Computing optimal pairings on abelian varieties with theta functions

18/04/2013 — Grenoble

David Lubicz, Damien Robert
Outline

1. Curves, pairings and cryptography
2. Abelian varieties
3. Theta functions
4. Pairings with theta functions
5. Performance
Curves, pairings and cryptography
Elliptic curves

Definition \((\text{char } k \neq 2, 3)\)

An elliptic curve is a plane curve with equation

\[ y^2 = x^3 + ax + b \quad 4a^3 + 27b^2 \neq 0. \]

Exponentiation:

\((\ell, P) \mapsto \ell P\)

Discrete logarithm:

\((P, \ell P) \mapsto \ell\)
Pairing-based cryptography

Definition

A pairing is a non-degenerate bilinear application \( e : G_1 \times G_1 \rightarrow G_2 \) between finite abelian groups.

Example

- If the pairing \( e \) can be computed easily, the difficulty of the DLP in \( G_1 \) reduces to the difficulty of the DLP in \( G_2 \).

\[ \Rightarrow \text{MOV attacks on supersingular elliptic curves.} \]

- Identity-based cryptography [BF03].
- Short signature [BLS04].
- One way tripartite Diffie–Hellman [Jou04].
- Self-blindable credential certificates [Ver01].
- Attribute based cryptography [SW05].
- Broadcast encryption [Goy+06].
The Weil pairing on elliptic curves

- Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $k$ (char $k \neq 2, 3$).
- Let $P, Q \in E[\ell]$ be points of $\ell$-torsion.
- Let $f_P$ be a function associated to the principal divisor $\ell(P) - \ell(0)$, and $f_Q$ to $\ell(Q) - \ell(0)$. We define:

\[
e_{W, \ell}(P, Q) = \frac{f_P((Q) - (0))}{f_Q((P) - (0))}.
\]

- The application $e_{W, \ell} : E[\ell] \times E[\ell] \to \mu_\ell(k)$ is a non degenerate pairing: the Weil pairing.

Definition (Embedding degree)

The embedding degree $d$ is the smallest number thus that $\ell \mid q^d - 1$; $\mathbb{F}_{q^d}$ is then the smallest extension containing $\mu_\ell(k)$. 
The Tate pairing on elliptic curves over $\mathbb{F}_q$

Definition

The Tate pairing is a non degenerate (on the right) bilinear application given by

$$e_T : E_0[\ell] \times E(\mathbb{F}_q)/\ell E(\mathbb{F}_q) \rightarrow \mathbb{F}_{q^d}^*/\mathbb{F}_{q^d}^\ell.$$

$$(P,Q) \mapsto f_P((Q) - (0))$$

where

$$E_0[\ell] = \{P \in E[\ell](\mathbb{F}_{q^d}) \mid \pi(P) = [q]P\}.$$ 

- On $\mathbb{F}_{q^d}$, the Tate pairing is a non degenerate pairing
  $$e_T : E[\ell](\mathbb{F}_{q^d}) \times E(\mathbb{F}_{q^d})/\ell E(\mathbb{F}_{q^d}) \rightarrow \mathbb{F}_{q^d}^*/\mathbb{F}_{q^d}^\ell \simeq \mu_{\ell};$$

- If $\ell^2 \nmid E(\mathbb{F}_{q^d})$ then $E(\mathbb{F}_{q^d})/\ell E(\mathbb{F}_{q^d}) \simeq E[\ell](\mathbb{F}_{q^d})$;

- We normalise the Tate pairing by going to the power of $(q^d - 1)/\ell$.

- This final exponentiation allows to save some computations.
  For instance if $d = 2d'$ is even, we can suppose that $Q = (x_2, y_2)$ with $x_2 \in E(\mathbb{F}_{q^{d'}})$. Then the denominators of $f_{\lambda, \mu, P}(Q)$ are $\ell$-th powers and are killed by the final exponentiation.
Miller’s functions

• We need to compute the functions $f_P$ and $f_Q$. More generally, we define the Miller’s functions:

Definition
Let $\lambda \in \mathbb{N}$ and $X \in E[\ell]$, we define $f_{\lambda,X} \in k(E)$ to be a function thus that:

$$(f_{\lambda,X}) = \lambda(X) - ([\lambda]X) - (\lambda - 1)(0).$$

• We want to compute (for instance) $f_{\ell,P}((Q) - (0))$. 
Miller’s algorithm

• The key idea in Miller’s algorithm is that

\[ f_{\lambda+\mu,X} = f_{\lambda,X} f_{\mu,X} f_{\lambda,\mu,X} \]

where \( f_{\lambda,\mu,X} \) is a function associated to the divisor

\[(\lfloor \lambda + \mu \rfloor X) - (\lfloor \lambda \rfloor X) - (\lfloor \mu \rfloor X) + (0).\]

• We can compute \( f_{\lambda,\mu,X} \) using the addition law in \( E \): if \( \lfloor \lambda \rfloor X = (x_1, y_1) \) and \( \lfloor \mu \rfloor X = (x_2, y_2) \) and \( \alpha = (y_1 - y_2)/(x_1 - x_2) \), we have

\[ f_{\lambda,\mu,X} = \frac{y - \alpha(x - x_1) - y_1}{x + (x_1 + x_2) - \alpha^2}. \]
Miller’s algorithm on elliptic curves

