Counting points on hyperelliptic curves in large characteristic: algorithms and complexity

Simon Abelard
Université de Lorraine
September 7, 2018

/* CARAMBA */

d[5],Q[999];
for (i=5, 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 --;)

/* cc caramba.c; echo f3 f2 f1 f0 p | ./a.out */

/C Caramba C; echo f3 f2 f1 f0 p | ./a.out */
## An example

How many solutions of $Y^2 = X^7 - 7X^5 + 14X^3 - 7X + 1$?
An example

How many solutions of $Y^2 = X^7 - 7X^5 + 14X^3 - 7X + 1$?

But what is a solution? Where does it live?
Solutions in $\mathbb{Z}$: diophantine equations, undecidable.
An example

How many solutions of $Y^2 = X^7 - 7X^5 + 14X^3 - 7X + 1$?

But what is a solution? Where does it live?
Solutions in $\mathbb{Z}$: diophantine equations, undecidable.

Our problem

Count solutions of $f(X, Y) = 0$ in a finite field $\mathbb{F}_{p^n}$. 
An example

How many solutions of $Y^2 = X^7 - 7X^5 + 14X^3 - 7X + 1$?

But what is a solution? Where does it live? Solutions in $\mathbb{Z}$: diophantine equations, undecidable.

Our problem

Count solutions of $f(X, Y) = 0$ in a finite field $\mathbb{F}_{p^n}$.

Naive approach: try all possibilities for $(x, y) \in \mathbb{F}_{p^n}^2$. When $p$ large (hundreds of bits), not the best idea.
Complexity of point-counting

Parameters of the problem

Equation $Y^2 = f(X)$ with $f$ polynomial over $\mathbb{F}_{p^n}$.
Input size: $\deg f \times n \log p$.

**Question**: dependency on $n$, $p$ and $\deg f$?

**Holy grail**: polynomial-time algorithm in input size.

Naive approach exponential in all.

Partly polynomial-time approaches

We will see algorithms polynomial either $n \log p$ or in $\deg f$.
No classical algorithm polynomial (yet) in all (quantum by [Kedlaya’05]).
When fixed $f$ and many $p$’s, polynomial **on average** [Harvey’14].
Our favorite geometrical object

The case of hyperelliptic curves

Count solutions of $Y^2 = f(X)$ with $f \in \mathbb{F}_q[X]$ monic squarefree. Assume $\deg f = 2g + 1$, call $g$ the genus of the curve.

Equation of hyperelliptic curve $C$, solutions are points on $C$.

Curve of equation $Y^2 = X^5 - 2X^4 - 7X^3 + 8X^2 + 12X$
Let $C$ be a hyperelliptic curve of genus $g$.

**Weil conjectures to the rescue**

Point counting over $\mathbb{F}_q$ is computing the local $\zeta$ function of $C$:

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

Where polynomial $\Lambda$ has degree $2g$ and integer coefficients.
Let $C$ be a hyperelliptic curve of genus $g$.

**Weil conjectures to the rescue**

Point counting over $\mathbb{F}_q$ is computing the local $\zeta$ function of $C$:

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

Where polynomial $\Lambda$ has degree $2g$ and integer coefficients.

**Point counting**

Input: $f \in \mathbb{F}_q[X]$ defining a hyperelliptic curve $Y^2 = f(X)$.  
Output: the polynomial $\Lambda$.  

Simon Abelard  
Point counting  
September 7, 2018
Point counting II

Let \( C \) be a hyperelliptic curve of genus \( g \).

Weil conjectures to the rescue

Point counting over \( \mathbb{F}_q \) is computing the local \( \zeta \) function of \( C \):

\[
\zeta(s) = \exp \left( \sum_k \#C(\mathbb{F}_{q^k}) \frac{s^k}{k} \right) \xrightarrow{\text{thm}} \frac{\Lambda(s)}{(1 - s)(1 - qs)}.
\]

Where polynomial \( \Lambda \) has degree \( 2g \) and integer coefficients.

Point counting

Input: \( f \in \mathbb{F}_q[X] \) defining a hyperelliptic curve \( Y^2 = f(X) \).
Output: the polynomial \( \Lambda \).

Example \( C : Y^2 = X^7 - 7X^5 + 14X^3 - 7X + 1 \) defined over \( \mathbb{F}_{23} \).
The associated \( \Lambda \) is 12167\( X^6 - 198 \)\( X^3 + 1 \).
A first application
Why counting points?

**Cryptographic purposes (genus \( \leq 2 \))**

Curves provide groups with no known subexponential algorithm for DLP. Size of group determines security level [Pohlig-Hellman’78].

**In other algorithms**

Primality proving with proven complexity [Adleman-Huang’01]. Deterministic factorization in \( \mathbb{F}_q[X] \)? (ongoing [Kayal’06, Poonen’17])

**Arithmetic geometry**

Conjectures in number theory e.g. Sato-Tate in genus \( \geq 2 \).

\( L \)-functions associated: \( L(s, C) = \sum_p A_p/p^s \) with \( A_p = \#C(\mathbb{F}_p)/\sqrt{p} \).

Computing them relies on point-counting primitives.
Algorithms for point counting

Let $C$ 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 $g$ and $n$, exponential in $\log p$.

$\ell$-adic methods

Elliptic curves (Schoof’85) extended to Abelian varieties (Pila’90).
Asymptotic complexity: polynomial in $\log p$ and $n$, exponential in $g$. 

Schoof’s algorithm in genus $\leq 2$

[Pila’90] is polynomial but with 23-bit exponent for $\log q$ when $g = 2$.

### Asymptotic complexities

<table>
<thead>
<tr>
<th>Genus</th>
<th>Complexity</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = 1$</td>
<td>$\tilde{O}(\log^4 q)$</td>
<td>Schoof-Elkies-Atkin (~ 1990)</td>
</tr>
<tr>
<td>$g = 2$</td>
<td>$\tilde{O}(\log^8 q)$</td>
<td>Gaudry-Harley-Schost (2000)</td>
</tr>
<tr>
<td>$g = 2$ with RM</td>
<td>$\tilde{O}(\log^5 q)$</td>
<td>Gaudry-Kohel-Smith (2011)</td>
</tr>
</tbody>
</table>

RM: real multiplication
Schoof’s algorithm in genus \( \leq 2 \)

[Pila’90] is polynomial but with 23-bit exponent for \( \log q \) when \( g = 2 \).

