Counting points on genus-3 hyperelliptic curves with real multiplication

Simon Abelard

Joint work with P. Gaudry and P.-J. Spaenlehauer

July 18, 2018

```
                                                       E,C,
/* CARAMBA */                                         c,r,
                                                     u,l,
                                                    e,s,
                                                   i=5,
   d[5],Q[999             ]={0};main(n            ){for
 (;i--;e=scanf("%"       "d",d+i));for(C          =*d;
 ++i<C        ;++Q[     i*i%          C],c=      i[Q]?
c:i);           for(;i  --;)           for(u      =C;u
--;n            +=!u*Q  [l%C           ],e+=      Q[(C
+l*l-         c*s*      s%C)           %C])        for(
  l=i,s=u,r=4;r;E=       i*l+c*u*s,s=(u*l          +i*s)
   %C,l=E%C+r             --[d]);printf             ("%d"
                                                     "\n",
                                                      (e+n*
                                                        n)/2
  /* cc caramba.c; echo f3 f2 f1 f0 p | ./a.out */     -C);}
```

What? Where?

Our favorite geometrical object:

Hyperelliptic curves $\mathcal{C}$ given by equation $Y^2 = f(X)$.
Polynomial $f \in \mathbb{F}_q[X]$ monic squarefree of degree $2g+1$.
The genus of the curve is the integer g.

Point counting

If $\mathcal{C}$ defined over $\mathbb{F}_q$, $P = (x, y)$ is rational if $(x, y) \in (\mathbb{F}_q)^2$.
Let $\mathcal{C}(\mathbb{F}_{q^i}) = \left\{(x, y) \in (\mathbb{F}_{q^i})^2 \,|y^2 = f(x)\right\} \cup \{\infty\}$,
Point counting over $\mathbb{F}_q$ is computing the local ζ function of $\mathcal{C}$:

$$\zeta(s) = \exp\left(\sum_k \#\mathcal{C}(\mathbb{F}_{q^k})\frac{s^k}{k}\right) \overset{thm}{=} \frac{\Lambda(s)}{(1-s)(1-qs)}$$

With $\Lambda \in \mathbb{Z}[T]$ of degree $2g$ with bounded coefficients.
In practice, we want the coeffs of the polynomial Λ, or simply $\Lambda(1)$.

Why counting points?

Cryptographic purposes (genus ≤ 2)

Curves provide groups with no known subexponential algorithm for DLP. Size of group determines security level [*Pohlig-Hellman*].

In other algorithms

Primality proving with proven complexity [*Adleman-Huang*]
Deterministic factorization in $\mathbb{F}_q[X]$? (ongoing [*Kayal, Poonen*])

Arithmetic geometry

Conjectures in number theory e.g. Sato-Tate in genus ≥ 2.
L-functions associated: $L(s, \mathcal{C}) = \sum_p A_p / p^s$ with $A_p = \#\mathcal{C}(\mathbb{F}_p)/\sqrt{p}$.
Computing them relies on point-counting primitives.

Algorithms for point counting

Let $\mathcal{C}$ be a curve over $\mathbb{F}_q$ with $q = p^n$.

p-adic methods

- elliptic curves: *Satoh'99, Mestre'00*
- hyp. curves: *Kedlaya'01, Denef-Vercauteren'06, Lauder-Wan'06*
- more general curves: *Castryck-Denef-Vercauteren'06, Tuitman'17*

Asymptotic complexity: polynomial in g, exponential in $\log p$.

ℓ-adic methods

Elliptic curves (*Schoof'85*) extended to Abelian varieties (*Pila'90*).
Asymptotic complexity: exponential in g, polynomial in $\log q$.

Algorithms for point counting

Let $\mathcal{C}$ be a curve over $\mathbb{F}_q$ with $q = p^n$.

p-adic methods

- elliptic curves: *Satoh'99, Mestre'00*
- hyp. curves: *Kedlaya'01, Denef-Vercauteren'06, Lauder-Wan'06*
- more general curves: *Castryck-Denef-Vercauteren'06, Tuitman'17*

Asymptotic complexity: polynomial in g, exponential in $\log p$.