Algorithm (Computing the Tate pairing)

**Input:** \( \ell \in \mathbb{N}, P = (x_1, y_1) \in E[\ell](\mathbb{F}_q), Q = (x_2, y_2) \in E(\mathbb{F}_{q^d}). \)

**Output:** \( e_T(P, Q). \)

1. **Compute the binary decomposition:** \( \ell := \sum_{i=0}^{I} b_i 2^i. \) Let \( T = P, f_1 = 1, f_2 = 1. \)
2. **For i in [I..0] compute**
   2.1 \( \alpha, \) the slope of the tangent of \( E \) at \( T. \)
   2.2 \( T = 2T. \) \( T = (x_3, y_3). \)
   2.3 \( f_1 = f_1^2(y_2 - \alpha(x_2 - x_3) - y_3), f_2 = f_2^2(x_2 + (x_1 + x_3) - \alpha^2). \)
   2.4 **If** \( b_i = 1, \) **then compute**
      2.4.1 \( \alpha, \) the slope of the line going through \( P \) and \( T. \)
      2.4.2 \( T = T + Q. \) \( T = (x_3, y_3). \)
      2.4.3 \( f_1 = f_1^2(y_2 - \alpha(x_2 - x_3) - y_3), f_2 = f_2(x_2 + (x_1 + x_3) - \alpha^2). \)

**Return**

\[
\left( \frac{f_1}{f_2} \right)^{\frac{q^d - 1}{\ell}}.
\]
Jacobian of curves

C a smooth irreducible projective curve of genus g.

- Divisor: formal sum \( D = \sum n_i P_i \), \( P_i \in C(\bar{k}) \).
  \[ \deg D = \sum n_i. \]

- Principal divisor: \( \sum_{P \in C(\bar{k})} v_P(f).P; \quad f \in \bar{k}(C). \)

Jacobian of \( C \) = Divisors of degree 0 modulo principal divisors
- + Galois action
  = Abelian variety of dimension \( g \).

- Divisor class of a divisor \( D \in \text{Jac}(C) \) is generically represented by a sum of \( g \) points.
Example of Jacobians

**Dimension 2**: Addition law on the Jacobian of an hyperelliptic curve of genus 2:

\[ y^2 = f(x), \quad \deg f = 5. \]

\[ D = P_1 + P_2 - 2\infty \]
\[ D' = Q_1 + Q_2 - 2\infty \]
Example of Jacobians

**Dimension 2:** Addition law on the Jacobian of an hyperelliptic curve of genus 2:

\[ y^2 = f(x), \quad \deg f = 5. \]

\[
D = P_1 + P_2 - 2\infty
\]

\[
D' = Q_1 + Q_2 - 2\infty
\]
Example of Jacobians

**Dimension 2:** Addition law on the Jacobian of an hyperelliptic curve of genus 2:

\[ y^2 = f(x), \quad \deg f = 5. \]

\[ D = P_1 + P_2 - 2\infty \]
\[ D' = Q_1 + Q_2 - 2\infty \]
\[ D + D' = R_1 + R_2 - 2\infty \]
Example of Jacobians

**Dimension 3**

Pairings on Jacobians

- Let $P \in \text{Jac}(C)[\ell]$ and $D_P$ a divisor on $C$ representing $P$;
- By definition of $\text{Jac}(C)$, $\ell D_P$ corresponds to a principal divisor $(f_P)$ on $C$;
- The same formulas as for elliptic curve define the Weil and Tate pairings:

$$e_W(P, Q) = \frac{f_P(D_Q)}{f_Q(D_P)}$$

$$e_T(P, Q) = f_P(D_Q).$$
Pairings on Jacobians

• Let $P \in \text{Jac}(C)[\ell]$ and $D_P$ a divisor on $C$ representing $P$;
• By definition of $\text{Jac}(C)$, $\ell D_P$ corresponds to a principal divisor $(f_P)$ on $C$;
• The same formulas as for elliptic curve define the Weil and Tate pairings:

$$e_W(P, Q) = \frac{f_P(D_Q)}{f_Q(D_P)}$$

$$e_T(P, Q) = f_P(D_Q).$$

• A key ingredient for evaluating $f_P(D_Q)$ comes from Weil reciprocity theorem.

Theorem (Weil)

Let $D_1$ and $D_2$ be two divisors with disjoint support linearly equivalent to $(0)$ on a smooth curve $C$. Then

$$f_{D_1}(D_2) = f_{D_2}(D_1).$$
Pairings on Jacobians

- Let $P \in \text{Jac}(C)[\ell]$ and $D_P$ a divisor on $C$ representing $P$;
- By definition of $\text{Jac}(C)$, $\ell D_P$ corresponds to a principal divisor $(f_P)$ on $C$;
- The same formulas as for elliptic curve define the Weil and Tate pairings:

$$e_W(P, Q) = \frac{f_P(D_Q)}{f_Q(D_P)}$$
$$e_T(P, Q) = f_P(D_Q).$$

- The extension of Miller’s algorithm to Jacobians is “straightforward”;
- For instance if $g = 2$, the function $f_{\lambda, \mu, P}$ is of the form

$$\frac{y - l(x)}{(x - x_1)(x - x_2)}$$

where $l$ is of degree 3.
2

Abelian varieties
Abelian varieties

Definition
An Abelian variety is a complete connected group variety over a base field $k$.