### Asymptotic complexities

<table>
<thead>
<tr>
<th>Genus</th>
<th>Complexity</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g = 1 )</td>
<td>( \tilde{O}(\log^4 q) )</td>
<td>Schoof-Elkies-Atkin (~ 1990)</td>
</tr>
<tr>
<td>( g = 2 )</td>
<td>( \tilde{O}(\log^8 q) )</td>
<td>Gaudry-Harley-Schost (2000)</td>
</tr>
<tr>
<td>( g = 2 ) with RM</td>
<td>( \tilde{O}(\log^5 q) )</td>
<td>Gaudry-Kohel-Smith (2011)</td>
</tr>
</tbody>
</table>

RM: real multiplication

### Practical results

In genus 1, SEA record with \( p \) a 16645-bit prime (Sutherland’10).
In genus 2, heavy computations yield 256-bit cryptographic Jacobian.
In genus 2 with RM, can go up to 1024-bit Jacobians.
Schoof’s algorithm in genus $\leq 2$

[Pila’90] is polynomial but with 23-bit exponent for $\log q$ when $g = 2$.

<table>
<thead>
<tr>
<th>Genus</th>
<th>Complexity</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = 1$</td>
<td>$\tilde{O}(\log^4 q)$</td>
<td>Schoof-Elkies-Atkin (~ 1990)</td>
</tr>
<tr>
<td>$g = 2$</td>
<td>$\tilde{O}(\log^8 q)$</td>
<td>Gaudry-Harley-Schost (2000)</td>
</tr>
<tr>
<td>$g = 2$ with RM</td>
<td>$\tilde{O}(\log^5 q)$</td>
<td>Gaudry-Kohel-Smith (2011)</td>
</tr>
</tbody>
</table>

RM: real multiplication

Practical results

In genus 1, SEA record with $p$ a 16645-bit prime (Sutherland’10).
In genus 2, heavy computations yield 256-bit cryptographic Jacobian.
In genus 2 with RM, can go up to 1024-bit Jacobians.

What about genus 3?
Main results

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

Complexities

<table>
<thead>
<tr>
<th>Genus</th>
<th>Complexity</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = 1$</td>
<td>$\tilde{O}(\log^4 q)$</td>
<td>Schoof-Elkies-Atkin</td>
</tr>
<tr>
<td>$g = 2$</td>
<td>$\tilde{O}(\log^8 q)$</td>
<td>Gaudry-Schost</td>
</tr>
<tr>
<td>$g = 2$ with RM</td>
<td>$\tilde{O}(\log^5 q)$</td>
<td>Gaudry-Kohel-Smith</td>
</tr>
<tr>
<td>$g = 3$</td>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>this thesis</td>
</tr>
<tr>
<td>$g = 3$ with RM</td>
<td>$\tilde{O}(\log^6 q)$</td>
<td>this thesis</td>
</tr>
</tbody>
</table>
## Asymptotic complexities

<table>
<thead>
<tr>
<th>Authors (year)</th>
<th>Complexity</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pila (1990)</td>
<td>$O\left((\log q)^{O(g)}\right)$</td>
<td>Abelian varieties</td>
</tr>
<tr>
<td>Huang-Ierardi (1998)</td>
<td>$O\left((\log q)^{O(1)}\right)$</td>
<td>Plane curves</td>
</tr>
<tr>
<td>Adleman-Huang (2001)</td>
<td>$O\left((\log q)^{O(1)}\right)$</td>
<td>Abelian varieties</td>
</tr>
<tr>
<td>Adleman-Huang (2001)</td>
<td>$O\left((\log q)^{O(g^2 \log g)}\right)$</td>
<td>Hyperelliptic curves</td>
</tr>
<tr>
<td>this thesis</td>
<td>$O_g \left((\log q)^{O(g)}\right)$</td>
<td>Hyperelliptic curves</td>
</tr>
<tr>
<td>this thesis</td>
<td>$\tilde{O}_g \left(((\log q)^8\right)$</td>
<td>with explicit RM</td>
</tr>
</tbody>
</table>
A prototype of Schoof’s algorithm

Let $C : y^2 = f(x)$ 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 $\Lambda \mod \ell$
2. $\ell$-torsion $J[\ell] = \{ D \in J | \ell D = 0 \} \cong (\mathbb{Z}/\ell \mathbb{Z})^{2g}$
3. action on Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ on $J[\ell]$ yields $\Lambda \mod \ell$

Algorithm a la Schoof

For sufficiently many primes $\ell$
- Describe $I_\ell$ the ideal of $\ell$-torsion
- Compute action of $\pi$ on $I_\ell$
- Deduce $\Lambda \mod \ell$
- Recover $\Lambda$ by CRT
A prototype of Schoof’s algorithm

Let $C : y^2 = f(x)$ 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 $\Lambda \mod \ell$
2. $\ell$-torsion $J[\ell] = \{D \in J | \ell D = 0\} \simeq (\mathbb{Z}/\ell\mathbb{Z})^{2g}$
3. action on Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ on $J[\ell]$ yields $\Lambda \mod \ell$

Algorithm a la Schoof

For sufficiently many primes $\ell$
- Describe $I_\ell$ the ideal of $\ell$-torsion
- Compute action of $\pi$ on $I_\ell$
- Deduce $\Lambda \mod \ell$
- Recover $\Lambda$ by CRT
A prototype of Schoof’s algorithm

Let $C : y^2 = f(x)$ 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 $\Lambda \mod \ell$
2. $\ell$-torsion $J[\ell] = \{ D \in J | \ell D = 0 \} \simeq (\mathbb{Z}/\ell\mathbb{Z})^{2g}$
3. action on Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ on $J[\ell]$ yields $\Lambda \mod \ell$

Algorithm a la Schoof

For sufficiently many primes $\ell$

- Describe $I_\ell$ the ideal of $\ell$-torsion
- Compute action of $\pi$ on $I_\ell$
- Deduce $\Lambda \mod \ell$
- Recover $\Lambda$ by CRT
Real multiplication

Explicit real multiplication

Famous endomorphisms: scalar multiplications and Frobenius $\pi$. Ask for additional endomorphism $\eta$ with explicit expression. Then $\mathbb{Z}[\eta] \hookrightarrow \text{End}(J)$ and we say $C$ has RM by $\mathbb{Z}[\eta]$.

