$$
$$c_Z = X_1 Y_2 - Y_1 X_2$$
2. The addition
$(S_1 - T_1\omega : S_1 + T_1\omega : V_1) + (S_2 - T_2\omega : S_2 + T_2\omega : V_2) = (S_3 - T_3\omega : S_3 + T_3\omega :$

$$\begin{cases} S_3 = & (V_1S_2 - V_2S_1)(S_1S_2 - 2T_1T_2\omega^2) + (V_1S_1T_2^2 - V_2S_2T_1^2)\omega^2 \\ T_3 = & (V_1T_2 - V_2T_1)(T_1T_2\omega^2 - 2S_1S_2) + V_1S_2^2T_1 - V_2S_1^2T_2 \\ V_3 = & S_1V_2 - S_2V_1)(S_1V_2 + S_2V_1) + (V_1T_2 - V_2T_1)(V_1T_2 + V_2T_1)\omega^2 \end{cases} \qquad (8)$$

Operations	Values	Costs
$A := V_1S_2, B := V_2S_1$	$A = V_1S_2, B = V_2S_1$	$2m_e$
$C := S_1S_2,\ D := T_1T_2$	$C = S_1S_2, D = T_1T_2$	$2m_e$
$E := V_1T_2, F := S_1T_2$	$E = V_1T_2,\ F = S_1T_2$	$2m_e$
$G := V_2T_1, H := S_2T_1$	$G = V_2T_1, H = S_2T_1$	$2m_e$
$L := A - B, M_1 := Dw^2$	$L = V_1S_2 - V_2S_1,\ M_1 = T_1T_2w^2$	$1m_c$
$M := L(C - 2M_1)$	$M = (V_1S_2 - V_2S_1)(S_1S_2 - 2T_1T_2w^2)$	$1m_e$
$N_1 := EF,\ N_2 := GH$	$N_1 = V_1T_2^2S_1,\ N_2 = V_2S_2T_1^2$	$2m_e$
$N := (N_1 - N_2)w^2$	$N = (V_1S_1T_2^2 - V_2S_2T_1^2)w^2$	$1m_c$
$O := E - G$	$O = V_1T_2 - V_2T_1$	-
$P := M_1 - 2C$	$P = T_1T_2w^2 - 2S_1S_2$	-
$Q := OP$	$Q = (V_1T_2 - V_2T_1)(T_1T_2w^2 - 2S_1S_2)$	$1m_e$
$R_1 := AH,\ R_2 := BF$	$R_1 = V_1S_2^2T_1,\ R_2 = V_2S_1^2T_2$	$2m_e$
$U_1 := A + B,\ U_2 := E + G$	$U_1 = V_1S_2 + V_2S_1,\ U_2 = V_1T_2 + V_2T_1$	-
$U_3 := OU_2w^2$	$U_3 = (V_1T_2 - V_2T_1)(V_1T_2 + V_2T_1)w^2$	$1m_e + 1m_c$

Table: Combined formulas for the point addition and Miller's function

Operations	Values	Costs
$U_4 := -LU_1$	$U_4 = -(V_1S_2 - V_2S_1)(V_1S_2 + V_2S_1)$	$1m_e$
$X_3 := M + N-$ $(Q + R_1 - R_2)w$		-
$Y_3 := M + N-$ $(Q + R_1 + R_2)w$		-
$Z_3 := U_4 + U_3$		-
$c_X := -L - Ow$	$c_X = -(V_1S_2 - V_2S_1) - (V_1T_2 - V_2T_1)w$	-
$c_Y := L - Ow$	$c_Y = (V_1S_2 - V_2S_1) - (V_1T_2 - V_2T_1)w$	-
$c_Z := 2(F - H)w$	$c_Z = 2(S_1T_2 - S_2T_1)w$	-
$h_{R,Q}(P) := c_Xx_P +$ $c_Yy_P + c_Z$	-	$\frac{k}{2}m_1 + \frac{k}{2}m_1 = km_1$
$f := f.h_{R,Q}(P)$		$1m_k$
Total cost :	$16m_e + 3m_c + km_1 + 1m_k$	