ℓ-adic methods

Elliptic curves (*Schoof'85*) extended to Abelian varieties (*Pila'90*).
Asymptotic complexity: exponential in g, polynomial in $\log q$.

No classical polynomial algorithm in both g and $\log p$,
Average polynomial for reductions modulo p of a curve (*Harvey'14*).

Algorithms for point counting

Let $\mathcal{C}$ be a curve over $\mathbb{F}_q$ with $q = p^n$.

p-adic methods

- elliptic curves: *Satoh'99, Mestre'00*
- hyp. curves: *Kedlaya'01, Denef-Vercauteren'06, Lauder-Wan'06*
- more general curves: *Castryck-Denef-Vercauteren'06, Tuitman'17*

Asymptotic complexity: polynomial in g, exponential in $\log p$.

ℓ-adic methods

Elliptic curves (*Schoof'85*) extended to Abelian varieties (*Pila'90*).
Asymptotic complexity: exponential in g, polynomial in $\log q$.

No classical polynomial algorithm in both g and $\log p$,
Average polynomial for reductions modulo p of a curve (*Harvey'14*).
Exponential algorithms are also efficient in practice (*Sutherland'09*).

Schoof's algorithm in genus ≤ 2

Pila's algorithm is highly impractical (23-bit exponent for $\log q$).

Asymptotic complexities

Genus	Complexity	Authors
$g = 1$	$\widetilde{O}(\log^4 q)$	Schoof-Elkies-Atkin ($\sim$ 1990)
$g = 2$	$\widetilde{O}(\log^8 q)$	Gaudry-Harley-Schost (2000)
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	Gaudry-Kohel-Smith (2011)

Schoof's algorithm in genus ≤ 2

Pila's algorithm is highly impractical (23-bit exponent for $\log q$).

Asymptotic complexities

Genus	Complexity	Authors
$g = 1$	$\widetilde{O}(\log^4 q)$	Schoof-Elkies-Atkin ($\sim$ 1990)
$g = 2$	$\widetilde{O}(\log^8 q)$	Gaudry-Harley-Schost (2000)
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	Gaudry-Kohel-Smith (2011)

Records

Genus 1: 16645-bit Jacobian using SEA (*Sutherland'10*).
Genus 2: 256-bit cryptographic Jacobian (*Gaudry-Schost'12*).
Genus 2 with RM: 1024-bit Jacobian (*Gaudry-Kohel-Smith'11*).

Schoof's algorithm in genus ≤ 2

Pila's algorithm is highly impractical (23-bit exponent for $\log q$).

Asymptotic complexities

Genus	Complexity	Authors
$g = 1$	$\widetilde{O}(\log^4 q)$	Schoof-Elkies-Atkin ($\sim$ 1990)
$g = 2$	$\widetilde{O}(\log^8 q)$	Gaudry-Harley-Schost (2000)
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	Gaudry-Kohel-Smith (2011)

Records

Genus 1: 16645-bit Jacobian using SEA (*Sutherland'10*).
Genus 2: 256-bit cryptographic Jacobian (*Gaudry-Schost'12*).
Genus 2 with RM: 1024-bit Jacobian (*Gaudry-Kohel-Smith'11*).

What about genus 3? With RM?

Schoof's algorithm in genus 3

What about genus 3? Asymptotic complexity? Practicality?

Main results

For $\mathcal{C}$ a genus-3 hyperelliptic curve with explicit RM, we give a Las Vegas algorithm to compute the local zeta function in $\widetilde{O}(\log^6 q)$ bit ops. Without RM, the algorithm runs in $\widetilde{O}(\log^{14} q)$ bit ops.
Experiments: $g = 3$ and $p = 2^{64} - 59$, 192-bit RM-Jacobian.

