Speeding up the CRT method to compute class polynomials in genus 2

MSR end of internship talk

Damien Robert[1]

[1]Mentor: Kristin Lauter

09/23/2010

Hyperelliptic curve cryptography

- $H : y^2 = f(x)$ hyperelliptic curve of genus 2 over $\mathbb{F}_q$ ($\deg f = 5, 6$).
- The Jacobian J of H is a finite abelian group of cardinal $n \approx q^2$.

⇒ Public key cryptosystem based on the discrete logarithm problem.

⇒ Pairings.

- We want to find a secure hyperelliptic curve of genus 2.
- Security: $\sqrt{n_0}$ where n_0 is the largest prime dividing n.

⇒ Take a random curve and count $\#J$.

⇒ Generate a curve with a prescribed number of points (also useful for pairings).

Hyperelliptic curve cryptography

- $H : y^2 = f(x)$ hyperelliptic curve of genus 2 over $\mathbb{F}_q$ ($\deg f = 5, 6$).
- The Jacobian J of H is a finite abelian group of cardinal $n \approx q^2$.

$\Rightarrow$ Public key cryptosystem based on the discrete logarithm problem.

$\Rightarrow$ Pairings.

- We want to find a secure hyperelliptic curve of genus 2.
- Security: $\sqrt{n_0}$ where n_0 is the largest prime dividing n.

$\Rightarrow$ Take a random curve and count $\#J$.

$\Rightarrow$ Generate a curve with a prescribed number of points (also useful for pairings).

Class polynomials

- Let K be a primitive CM field of degree 4: K is a totally imaginary quadratic extension of a totally real field K_0. (K is then cyclic Galois, or dihedral)
- The class polynomials H_1, H_2, H_3 parametrize the Igusa invariants of Jacobians J whose endomorphism rings is isomorphic to O_K, the maximal ring of K.
  These Jacobians are defined over the Hilbert class field HK_r of the reflex class field K_r of K.
- If $\mathfrak{P}$ is a prime of good reduction in HK_r, the typenorm of $\mathfrak{P}$ give the Frobenius polynomial of $J_{\mathfrak{P}}$.

$\Rightarrow$ select $p \in \mathbb{Z}$ of cryptographic size such that $\#J_{\mathbb{F}_p}$ is prime.

$\Rightarrow$ Reduce H_1, H_2, H_3 modulo p to find $J_{\mathbb{F}_p}$.

Constructing class polynomials

- Analytic method: compute the Igusa invariants in $\mathbb{C}$ with sufficient precision to recover the class polynomials.
- p-adic lifting: lift the Igusa invariants in $\mathbb{Q}_p$ with sufficient precision to recover the class polynomials (require specific splitting behavior of p in K).
- CRT: compute the class polynomials modulo small primes, and use the CRT to reconstruct the class polynomials.

Remark

In genus 1, the analytic and CRT method are quasi-linear in the size of the output $\Rightarrow$ computation bounded by memory. But we can construct directly the class polynomials modulo p with the explicit CRT.

Complexity of constructing class polynomials in genus 2

Let k be the precision needed.

- Analytic method: compute the invariants using theta functions $\widetilde{O}(k^2)$. (Remark: available implementation for K_0 of class number one, huge precision loss.)
- p-adic lifting: lifting $\widetilde{O}(k)$, recovery $\widetilde{O}(k^2)$.
- CRT method: we need to use $O(k)$ prime of size $O(k)$. For each prime we check all isomorphism classes of curves: $O(k^3)$. We need to speed up the CRT!

Review of the CRT algorithm

1. Select a prime p.
2. For each Jacobian J in the p^3 isomorphic classes:
   2.1 Check if J is in the right isogeny class by computing the characteristic polynomial of the Frobenius (do some trial tests to check for $\#J$ before).
   2.2 Check if $\mathrm{End}(J) = O_K$.
3. From the invariants of the maximal curves, reconstruct $H_i \mod p$.

Remark

Algorithm developed by EISENTRÄGER, FREEMAN *and* LAUTER, *with ameliorations from* BRÖKER, GRUENEWALD *and* LAUTER *by using the* $(3,3)$-*Galois action.*

Selecting the prime p

- Usual method: find a prime p that splits completely into principal ideals in K_r, and splits completely in K.
- But we only need the typenorm of the ideals above p to be principal ideals.

$\Rightarrow$ We can work with more prime!