Table: Combined formulas for the point addition and Miller's function

Ate pairing on Selmer Curves : Miller function and denominator elimination

Doubling step :

$$h_{R,R}(P) = \frac{c_X x_P + c_Y y_P + c_Z}{Z_3(x_P + y_P) - (X_3 + Y_3)} = \frac{l_1(P)}{l_2(P)} \quad (9)$$

The denominator reduces to $(-6V_1S_1^2T_1 - 2V_1T_1^3w^2)(x_P + y_P) + 2T_3 \in \mathbb{F}_{q^{k/2}}$.
The doubling step then consists in the computation of :

1. $h_{R,R}(P) = c_X x_P + c_Y y_P + c_Z$ with

$$c_X = c_Y = Y_1Z_1 - X_1Z_1,$$
$$c_Z = X_1^2 - Y_1^2.$$

2. The doubling

$2(S_1 - T_1\omega : S_1 + T_1\omega : V_1) = (S_3 - T_3\omega : S_3 + T_3\omega : V_3)$

$$\begin{cases} S_3 = & -8S_1T_1^3\omega^2 \\ T_3 = & T_1^4\omega^2 - 6S_1^2T_1^2 - 3\frac{S_1^4}{\omega^2} \\ V_3 = & (-6V_1S_1^2T_1 - 2V_1T_1^3\omega^2) \end{cases} \quad (10)$$

Ate Pairing on Selmer curves : cost of the combined doubling Miller step

Operations	Values	Costs
$A := S_1^2$	$A = S_1^2$	$1s_e$
$B := T_1^2$	$B = T_1^2$	$1s_e$
$C := ((S_1 + T_1)^2 - A - B)/2$	$C = S_1 T_1$	$1s_e$
$D := A^2$	$D = X_1^4$	$1s_e$
$E := Bw^2$	$E = T_1^2 w^2$	$1m_c$
$T_3 := -12D + (3A - E)^2$		$1s_e$
$S_3 := 8CE$		$1m_e$
$F := V_1 T_1$		$1m_e$
$V_3 := (-2F(3A + E))w$		$1m_e$
$X_3 := S_3 - T_3 w$		-
$Y_3 := S_3 + T_3 w$		-
$Z_3 := V_3$		-
$h_{R,R}(P) := (2C(y_P - x_P))w$		$km_1 + +1s_e + 2m_c$
$+(A + Bw^2)(x_P + y_P) - dV_1^2$		
$f := f^2.h_{R,R}(P)$		$1s_k + 1m_k$
Total cost :	$4m_e + 5s_e + 3m_c + km_1 + s_k + m_k$	

Table: Combined formulas for point doubling and Miller' function

Processor 1	Processor 2	Processor 3	Cost
$m_1 = V_1 S_2$	$m_2 = V_2 S_1$	$m_3 = S_1 S_2$	$1 m_e$
$m_4 = T_1 T_2$	$m_5 = V_1 T_2$	$m_6 = S_1 T_2$	$1 m_e$
$m_7 = V_2 T_1$	$m_8 = S_2 T_1$	$m_9 = m_5 m_6$	$1 m_e$
$m_{10} = m_7 m_8$	$m_{11} = m_1 m_8$	$m_{12} = m_2 m_6$	$1 m_e$
$a_1 = m_5 - m_7$	$a_2 = m_5 + m_7$	$--$	$--$
$m_{13} = a_1 a_2$	$--$	$--$	$1 m_e$
$c_1 = m_4 w^2$	$c_2 = (m_9 - m_{10}) w^2$	$c_3 = m_{12} w^2$	$1 m_c$
$a_3 = m_1 - m_2$	$a_4 = c_1 - 2 m_3$	$a_5 = m_1 + m_2$	$--$
$m_{14} = a_3 (m_3 - 2 c_1)$	$m_{15} = a_1 a_4$	$m_{16} = -a_3 a_5$	$1 m_e$
$X_3 = m_{14} + c_2$ $-(m_{15} + m_{11} - m_{12}) w$	$Y_3 = m_{14} + c_2$ $+(m_{15} + m_{11} - m_{12}) w$	$Z_3 = c_3 + m_{16}$	$--$
$c_X = -a_3 - a_1 w$	$c_Y = a_3 - a_1 w$	$c_Z = 2(m_6 - m_8) w$	$--$
$t_1 = c_X x_P$	$t_2 = c_Y y_P$	$--$	$\frac{k}{2} m_1$
$f = f \cdot (t_1 + t_2 + c_Z)$	$--$	$--$	$1 m_k$
Total cost : $6 m_e + 1 m_c + \frac{k}{2} m_1 + 1 m_k$			