- Abelian variety = points on a projective space (locus of homogeneous polynomials) + an abelian group law given by rational functions.

Example
- Elliptic curves = Abelian varieties of dimension 1;
- If $C$ is a (smooth) curve of genus $g$, its Jacobian is an abelian variety of dimension $g$;
- In dimension $g \geq 4$, not every abelian variety is a Jacobian.
Isogenies and pairings

Let \( f : A \rightarrow B \) be a separable isogeny with kernel \( K \) between two abelian varieties defined over \( k \):

\[
\begin{array}{cccc}
0 & \rightarrow & K & \rightarrow & A & \xrightarrow{f} & B & \rightarrow & 0 \\
& & & & f & & & & \\
& & & & 0 & \leftarrow & \hat{A} & \xleftarrow{\hat{f}} & \hat{B} & \leftarrow & \hat{K} & \leftarrow & 0
\end{array}
\]

- \( \hat{K} \) is the Cartier dual of \( K \), and we have a non degenerate pairing \( e_f : K \times \hat{K} \rightarrow k^* \):
  1. If \( Q \in \hat{K}(k) \), \( Q \) defines a divisor \( D_Q \) on \( B \);
  2. \( \hat{f}(Q) = 0 \) means that \( f^*D_Q \) is equal to a principal divisor \( (g_Q) \) on \( A \);
  3. \( e_f(P, Q) = g_Q(x)/g_Q(x + P) \). (This last function being constant in its definition domain).

- The Weil pairing is the pairing associated to the isogeny \( [\ell] : A \rightarrow A \).
Pairings and polarisations

- If $\Theta$ is an ample divisor, the polarisation $\varphi_\Theta$ is a morphism $A \to \hat{A}$, $x \mapsto t_x^* \Theta - \Theta$.
- We can then compose the Weil pairing with $\varphi_\Theta$:

  \[ e_{W,\Theta,\ell} : A[\ell] \times A[\ell] \to \mu_\ell(k) \]

  \[ (P, Q) \mapsto e_{W,\ell}(P, \varphi_\Theta(Q)) \]

- If $\Theta$ corresponds to the ample line bundle $\mathcal{L}$, $e_{W,\Theta,\ell}$ can also be seen as the pairing coming from the polarisation $\varphi_{\ell \Theta}$ or as the commutator pairing $e_{\mathcal{L}^\ell}$.
The Tate pairings on abelian varieties over finite fields

• From the exact sequence

\[ 0 \to A[\ell](\overline{\mathbb{F}}_{q^d}) \to A(\overline{\mathbb{F}}_{q^d}) \to [\ell] A(\overline{\mathbb{F}}_{q^d}) \to 0 \]

we get from Galois cohomology a connecting morphism

\[ \delta : A(\overline{\mathbb{F}}_{q^d})/\ell A(\overline{\mathbb{F}}_{q^d}) \to H^1(\text{Gal}(\overline{\mathbb{F}}_{q^d}/\mathbb{F}_{q^d}), A[\ell]); \]

• Composing with the Weil pairing, we get a bilinear application

\[ A[\ell](\overline{\mathbb{F}}_{q^d}) \times A(\overline{\mathbb{F}}_{q^d})/\ell A(\overline{\mathbb{F}}_{q^d}) \to H^1(\text{Gal}(\overline{\mathbb{F}}_{q^d}/\mathbb{F}_{q^d}), \mu_\ell) \cong \mathbb{F}_{q^d}^*/\mathbb{F}_{q^d}^\ell \cong \mu_\ell \]

where the last isomorphism comes from the Kummer sequence

\[ 1 \to \mu_\ell \to \overline{\mathbb{F}}_{q^d}^* \to \overline{\mathbb{F}}_{q^d}^* \to 1 \]

and Hilbert 90;

• Explicitely, if \( P \in A[\ell](\overline{\mathbb{F}}_{q^d}) \) and \( Q \in A(\overline{\mathbb{F}}_{q^d}) \) then the (reduced) Tate pairing is given by

\[ e_T(P, Q) = e_W(\pi(P_0) - P_0, Q) \]

where \( P_0 \) is any point such that \( P = [\ell]P_0 \) and \( \pi \) is the Frobenius of \( \overline{\mathbb{F}}_{q^d} \).
Cycles and Lang reciprocity

- Let $(A, \Theta)$ be a principally polarized abelian variety;
- To a degree 0 cycle $\sum(P_i)$ on $A$, we can associate the divisor $\sum t_{P_i}^* \Theta$ on $A$;
- The cycle $\sum(P_i)$ corresponds to a trivial divisor iff $\sum P_i = 0$ in $A$;
- If $f$ is a function on $A$ and $D = \sum(P_i)$ a cycle whose support does not contain a zero or pole of $f$, we let
  \[ f(D) = \prod f(P_i). \]

(In the following, when we write $f(D)$ we will always assume that we are in this situation.)