Real multiplication: $\mathbb{Z}[\eta]$ is in a totally real number field.
Real multiplication

Explicit real multiplication

Famous endomorphisms: scalar multiplications and Frobenius $\pi$. Ask for additional endomorphism $\eta$ with explicit expression. Then $\mathbb{Z}[\eta] \hookrightarrow \text{End}(J)$ and we say $C$ has RM by $\mathbb{Z}[\eta]$.

Real multiplication: $\mathbb{Z}[\eta]$ is in a totally real number field.

An RM family (Mestre’91, Tautz-Top-Verberkmoes’91, Kohel-Smith’06)

Family $C_t : y^2 = x^7 - 7x^5 + 14x^3 - 7x + t$ with $t \in \mathbb{F}_q$.

$\longrightarrow$ hyperelliptic curves of genus 3.
Real multiplication

Explicit real multiplication

Famous endomorphisms: scalar multiplications and Frobenius \( \pi \).
Ask for additional endomorphism \( \eta \) with explicit expression.
Then \( \mathbb{Z}[\eta] \hookrightarrow \text{End}(J) \) and we say \( C \) has RM by \( \mathbb{Z}[\eta] \).

Real multiplication: \( \mathbb{Z}[\eta] \) is in a totally real number field.

An RM family (Mestre’91, Tautz-Top-Verberkmoes’91, Kohel-Smith’06)

Family \( C_t : y^2 = x^7 - 7x^5 + 14x^3 - 7x + t \) with \( t \in \mathbb{F}_q \).
\( \rightarrow \) hyperelliptic curves of genus 3.

For \( P = (x, y) \) generic point on \( C \), \( \eta(P - \infty) = P_+ + P_- - 2\infty \) with

\[
P_\pm = \left(-\frac{11}{4}x \pm \sqrt{\frac{105}{16}x^2 + \frac{16}{9}}, y\right).
\]
Real multiplication

Explicit real multiplication

Famous endomorphisms: scalar multiplications and Frobenius $\pi$. Ask for additional endomorphism $\eta$ with explicit expression. Then $\mathbb{Z}[\eta] \hookrightarrow \text{End}(J)$ and we say $C$ has RM by $\mathbb{Z}[\eta]$.

Real multiplication: $\mathbb{Z}[\eta]$ is in a totally real number field.

An RM family (Mestre’91, Tautz-Top-Verberkmoes’91, Kohel-Smith’06)

Family $C_t : y^2 = x^7 - 7x^5 + 14x^3 - 7x + t$ with $t \in \mathbb{F}_q$.

$\longrightarrow$ hyperelliptic curves of genus 3.

For $P = (x, y)$ generic point on $C$, $\eta(P - \infty) = P_+ + P_- - 2\infty$ with

$$P_\pm = \left(-\frac{11}{4}x \pm \sqrt{\frac{105}{16}x^2 + \frac{16}{9}}, y\right).$$

Element $\eta$ has minimal polynomial $X^3 + X^2 - 2X - 1$. 
Directions


Chapter VII of the manuscript, to be submitted.
### Contributions

<table>
<thead>
<tr>
<th>$g = 3$</th>
<th>arbitrary $g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>$O_g((\log q)^{O(g)})$</td>
</tr>
<tr>
<td>$\tilde{O}(\log^6 q)$</td>
<td>$\tilde{O}_g(\log^8 q)$</td>
</tr>
</tbody>
</table>
A one-slide summary

Contributions

<table>
<thead>
<tr>
<th>$g = 3$</th>
<th>hyperelliptic $\tilde{O}(\log^{14} q)$</th>
<th>with RM $\tilde{O}(\log^6 q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>arbitrary $g$</td>
<td>$O_g((\log q)^{O(g)})$</td>
<td>$\tilde{O}_g(\log^8 q)$</td>
</tr>
</tbody>
</table>

All our results are based on 3 steps:

- **modelling** (subgroups of) the $\ell$-torsion by polynomial systems
- **bounding** their sizes (number of variables, degrees)
- **solving** them (and bounding complexity)
A one-slide summary

Contributions

<table>
<thead>
<tr>
<th>$g = 3$</th>
<th>arbitrary $g$</th>
<th>hyperelliptic</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>$O_g((\log q)^{O(g)})$</td>
<td>$\tilde{O}(\log^{6} q)$</td>
<td>$\tilde{O}_g(\log^{8} q)$</td>
</tr>
</tbody>
</table>

All our results are based on 3 steps:
- **modelling** (subgroups of) the $\ell$-torsion by polynomial systems
- **bounding** their sizes (number of variables, degrees)
- **solving** them (and bounding complexity)

Keys to each result

Genus 3: use RM to split the torsion $\Rightarrow$ decrease the degrees.
Genus $g$: different modelling, exploit multihomogeneity.
Genus $g$ with RM: combine both approaches.
A one-slide summary

Contributions

<table>
<thead>
<tr>
<th>$g = 3$</th>
<th>arbitrary $g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>$O_g((\log q)^{O(g)})$</td>
</tr>
<tr>
<td>$\tilde{O}(\log^6 q)$</td>
<td>$\tilde{O}_g(\log^8 q)$</td>
</tr>
</tbody>
</table>

All our results are based on 3 steps:

- **modelling** (subgroups of) the $\ell$-torsion by polynomial systems
- **bounding** their sizes (number of variables, degrees)
- **solving** them (and bounding complexity)

Keys to each result

Genus 3: use RM to split the torsion $\Rightarrow$ decrease the degrees.
Genus $g$: different modelling, exploit multihomogeneity.
Genus $g$ with RM: combine both approaches.
Plan

1. Introduction
2. Genus 3
3. Arbitrary $g$
4. RM in any genus
Counting points on genus-3 hyperelliptic curves

Contents

- Model the $\ell$-torsion
- Use RM to split $J[\ell]$  
- Model the ‘parts’ of $J[\ell]$  
- Bound size of input systems  
- Solve them with resultants  
- Practical results
Modelling the $\ell$-torsion

To model the $\ell$-torsion, consider a divisor $D = \sum_{i=1}^{g} (P_i - \infty)$. Compute $\ell D = \sum_{i=1}^{g} \ell (P_i - \infty)$ formally.