$\Rightarrow$ And the typenorm are generated by the frobenius!

Checking if a curve is maximal

- Let J be the Jacobian of a curve in the right isogeny class. Then $\mathbb{Z}[\pi,\overline{\pi}] \subset \mathrm{End}(J) \subset O_K$.
- Let $\gamma \in O_K \backslash \mathbb{Z}[\pi,\overline{\pi}]$. We want to check if $\gamma \in \mathrm{End}(J)$.
- Since $(O_K : \mathbb{Z}[\pi,\overline{\pi}])$ is prime to p we have $\gamma \in \mathrm{End}(J) \Leftrightarrow p\gamma \in \mathrm{End}(J)$.
- Let n be the smallest integer thus that $n\gamma \in \mathbb{Z}[\pi,\overline{\pi}]$. Since $(\mathbb{Z}[\pi,\overline{\pi}] : \mathbb{Z}[\pi]) = p$, we can write $np\gamma = P(\pi)$.
- Then $\gamma \in \mathrm{End}(J) \Leftrightarrow P(\pi) = 0$ on $J[n]$.
- In practice: compute $J[\ell^d]$ for $\ell^d \mid (O_K : \mathbb{Z}[\pi,\overline{\pi}])$ and check the action of the generators of O_K on it.

Remark

If $1, \alpha, \beta, \gamma$ are generators of O_K as a $\mathbb{Z}$-module, it can happen that $\gamma = P(\alpha, \beta)$, so that we don't need to check that $\gamma \in \mathrm{End}(J)$.

Field of definition of the ℓ^d-torsion

Proposition

- *The geometric points of* $J[\ell^d]$ *are defined over* $\mathbb{F}_{p^{\alpha_d}} \Leftrightarrow \pi^{\alpha_d} - 1 \in \ell^d \operatorname{End}(J)$.
- $\alpha_d \mid \alpha_1 \ell^{d-1}$. *If* $\operatorname{End}(J) = O_K$ *this is an equality:* $\alpha_d = \alpha_1 \ell^{d-1}$.

Corollary

Let α *be thus that* $\pi^\alpha - 1 \in \ell O_K$. *We first check that* $(\pi^\alpha - 1)/\ell$ *is an element of* $\operatorname{End}(J)$ *(*$\Leftrightarrow J[\ell]$ *defined over* $\mathbb{F}_{p^\alpha}$*). Then* $J[\ell^d]$ *is defined over* $\mathbb{F}_{p^{\alpha \ell^{d-1}}}$.

Remark

It may happen that we get a factor two on the degrees by working over the twist: that is by working with $-\pi$.

Computing the ℓ^d-torsion

- We compute $\#J(\mathbb{F}_{p^{\alpha_d}}) = \ell^\beta c$.
- If P_0 is a random point of $J(\mathbb{F}_{p^\alpha})$, then $P = cP_0$ is a random point of ℓ^∞-torsion, and P multiplied by a suitable power of ℓ is a random point of ℓ^d-torsion.
- Usual method: take a lot of random points of ℓ^d-torsion, and hope they generate it over $\mathbb{F}_{p^{\alpha_d}}$.
- Problems: the random points of ℓ^d-torsion are not uniform $\Rightarrow$ require a lot of random points, and the result is probabilistic.
- Our solution: Compute the whole ℓ^∞-torsion. "Correct" points to find uniform points of ℓ^d-torsion. Use pairings to save memory.

$\Rightarrow$ We can check if a curve is maximal faster.

$\Rightarrow$ We can abort early.

Obtaining all the maximal curves

- If J is a maximal curve, and ℓ does not divide $(O_K : \mathbb{Z}[\pi, \overline{\pi}])$, then any (ℓ, ℓ)-isogenous curve is maximal.
- The maximal Jacobians form a principal homogeneous space under the Shimura class group $\mathfrak{C}(O_K) = \{(I, \rho) \mid I\overline{I} = (\rho) \text{ and } \rho \in K_0^+\}$.
- (ℓ, ℓ)-isogenies between maximal Jacobians correspond to element of the form $(I, \ell) \in \mathfrak{C}(O_K)$. We can use the structure of $\mathfrak{C}(O_K)$ to determine the number of new curves we will obtain with (ℓ, ℓ)-isogenies.
  $\Rightarrow$ Don't compute unneeded isogenies.
- It can be faster to compute (ℓ, ℓ)-isogenies with $\ell \mid (O_K : \mathbb{Z}[\pi, \overline{\pi}])$ to find new maximal Jacobians when ℓ and $\mathrm{val}_\ell((O_K : \mathbb{Z}[\pi, \overline{\pi}]))$ is small.