Complexities

Genus	Complexity	Authors
$g = 1$	$\widetilde{O}(\log^4 q)$	Schoof-Elkies-Atkin
$g = 2$	$\widetilde{O}(\log^8 q)$	Gaudry-Schost
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	Gaudry-Kohel-Smith
$g = 3$	$\widetilde{O}(\log^{14} q)$	A.-Gaudry-Spaenlehauer
$g = 3$ with RM	$\widetilde{O}(\log^6 q)$	A.-Gaudry-Spaenlehauer

Jacobians, real multiplication

Let $\mathcal{C} : y^2 = f(x)$ be a hyperelliptic curve of genus g over $\mathbb{F}_q$.

Mumford form

Any $D \in \operatorname{Jac} \mathcal{C}(\overline{\mathbb{F}}_q)$ can be represented as a sum of $w \leq g$ points.
Unique representation of $D \in \operatorname{Jac} \mathcal{C}(\mathbb{F}_q)$ by $\langle u, v \rangle$ in $\mathbb{F}_q[X]^2$ such that:

- $\deg u = w$
- $u | v^2 - f$

Jacobians, real multiplication

Let $\mathcal{C} : y^2 = f(x)$ be a hyperelliptic curve of genus g over $\mathbb{F}_q$.

Mumford form

Any $D \in \operatorname{Jac}\mathcal{C}(\overline{\mathbb{F}}_q)$ can be represented as a sum of $w \leq g$ points.
Unique representation of $D \in \operatorname{Jac}\mathcal{C}(\mathbb{F}_q)$ by $\langle u, v\rangle$ in $\mathbb{F}_q[X]^2$ such that:

- $\deg u = w$
- $u | v^2 - f$

Explicit real multiplication

We say that $\mathcal{C}$ has RM by an order $\mathbb{Z}[\eta]$ if $\mathbb{Z}[\eta] \hookrightarrow \operatorname{End}(J)$ with $\mathbb{Q}(\eta)$ is a degree-g totally real field.
Over finite fields all curves have RM by $\psi = \pi + \pi^\vee$.
We ask for an explicit expression of $\eta(P - \infty) = \langle u, v\rangle$.

A prototype of Schoof's algorithm

Let $\mathcal{C} : y^2 = f(x)$ be a hyperelliptic curve over $\mathbb{F}_q$.
Let J be its Jacobian and g its genus.

1. (Hasse-Weil) bounds on coeffs of $\Lambda \Rightarrow$ compute $\Lambda \bmod \ell$
2. number and size of ℓ bounded by $O(g \log q)$
3. ℓ-torsion $J[\ell] = \{D \in J | \ell D = 0\} \simeq (\mathbb{Z}/\ell\mathbb{Z})^{2g}$
4. action on Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ on $J[\ell]$ yields $\Lambda \bmod \ell$

Algorithm *a la* Schoof

For sufficiently many primes ℓ

Describe I_ℓ the ideal of ℓ-torsion

Compute $\chi \bmod \ell$ by testing char. eq. of π in I_ℓ

Deduce $\Lambda \bmod \ell$

Recover Λ by CRT

A prototype of Schoof's algorithm

Let $\mathcal{C} : y^2 = f(x)$ be a hyperelliptic curve over $\mathbb{F}_q$.
Let J be its Jacobian and g its genus.

1. (Hasse-Weil) bounds on coeffs of $\Lambda \Rightarrow$ compute $\Lambda \bmod \ell$
2. number and size of ℓ bounded by $O(g \log q)$
3. ℓ-torsion $J[\ell] = \{D \in J | \ell D = 0\} \simeq (\mathbb{Z}/\ell\mathbb{Z})^{2g}$
4. action on Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ on $J[\ell]$ yields $\Lambda \bmod \ell$

Algorithm *a la* Schoof

For sufficiently many primes ℓ
  Describe I_ℓ the ideal of ℓ-torsion
  Compute $\chi \bmod \ell$ by testing char. eq. of π in I_ℓ
  Deduce $\Lambda \bmod \ell$
Recover Λ by CRT

A prototype of Schoof's algorithm

Let $\mathcal{C} : y^2 = f(x)$ be a hyperelliptic curve over $\mathbb{F}_q$.
Let J be its Jacobian and g its genus.