Then write a system equivalent to $\ell D = 0$ in $J$, and ‘solve’ it.
Modelling the $\ell$-torsion

To model the $\ell$-torsion, consider a divisor $D = \sum_{i=1}^{g}(P_i - \infty)$. Compute $\ell D = \sum_{i=1}^{g} \ell(P_i - \infty)$ 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 $\ell^6$. Complexity bound: square of the degree, i.e. $\ell^{12}$ field ops. $\Rightarrow$ Even $\ell = 5$ already seems out of reach...
Modelling the $\ell$-torsion

To model the $\ell$-torsion, consider a divisor $D = \sum_{i=1}^{g} (P_i - \infty)$. Compute $\ell D = \sum_{i=1}^{g} \ell (P_i - \infty)$ 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 $\ell^6$. Complexity bound: square of the degree, i.e. $\ell^{12}$ field ops. $\Rightarrow$ Even $\ell = 5$ already seems out of reach…

Wishful thinking

Can we split $J[\ell]$ into small ($\pi$-stable) subspaces? For curves with explicit RM, it is possible.
Tuning Schoof’s algorithm using RM

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

### Splitting $J[\ell]$

For totally split $\ell$, decompose $\ell = p_1p_2p_3$ in $\mathbb{Z}[\eta]$.

Find well-chosen $\epsilon_i$ in $p_i$ (i.e. of ‘size’ $\ell^{1/3}$).

The action of $\pi$ on all the Ker $\epsilon_i$ uniquely determines $\Lambda$ mod $\ell$.

**Advantage:** model Ker $\epsilon_i$ instead of $J[\ell]$, degree $O(\ell^2)$ vs $\ell^6$. 
Cantor’s division polynomials (Cantor’94)

Problem

We have to compute $\ell D$ or $\epsilon_i(D)$ to write our systems. The $\epsilon_i$ are ‘close to’ multiplication by $\ell^{1/3} \Rightarrow$ scalar multiplication?
Cantor’s division polynomials (*Cantor’94*)

**Problem**

We have to compute $\ell D$ or $\epsilon_i(D)$ to write our systems. The $\epsilon_i$ are ‘close to’ multiplication by $\ell^{1/3} \Rightarrow$ scalar multiplication?

**Answer: Cantor’s $n$-division polynomials**

For $n > g$ and $P = (x, y)$ a generic point on $\mathcal{C}$, $n(P - \infty)$ is described by $2g + 2$ univariate polynomials in $x$.

In genus 1 and 2, it is known that their degrees are in $O(n^2)$. 
Cantor’s division polynomials (Cantor’94)

Problem
We have to compute $\ell D$ or $\epsilon_i(D)$ to write our systems. The $\epsilon_i$ are ‘close to’ multiplication by $\ell^{1/3} \Rightarrow$ scalar multiplication.

Answer: Cantor’s $n$-division polynomials
For $n > g$ and $P = (x, y)$ a generic point on $C$, $n(P - \infty)$ is described by $2g + 2$ univariate polynomials in $x$.

In genus 1 and 2, it is known that their degrees are in $O(n^2)$.

Quadratic bound (this thesis)
In genus 3, Cantor’s $n$-division polynomials have degrees in $O(n^2)$. 
Counting points on genus-3 hyperelliptic curves

Contents

- Model the $\ell$-torsion
- Use RM to split $J[\ell]$
- Model the ‘parts’ of $J[\ell]$
- Bound size of input systems
- Solve them with resultants
- Practical results
Counting points on genus-3 hyperelliptic curves

Contents
- Model the $\ell$-torsion
- Use RM to split $J[\ell]$  
- Model the ‘parts’ of $J[\ell]$  
- Bound size of input systems  
- Solve them with resultants  
- Practical results
Successive elimination by resultants

System modelling kernel: trivariate with degrees bounded by some $d$. Compute tri- then bi-variate resultants to put in triangular form. Final complexity in $\tilde{O}(d^6)$ field operations.
Solving the systems, in theory

Successive elimination by resultants
System modelling kernel: trivariate with degrees bounded by some $d$. Compute tri- then bi-variate resultants to put in triangular form. Final complexity in $\tilde{O}(d^6)$ field operations.

Complexities
For $\ell$ inert, $d = O(\ell^2)$ and $J[\ell]$ is computed in $\tilde{O}(\ell^{12})$ field ops. For $\ell$ totally split, $d = O(\ell^{2/3})$ and cost decreased to $\tilde{O}(\ell^4)$ field ops. (The $\epsilon_i$ amount to multiplication by $\ell^{1/3}$)
Solving the systems, in theory

Successive elimination by resultants

System modelling kernel: trivariate with degrees bounded by some $d$. Compute tri- then bi-variate resultants to put in triangular form. Final complexity in $\tilde{O}(d^6)$ field operations.

Complexities

For $\ell$ inert, $d = O(\ell^2)$ and $J[\ell]$ is computed in $\tilde{O}(\ell^{12})$ field ops. For $\ell$ totally split, $d = O(\ell^{2/3})$ and cost decreased to $\tilde{O}(\ell^4)$ field ops. (The $\epsilon_i$ amount to multiplication by $\ell^{1/3}$)

Overall complexities of $\tilde{O}(\log^{14} q)$ in general and $\tilde{O}(\log^6 q)$ with RM.
A practical example

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

Retrieving modular information

With general (non-RM related) techniques: \( \Lambda \) modulo 12 = 3 × 4.
Smallest totally-split prime: \( \Lambda \) modulo \( \ell = 13 \).
From theory to practice

**Timing estimates for resultants**

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

<table>
<thead>
<tr>
<th>ℓ</th>
<th>Cost (NTL)</th>
<th>Cost (FLINT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1,850 days</td>
<td>735 days</td>
</tr>
<tr>
<td>29</td>
<td>310,000 days</td>
<td>190,000 days</td>
</tr>
</tbody>
</table>
From theory to practice

Timing estimates for resultants

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

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>Cost (NTL)</th>
<th>Cost (FLINT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1,850 days</td>
<td>735 days</td>
</tr>
<tr>
<td>29</td>
<td>310,000 days</td>
<td>190,000 days</td>
</tr>
</tbody>
</table>

Successful attempt (F4, FGLM in Magma)

<table>
<thead>
<tr>
<th>mod ( \ell^k )</th>
<th>#var</th>
<th>degree bounds</th>
<th>time</th>
<th>memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>4 (inert(^2))</td>
<td>6</td>
<td>15</td>
<td>1 min</td>
<td>negl.</td>
</tr>
<tr>
<td>3 (inert)</td>
<td>5</td>
<td>55</td>
<td>14 days</td>
<td>140 GB</td>
</tr>
<tr>
<td>13 = p_1p_2p_3</td>
<td>5</td>
<td>52</td>
<td>3 × 3 days</td>
<td>41 GB</td>
</tr>
</tbody>
</table>
A practical example

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

Retrieving modular information

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

We deduce \( \Lambda \) modulo \( m = 156 \), still far from sufficient...
A practical example

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

Retrieving modular information

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

We deduce \( \Lambda \) modulo \( m = 156 \), still far from sufficient...

