Arithmetic on Abelian and Kummer varieties

2013/12/03 — **PEACE** — Rennes

David Lubicz, **Damien Robert**

The context

- We want efficient arithmetic on abelian varieties (cryptography);
- We use theta functions (of level n);
- For the slides we will work with an abelian surface A, but everything work in higher dimensions g (or in dimension 1).

Problem:

- The $2^g = 4$ level two theta functions give a projective embedding of the Kummer variety $K = A/\pm 1$. Compact representation, fast arithmetic, but only(?) for scalar multiplication;
- The $4^g = 16$ level four theta functions give a projective embedding of A. Can compute any arithmetic operation, but much slower.

Questions:

- How much arithmetic descend on K? Can we compute it explicitly?
- Are there more compact/efficient representations to work directly on A? Something like level 2 + some extra information?

Outline

1. Arithmetic of theta functions
2. Compatible additions
3. Coordinates compression
4. Efficient representation
5. Formulas

The tools

Let $\mathscr{L}$ be an ample totally symmetric line bundle.

Theorem (Duplication formula)

$$\vartheta_i^{\mathscr{L}}(x+y)\vartheta_j^{\mathscr{L}}(x-y)=\sum_{\substack{u+v=i\\ u-v=j}}\vartheta_u^{\mathscr{L}^2}(x)\vartheta_v^{\mathscr{L}^2}(y)$$

- The duplication formula express the isogeny

$$\begin{aligned} f: A\times A &\longrightarrow A\times A \\ (x,y) &\longmapsto (x+y,x-y) \end{aligned} ;$$

- Gives a link between theta functions of level n and theta functions of level $2n$ (Koizumi-Kempf's formulas are a generalisation to higher level).

Riemann relations

Theorem (Application of the duplication formula)

$$\Big(\sum_{t\in Z(\overline{2})}\chi(t)\vartheta_{i+t}(x_1)\vartheta_{j+t}(y_1)\Big).\Big(\sum_{t\in Z(\overline{2})}\chi(t)\vartheta_{k+t}(u_1)\vartheta_{l+t}(v_1)\Big)=$$
$$\Big(\sum_{t\in Z(\overline{2})}\chi(t)\vartheta_{i'+t}(x_2)\vartheta_{j'+t}(y_2)\Big).\Big(\sum_{t\in Z(\overline{2})}\chi(t)\vartheta_{k'+t}(u_2)\vartheta_{l'+t}(v_2)\Big).$$

Where $x_1, y_1, u_1, v_1, z \in A(\overline{k})$ *with* $2z = x_1 + y_1 + u_1 + v_1$, $x_2 = z - x_1$, $y_2 = z - y_1$, $u_2 = z - u_1$ *and* $v_2 = z - v_1$,
for all $\chi \in \hat{Z}(\overline{2})$, $i, j, k, l, m \in Z(\overline{n})$ *with* $2m = i + j + k + l$, $i' = m - i$, $j' = m - j$, $k' = m - k$ *and* $l' = m - l$.

Remark

- *When* $4 \mid n$ *Riemann relations encode all the arithmetic of the abelian variety (Mumford's description of the corresponding modular space);*
- *When* $n = 2$ *we assume that the even theta constants are non zero, or equivalently that the embedding of the Kummer variety is projectively normal (Koizumi-Kempf).*
  $\Rightarrow A$ *is absolutely simple, and not a Jacobian of an hyperelliptic curve when* $g \geqslant 3$.

Arithmetic from Riemann relations

Given $x = (\vartheta_i(x))$ and $y = (\vartheta_i(y))$, one can recover

- All $\vartheta_i(x+y)\vartheta_j(x-y)$ when $4 \mid n$;
- All $\vartheta_i(x+y)\vartheta_j(x-y) + \vartheta_j(x+y)\vartheta_i(x-y)$ when $n = 2$.

Proposition ($2 \mid n$)

- *Given $x = (\vartheta_i(x))$, one can compute $-x = (\vartheta_{-i}(x)$ (Opposite);*
- *Given the points x, y and $x-y$, one can compute $x+y$ (Differential addition);*
- *Given the points $x_1, \ldots, x_n$ and the two by two sums $x_i + x_j$, one can recover $x_1 + \ldots + x_n$ (Multiway addition).*

Remark

The previous arithmetic actually can be defined over affine lifts of the projective theta coordinates. These lifts correspond to the lift of the projection $\mathbb{C}^g \to \mathbb{C}^g/\Lambda$ when $\overline{k} = \mathbb{C}$, or in general to a choice of projective system of compatible theta structures. This extra affine data is crucial for isogenies or pairings computations [LR10; LR13].