Theorem ([Lan58])

Let $D_1$ and $D_2$ be two cycles equivalent to 0, and $f_{D_1}$ and $f_{D_2}$ be the corresponding functions on $A$. Then

\[ f_{D_1}(D_2) = f_{D_2}(D_1) \]
The Weil and Tate pairings on abelian varieties

**Theorem**

Let $P, Q \in A[\ell]$. Let $D_P$ and $D_Q$ be two cycles equivalent to $(P) - (0)$ and $(Q) - (0)$. The Weil pairing is given by

$$e_W(P, Q) = \frac{f_{\ell D_P}(D_Q)}{f_{\ell D_Q}(D_P)}.$$

**Theorem**

Let $P \in A[\ell](\mathbb{F}_{q^d})$ and $Q \in A(\mathbb{F}_{q^d})$, and let $D_P$ and $D_Q$ be two cycles equivalent to $(P) - (0)$ and $(Q) - (0)$. The (non reduced) Tate pairing is given by

$$e_T(P, Q) = f_{\ell D_P}(D_Q).$$
Cryptographic usage of pairings on abelian varieties

- The moduli space of abelian varieties of dimension $g$ is a space of dimension $\frac{g(g + 1)}{2}$. We have more liberty to find optimal abelian varieties in function of the security parameters.
- Supersingular elliptic curves have a too small embedding degree. [RS09] says that for the current security parameters, optimal supersingular abelian varieties of small dimension are of dimension 4.
- If $A$ is an abelian variety of dimension $g$, $A[\ell]$ is a $(\mathbb{Z}/\ell\mathbb{Z})$-module of dimension $2g \Rightarrow$ the structure of pairings on abelian varieties is richer.
Theta functions
Complex abelian variety

- A complex abelian variety is of the form $A = V / \Lambda$ where $V$ is a $\mathbb{C}$-vector space and $\Lambda$ a lattice, with a polarization (actually an ample line bundle) $\mathcal{L}$ on it;
- The Chern class of $\mathcal{L}$ corresponds to a symplectic real form $E$ on $V$ such that $E(ix, iy) = E(x, y)$ and $E(\Lambda, \Lambda) \subset \mathbb{Z}$;
- The commutator pairing $e_\mathcal{L}$ is then given by $\exp(2i\pi E(\cdot, \cdot))$;
- A principal polarization on $A$ corresponds to a decomposition $\Lambda = \Omega \mathbb{Z}^g + \mathbb{Z}^g$ with $\Omega \in \mathfrak{H}_g$ the Siegel space;
- The associated Riemann form on $A$ is then given by $E(\Omega x_1 + x_2, \Omega y_1 + y_2) = {}^t x_1 \cdot y_2 - {}^t y_1 \cdot x_2$. 
Theta coordinates on abelian varieties

• Every abelian variety (over an algebraically closed field) can be described by theta coordinates of level $n > 2$ even. (The level $n$ encodes information about the $n$-torsion).
• The theta coordinates of level 2 on $A$ describe the Kummer variety of $A$.
• For instance if $A = \mathbb{C}^g / (\mathbb{Z}^g + \Omega \mathbb{Z}^g)$ is an abelian variety over $\mathbb{C}$, the theta coordinates on $A$ come from the analytic theta functions with characteristic:

$$\vartheta \left( \begin{bmatrix} a \\ b \end{bmatrix} \right)(z, \Omega) = \sum_{n \in \mathbb{Z}^g} e^{\pi i t(n+a)\Omega(n+a) + 2\pi i t(n+a)(z+b)} \quad a, b \in \mathbb{Q}^g$$

Remark

Working on level $n$ mean we take a $n$-th power of the principal polarisation. So in the following we will compute the $n$-th power of the usual Weil and Tate pairings.
The differential addition law \((k = \mathbb{C})\)

\[
\left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \theta_{i+t}(x + y) \theta_{j+t}(x - y) \right) \cdot \left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \theta_{k+t}(0) \theta_{l+t}(0) \right) = \\
\left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \theta_{-i'+t}(y) \theta_{j'+t}(y) \right) \cdot \left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \theta_{k'+t}(x) \theta_{l'+t}(x) \right).
\]

where \(\chi \in \hat{\mathbb{Z}}(2), i, j, k, l \in \mathbb{Z}(n)\)

\((i', j', k', l') = A(i, j, k, l)\)

\[
A = \frac{1}{2} \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1
\end{pmatrix}
\]
Example: differential addition in dimension 1 and in level 2

Algorithm

**Input** \( z_p = (x_0, x_1), z_Q = (y_0, y_1) \) and \( z_{P-Q} = (z_0, z_1) \) with \( z_0z_1 \neq 0 \);

\( z_0 = (a, b) \) and \( A = 2(a^2 + b^2), B = 2(a^2 - b^2) \).

**Output** \( z_{P+Q} = (t_0, t_1) \).

1. \( t'_0 = (x_0^2 + x_1^2)(y_0^2 + y_1^2)/A \)
2. \( t'_1 = (x_0^2 - x_1^2)(y_0^2 - y_1^2)/B \)
3. \( t_0 = (t'_0 + t'_1)/z_0 \)
4. \( t_1 = (t'_0 - t'_1)/z_1 \)

Return \( (t_0, t_1) \)
Cost of the arithmetic with low level theta functions \((\text{char } k \neq 2)\)

<table>
<thead>
<tr>
<th></th>
<th>Montgomery</th>
<th>Level 2</th>
<th>Jacobians coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doubling</td>
<td>(5M + 4S + 1m_0)</td>
<td>(3M + 6S + 3m_0)</td>
<td>(3M + 5S)</td>
</tr>
<tr>
<td>Mixed Addition</td>
<td>(7M + 6S + 1m_0)</td>
<td></td>
<td>(7M + 6S + 1m_0)</td>
</tr>
</tbody>
</table>

Multiplication cost in genus 1 (one step).