Finishing the computation

Action of \( \pi \) on \( J \) (not on \( J[\ell] \)), by collision search.
[Matsuo-Chao-Tsujii’02,Gaudry-Schost’04,Galbraith-Ruprai’09].
Main drawback: \textit{exponential} complexity.
Advantages: memory efficient, \textit{massively run in parallel}.
And a factor \( 156^{3/2} \approx 1950 \) speed-up via modular info.
In our experiments, it represents \( 105 \) CPU-days done in a few hours.
## Summary of hyperelliptic genus-3 case

### Complexities

<table>
<thead>
<tr>
<th>Object to model</th>
<th>Genus 3 hyperelliptic</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation</td>
<td>$\ell$-torsion $J[\ell]$</td>
<td>Ker $\epsilon_i$ where $\ell = \prod \epsilon_i$</td>
</tr>
<tr>
<td>Degrees</td>
<td>$\ell D = 0$</td>
<td>$\epsilon_i(D) = 0$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$O(\ell^2)$</td>
<td>$O(\ell^{2/3})$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$\tilde{O}((\log q)^{14})$</td>
<td>$\tilde{O}((\log q)^6)$</td>
</tr>
</tbody>
</table>

### 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.
Summary of hyperelliptic genus-3 case

Complexities

<table>
<thead>
<tr>
<th>Object to model</th>
<th>Genus 3 hyperelliptic</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation</td>
<td>$\ell$-torsion $J[\ell]$</td>
<td>Ker $\epsilon_i$ where $\ell = \prod \epsilon_i$</td>
</tr>
<tr>
<td>Degrees</td>
<td>$\ell D = 0$</td>
<td>$\epsilon_i(D) = 0$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$O(\ell^2)$</td>
<td>$O(\ell^{2/3})$</td>
</tr>
<tr>
<td></td>
<td>$\tilde{O}((\log q)^{14})$</td>
<td>$\tilde{O}((\log q)^{6})$</td>
</tr>
</tbody>
</table>

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.
Summary of hyperelliptic genus-3 case

Complexities

<table>
<thead>
<tr>
<th>Object to model</th>
<th>Genus 3 hyperelliptic</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation</td>
<td>$\ell$-torsion $J[\ell]$</td>
<td>$\ker \epsilon_i$ where $\ell = \prod \epsilon_i$</td>
</tr>
<tr>
<td>Degrees</td>
<td>$\ell D = 0$</td>
<td>$\epsilon_i(D) = 0$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$O(\ell^2)$</td>
<td>$O(\ell^{2/3})$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$\tilde{O}((\log q)^{14})$</td>
<td>$\tilde{O}((\log q)^6)$</td>
</tr>
</tbody>
</table>

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.

→ genus-3 point-counting in large characteristic is challenging.
# Perspective on Schoof’s algorithm for $g \leq 3$

<table>
<thead>
<tr>
<th>Genus</th>
<th>Usual resultants</th>
<th>Villard’s algorithm</th>
<th>With $\omega = 2.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = 2$</td>
<td>$\tilde{O}(\log^8 q)$</td>
<td>$\tilde{O}((\log q)^{8-2/\omega})$</td>
<td>$\tilde{O}((\log q)^{7.3})$</td>
</tr>
<tr>
<td>$g = 2$ RM</td>
<td>$\tilde{O}(\log^5 q)$</td>
<td>$\tilde{O}((\log q)^{5-1/\omega})$</td>
<td>$\tilde{O}((\log q)^{4.6})$</td>
</tr>
<tr>
<td>$g = 3$</td>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>$\tilde{O}((\log q)^{14-4/\omega})$</td>
<td>$\tilde{O}((\log q)^{12.6})$</td>
</tr>
<tr>
<td>$g = 3$ RM</td>
<td>$\tilde{O}(\log^6 q)$</td>
<td>$\tilde{O}((\log q)^{6-4/(3\omega)})$</td>
<td>$\tilde{O}((\log q)^{5.5})$</td>
</tr>
</tbody>
</table>

Further improvements:
- Extension of the SEA algorithm using modular polynomials.
- Work of Milio and Martindale, in particular in RM case.

Still large objects (both degrees and coefficients).

Ongoing in genus 2, not tomorrow in genus 3.
Perspective on Schoof’s algorithm for $g \leq 3$

### Villard’s algorithm for bivariate resultant (ISSAC 2018)

<table>
<thead>
<tr>
<th>Genus $g$</th>
<th>Usual resultants</th>
<th>Villard’s algorithm</th>
<th>With $\omega = 2.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = 2$</td>
<td>$\tilde{O}(\log^8 q)$</td>
<td>$\tilde{O}((\log q)^{8-2/\omega})$</td>
<td>$\tilde{O}((\log q)^{7.3})$</td>
</tr>
<tr>
<td>$g = 2$ RM</td>
<td>$\tilde{O}(\log^5 q)$</td>
<td>$\tilde{O}((\log q)^{5-1/\omega})$</td>
<td>$\tilde{O}((\log q)^{4.6})$</td>
</tr>
<tr>
<td>$g = 3$</td>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>$\tilde{O}((\log q)^{14-4/\omega})$</td>
<td>$\tilde{O}((\log q)^{12.6})$</td>
</tr>
<tr>
<td>$g = 3$ RM</td>
<td>$\tilde{O}(\log^6 q)$</td>
<td>$\tilde{O}((\log q)^{6-4/(3\omega)})$</td>
<td>$\tilde{O}((\log q)^{5.5})$</td>
</tr>
</tbody>
</table>