"Going up"

- There is p^3 classes of isomorphic curves, but only a very small number ($\#\mathfrak{C}(O_K)$) with $\mathrm{End}(J) = O_K$.
- But there is at most $16p^{3/2}$ isogeny class.

$\Rightarrow$ On average, there is $\approx p^{3/2}$ curves in a given isogeny class.

$\Rightarrow$ If we have a curve in the right isogeny class, try to find isogenies giving a maximal curve!

An algorithm for "going up"

1. Let $\gamma \in O_K \setminus \mathrm{End}(J)$. We can assume that $\ell^\infty \gamma \in \mathbb{Z}[\pi, \overline{\pi}]$.
2. Let d be the minimum such that $\gamma(J[\ell^d]) \neq \{0\}$, and let $K = \gamma(J[\ell^d])$. By definition, $K \subset J[\ell]$.
3. We compute all (ℓ, ℓ)-isogeneous Jacobians J' where the kernel intersect K. Keep J' if $\#\gamma(J'[\ell^d]) < \#K$ (and be careful to prevent cycles).

- First go up for $\gamma = (\pi^\alpha - 1)/\ell$: this minimize the extensions we have to work with.
- It is not always possible to go up. We would need more general isogenies than (ℓ, ℓ)-isogenies. Most frequent case: we can't go up because there is no (ℓ, ℓ)-isogenies at all! (And we can detect this).

Sieving the primes

- We throw a prime p for the CRT if detecting if a curve is maximal is too costly, or there is not enough curves where we can "go up".
- How to estimate this number?
    1. Compute the lattice of orders between $\mathbb{Z}[\pi, \overline{\pi}]$ and O_K. For all such order O such that $(O_K : O)$ is not divisible by any ℓ where there is no (ℓ, ℓ)-isogeny, compute $\mathfrak{C}(O)$.
    This is too costly! (Even computing $\operatorname{Pic}(\mathbb{Z}[\pi, \overline{\pi}])$ is too costly!)
    2. Compute

$$\#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}]) = \frac{c(O_K : Z[\pi, \overline{\pi}]) \# \operatorname{Cl}(O_K) \operatorname{Reg}(O_K)(\widehat{O}_K^* : \widehat{\mathbb{Z}}[\pi, \overline{\pi}]^*)}{2\# \operatorname{Cl}(\mathbb{Z}[\pi + \overline{\pi}]) \operatorname{Reg}(\mathbb{Z}[\pi + \overline{\pi}])}$$

and estimate the number of curves as

$$\sum_{d \mid \#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}])} d$$

(for d not divisible by a ℓ where we can't go up).

Exploring the curves

1. Go sequentially through the p^3 Igusa invariants j_1, j_2, j_3. But constructing the curve from the invariants is costly.
2. Construct random curves in Weierstrass form

$$y^2 = a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

3. If the two torsion is rational (check where $\frac{\pi-1}{2}$ live), construct curves in Rosenhain form

$$y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu).$$

4. If the Hilbert moduli space is rational, construct the j-invariants from the Gundlach invariants (only p^2 invariants, parametrizing the space of curves with real multiplication by K_0).

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
7	2^2	4	7	8	0.5 + 0.3	0 + 0.2
17	2	1	39	32	4 + 0.2	0 + 0.1
23	$2^2, 7$	4, 3	49	51	9 + 2.3	0 + 0.2
71	2^2	4	7	8	255 + 0.7	5.3 + 0.2
97	2	1	39	32	680 + 0.3	2 + 0.1
103	$2^2, 17$	4, 16	119	127	829 + 17.6	0.5 + 1
113	$2^5, 7$	16, 6	1281	877	1334 + 28.8	0.2 + 1.3
151	$2^2, 7, 17$	4, 3, 16	-	-	0	0
					$3162s$	$13s$

Computing the class polynomial for $K = \mathbb{Q}(i\sqrt{2+\sqrt{2}})$, $\mathfrak{C}(O_K) = \{0\}$.