<table>
<thead>
<tr>
<th></th>
<th>Mumford</th>
<th>Level 2</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doubling</td>
<td>(34M + 7S)</td>
<td>(7M + 12S + 9m_0)</td>
<td>(49M + 36S + 27m_0)</td>
</tr>
<tr>
<td>Mixed Addition</td>
<td>(37M + 6S)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Multiplication cost in genus 2 (one step).
Proposition ([LR13])

- For $P \in A$ we note $z_P$ a lift to $\mathbb{C}^g$. We call $P$ a projective point and $z_P$ an affine point (because we describe them via their projective, resp affine, theta coordinates);
- We have (up to a constant)

$$f_{\lambda, p}(z) = \frac{\theta(z)}{\theta(z + \lambda z_P)} \left( \frac{\theta(z + z_P)}{\theta(z)} \right)^\lambda;$$

- So (up to a constant)

$$f_{\lambda, \mu, p}(z) = \frac{\theta(z + \lambda z_P) \theta(z + \mu z_P)}{\theta(z) \theta(z + (\lambda + \mu)z_P)}.$$
Three way addition

Proposition ([LR13])

From the affine points \(z_P, z_Q, z_R, z_{P+Q}, z_{P+R}\) and \(z_{Q+R}\) one can compute the affine point \(z_{P+Q+R}\).
(In level 2, the proposition is only valid for “generic” points).

Proof.

We can compute the three way addition using a generalised version of Riemann’s relations:

\[
\left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \vartheta_{i+t}(z_{P+Q+R}) \vartheta_{j+t}(z_P) \right) \left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \vartheta_{k+t}(z_Q) \vartheta_{l+t}(z_R) \right) = \\
\left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \vartheta_{-i'+t}(z_0) \vartheta_{j'+t}(z_{Q+R}) \right) \left( \sum_{t \in \mathbb{Z}(2)} \chi(t) \vartheta_{k'+t}(z_{P+R}) \vartheta_{l'+t}(z_{P+Q}) \right).
\]
Three way addition in dimension 1 level 2

Algorithm

**Input** The points \( x, y, z, X = y + z, Y = x + z, Z = x + y; \)

**Output** \( T = x + y + z. \)

**Return**

\[
T_0 = \frac{(aX_0 + bX_1)(Y_0Z_0 + Y_1Z_1)}{x_0(y_0z_0 + y_1z_1)} + \frac{(aX_0 - bX_1)(Y_0Z_0 - Y_1Z_1)}{x_0(y_0z_0 - y_1z_1)}
\]

\[
T_1 = \frac{(aX_0 + bX_1)(Y_0Z_0 + Y_1Z_1)}{x_1(y_0z_0 + y_1z_1)} - \frac{(aX_0 - bX_1)(Y_0Z_0 - Y_1Z_1)}{x_1(y_0z_0 - y_1z_1)}
\]
Computing the Miller function $f_{\lambda, \mu, P}(Q - (0))$

Algorithm

**Input** $\lambda P, \mu P$ and $Q$;

**Output** $f_{\lambda, \mu, P}(Q - (0))$

1. Compute $(\lambda + \mu)P, Q + \lambda P, Q + \mu P$ using normal additions and take any affine lifts $z_{(\lambda+\mu)P}, z_{Q+\lambda P}$ and $z_{Q+\mu P}$;

2. Use a three way addition to compute $z_{Q+(\lambda+\mu)P}$;

Return

$$f_{\lambda, \mu, P}(Q - (0)) = \frac{\theta(z_Q + \lambda z_P) \theta(z_Q + \mu z_P)}{\theta(z_Q) \theta(z_Q + (\lambda + \mu) z_P)} \cdot \frac{\theta(\lambda + \mu) z_P \theta(z_P)}{\theta(\lambda z_P) \theta(\mu z_P)}.$$

Lemma

The result does not depend on the choice of affine lifts in Step 2.

😊 This allow us to evaluate the Weil and Tate pairings and derived pairings;

😊 Not possible *a priori* to apply this algorithm in level 2.
The Tate pairing with Miller’s functions and theta coordinates

• Let $P \in A[\ell](\mathbb{F}_{q^d})$ and $Q \in A(\mathbb{F}_{q^d})$; choose any lift $z_P$, $z_Q$ and $z_{P+Q}$.

• The algorithm loop over the binary expansion of $\ell$, and at each step does a doubling step, and if necessary an addition step.

  - **Given** $z_{\lambda P}$, $z_{\lambda P+Q}$;
  - **Doubling** Compute $z_{2\lambda P}$, $z_{2\lambda P+Q}$ using two differential additions;
  - **Addition** Compute $(2\lambda + 1)P$ and take an arbitrary lift $z_{(2\lambda+1)P}$. Use a three way addition to compute $z_{(2\lambda+1)P+Q}$.

• At the end we have computed affine points $z_{\ell P}$ and $z_{\ell P+Q}$. Evaluating the Miller function then gives exactly the quotient of the projective factors between $z_{\ell P}$, $z_0$ and $z_{\ell P+Q}$, $z_Q$.