### Further improvements

Extension of the SEA algorithm using **modular polynomials**.
Work of Milio and Martindale, in particular in RM case.
Still large objects (both degrees and coefficients).
Ongoing in genus 2, not tomorrow in genus 3.
Plan

1. Introduction
2. Genus 3
3. Arbitrary $g$
4. RM in any genus
Hyperelliptic point-counting in any genus

Strategy

- Extend degree bounds for Cantor’s polynomials
- New modelling for $J[\ell]$ with multihomogeneous structure
- Exploit multihomogeneity with geometric resolution

Complexity result

<table>
<thead>
<tr>
<th>Hyperelliptic</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O}(\log^{14} q)$</td>
<td>$\tilde{O}(\log^6 q)$</td>
</tr>
<tr>
<td>$O_g((\log q)^{O(g)})$</td>
<td>$\tilde{O}_\eta(\log^8 q)$</td>
</tr>
</tbody>
</table>
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$.

Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- extend degree-bounds on Cantor’s polynomials to any $g$
For $\ell > g$ and $P = (x, y)$ a generic point on $C$, Recall that $\ell(P - \infty)$ is given by Cantor’s polynomials.

Cubic bound for any $g$ (this thesis)

Cantor’s $\ell$-division polynomials have degrees in $O_g(\ell^3)$. 
For \( \ell > g \) and \( P = (x, y) \) a generic point on \( C \),
Recall that \( \ell(P - \infty) \) is given by Cantor’s polynomials.

Cubic bound for any \( g \) (this thesis)
Cantor’s \( \ell \)-division polynomials have degrees in \( O_g(\ell^3) \).

Conjecture: quadratic bound
Cantor proved two of the polynomials had degrees \( g \ell^2 + O_g(1) \).
Experiments: the degrees of Cantor’s polynomials are consecutive.
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$.

Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- need to bound the degrees of Cantor’s polynomials
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$.

Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- need to bound the degrees of Cantor’s polynomials
- degrees grow at each composition of $\ell(P_i - \infty) + \ell(P_j - \infty)$
Another look at the $\ell$-torsion

Writing $\ell D = 0$

Still write $D = P_1 + \cdots + P_g - g \infty$ and compute $\ell(P_i - \infty)$. 
Another look at the \( \ell \)-torsion

Writing \( \ell D = 0 \)

Still write \( D = P_1 + \cdots + P_g - g\infty \) and compute \( \ell(P_i - \infty) \).
Adding the \( \ell(P_i - \infty) \) is avoided by different modelling.
But this introduces additional variables.
Another look at the $\ell$-torsion

**Writing $\ell D = 0$**

Still write $D = P_1 + \cdots + P_g - g\infty$ and compute $\ell(P_i - \infty)$. Adding the $\ell(P_i - \infty)$ is avoided by different modelling. But this introduces additional variables.

**Our polynomial system**

Degrees are bounded by $O_g(\ell^3)$ (Cantor’s polynomials). About $g^2$ equations in $g^2$ variables $\Rightarrow$ Bézout bound in $\ell g^2$. $\Rightarrow$ seems hard to improve previous bound in $(\log q)^{O(g^2)}$. But not all these variables appear with high degrees.
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$. Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- need to bound the degrees of Cantor’s polynomials
- degrees grow at each composition of $\ell(P_i - \infty) + \ell(P_j - \infty)$

$\Rightarrow$ Different model, more variables but multihomogeneous structure.
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$. Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- need to bound the degrees of Cantor’s polynomials
- degrees grow at each composition of $\ell(P_i - \infty) + \ell(P_j - \infty)$

$\Rightarrow$ Different model, more variables but multihomogeneous structure.

Solving the system

Resultants: exponential degree growth.
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$. Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- need to bound the degrees of Cantor’s polynomials
- degrees grow at each composition of $\ell(P_i - \infty) + \ell(P_j - \infty)$

$\Rightarrow$ Different model, more variables but multihomogeneous structure.

Solving the system

Resultants: exponential degree growth.
Gröbner bases: unusable complexity bounds.
Modelling the $\ell$-torsion

Write $\ell D = 0$ with $D = P_1 + \cdots + P_g - g\infty$.

Use Cantor’s polynomials for $\ell(P_i - \infty)$ and add them.

- need to bound the degrees of Cantor’s polynomials
- degrees grow at each composition of $\ell(P_i - \infty) + \ell(P_j - \infty)$

$\Rightarrow$ Different model, more variables but multihomogeneous structure.

Solving the system

Resultants: exponential degree growth.
Gröbner bases: unusable complexity bounds.
Geometric resolution: takes advantage of structure.
Multihomogeneity and complexity

- $g$ variables $x_i$
- $O(g^2)$ equations
- Degree $O_g(\ell^3)$ in $x_i$

- $g$ variables $y_i$
- $g^2 - g$ variables for $\varphi$
- $O(g^2)$ equations
- Degrees in $O_g(1)$
Multihomogeneity and complexity

- $g$ variables $x_i$
  - $O(g^2)$ equations
  - degree $O_g(\ell^3)$ in $x_i$

- $g$ variables $y_i$
  - $g^2 - g$ variables for $\varphi$
  - $O(g^2)$ equations
  - degrees in $O_g(1)$

**Geometric resolution** *(Giusti-Lecerf-Salvy’01, Cafure-Matera’06)*

Assume $f_1, \ldots, f_n$ have degrees $\leq d$ and form a reduced regular sequence, and let $\delta = \max_i \deg \langle f_1, \ldots, f_i \rangle$. There is an algorithm computing a geometric resolution in time polynomial in $\delta, d, n$. 
Multihomogeneity and complexity

**Geometric resolution**  
(Giusti-Lecerf-Salvy’01, Cature-Matera’06)

Assume $f_1, \cdots, f_n$ have degrees $\leq d$ and form a reduced regular sequence, and let 
\[ \delta = \max_i \deg \langle f_1, \ldots, f_i \rangle. \] There is an algorithm computing a geometric resolution in time polynomial in $\delta$, $d$, $n$.