$$H_1 = X - 1836660096, \quad H_2 = X - 28343520, \quad H_3 = X - 9762768$$

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
29	$3, 23$	$2, 264$	-	-	-	-
53	$3, 43$	$2, 924$	-	-	-	-
61	3	2	9	6	$167 + 0.2$	$0.2 + 0.5$
79	3^3	18	81	54	$376 + 8.1$	$0.3 + 0.9$
107	$3^2, 43$	$6, 308$	-	-	-	-
113	$3, 53$	$1, 52$	159	155	$1118 + 137.2$	$0.8 + 25$
131	$3^2, 53$	$6, 52$	477	477	$1872 + 127.4$	$2.2 + 44.4$
139	3^5	81	?	486	-	$1 + 36.7$
157	3^4	27	243	164	$3147 + 16.5$	-
					$6969s$	$114s$

Computing the class polynomial for $K = \mathbb{Q}(i\sqrt{13 + 2\sqrt{29}})$, $\mathfrak{C}(O_K) = \{0\}$.

$$H_1 = X - 268435456, \quad H_2 = X + 5242880, \quad H_3 = X + 2015232.$$

Checking if a curve is maximal

- Let $H : y^2 = 80x^6 + 51x^5 + 49x^4 + 3x^3 + 34x^2 + 40x + 12$ over $\mathbb{F}_{139}$ and J the Jacobian of H. We have $\mathrm{End}(J) \otimes \mathbb{Q} = \mathbb{Q}(i\sqrt{13 + 2\sqrt{29}})$ and we want to check if $\mathrm{End}(J) = O_K$.
- For that we need to compute $J[3^5]$, that lives over an extension of degree 81 (for the twist it lives over an extension of degree 162).
- With the old randomized algorithm, this computation takes 470 seconds (with 12 Frobenius trials over $\mathbb{F}_{139^{162}}$).
- With the new algorithm computing the ℓ^∞-torsion, it only takes 17.3 seconds (needing only 4 random points over $\mathbb{F}_{139^{81}}$, approx 4 seconds needed to get a new random point of ℓ^∞-torsion).

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
7	-	-	1	1	0.3	0 + 0.1
23	**13**	84	15	2 (16)	9 + 70.7	0.4 + 24.6
53	7	3	7	7	105 + 0.5	7.7 + 0.5
59	2, **5**	1, 12	322	48 (286)	164 + 6.4	1.4 + 0.6
83	3, 5	4, 24	77	108	431 + 9.8	2.4 + 1.1
103	67	1122	-	-	-	-
107	7, **13**	3, 21	105	8 (107)	963 + 69.3	-
139	$\mathbf{5}^2$, 7	60, 2	259	9 (260)	2189 + 62.1	-
181	3	1	161	135	5040 + 3.6	4.5 + 0.2
197	5, 109	24, 5940	-	-	-	-
199	$\mathbf{5}^2$	60	37	2 (39)	10440 + 35.1	-
223	2, *23*	1, 11	1058	39 (914)	10440 + 35.1	-
227	109	1485	-	-	-	-
233	5, 7, **13**	8, 3, 28	735	55 (770)	11580 + 141.6	88.3 + 29.4
239	7, 109	6, 297	-	-	-	-
257	3, 7, **13**	4, 6, 84	1155	109 (1521)	17160 + 382.8	-
313	3, **13**	1, 14	?	146 (2035)	-	165 + 14.7
373	5, 7	6, 24	?	312	-	183.4 + 3.8
541	2, 7, **13**	1, 3, 14	?	294 (4106)	-	91 + 5.5
571	3, **5**, 7	2, 6, 6	?	1111 (6663)	-	96.6 + 3.1
					56585s	776s

Computing the class polynomial for $K = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$, $\mathfrak{C}(O_K) = \{0\}$.
(The new algorithm also skipped the primes 277, 281, 349, 397, 401, 431, 487, 509, 523.)

$$H_1 = 244140625X - 2614061544410821165056$$

Checking if a curve is maximal (2)

- Let $H : y^2 = 10x^6 + 57x^5 + 18x^4 + 11x^3 + 38x^2 + 12x + 31$ over $\mathbb{F}_{59}$ and J the Jacobian of H. We have $\mathrm{End}(J) \otimes \mathbb{Q} = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$ and we want to check if $\mathrm{End}(J) = O_K$.
- O_K is generated as a $\mathbb{Z}$-module by $1, \alpha, \beta, \gamma$. α is of index 2 in $O_K/\mathbb{Z}[\pi, \overline{\pi}]$, β of index 4 and γ of index 40.
- So the old algorithm will check $J[2^3]$ and $J[5]$.
- But $O_K = \mathbb{Z}_2[\pi, \overline{\pi}, \alpha]$, so we only need to check $J[2]$ and $J[5]$.