1. (Hasse-Weil) bounds on coeffs of $\Lambda \Rightarrow$ compute Λ mod ℓ
2. number and size of ℓ bounded by $O(g \log q)$
3. ℓ-torsion $J[\ell] = \{D \in J | \ell D = 0\} \simeq (\mathbb{Z}/\ell\mathbb{Z})^{2g}$
4. action on Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ on $J[\ell]$ yields Λ mod ℓ

Algorithm *a la* Schoof

For sufficiently many primes ℓ
    Describe I_ℓ the ideal of ℓ-torsion
    Compute χ mod ℓ by testing char. eq. of π in I_ℓ
    Deduce Λ mod ℓ
Recover Λ by CRT

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
- Using RM to split $J[\ell]$ for $g = 3$.

Handling the system

- Writing and bounding degrees of input system.
- Solving the system.
- And in practice ?

Modelling the ℓ-torsion

To model the ℓ-torsion, consider a divisor D, compute ℓD formally,
Then write a system equivalent to $\ell D = 0$ in J, and “solve” it.

Modelling the ℓ-torsion

To model the ℓ-torsion, consider a divisor D, compute ℓD formally,
Then write a system equivalent to $\ell D = 0$ in J, and "solve" it.

Bad news

In genus 3, the ideal $J[\ell]$ has degree ℓ^6.
The size of the torsion impacts the complexity of solving part.
Hard to go lower than quadratic in the degree, i.e. ℓ^{12} field ops.
$\Rightarrow$ Even $\ell = 5$ already seems out of reach…

Modelling the ℓ-torsion

To model the ℓ-torsion, consider a divisor D, compute ℓD formally,
Then write a system equivalent to $\ell D = 0$ in J, and "solve" it.

Bad news

In genus 3, the ideal $J[\ell]$ has degree ℓ^6.
The size of the torsion impacts the complexity of solving part.
Hard to go lower than quadratic in the degree, i.e. ℓ^{12} field ops.
$\Rightarrow$ Even $\ell = 5$ already seems out of reach...

Wishful thinking

Can we find curves with smaller ℓ-torsion? No.
Can we split $J[\ell]$ into small (π-stable) subspaces?
(i.e. does Λ factors modulo ℓ?)
For curves with explicit RM, it is possible.

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.

Handling the system

- Writing and bounding degrees of input system.
- Solving the system.
- And in practice ?

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.

Handling the system

- Writing and bounding degrees of input system.
- Solving the system.
- And in practice ?

Tuning Schoof's algorithm using RM

Let $\mathcal{C}$ be a genus-3 hyperelliptic curve with explicit RM by $\mathbb{Z}[\eta]$.

Replacing χ_π

Let $\psi = \pi + \pi^\vee$, $\psi \in \mathbb{Z}[\eta]$ so we write $\psi = \alpha + \beta\eta + \gamma\eta^2$.
We write a system to express Λ from knowledge of η and (α, β, γ).
Replace $\chi_\pi(D) = 0 \bmod J[\ell]$ by $\psi\pi(D) = \pi^2(D) + q^2 D$.
Advantage: α, β, γ are in $O(\sqrt{q})$ vs Weil's bounds in $O(q^{3/2})$.

Tuning Schoof's algorithm using RM

Let $\mathcal{C}$ be a genus-3 hyperelliptic curve with explicit RM by $\mathbb{Z}[\eta]$.

Replacing χ_π

Let $\psi = \pi + \pi^\vee$, $\psi \in \mathbb{Z}[\eta]$ so we write $\psi = \alpha + \beta\eta + \gamma\eta^2$.
We write a system to express Λ from knowledge of η and (α, β, γ).
Replace $\chi_\pi(D) = 0 \bmod J[\ell]$ by $\psi\pi(D) = \pi^2(D) + q^2 D$.
Advantage: α, β, γ are in $O(\sqrt{q})$ vs Weil's bounds in $O(q^{3/2})$.