(Projective) additions

Given x and y, we want to compute $x+y$.

- When $4 \mid n$, we can always compute $x+y$ by using Riemann relations;
- When $n=2$, we can compute the (sub-scheme) $\{x+y, x-y\}$ as follows:
- Let $\kappa_{ij} = \vartheta_i(x+y)\vartheta_j(x-y) + \vartheta_j(x+y)\vartheta_i(x-y)$;
- The roots of $\mathfrak{P}_i(X) = X^2 - 2\frac{\kappa_{i0}}{\kappa_{00}}X + \frac{\kappa_{ii}}{\kappa_{00}}$ are $\frac{\vartheta_i(z_P+z_Q)}{\vartheta_0(z_P+z_Q)}$ and $\frac{\vartheta_i(z_P-z_Q)}{\vartheta_0(z_P-z_Q)}$;
- We recover the subscheme $\{x+y, x-y\}$ via the equation $\mathfrak{P}_\alpha(X)=0$ and the linear relations coming from
$$\begin{pmatrix} \vartheta_0(x+y) & \vartheta_0(x-y) \\ \vartheta_\alpha(x+y) & \vartheta_\alpha(x-y) \end{pmatrix} \begin{pmatrix} \vartheta_i(x-y) \\ \vartheta_i(x+y) \end{pmatrix} = \begin{pmatrix} \kappa_{0i} \\ \kappa_{\alpha i} \end{pmatrix};$$
- Recovering the set $\{x+y, x-y\}$ explicitly costs a square root in k.

Compatible additions

We work on the Kummer variety $K = A/\pm 1$.

Theorem

Let x, y, z, t be geometric points on A such that $x + y = z + t$ and $x - y \neq z - t$. Then one can compute $x + y = z + t$ on K.

Proof.

The corresponding point is just the intersection of $\{x + y, x - y\}$ and $\{z + t, z - t\}$. In practice this is just a gcd computation between two quadratic polynomials! □

Projective multiway additions

Corollary (Projective multiway addition)

Let x_0 be a point not of 2-torsion. Then from $x_1, \dots, x_n \in K$ and $x_0 + x_1, \dots, x_0 + x_n \in K$, one can compute $x_1 + \dots x_n$ and $x_0 + x_1 + \dots x_n$.

Proof.

By an easy recursion, it suffices to look at the case $n = 2$. In the previous theorem set $x = x_1, y = x_2, z = x_0 + x_1, t = -x_0 + x_2$ to recover $x_1 + x_2$, and $x = x_1, y = x_0 + x_2, z = x_2, t = x_0 + x_1$ to recover $x_0 + x_1 + x_2$. □

Remark

- *The arithmetic here works only in the projective setting, that's why the projective multiway addition needs less input than the affine multiway addition;*
- *In the $n = 2$ case above, one can also recover the point $x_0 + x_1 + x_2$ or $x_1 + x_2$ once the other is computed by using Riemann relations for the three-way addition.*

Double scalar multiplication

In a Kummer variety, how to compute $\alpha P + \beta Q$? (Think GLV/GLS). We assume that we are given P,Q and $P+Q$.

1. A Montgomery square $mP+nQ$, $(m+1)P+nQ$, $mP+(n+1)Q$, $(m+1)P+(n+1)Q$, adding the correct element to the square depending on the current bits of (α,β);
2. A cleverer way is to use a triangle (DJB);
3. But actually we only need to keep track of two elements in the square.

Example

From $nP+(m+1)Q,(n+1)P+mQ$, one can recover $nP+mQ$ by using a compatible addition with $x=nP+(m+1)Q$, $y=-Q$, $z=(n+1)P+mQ$, $t=-P$.