CRT for dihedral fields

- $K = \mathbb{Q}(X)/(X^4 + 13X^2 + 41)$ dihedral, $\mathfrak{C}(K) \simeq \{0\}$.
- Primes used: 59, 859, 911, 1439, 2029, 3079.
  (Primes skipped: 131, 139, 241, 269, 271, 359, 409, 541, 569, 599, 661, 701, 761, ...)
- Time: 5956 seconds.
- Class polynomials:

$$H_1 = 64X^2 + 14761305216X - 111577100832000000,$$

$$H_2 = 16X^2 + 72590904X - 8609344200000,$$

$$H_3 = 16X^2 + 28820286X - 303718531500.$$

CRT for non principal fields

- $K = \mathbb{Q}(X)/(X^4 + 238X^2 + 833)$ cyclic. $\mathfrak{C}(K) \simeq \mathbb{Z}/2\mathbb{Z}$ is generated by $(7,7)$-isogenies.
- Primes used: 19 , 59 , 67, 83, 149, 191, 223, 229, 239, 257, 349, 463, 557, 613, 661, 733, 859, 1039, 1373, 1613, 1657, 1667, 1733, 1753, 1801, 1871, 1879, 2399, 3449, 3469, 3761, 3931, 4259, 4691, 5347, 5381, 6427, 6571, 6781.
- For $p \approx 6000$, we keep p if we expect more than $\frac{p^{3/2}}{32} \approx 15 \times 10^6$ curves. At this size, it takes around 6 seconds to test 10000 curves, so around 2.5 hours are needed for p.
- Total time: 44062 second (not the latest version of the code).
- Class polynomials:

$$
\begin{aligned}
H_1(X) = {} & 168451200633545364243594910146286907316572281862280871005795423612829696X^2 \\
& +158582528695513934970693031198523489269724119094630145672062735632518026507497890643968X \\
& -20148439779616498933576752193721158991703786695904651875585742599422503529550925413744 64.
\end{aligned}
$$

- $K = \mathbb{Q}(X)/(X^4 + 185X^2 + 8325)$. $\mathfrak{C}(K) \simeq \mathbb{Z}/10\mathbb{Z}$ is generated by $(3,3)$-isogenies (generating a subgroup $\simeq \mathbb{Z}/5\mathbb{Z}$) and $(5,5)$-isogenies (generating a subgroup $\simeq \mathbb{Z}/2\mathbb{Z}$).
- Primes used for now: 263, 271, 317, 337, 397, 641, 941, 1103, 11699, 1259, 2293, 2341, 2393, 2803, 3203, 3319, 3919, 6151, 6367, 7669, 7759, 9949.
- Time currently spent: $\approx$ 150000 seconds.
  We have $\approx$ 216 bits of precision, but the denominator are of size $\approx$ 588 bits.
- Current class polynomials:

$$
\begin{aligned}
H_1 = &-2148061154236176250872355746833546154293069021734542210143570 7227X^{10} \\
&+ 1312267233956977280466457447356683385775372099038401531675512 82021X^9 \\
&+ 1199459772554977332188737103604932493410559381817989365966236 83383X^8 \\
&- 1537142137801790603683482341701748032892008994822685208787932 09046X^7 \\
&+ 6263874479359993979349589228551770130375396757888438666331522 5591X^6 \\
&- 9367781644606331484241836458072043058135031972618764279234050 8326X^5 \\
&- 7169184216574133822561018629789731781493822809290499861660826 5551X^4 \\
&+ 1369815271122646110434851597843323060157085026247695921168481 81204X^3 \\
&- 3947701035212686018560301000460464226956697965915593433171515 3441X^2 \\
&- 1513714522524486946465931170876352983166505269951944719281880 77417X \\
&- 3699326571758938480406710643683761432168295010151303199445539 4382.
\end{aligned}
$$

Perspectives

- 6 seconds for 10000 curves is way too slow! Implement this part with C.
- Better estimates for the precision required.
- Compute Gundlach invariants for more real quadratic fields.
- More general isogenies than (ℓ, ℓ)-isogenies!