Splitting $J[\ell]$

For some ℓ, decompose multiplication as $\ell = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3$ in $\mathbb{Z}[\eta]$,
Find $\epsilon_i = a_i + b_i\eta + c_i\eta^2$ in $\mathfrak{p}_i$ with $|a_i|$, $|b_i|$, $|c_i|$ in $O(\ell^{1/3})$.
The action of π on all the $J[\epsilon_i]$ uniquely determines ψ hence Λ.
Advantage: model $\operatorname{Ker} \epsilon_i$ instead of $J[\ell]$, degree $O(\ell^2)$ vs ℓ^6.

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
- Solving the system.
- And in practice ?

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
- Solving the system.
- And in practice ?

Cantor's division polynomials (*Cantor'94*)

Problem

We have to compute ℓD or $\epsilon_i(D)$ to write our systems.
Recall $\epsilon_i = a_i + b_i\eta + c_i\eta^2$ with η known $\Rightarrow$ scalar multiplication ?

Cantor's division polynomials (*Cantor'94*)

Problem

We have to compute ℓD or $\epsilon_i(D)$ to write our systems.
Recall $\epsilon_i = a_i + b_i\eta + c_i\eta^2$ with η known $\Rightarrow$ scalar multiplication ?

For $n > g$ and $P = (x, y)$ a generic point on $\mathcal{C}$, $n(P - \infty)$ equals

$$\left\langle X^g + \frac{d_{g-1}(x)}{d_g(x)}X^{g-1} + \cdots + \frac{d_0(x)}{d_g(x)}, y\left(\frac{e_{g-1}(x)}{e_g(x)}X^{g-1} + \cdots + \frac{e_0(x)}{e_g(x)}\right)\right\rangle$$

The d_i and e_i are called Cantor's n-division polynomials.
In genus 1 and 2, it is know that their degrees are in $O(n^2)$.

Cantor's division polynomials (*Cantor'94*)

Problem

We have to compute ℓD or $\epsilon_i(D)$ to write our systems.
Recall $\epsilon_i = a_i + b_i\eta + c_i\eta^2$ with η known $\Rightarrow$ scalar multiplication ?

For $n > g$ and $P = (x, y)$ a generic point on $\mathcal{C}$, $n(P - \infty)$ equals

$$\left\langle X^g + \frac{d_{g-1}(x)}{d_g(x)}X^{g-1} + \cdots + \frac{d_0(x)}{d_g(x)}, y\left(\frac{e_{g-1}(x)}{e_g(x)}X^{g-1} + \cdots + \frac{e_0(x)}{e_g(x)}\right)\right\rangle$$

The d_i and e_i are called Cantor's n-division polynomials.
In genus 1 and 2, it is know that their degrees are in $O(n^2)$.

Theorem (A.-Gaudry-Spaenlehauer)

In genus 3, Cantor's n-division polynomials have degrees in $O(n^2)$.

Bounding degrees

We need to solve the system $\epsilon_i(D) = 0$, let us write it.

Bounding degrees

We need to solve the system $\epsilon_i(D) = 0$, let us write it.

Our polynomial systems

Write $\epsilon_i(P_1 - \infty) + \epsilon_i(P_2 - \infty) = -\epsilon_i(P_3 - \infty)$:

$$\tilde{d}_1(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_1(x_3) = 0,$$
$$\tilde{d}_2(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_2(x_3) = 0,$$
$$\tilde{d}_3(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_3(x_3) = 0.$$

Bounding degrees

We need to solve the system $\epsilon_i(D) = 0$, let us write it.

Our polynomial systems

Write $\epsilon_i(P_1 - \infty) + \epsilon_i(P_2 - \infty) = -\epsilon_i(P_3 - \infty)$:

$$\begin{aligned} \tilde{d}_1(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_1(x_3) &= 0, \\ \tilde{d}_2(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_2(x_3) &= 0, \\ \tilde{d}_3(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_3(x_3) &= 0. \end{aligned}$$

Recall that $\epsilon_i = a_i + b_i\eta + c_i\eta^2$ amounts to multiplication by $\ell^{1/3}$.
Cantor's polynomials $\Rightarrow$ degrees of the d_i's and $\tilde{d}_i$'s are in $O(\ell^{2/3})$.