With $\delta = O_g(\ell^3 g)$ bounded by multihomogeneous Bézout bound. Both $d = O_g(\ell^3)$ and $n = O_g(1)$ are harmless for our complexity result.
Overall complexity bound

**Overall result**

Model the $\ell$-torsion with complexity $O_g(\ell^{O(g)})$.
Recall the largest $\ell$ is in $O_g(\log q)$.

$\Rightarrow$ we compute the local zeta function in $O_g((\log q)^{O(g)})$. 
Overall complexity bound

**Overall result**

Model the $\ell$-torsion with complexity $O_g(\ell^{O(g)})$.
Recall the largest $\ell$ is in $O_g(\log q)$.

$\Rightarrow$ **we compute the local zeta function in** $O_g((\log q)^{O(g)})$.

**State of the art**

<table>
<thead>
<tr>
<th></th>
<th>hyperelliptic case</th>
<th>plane curves</th>
<th>Abelian var</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Adleman-Huang’01</em></td>
<td>$(\log q)^{O(g^2 \log g)}$</td>
<td>$(\log q)^{O(1)}$</td>
<td>$(\log q)^{O(1)}$</td>
</tr>
<tr>
<td><em>This thesis</em></td>
<td>$O_g((\log q)^{O(g)})$</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
1. Introduction
2. Genus 3
3. Arbitrary $g$
4. RM in any genus
Hyperelliptic point-counting with RM in any genus

Contents
- Extend genus-3 case
- Use multihomogeneous modelling for $\text{Ker } \epsilon_i$
- Dependency on $g$?

Complexity result
| $g = 3$ | hyperelliptic $\tilde{O}(\log^{14} q)$ | with RM $\tilde{O}(\log^{6} q)$ |
| any $g$ | $O_g((\log q)^{cg})$ | $\tilde{O}_\eta(\log^{8} q)$ |
Explicit RM for arbitrary large $g$

RM families in any genus (Tautz-Top-Verberkmoes’91)

Consider curves with affine model $C_{n,t} : Y^2 = D_n(X) + t$. With $t$ a parameter and $D_n$ the $n$-th Dickson polynomial. For $n = 2g + 1$, yields genus-$g$ imaginary hyperelliptic curves. Explicit expression for $\eta$ is computable in $\tilde{O}_\eta(\log q)$ (Kohel-Smith’06).
Explicit RM for arbitrary large $g$

Consider curves with affine model $C_{n,t} : Y^2 = D_n(X) + t$. With $t$ a parameter and $D_n$ the $n$-th Dickson polynomial. For $n = 2g + 1$, yields genus-$g$ imaginary hyperelliptic curves. Explicit expression for $\eta$ is computable in $\tilde{O}_\eta(\log q)$ (Kohel-Smith’06).

<table>
<thead>
<tr>
<th>$\text{Split } \ell$</th>
<th>Genus 3</th>
<th>Genus $g$ with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree bounds</td>
<td>$p_1p_2p_3$ in $O(\ell^{1/3})$</td>
<td>$\prod_{i=1}^{g} p_i$ in $O(\ell^{1/g})$</td>
</tr>
</tbody>
</table>
Explicit RM for arbitrary large $g$

**RM families in any genus (Tautz-Top-Verberkmoes’91)**

Consider curves with affine model $C_{n,t} : Y^2 = D_n(X) + t$.
With $t$ a parameter and $D_n$ the $n$-th Dickson polynomial.
For $n = 2g + 1$, yields genus-$g$ imaginary hyperelliptic curves.

Explicit expression for $\eta$ is computable in $\tilde{O}_\eta(\log q)$ (Kohel-Smith’06).

<table>
<thead>
<tr>
<th></th>
<th>Genus 3</th>
<th>Genus $g$ with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Split $\ell$</strong></td>
<td>$p_1 p_2 p_3$</td>
<td>$\prod_{i=1}^g p_i$</td>
</tr>
<tr>
<td><strong>Degree bounds</strong></td>
<td>in $O(\ell^{1/3})$</td>
<td>in $O(\ell^{1/g})$</td>
</tr>
</tbody>
</table>
### Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th>Model</th>
<th>Genus $g$ with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J[\ell]$ (degree $\ell^{2g}$)</td>
</tr>
</tbody>
</table>

Remark: assuming quadratic degrees for Cantor's polynomials, we get a complexity in $\tilde{O}(\eta \log 8 q)$ similar to the case $g = 3$.

Practical use? Smallest case: $g = 5$ and $\ell = 23$.

Warning: even the size of the system is exponential in $g$. 

| Complexity | $O(g ((\log q)^{O(g)}))$ | $\tilde{O}(\eta \log 8 q)$ |

Simon Abelard  
Point counting  
September 7, 2018
Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th>Model Equations</th>
<th>Genus $g$ with degree $\ell^{2g}$</th>
<th>with RM $\text{Ker} \epsilon$ with degree $\ell^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J[\ell]$</td>
<td>$\ell D = 0$</td>
<td>$\epsilon(D) = 0$</td>
</tr>
</tbody>
</table>
### Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th>Model Equations</th>
<th>Genus $g$ ( J[\ell] ) (degree $\ell^{2g}$)</th>
<th>with RM ( \text{Ker} , \epsilon ) (degree $\ell^2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell D = 0$</td>
<td></td>
<td>$\epsilon(D) = 0$</td>
</tr>
</tbody>
</table>
Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th>Model Equations</th>
<th>Genus $g$</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Equations</td>
<td>$J[\ell]$ (degree $\ell^{2g}$)</td>
<td>Ker $\epsilon$ (degree $\ell^2$)</td>
</tr>
<tr>
<td>Model Equations</td>
<td>$\ell D = 0$</td>
<td>$\epsilon(D) = 0$</td>
</tr>
<tr>
<td>Variables</td>
<td>$g$ with degree $O_g(\ell^3)$</td>
<td>$g$ with degree $O_g(\ell^{3/g})$</td>
</tr>
<tr>
<td>Variables</td>
<td>$O(g^2)$ with degree $O_g(1)$</td>
<td>$O(g^2)$ with degree $O_{\eta}(1)$</td>
</tr>
</tbody>
</table>
**Modelling kernels of endomorphisms**