Remark

We expect to need to reconstruct a missing element in the square with probability $1/2$*, but when we do that we can be clever in the two elements we keep, so the probability is actually* $9/16$*. The final cost is* 2 *differential additions +* $7/16$ *compatible additions by bits.*

Multi scalar multiplication

- In a Kummer variety, we want to compute $\sum \alpha_i P_i$. (Think higher dimensional GLV/GLS).
- We assume that we are given the two by two sums $P_i + P_j$ (actually, we just need the $P_1 + P_i$, we can recover the others via compatible additions);
- The trivial way would be to use an hypercube;
- But as previously, we just need two elements in the hypercube, say $\sum m_i P_i$ and $P_1 + \sum m_i P_i$;
- At each step we do one compatible addition to recover the element we need in the hypercube, and then use it for two differential additions;
- The total cost is 2 differential additions +1 compatible addition by bits.

Isogenies and affine lifts

- $f : (x_i)_{i \in Z(\overline{\ell n})} \mapsto (x_i)_{i \in Z(\overline{n})}$ is an ℓ-isogeny between an abelian variety A given by level ℓn theta functions and an abelian variety given by level n theta functions;
- Let T_i be a basis of $A_1[\ell n]$, the kernel of this isogeny is generated by the $n T_i$;
- One can lift f to a morphism $\widetilde{f}$ on the affine lifts of the geometric points;
- Then $x \in A(\overline{k})$ is uniquely determined by the $\widetilde{f}(\widetilde{x} + \sum_{i=1}^{g} \alpha_i \widetilde{T_i})$.

Example ($g = 1$, $\ell = 3$, $n = 4$)

- $\widetilde{0}_A = (a_0, \ldots, a_{11})$, $\widetilde{T_1} = (a_1, a_2, \ldots, a_{11}, a_0)$;
- $\widetilde{f} : ((x_0, \ldots, x_{11}) \mapsto ((x_0, x_3, x_6, x_9))$;
- $\widetilde{f}(\widetilde{x} + \widetilde{T_1}) = (x_1, x_4, x_7, x_{10})$;
- $\widetilde{f}(\widetilde{x} + 2\widetilde{T_1}) = (x_2, x_5, x_8, x_{11})$.

Isogenies and differential additions

Proposition

- *From $\widetilde{f}(\widetilde{T}_i)$ and $\widetilde{f}(\widetilde{T}_i + \widetilde{T}_j)$, one can use differential additions and (affine) multi-way additions to recover all $\widetilde{f}(\sum_{i=1}^{g} \alpha_i \widetilde{T}_i)$, hence $\widetilde{0}_A$.*
- *From $\widetilde{f}(\widetilde{x})$ and $\widetilde{f}(\widetilde{x} + \widetilde{T}_i)$, one can use differential additions and (affine) multi-way additions to recover all $\widetilde{f}(\widetilde{x} + \sum_{i=1}^{g} \alpha_i \widetilde{T}_i)$, hence $\widetilde{x}$.*

Remark

This idea is at the heart of the explicit isogenies computations in [LR12; CR13; Rob10].

Point compression

Compressing coordinates:

- Level $2m$ theta null point $\Rightarrow 1$ level 2 theta null point $+g(g+1)/2$ level 2 theta points.
- Level $2m \Rightarrow 1+g$ level 2 theta functions.
- $O(1)$

Decompression:

- $O(n^g)$ differential or multi-way additions.

Remark

- *One can see a differential addition of level $2m$ as m^g differential additions of level 2; the compressed representation needing only $1+g$ differential additions of level 2 also gives a more efficient arithmetic!*
- *The same remark applies for (affine) multiway additions.*

Isogenies and projective points

- When we have just the projective points $f(x)$ and $f(T_i)$, we can compute (if $4 \mid n$) $f(x+T_i)$;
- Using differential additions, one can compute an affine lift $\widetilde{f}(x+T_i)$ up to a ℓ-root of unity ζ_i;
- By decompressing the coordinates, we recover one of the ℓ^g preimage $x_0 \in A(\overline{k})$ of $f(x)$.

What about $n=2$? We can't distinguish $f(x)$ from $f(-x)$.

- We compute $f(x \pm T_1)$ using a normal addition, we have to make a choice here (which corresponds to a choice of a sign of x);
- Once we have done this choice, we can compute $f(x+T_i)$ (projectively) exactly by using a compatible addition

$$f(x+T_i) = f(x) + f(T_i) = f(x+T_1) + f(T_i - T_1)!$$

- This is how we first used compatible additions with David;
- Since we can go back to an abelian variety, morally this mean that from the tools of normal additions, compatible additions, differential additions and multiway additions, we can recover all possible arithmetic on the Kummer variety.