😊 Described this way can be extended to level 2 by using compatible additions;

😊 Three way additions and normal (or compatible) additions are quite cumbersome, is there a way to only use differential additions?
Pairings with theta functions
The Weil and Tate pairing with theta coordinates \([LR10]\)

\(P\) and \(Q\) points of \(\ell\)-torsion.

\[
\begin{array}{cccccc}
& z_0 & z_P & 2z_P & \ldots & \ell z_P = \lambda_P^0 z_0 \\
& z_Q & z_P \oplus z_Q & 2z_P + z_Q & \ldots & \ell z_P + z_Q = \lambda_P^1 z_Q \\
2z_Q & & z_P + 2z_Q & \\
\ldots & & \ldots & \\
\ell Q = \lambda_Q^0 0_A & z_P + \ell z_Q = \lambda_Q^1 z_P
\end{array}
\]

- \(e_{W,\ell}(P,Q) = \frac{\lambda_P^1 \lambda_Q^0}{\lambda_P^0 \lambda_Q^1}\).
- \(e_{T,\ell}(P,Q) = \frac{\lambda_P^1}{\lambda_Q^0}\).
Why does it works?

\[
\begin{align*}
& z_0 \quad \alpha z_p \quad \alpha^4(2z_p) \quad \ldots \quad \alpha^2(\ell z_p) = \lambda_0^0 z_0 \\
& \beta z_Q \quad \gamma(z_p \oplus z_Q) \quad \frac{\gamma^2\alpha^2}{\beta}(2z_p + z_Q) \quad \ldots \quad \frac{\gamma\alpha^{\ell(\ell-1)}}{\beta^{\ell-1}}(\ell z_p + z_Q) = \lambda_1^1 \beta z_Q \\
& \beta^4(2z_Q) \quad \frac{\gamma^2\beta^2}{\alpha}(z_p + 2z_Q) \quad \ldots \quad \ldots \\
& \beta^{\ell^2}(\ell z_Q) = \lambda_0^0 z_0 \quad \frac{\gamma^l\beta^{l(l-1)}}{\alpha^{l-1}}(z_p + \ell z_Q) = \lambda_1^1 \alpha z_p
\end{align*}
\]

We then have

\[
\begin{align*}
\lambda_0^0 = \alpha^2 \lambda_0^0, & \quad \lambda_0^1 = \beta^2 \lambda_0^0, & \quad \lambda_1^0 = \frac{\gamma^l\alpha^{(l^2)}}{\beta^l} \lambda_1^1, & \quad \lambda_1^1 = \frac{\gamma^l\beta^{l(l-1)}}{\alpha^l} \lambda_1^1, \\
e'_W(P,Q) = \frac{\lambda_0^0 \lambda_1^0}{\lambda_0^0 \lambda_1^0} = \frac{\lambda_1^0 \lambda_0^0}{\lambda_0^0 \lambda_1^0} = e_W(P,Q), & \\
e'_T(P,Q) = \frac{\lambda_0^1}{\lambda_0^0} = \frac{\gamma^l}{\alpha^l \beta^l} \frac{\lambda_1^1}{\lambda_0^1} = \frac{\gamma^l}{\alpha^l \beta^l} e_T(P,Q).
\end{align*}
\]
The case $n = 2$

- If $n = 2$ we work over the Kummer variety $K$ over $k$, so $e(P, Q) \in \overline{k}^{*, \pm 1}$.
- We represent a class $x \in \overline{k}^{*, \pm 1}$ by $x + 1/x \in \overline{k}^{*}$. We want to compute the symmetric pairing
  $$e_s(P, Q) = e(P, Q) + e(-P, Q).$$
- From $\pm P$ and $\pm Q$ we can compute $\{\pm (P + Q), \pm (P - Q)\}$ (need a square root), and from these points the symmetric pairing.
- $e_s$ is compatible with the $\mathbb{Z}$-structure on $K$ and $\overline{k}^{*, \pm 1}$.
- The $\mathbb{Z}$-structure on $\overline{k}^{*, \pm}$ can be computed as follow:
  $$(x^\ell_1 + \ell_2 + \frac{1}{x^{\ell_1 + \ell_2}}) + (x^{\ell_1 - \ell_2} + \frac{1}{x^{\ell_1 - \ell_2}}) = (x^{\ell_1} + \frac{1}{x^{\ell_1}})(x^{\ell_2} + \frac{1}{x^{\ell_2}})$$
Ate pairing

Definition

Let \( G_1 = E[\ell] \cap \text{Ker}(\pi_q - 1) \) and \( G_2 = E[\ell] \cap \text{Ker}(\pi_q - [q]) \).

Let \( \lambda \equiv q \mod \ell \), the (reduced) ate pairing is defined by

\[
a_{\lambda} : G_2 \times G_1 \to \mu_\ell, (P, Q) \mapsto f_{\lambda,P}(Q)^{(q^d - 1)/\ell}.
\]

It is non degenerate if \( \ell^2 \nmid (\lambda^k - 1) \).

We expect the Miller loop to be half the length as for the Tate pairing;

We need to work over \( \mathbb{F}_{q^d} \) rather than \( \mathbb{F}_q \) for computing Miller’s functions;

Can use twists to alleviate the problem (this was not possible with non elliptic Jacobians).
Ate pairing with theta functions

• Let \( P \in G_2 \) and \( Q \in G_1 \).
• In projective coordinates, we have \( \pi_q^d(P + Q) = \lambda^d P + Q = P + Q \);
• Unfortunately, in affine coordinates, \( \pi_q^d(z_{P+Q}) \neq \lambda^d z_P + z_Q \).
• But if \( \pi_q(z_{P+Q}) = C \ast (\lambda z_P + z_Q) \), then \( C \) is exactly the (non reduced) ate pairing!