Bounding degrees

We need to solve the system $\epsilon_i(D) = 0$, let us write it.

Our polynomial systems

Write $\epsilon_i(P_1 - \infty) + \epsilon_i(P_2 - \infty) = -\epsilon_i(P_3 - \infty)$:

$$\begin{aligned}
\tilde{d}_1(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_1(x_3) &= 0,\\
\tilde{d}_2(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_2(x_3) &= 0,\\
\tilde{d}_3(x_1, x_2, y)d_3(x_3) - \tilde{d}_3(x_1, x_2)d_3(x_3) &= 0.
\end{aligned}$$

Recall that $\epsilon_i = a_i + b_i\eta + c_i\eta^2$ amounts to multiplication by $\ell^{1/3}$.
Cantor's polynomials $\Rightarrow$ degrees of the d_i's and $\tilde{d}_i$'s are in $O(\ell^{2/3})$.

Remark: without splitting $J[\ell]$, degrees would be in $O(\ell^2)$.

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
  $\rightarrow$ Cantor's n-division polynomials.
- Solving the system.
- And in practice ?

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
  $\rightarrow$ Cantor's n-division polynomials.
- Solving the system.
- And in practice ?

Solving the systems, in theory

Successive elimination by resultants

Input system is trivariate of degree d in each variable.

Compute tri- then bi-variate resultants to get a triangular system.

Final complexity in $\widetilde{O}(d^6)$ field operations.

Solving the systems, in theory

Successive elimination by resultants

Input system is trivariate of degree d in each variable.
Compute tri- then bi-variate resultants to get a triangular system.
Final complexity in $\widetilde{O}(d^6)$ field operations.

Complexities

For ℓ inert, $d = O(\ell^2)$ and $J[\ell]$ is computed in $\widetilde{O}(\ell^{12})$ field ops.
For ℓ totally split, $d = O(\ell^{2/3})$, cost decreased to $\widetilde{O}(\ell^4)$ field ops.
Overall complexities of $\widetilde{O}(\log^{14} q)$ in general and $\widetilde{O}(\log^6 q)$ with RM.

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
  $\rightarrow$ Cantor's n-division polynomials.
- Solving the system.
  $\rightarrow$ Successive resultants.
- And in practice ?

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
  $\rightarrow$ Cantor's n-division polynomials.
- Solving the system.
  $\rightarrow$ Successive resultants.
- And in practice ?

A genus-3 family with explicit RM

An RM family [Mestre'91,Tautz-Top-Verberkmoes'91]

Family $\mathcal{C}_t : y^2 = x^7 - 7x^5 + 14x^3 - 7x + t$ with $t \in \mathbb{F}_q$.
$\longrightarrow$ hyperelliptic curves of genus 3, with explicit RM.

A genus-3 family with explicit RM

An RM family [Mestre'91,Tautz-Top-Verberkmoes'91]

Family $\mathcal{C}_t : y^2 = x^7 - 7x^5 + 14x^3 - 7x + t$ with $t \in \mathbb{F}_q$.
$\longrightarrow$ hyperelliptic curves of genus 3, with explicit RM.

An explicit endomorphism [Kohel-Smith'06]

Plus, $\mathbb{Z}[\eta] \cong \mathbb{Z}[2\cos(2\pi/7)] \subset \mathbb{Q}(\zeta_7)$ and η has explicit expression:
For $P = (x, y)$ a generic point on $\mathcal{C}$,

$$\eta(P - \infty) = \left\langle X^2 + \frac{11}{2}xX + x^2 - \frac{16}{9}, y \right\rangle.$$

A practical example

$\mathcal{C} : y^2 = x^7 - 7x^5 + 14x^3 - 7x + 42$ over $\mathbb{F}_p$ with $p = 2^{64} - 59$.

Retrieving modular information

With general (non-RM related) techniques: Λ modulo $12 = 3 \times 4$.
Smallest totally-split prime: Λ modulo $\ell = 13$.