The level $(2,2,\ldots,4)$ theta structure

- The level 2 theta structure will not give a projective embedding of the abelian variety A, but a level $(2,4)$ theta structure will (assuming A is absolutely simple, ...);
- The compressed representation shows that a point of A can be represented as level 2 data $(x, x+T)$ on a suitable $(1,2)$ isogenous abelian variety B, where T is a point of four torsion;
- Differential additions (or affine multiway additions) are straightforward on this representation;
- But the addition of $(x, x+T)$ and $(y, y+T)$ is just a matter of two compatible additions to recover $(x+y, x+y+T)$ since T is not of two torsion!
- Still a level $(2,4)$ polarisation is not natural, it comes from a principally polarised line bundle $\mathscr{L}$ from the action of a real endomorphism φ splitting 2;
- But in fact, if we work on B rather than A, we can work on projective coordinates, and use any point T that is not of two torsion!

An efficient representation

Definition

Let A be an abelian variety with a point $T \in A(k)$ not of two torsion, and let $K = A/\pm 1$ be the associated Kummer variety. We represent a point $x \in A(\overline{k})$ by the couple $(x, x+T) \in K^2$.

Remark

To represent $x+T$ we just need to give a root of $\mathfrak{P}_1(X)$, hence this representation needs only $1+2^g$ coordinates.

Efficient arithmetic

- Differential addition: From $(x, x+T), y, (x-y, x-y+T)$, recover $(x+y, x+y+T)$ via two level 2 differential additions;
- Addition: this uses two compatible additions (or one compatible addition + one threeway addition);
- Scalar multiplication:
  1. Do a Montgomery ladder: One doubling and two differential additions at each step (adding the same point, so some savings – $23M+13S$ by bits);
  2. Use a standard level 2 multiplication to compute $(m-1)P, mP$ ($16M+9S$ by bits) and recover $mP+T$ as a compatible addition

$$mP+T=(mP)+T=(m-1)P+(P+T);$$

- Multi scalar multiplication: likewise, do a level 2 multiscalar multiplication to compute $(\sum m_i P_i)-P_1, \sum m_i P_i$ and recover $\sum m_i P_i + T$ as

$$\sum m_i P_i + T = (\sum m_i P_i) + T = ((\sum m_i P_i) - P_1) + (P_1 + T);$$

⇒ This representation only add a small overhead compared to the level 2 representation, but allows to compute additions!

Differential addition

- Notations: $x, y, X = x + y, Y = x - y$, $0_A = (a_i)$;
- $$z_\chi^i = \Big(\sum_t \chi(t) x_{i+t} x_t\Big)\Big(\sum_t \chi(t) y_{i+t} y_t\Big) / \Big(\sum_t \chi(t) a_{i+t} a_t\Big).$$
- $$\begin{aligned} 4X_{00}Y_{00} &= z_{00}^{00} + z_{01}^{00} + z_{10}^{00} + z_{11}^{00};\\ 4X_{01}Y_{01} &= z_{00}^{00} - z_{01}^{00} + z_{10}^{00} + z_{11}^{00};\\ 4X_{10}Y_{10} &= z_{00}^{00} + z_{01}^{00} - z_{10}^{00} - z_{11}^{00};\\ 4X_{11}Y_{11} &= z_{00}^{00} - z_{01}^{00} - z_{10}^{00} + z_{11}^{00}; \end{aligned}$$

⇒ $8S + 4M + 4I = 14M + 8S$ for the differential addition (here we neglect multiplications by constants).

Remark

$\big(\sum_t \chi(t) a_{i+t} a_t\big)$ is simply the classical theta null point $\vartheta\left[\begin{smallmatrix}\chi/2\\ i/2\end{smallmatrix}\right](0,\Omega)^2$.