<table>
<thead>
<tr>
<th></th>
<th>Genus $g$</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Equations</strong></td>
<td>$J[\ell]$ (degree $\ell^{2g}$)</td>
<td>$\text{Ker} , \epsilon$ (degree $\ell^2$)</td>
</tr>
<tr>
<td></td>
<td>$\ell D = 0$</td>
<td>$\epsilon(D) = 0$</td>
</tr>
<tr>
<td></td>
<td>$g$ with degree $O_g(\ell^3)$</td>
<td>$g$ with degree $O_g(\ell^3/g)$</td>
</tr>
<tr>
<td></td>
<td>$O(g^2)$ with degree $O_g(1)$</td>
<td>$O(g^2)$ with degree $O_\eta(1)$</td>
</tr>
<tr>
<td></td>
<td>$O_g((\log q)^O(g))$</td>
<td>$\tilde{O}_\eta(\log^8 q)$</td>
</tr>
<tr>
<td><strong>Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th>Model Equations</th>
<th>Genus $g$ with degree $O_g(\ell^3)$</th>
<th>with RM $\text{Ker} \epsilon$ with degree $\ell^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J[\ell]$ (degree $\ell^{2g}$)</td>
<td>$\ell D = 0$</td>
<td>$\epsilon(D) = 0$</td>
</tr>
<tr>
<td>$g$ with degree $O_g(\ell^3)$</td>
<td>$O(g^2)$ with degree $O_g(1)$</td>
<td>$g$ with degree $O_g(\ell^3/g)$</td>
</tr>
<tr>
<td>$O(g^2)$ with degree $O_g(1)$</td>
<td>$O_g((\log q)^{O(g)})$</td>
<td>$O_g((\log q)^{O(g)})$</td>
</tr>
</tbody>
</table>

Complexity

- $O(g^2)$ with degree $O_g(1)$
- $O_g((\log q)^{O(g)})$
- $\tilde{O}_\eta(\log^8 q)$

Remark: assuming quadratic degrees for Cantor’s polynomials, we get a complexity in $\tilde{O}_\eta(\log^6 q)$ similar to the case $g = 3$. 

Practical use? Smallest case: $g = 5$ and $\ell = 23$.

Warning: even the size of the system is exponential in $g$.
Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th>Model</th>
<th>Genus $g$</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equations</td>
<td>$J[\ell]$ (degree $\ell^{2g}$)</td>
<td>$\Ker \epsilon$ (degree $\ell^2$)</td>
</tr>
<tr>
<td>Variables</td>
<td>$\ell D = 0$</td>
<td>$\epsilon(D) = 0$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$g$ with degree $O_g(\ell^3)$</td>
<td>$g$ with degree $O_g(\ell^3/g)$</td>
</tr>
<tr>
<td></td>
<td>$O(g^2)$ with degree $O_g(1)$</td>
<td>$O(g^2)$ with degree $O_\eta(1)$</td>
</tr>
<tr>
<td></td>
<td>$O_g((\log q)^{O(g)})$</td>
<td>$\tilde{O}_\eta(\log^8 q)$</td>
</tr>
</tbody>
</table>

Remark: assuming quadratic degrees for Cantor’s polynomials, we get a complexity in $\tilde{O}_\eta(\log^6 q)$ similar to the case $g = 3$. Practical use? Smallest case: $g = 5$ and $\ell = 23$. 

Warning: even the size of the system is exponential in $g$. 

Simon Abelard
Modelling kernels of endomorphisms

<table>
<thead>
<tr>
<th></th>
<th>Genus ( g )</th>
<th>with RM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Equations</strong></td>
<td>( J[\ell] ) (degree ( \ell^{2g} ))</td>
<td>( \operatorname{Ker} \epsilon ) (degree ( \ell^{2} ))</td>
</tr>
<tr>
<td><strong>Variables</strong></td>
<td>( \ell D = 0 )</td>
<td>( \epsilon(D) = 0 )</td>
</tr>
<tr>
<td></td>
<td>( g ) with degree ( O_g(\ell^{3}) )</td>
<td>( g ) with degree ( O_g(\ell^{3}/g) )</td>
</tr>
<tr>
<td></td>
<td>( O(g^{2}) ) with degree ( O_g(1) )</td>
<td>( O(g^{2}) ) with degree ( O_\eta(1) )</td>
</tr>
<tr>
<td></td>
<td>( O_g((\log q)^{O(g)}) )</td>
<td>( \tilde{O}_\eta(\log^{8} q) )</td>
</tr>
</tbody>
</table>

Remark: assuming quadratic degrees for Cantor’s polynomials, we get a complexity in \( \tilde{O}_\eta(\log^{6} q) \) similar to the case \( g = 3 \).

Practical use? Smallest case: \( g = 5 \) and \( \ell = 23 \).

Warning: even the size of the system is exponential in \( g \).
Summary of results

Three questions to address:

- **modelling** (subgroups of) the $\ell$-torsion by polynomial systems
- **bounding** their sizes (number of variables, degrees)
- **solving** them (and bounding complexity)

Answers provided

- quadratic and cubic bounds for Cantor’s polynomials
- multihomogeneous modelling for $J[\ell]$ (includes non-genericity)
- exploiting structure via geometric resolution
- when possible (RM) model subgroups of $J[\ell]$
Future work

Beyond the hyperelliptic case

Goal: explicit value for the $g^{O(1)}$, maybe even reach $O_g ((\log q)^{O(g)})$. Main obstacle: need analogue of Cantor’s polynomials.
Future work

**Beyond the hyperelliptic case**

Goal: explicit value for the $g^{O(1)}$, maybe even reach $O_g \left((\log q)^{O(g)}\right)$. Main obstacle: need analogue of Cantor’s polynomials.

**Splitting the torsion without RM**

Model kernels of $\ell$-isogenies, as in SEA. Fast evaluation of modular polynomials? ($g = 1$ in *Sutherland’12*)
Future work

**Beyond the hyperelliptic case**

Goal: explicit value for the $g^{O(1)}$, maybe even reach $O_g ((\log q)^{O(g)})$.
Main obstacle: need analogue of Cantor’s polynomials.

**Splitting the torsion without RM**

Model kernels of $\ell$-isogenies, as in SEA.
Fast evaluation of modular polynomials? ($g = 1$ in Sutherland’12)

**Better handling non-genericity?**

Elements of $J[\ell]$ of weight $< g$ and other pathological cases?
Problem: when these elements contain a proper subgroup of $J[\ell]$.
Can this happen for any curve or any $\ell$? In what proportions?
Thanks for your attention