Our plan

Describing the ℓ-torsion

- How to describe the ℓ-torsion ?
  $\rightarrow$ Write $\ell D = 0$ and solve this system.
- Using RM to split $J[\ell]$ for $g = 3$.
  $\rightarrow$ Split it in the RM case.

Handling the system

- Writing and bounding degrees of input system.
  $\rightarrow$ Cantor's n-division polynomials.
- Solving the system.
  $\rightarrow$ Successive resultants.
- And in practice ?
  $\rightarrow$ Gröbner bases and final collision search.

From theory to practice

Timing estimates for resultants

Evaluation/Interpolation: many not-so-small univariate resultants.

ℓ	#res	Deg	Cost (NTL)	Cost (FLINT)
13	525M	16,000	1,850 days	735 days
29	12.8G	80,000	310,000 days	190,000 days

Recovering modular information (F4,FGLM in Magma)

mod ℓ^k	#var	degree bounds	time	memory
2	—	—	—	—
4 (inert^2)	6	15	1 min	negl.
3 (inert)	5	55	14 days	140 GB
$13 = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3$	5	52	3×3 days	41 GB

A practical example

$\mathcal{C} : y^2 = x^7 - 7x^5 + 14x^3 - 7x + 42$ over $\mathbb{F}_p$ with $p = 2^{64} - 59$.

Retrieving modular information

With general (non-RM related) techniques: Λ modulo $12 = 3 \times 4$.
Smallest totally-split prime: $\ell = 13$

We deduce Λ modulo $m = 156$, still far from sufficient...

A practical example

$\mathcal{C} : y^2 = x^7 - 7x^5 + 14x^3 - 7x + 42$ over $\mathbb{F}_p$ with $p = 2^{64} - 59$.

Retrieving modular information

With general (non-RM related) techniques: Λ modulo $12 = 3 \times 4$.
Smallest totally-split prime: $\ell = 13$

We deduce Λ modulo $m = 156$, still far from sufficient. . .

Finishing the computation

Testing $\psi\pi(D) = \pi^2(D) + qD$ in J (not in $J[\ell]$), by collision search.
[Matsuo-Chao-Tsujii'02,Gaudry-Schost'04,Galbraith-Ruprai'09].
Main drawback: exponential complexity in $O(q^{3/4}/m^{3/2})$
Advantages: memory efficient, massively run in parallel.
In our experiments, it represents 105 CPU-days.

Conclusion

Complexities

	Genus 3 hyperelliptic	with RM
Object to model	ℓ-torsion $J[\ell]$	Ker ϵ_i where $\ell = \prod \epsilon_i$
Equation	$\ell D = 0$	$\epsilon_i(D) = 0$
Degrees	$O(\ell^2)$	$O(\ell^{2/3})$
Complexity	$\widetilde{O}\left((\log q)^{14}\right)$	$\widetilde{O}((\log q)^6)$

Conclusion

Complexities

	Genus 3 hyperelliptic	with RM
Object to model	ℓ-torsion $J[\ell]$	Ker ϵ_i where $\ell = \prod \epsilon_i$
Equation	$\ell D = 0$	$\epsilon_i(D) = 0$
Degrees	$O(\ell^2)$	$O(\ell^{2/3})$
Complexity	$\widetilde{O}\left((\log q)^{14}\right)$	$\widetilde{O}((\log q)^6)$

Experiments

We count points in a 192-bit hyperelliptic Jacobian with RM.
Previously: 183-bit by Sutherland (generic group methods).
Both are for particular cases, although RM is less likely.

