Counting points on hyperelliptic curves over finite fields

P. Gaudry and R. Harley

PUT PICTURES HERE

ANTS IV, Leiden, 3rd July 2000.

Context, motivation

Problem: Find the number of solutions (x, y) of $y^2 = f(x)$ over a finite field.

- Naive method: try all (x, y) in exponential time.
- Polynomial-time algorithm by Schoof (1985) for elliptic curves $(\deg f = 3)$.
  Practical: Morain $10^{499} + 153$, Vercauteren 2^{1999}.
- New algorithm for elliptic curves by Satoh (1999).
  Practical: Fouquet, Gaudry, Harley 2^{4001}.
- Practical algorithm for hyperelliptic curves: this talk.

1. Background on hyperelliptic curves

Defs.: Hyperelliptic curve $\mathcal{C}$: $y^2 = f(x)$, $\deg f = 2g + 1$, genus g.

Coeffs in finite field $\mathbb{F}_q$, $q = p^n$ elements, characteristic $p \geq 3$.

Defs.: Divisor: sum $\sum_i P_i$ of points on $\mathcal{C}$. Opposite of (x, y) is $(x, -y)$.

Reduced divisor: $\leq g$ points, none opposite.

Mumford's representation: $u(x) = \prod_i (x - x_i)$ and $y_i = v(x_i)$.

Divisor with no points: $\mathcal{O} = \langle 1, 0 \rangle$.

Divisor with point P: $\langle x - x_P, y_P \rangle$.

Def.: Jacobian: $\{\ \langle u, v \rangle,\ u$ monic, $\deg v < \deg u \leq g,\ u \,|\, f - v^2\ \}$.

Frobenius endomorphism, field of definition

Def.: Frobenius automorphism: $\phi : x \mapsto x^q$ on $\mathbb{F}_q$ and extensions.

Extends to points: $\phi(P) = (\phi(x), \phi(y))$.

Extends to divisors: $\phi(\sum_i P_i) = \sum_i \phi(P_i)$.

With Mumford's repr.: $\phi(\langle u, v\rangle) = \langle \phi(u), \phi(v)\rangle$.

Def.: Divisor D defined over $\mathbb{F}_q$ iff $\phi(D) = D$.

Property: Jacobian is a group. Subset defined over $\mathbb{F}_q$ is a subgroup, denoted $\mathbf{J}$.

Property: Frobenius acts linearly on the Jacobian.

$\Rightarrow$ Characteristic polynomial $\chi(t)$, $\deg = 2g$, integer coeffs.

Thm.: (Weil) Roots of χ have absolute value $\sqrt{q}$.

Characteristic polynomial, Relation to cardinalities

Equivalence between $\chi(t)$ and $\#\mathcal{C}/\mathbb{F}_{q^i}$, $1 \leq i \leq g$.

Example, in genus 2:

$$\chi(t) = t^4 - s_1 t^3 + s_2 t^2 - s_1 q t + q^2$$

$$\#\mathcal{C}/\mathbb{F}_q = q - s_1 \text{ and } \#\mathcal{C}/\mathbb{F}_{q^2} = q^2 - s_1^2 + 2s_2 \ .$$

Have $\#\mathbf{J} = \chi(1)$.

Conversely $\#\mathbf{J}$ gives info about $\chi(t)$. In genus 2, $O(1)$ possibilities.

Will compute $\#\mathbf{J}$. By theorem, $\#\mathbf{J} \simeq q^g$ in interval of width $\simeq 4gq^{g-1/2}$.

2. Group law in the Jacobian

Intuition: Generalisation of chord-and-tangent operation on elliptic curves.

Calculation:

Use Mumford's repr.

General algorithm due to Cantor.

We derived an optimized algorithm for genus 2.

Result: Doubling a divisor, or adding two divisors, takes 30 mul + 2 inv in general.

(See Web page for C code).

3. Birthday paradox algorithm

Cardinality $N = \#\mathbf{J}$ in interval of width w.

Method:

Pick random divisor D.

Compute order n.

Group exponent is e.

Have $n \mid e$ and $e \mid N$ so $n \mid N$.

Usually $n > w$, get N.

Otherwise try several divisors until $\mathrm{lcm} > w$.

In rare case $e \leq w$, cannot determine N completely.

To find n, find multiple in interval, find smallest factor s.t. $[d]D = \mathcal{O}$.

Naive search of interval: $O(w)$ group operations.

Birthday paradox algorithm: $O(\sqrt{w})$ group operations.

Algorithm

Distributed version of Pollard's lambda method, with distinguished points.

c center of the search interval,

M (approximate) number of machines.

- Pick hash function $h : \mathbf{J} \to \{0, \ldots, 19\}$,
- Pick 20 step lengths $l_i > 0$, average $M\sqrt{w}$.

Calculation consists of pseudo-random walks:

- Pick random bit b;
  IF $b = 0$ THEN $R \leftarrow \mathcal{O}$ ELSE $R \leftarrow [c]D$;
- Pick random $r < w$;
  set $R \leftarrow R + [r]D$;
- WHILE R not distinguished DO
  $i \leftarrow h(R)$;
  $r \leftarrow r + l_i$;
  $R \leftarrow R + [l_i]D$;
- Store b, R and r on central server.

Finding the solution

Chains with $b = 0$ visit divisors in $S_0 = \{[r]D \mid 0 \le r < w\}$.

Chains with $b = 1$ visit divisors in $S_1 = \{[c + r]D \mid 0 \le r < w\}$.