Table: Parallel execution of addition step in Miller's function

Ate Pairing on Selmer curves : Parallelizing the doubling step

Processor 1	Processor 2	Processor 3	Cost
$s_1 = S_1^2$	$s_2 = T_1^2$	$s_3 = V_1^2$	$1s_e$
$c_1 = s_2 w^2$	$c_2 = s_1$	$c_3 = ds_3$	$1m_c$
$s_4 = (S_1 + T_1)^2$	$s_5 = \frac{s_1^2}{w^2}$	$s_6 = (3\frac{s_1^2}{w^2} - c_1 w^2)^2$	$1s_e$
$a_1 = (s_4 - s_1 - s_2)/2$	$a_2 = -12s_5 + s_6$	$m_1 = V_1 T_1$	$1m_e$
$m_2 = -8a_1 c_1$	$m_3 = (-2m_1(3s_1 + c_1))w$	$f_1 = f^2$	$1m_e + 1s_k$
$X_3 = m_2 - a_2 w$	$Y_3 = m_2 + a_2 w$	$Z_3 = m_3$	$--$
$t_1 = 2a_1(y_P - x_P)$	$t_2 = (c_2 + s_2 w^2)(x_P + y_P)$	$--$	$\frac{k}{2}m_1$
$f = f_1 \cdot (t_1 w + t_2 - c_3)$	$--$	$--$	$1m_k$
Total cost : $2m_e + 2s_e + 1m_c + \frac{k}{2}m_1 + 1s_k + 1m_k$			

Table: Parallel execution of doubling step in Miller's function

Ate Pairing on Selmer curves : Comparison with other work

In the Table we compare the costs for one iteration for Ate on the Selmer curve $S_d : x^3 + y^3 = d$ and on the Weierstrass curve $W : y^2 = x^3 + c^2$ ([1] Costello, Lange, Naehrig, PKC 2010).

Pairings	Doubling	Addition	Mixed Addition
Ate(Q, P) Weierstrass$(a = 0)$[1]	$4m_e + 7s_e + km_1 + 1m_k + 1s_k$	$16m_e + 2s_e + km_1 + 1m_k$	$12m_e + 2s_e + km_1 + 1m_k$
ate(Q, P) **This work**	$3m_e + 5s_e + km_1 + 1m_k + 1s_k$	$16m_e + km_1 + 1m_k$	$14m_e + km_1 + 1m_k$
ate(Q, P)(This work) **Parallelization**	$2m_e + 2s_e + \frac{k}{2}m_1 + 1m_k + 1s_k$	$6m_e + \frac{k}{2}m_1 + 1m_k$	–

Table: Comparison of costs of one iteration for Ate pairing on Selmer and Weierstrass Elliptic Curves

Pairing friendly Selmer curves

- Let p be a prime number with $p \equiv 1 \mod 3$. The $E_d : y^2 = x^3 + d^2$ is an ordinary elliptic curve.
- We have the following isomorphism

$$\begin{array}{rccc} \varphi : & E_d & \rightarrow & W_d \\ & (x, y) & \mapsto & (-12x, -24\sqrt{-3}y) \end{array}$$

  and the curve $W_d : y^2 = x^3 - 432 \cdot 4 \cdot d^2$ is birrationally equivalent to the Selmer curve $S_d : x^3 + y^3 = 2d$
- The construction of pairing friendly curve of the form $E_d : y^2 = x^3 + d^2$ is given by the construction 6.6 of Freemann with $\rho = 1.5$

Emmanuel Fouotsa, Abdoul Aziz Ciss, *Faster Ate Pairing Computation on Selmer's Model of Elliptic Curves*. In ***Groups, Complexity, Cryptology***, Vol.8(1) DeGruyter (2016)

Thanks for your attention !