Conclusion

Complexities

	Genus 3 hyperelliptic	with RM
Object to model	ℓ-torsion $J[\ell]$	Ker ϵ_i where $\ell = \prod \epsilon_i$
Equation	$\ell D = 0$	$\epsilon_i(D) = 0$
Degrees	$O(\ell^2)$	$O(\ell^{2/3})$
Complexity	$\widetilde{O}\left((\log q)^{14}\right)$	$\widetilde{O}((\log q)^6)$

Experiments

We count points in a 192-bit hyperelliptic Jacobian with RM.
Previously: 183-bit by Sutherland (generic group methods).
Both are for particular cases, although RM is less likely.
Further: $\ell = 7$ (ramified), $\ell = 29$ (next totally split, hot topic)

Ongoing and future work

Villard's algorithm for bivariate resultant (ISSAC 18)

Genus	Usual resultants	Villard's algorithm
$g = 2$	$\widetilde{O}(\log^8 q)$	$\widetilde{O}((\log q)^{8-2/\omega})$
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	$\widetilde{O}((\log q)^{5-1/\omega})$
$g = 3$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}((\log q)^{14-4/\omega})$
$g = 3$ with RM	$\widetilde{O}(\log^6 q)$	$\widetilde{O}((\log q)^{6-4/(3\omega)})$

Ongoing and future work

Villard's algorithm for bivariate resultant (ISSAC 18)

Genus	Usual resultants	Villard's algorithm
$g = 2$	$\widetilde{O}(\log^8 q)$	$\widetilde{O}((\log q)^{8-2/\omega})$
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	$\widetilde{O}((\log q)^{5-1/\omega})$
$g = 3$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}((\log q)^{14-4/\omega})$
$g = 3$ with RM	$\widetilde{O}(\log^6 q)$	$\widetilde{O}((\log q)^{6-4/(3\omega)})$

- Modular equations ([Martindale et al., Milio]) hope for RM-SEA ?
- Non-hyperelliptic curves ?
- Hyperelliptic curves of greater genus ?

Ongoing and future work

Villard's algorithm for bivariate resultant (ISSAC 18)

Genus	Usual resultants	Villard's algorithm
$g = 2$	$\widetilde{O}(\log^8 q)$	$\widetilde{O}((\log q)^{8-2/\omega})$
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	$\widetilde{O}((\log q)^{5-1/\omega})$
$g = 3$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}((\log q)^{14-4/\omega})$
$g = 3$ with RM	$\widetilde{O}(\log^6 q)$	$\widetilde{O}((\log q)^{6-4/(3\omega)})$

- Modular equations ([Martindale et al., Milio]) hope for RM-SEA ? Large objects and ongoing research already in genus 2.
- Non-hyperelliptic curves ?
- Hyperelliptic curves of greater genus ?

Ongoing and future work

Villard's algorithm for bivariate resultant (ISSAC 18)

Genus	Usual resultants	Villard's algorithm
$g = 2$	$\widetilde{O}(\log^8 q)$	$\widetilde{O}((\log q)^{8-2/\omega})$
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	$\widetilde{O}((\log q)^{5-1/\omega})$
$g = 3$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}((\log q)^{14-4/\omega})$
$g = 3$ with RM	$\widetilde{O}(\log^6 q)$	$\widetilde{O}((\log q)^{6-4/(3\omega)})$

- Modular equations ([Martindale et al., Milio]) hope for RM-SEA ?
  Large objects and ongoing research already in genus 2.
- Non-hyperelliptic curves ?
  Need analogues and bound for degrees of Cantor's polynomials.
- Hyperelliptic curves of greater genus ?

Ongoing and future work

Villard's algorithm for bivariate resultant (ISSAC 18)

Genus	Usual resultants	Villard's algorithm
$g = 2$	$\widetilde{O}(\log^8 q)$	$\widetilde{O}((\log q)^{8-2/\omega})$
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	$\widetilde{O}((\log q)^{5-1/\omega})$
$g = 3$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}((\log q)^{14-4/\omega})$
$g = 3$ with RM	$\widetilde{O}(\log^6 q)$	$\widetilde{O}((\log q)^{6-4/(3\omega)})$

- Modular equations ([Martindale et al., Milio]) hope for RM-SEA ?
  Large objects and ongoing research already in genus 2.
- Non-hyperelliptic curves ?
  Need analogues and bound for degrees of Cantor's polynomials.
- Hyperelliptic curves of greater genus ?
  Work in progress, hope for complexity in $O_g(\log^8 q)$.

Thanks for your attention