Algorithm (Computing the ate pairing)

**Input** \( P \in G_2, Q \in G_1 \);

1. Compute \( z_Q + \lambda z_P, \lambda z_P \) using differential additions;
2. Find the projective factors \( C_1 \) and \( C_0 \) such that \( z_Q + \lambda z_P = C_1 \ast \pi(z_{P+Q}) \) and \( \lambda z_P = C_0 \ast \pi(z_P) \) respectively;

**Return** \( (C_1/C_0)^{\frac{q^d-1}{\ell}} \).
Optimal ate pairing

- Let $\lambda = m\ell = \sum c_i q^i$ be a multiple of $\ell$ with small coefficients $c_i$. ($\ell \nmid m$)
- The pairing

  $a_\lambda : G_2 \times G_1 \rightarrow \mu_\ell$

  $$(P, Q) \mapsto \left(\prod_i f_{c_i, P}(Q)^{q^i} \prod_i f_{\sum_{j>i} c_j q^j, c_i q^i, P}(Q)\right)^{(q^d - 1)/\ell}$$

  is non-degenerate when $mdq^{d-1} \neq (q^d - 1)/r \sum_i ic_i q^{i-1} \mod \ell$.
- Since $\varphi_d(q) = 0 \mod \ell$ we look at powers $q, q^2, \ldots, q^{\varphi(d)-1}$.
- We can expect to find $\lambda$ such that $c_i \approx \ell^{1/\varphi(d)}$. 
Optimal ate pairing with theta functions

Algorithm (Computing the optimal ate pairing)

**Input** \( \pi_q(P) = [q]P, \pi_q(Q) = Q, \lambda = ml = \sum c_i q^i; \)

1. Compute the \( z_Q + c_i z_P \) and \( c_i z_P \);
2. Apply Frobeniuses to obtain the \( z_Q + c_i q^i z_P, c_i q^i z_P; \)
3. Compute \( c_i q^i z_P \oplus \sum_j c_j q^j z_P \) (up to a constant) and then do a three way addition to compute \( z_Q + c_i q^i z_P + \sum_j c_j q^j z_P \) (up to the same constant);
4. Recurse until we get \( \lambda z_P = C_0 * z_P \) and \( z_Q + \lambda z_P = C_1 * z_Q; \)

**Return** \( \left( \frac{C_1}{C_0} \right)^{\frac{q^d - 1}{\ell}}. \)
The case $n = 2$

- Computing $c_i q^i z_p \pm \sum_j c_j q^j z_p$ requires a square root (very costly);
- And we need to recognize $c_i q^i z_p + \sum_j c_j q^j z_p$ from $c_i q^i z_p - \sum_j c_j q^j z_p$.
- We will use compatible additions: if we know $x, y, z$ and $x + z, y + z$, we can compute $x + y$ without a square root;
- We apply the compatible additions with $x = c_i q^i z_p$, $y = \sum_j c_j q^j z_p$ and $z = z_Q$. 
Compatible additions

• Recall that we know $x, y, z$ and $x + z, y + z$;
• From it we can compute $(x + z) \pm (y + z) = \{x + y + 2z, x - y\}$ and of course $\{x + y, x - y\}$;
• Then $x + y$ is the element in $\{x + y, x - y\}$ not appearing in the preceding set;
• Since $x - y$ is a common point, we can recover it without computing a square root.
The compatible addition algorithm in dimension 1

Algorithm

**Input**  \( x, y, Y = x + z, X = y + z; \)

1. **Computing** \( x \pm y : \)
   
   \[
   \alpha = (x_0^2 + x_1^2)(y_0^2 + y_1^2)/A \\
   \beta = (x_0^2 - x_1^2)(y_0^2 - y_1^2)/B \\
   \kappa_{00} = (\alpha + \beta), \quad \kappa_{11} = (\alpha - \beta) \\
   \kappa_{10} := x_0x_1y_0y_1/ab
   \]

2. **Computing** \((x + z) \pm (y + z) : \)

   \[
   \alpha' = (Y_0^2 + Y_1^2)(X_0^2 + X_1^2)/A \\
   \beta' = (Y_0^2 - Y_1^2)(X_0^2 - X_1^2)/B \\
   \kappa'_{00} = \alpha' + \beta', \quad \kappa'_{11} = \alpha' - \beta' \\
   \kappa'_{10} = Y_1Y_2X_1X_2/ab
   \]

**Return**  \( x + y = [\kappa_{00}(\kappa_{10}\kappa'_{00} - \kappa'_{10}\kappa_{00}), \kappa_{10}(\kappa_{10}\kappa'_{00} - \kappa'_{10}\kappa_{00}) + \kappa_{00}(\kappa_{11}\kappa'_{00} - \kappa'_{11}\kappa_{00})] \)
Performance
One step of the pairing computation

Algorithm (A step of the Miller loop with differential additions)