Normal additions

- $$\begin{aligned} 2(X_{10}Y_{00}+X_{00}Y_{10}) &= z_{00}^{10}+z_{01}^{10}; \\ 2(X_{11}Y_{01}+X_{01}Y_{11}) &= z_{00}^{10}-z_{01}^{10}; \\ 2(X_{01}Y_{00}+X_{00}Y_{01}) &= z_{00}^{01}+z_{10}^{01}; \\ 2(X_{11}Y_{10}+X_{10}Y_{11}) &= z_{00}^{01}-z_{10}^{01}; \\ 2(X_{11}Y_{00}+X_{00}Y_{11}) &= z_{00}^{11}+z_{11}^{11}; \\ 2(X_{01}Y_{10}+X_{10}Y_{01}) &= z_{00}^{11}-z_{11}^{11}; \end{aligned}$$

$\Rightarrow$ $(8S+4M)+3\times(4M+2M)=22M+8S$ to compute all the κ_{ij}.

Normal additions, explicit coordinates

- We work with the polynomial $\mathfrak{P}_\alpha = Z^2 - 2\kappa_{\alpha 0} Z + \kappa_{\alpha\alpha}\kappa_{00}$, whose roots are $Z = X_\alpha Y_0$ and $Z' = X_0 Y_\alpha$;
- We can as well assume that $Y_0 = 1$ (projective coordinates);
- The equation to solve is then

$$\begin{pmatrix} \kappa_{00} & 1 \\ Z & Z'/\kappa_{00} \end{pmatrix} \begin{pmatrix} Y_i \\ X_i \end{pmatrix} = \begin{pmatrix} \kappa_{0i} \\ \kappa_{\alpha i} \end{pmatrix};$$

- We get $X_i = (-\kappa_{0i} + \kappa_{00}\kappa_{\alpha i})/(Z' - Z)$;

⇒ $24M + 8S + I = 26M + 8S$ to compute X once we know Z.

Compatible additions

- Les $P_1 = X^2 + aX + b$ and $P_2 = X^2 + cX + d$. Then P_1 and P_2 have a common root iff $(ad - bc)(c - a) = (d - b)^2$, in this case this root is $(d - b)/(a - c)$.
- A compatible addition amount to computing a normal addition $x + y$, and finding a root of $\mathfrak{P}_\alpha$ as a common root of the polynomial $\mathfrak{P}'_\alpha$ coming from the addition of $(x + t, y + t)$;
- So for a compatible addition we need the extra computation of $\mathfrak{P}'_\alpha$ $\Rightarrow 10M + 8S$;
- The common root is
$$\frac{\kappa'_{\alpha\alpha}\kappa'_{00} - \kappa_{\alpha\alpha}\kappa_{00}}{2(\kappa'_{\alpha 0} - \kappa_{\alpha 0})};$$

⇒ $36M + 16S + 2M + 1I = 41M + 16S$;

- In the $(x, x + t)$ representation, once we have computed $x + y$ via a compatible addition, we can reuse some operations in the computation of $x + y + t$, we gain $-4S - 6M - 4S - 2M$ for a cost of $33M + 8S$;
- Still, it may be more efficient to use a three way addition to compute $x + y + t$ rather than another compatible addition, since this cost $12M + 8I = 32M$;
- I have not used the projectivity all the time, probably a lot to gain...

Bibliography

R. Cosset and D. Robert. “An algorithm for computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2”. Accepted for publication at Mathematics of computation. Oct. 2013. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143 (cit. on p. 13).

D. Lubicz and D. Robert. “Efficient pairing computation with theta functions”. In: *Algorithmic Number Theory*. Lecture Notes in Comput. Sci. 6197 (July 2010). Ed. by G. Hanrot, F. Morain, and E. Thomé. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. DOI: 10.1007/978-3-642-14518-6_21. URL: http://www.normalesup.org/~robert/pro/publications/articles/pairings.pdf. Slides http://www.normalesup.org/~robert/publications/slides/2010-07-ants.pdf (cit. on p. 6).

D. Lubicz and D. Robert. “Computing isogenies between abelian varieties”. In: *Compositio Mathematica* 148.05 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG]. URL: http://www.normalesup.org/

~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062 (cit. on p. 13).

D. Lubicz and D. Robert. “A generalisation of Miller’s algorithm and applications to pairing computations on abelian varieties”. Mar. 2013. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192 (cit. on p. 6).

D. Robert. “Fonctions thêta et applications à la cryptographie”. PhD thesis. Université Henri-Poincarré, Nancy 1, France, July 2010. URL: http://www.normalesup.org/~robert/pro/publications/academic/phd.pdf. Slides http://www.normalesup.org/~robert/pro/publications/slides/2010-07-phd.pdf, TEL: tel-00528942. (Cit. on p. 13).