**Input** \( nP = (x_n, z_n); (n + 1)P = (x_{n+1}, z_{n+1}); (n + 1)P + Q = (x'_{n+1}, z'_{n+1}). \)

**Output** \( 2nP = (x_{2n}, z_{2n}); (2n + 1)P = (x_{2n+1}, z_{2n+1}); \)
\( (2n + 1)P + Q = (x'_{2n+1}, z'_{2n+1}). \)

1. \( \alpha = (x_n^2 + z_n^2); \beta = \frac{A}{B}(x_n^2 - z_n^2). \)
2. \( X_n = \alpha^2; X_{n+1} = \alpha(x_{n+1}^2 + z_{n+1}^2); X'_{n+1} = \alpha(x'_{n+1}^2 + z'_{n+1}^2); \)
3. \( Z_n = \beta(x_n^2 - z_n^2); Z_{n+1} = \beta(x_{n+1}^2 - z_{n+1}^2); Z'_{n+1} = \beta(x'_{n+1}^2 - z'_{n+1}^2); \)
4. \( x_{2n} = X_n + Z_n; x_{2n+1} = (X_{n+1} + Z_{n+1})/x_P; x'_{2n+1} = (X'_{n+1} + Z'_{n+1})/x_Q; \)
5. \( z_{2n} = \frac{a}{b}(X_n - Z_n); z_{2n+1} = (X_{n+1} - Z_{n+1})/z_P; z'_{2n+1} = (X'_{n+1} - Z'_{n+1})/z_Q; \)

**Return** \( (x_{2n}, z_{2n}); (x_{2n+1}, z_{2n+1}); (x'_{2n+1}, z'_{2n+1}). \)
Weil and Tate pairing over $\mathbb{F}_{q^d}$

\[
\begin{align*}
g &= 1 & 4M + 2m + 8S + 3m_0 \\
g &= 2 & 8M + 6m + 16S + 9m_0
\end{align*}
\]

Tate pairing with theta coordinates, $P, Q \in A[\ell](\mathbb{F}_{q^d})$ (one step)

Operations in $\mathbb{F}_q$: $M$: multiplication, $S$: square, $m$ multiplication by a coordinate of $P$ or $Q$, $m_0$ multiplication by a theta constant;

Mixed operations in $\mathbb{F}_q$ and $\mathbb{F}_{q^d}$: $M$, $m$ and $m_0$;

Operations in $\mathbb{F}_{q^d}$: $M$, $m$ and $S$.

Remark

• *Doubling step for a Miller loop with Edwards coordinates*: $9M + 7S + 2m_0$;

• *Just doubling a point in Mumford projective coordinates using the fastest algorithm* [Lan05]: $33M + 7S + 1m_0$;

• *Asymptotically the final exponentiation is more expensive than Miller’s loop, so the Weil’s pairing is faster than the Tate’s pairing!*
### Tate pairing

\[
\begin{align*}
g = 1 & \quad 1m + 2S + 2M + 2M + 1m + 6S + 3m_0 \\
g = 2 & \quad 3m + 4S + 4M + 4M + 3m + 12S + 9m_0
\end{align*}
\]

Tate pairing with theta coordinates, \( P \in A[\ell](\mathbb{F}_q), Q \in A[\ell](\mathbb{F}_{q^d}) \) (one step)

<table>
<thead>
<tr>
<th></th>
<th>Miller</th>
<th>Theta coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Doubling</td>
<td>Addition</td>
</tr>
<tr>
<td>( g = 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( d ) even</td>
<td>( 1M + 1S + 1M )</td>
<td>( 1M + 1M )</td>
</tr>
<tr>
<td>( d ) odd</td>
<td>( 2M + 2S + 1M )</td>
<td>( 2M + 1M )</td>
</tr>
<tr>
<td>( g = 2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( Q ) degenerate +</td>
<td>( 1M + 1S + 3M )</td>
<td>( 1M + 3M )</td>
</tr>
<tr>
<td>( d ) even</td>
<td></td>
<td></td>
</tr>
<tr>
<td>General case</td>
<td>( 2M + 2S + 18M )</td>
<td>( 2M + 18M )</td>
</tr>
</tbody>
</table>

\( P \in A[\ell](\mathbb{F}_q), Q \in A[\ell](\mathbb{F}_{q^d}) \) (counting only operations in \( \mathbb{F}_{q^d} \)).
Ate and optimal ate pairings

\[
\begin{align*}
g = 1 & \quad 4M + 1m + 8S + 1m + 3m_0 \\
g = 2 & \quad 8M + 3m + 16S + 3m + 9m_0
\end{align*}
\]

Ate pairing with theta coordinates, \( P \in G_2, Q \in G_1 \) (one step)

**Remark**

*Using affine Mumford coordinates in dimension 2, the hyperelliptic ate pairing costs* [Gra+07]:

**Doubling** \( 1I + 29M + 9S + 7M \)

**Addition** \( 1I + 29M + 5S + 7M \)

(where \( I \) denotes the cost of an affine inversion in \( \mathbb{F}_{q^d} \)).
Perspectives

- Look at supersingular abelian varieties in characteristic 2 (Just for fun, cryptographic applications are killed by the $L(1/4, \cdot)$ index calculus in $\mathbb{F}_{2^n}^\ast$ from A. Joux);
- Optimized implementations (FPGA, ...);
- Look at special points (degenerate divisors, ...).
Bibliography