Choice of c implies $I = S_0 \cap S_1$ has $|I| \ge w/2$.

Visit $O(\sqrt{w})$ divisors. $O(\sqrt{w})$ are in I.

Birthday paradox: likely that some D visited twice with $b = 0, 1$.

Server detects collision $R_0 = R_1$ at distinguished points.

Then $r_1 + c - r_0$ is multiple of n.

Done.

Speed up: Precompute $N \bmod m$. Restricts search to arithmetic progression mod m. Number of operations in B.P. reduced by $\sqrt{m}$.

Modifications: Choose the l_i to be multiples of m etc.

4. Counting modulo the characteristic

Elliptic curve case: Hasse invariant related to number of points mod p.

Hyperelliptic case, Cartier-Manin operator represented by Hasse-Witt matrix.

Def.: Let $f(x)^{(p-1)/2} = \sum_i c_i x^i$. Hasse-Witt matrix is $A = (c_{ip-j})_{1\leq i,j\leq g}$.

Let A^σ denote $(a_{i,j}^p)$. Manin proved:

Thm.: Let $\kappa(t)$ be the characteristic polynomial of $AA^\sigma A^{\sigma^2}\cdots A^{\sigma^{n-1}}$. Then

$$\chi(t) \equiv (-1)^g t^g \kappa(t) \bmod p.$$

$\Rightarrow$ Can compute $\#\mathbf{J} \bmod p$ up to $p \simeq 10^5$.

5. Counting modulo small powers of 2

Prop.: Divisors of order 1 or 2 are those with $v = 0$.

Recall $u \mid f - v^2$, monic, $\deg \leq g$.

When f irreducible, $\#\mathbf{J}$ odd.

Otherwise factor f: k irreducible factors, 2^k factors, half have $\deg \leq g$. Hence:

$$2^{k-1} \mid \#\mathbf{J}.$$

Better: Copy Schoof. Find $\#\mathbf{J} \bmod \ell$ for small primes ℓ.

Chinese Remainder Theorem gives:

$$\#\mathbf{J} \bmod \prod \ell.$$

$\Rightarrow$ Polynomial-time algorithm. In practice combine with B.P.

6. Hyperelliptic Schoof algorithm

General idea: To find $\#\mathbf{J} \bmod \ell$...

Compute ℓ-torsion divisors,

Plug into χ for all tuples $(s_i)_{1\le i\le g} \bmod \ell$,

Check $\chi = 0$,

Eliminate bad tuples,

Stop when one possibility for $\chi(1)$ remains.

Example: In genus 2

- Find ℓ-torsion divisor D,
- For $0 \le s_1 < \ell$, compute $[s_1]\phi^3(D) + [qs_1 \bmod \ell]\phi(D)$,
- For $0 \le s_2 < \ell$, compute $\phi^4(D) + [s_2]\phi^2(D) + [q^2 \bmod \ell]D$,
- Keep pairs (s_1, s_2) for which they are equal.

Hyperelliptic division polynomials

Need to construct ℓ-torsion divisors.

Characterize $D = \langle x^2 + u_1 x + u_2, v_0 x + v_1 \rangle \in \mathbf{J}[l]$ by division ideal I_ℓ.

Formulae for I_ℓ by Kampkötter.

Compute with multivariate polynomials modulo I_ℓ using Gröbner bases.

Problem: Slow.

Cantor found hyperelliptic division polynomials which allow to write explicit rational formula for $[\ell]P$.

Divisor $D = P + Q$ is ℓ-torsion divisor iff $[\ell]P = -[\ell]Q$.

Write equality using Cantor's formula, clear denominators

$\Rightarrow$ System of 4 polynomials in x_P, y_P, x_Q, y_Q.

Advantage: Two polys involve only x_P and x_Q.

Computing the ℓ-torsion (in genus 2)

The procedure for finding an ℓ–torsion divisor is:

1. Compute resultant of two bivariate polys.
2. Find small factor.
3. Construct P and Q.
4. Check to eliminate parasite solutions.

Optimization: Fast arithmetic, clever resultant, lazy factorization...

$\Rightarrow$ Arithmetic on univariate polynomials. Fast!

7. Practical results

Prime field: Curve over $\mathbb{F}_p$ with $p = 10^{19} + 51$:

$$\begin{aligned} y^2 = x^5 \quad & + 3141592653589793238\, x^4 \quad + 4626433832795028841\, x^3 \\ & + 9716939937510582097\, x^2 \quad + 4944592307816406286\, x \\ & + 2089986280348253421 \end{aligned}$$

2^k–torsion divisors give $176 \bmod 256$ in 9 hours.

Schoof for $l \in \{3, 5, 7, 11, 13\}$ gives $5170 \bmod 15015$ in 10 days.

$\Rightarrow$ Speed up B.P. by factor 2000. Use 50 days CPU time.

Result is: $\#\mathbf{J} = 99999999982871020671452277000281660080$.

Medium characteristic: Curve over $\mathbb{F}_{p^4} = \mathbb{F}_p[t]/(t^4 - 17)$ where $p = 2^{16} - 15$:

$$y^2 = x^5 + x^4 + x^3 + x^2 + x + t$$

Cartier-Manin gave $58976 \bmod p$ in 17 minutes!

Result is: $\#\mathbf{J} = 339659790214687297284652908385855015466$.

8. Perspectives

- Extension to characteristic 2.
- Extension of Schoof part to genus $g \geq 3$.

If/when we have modular equations for Siegel modular forms:

- Lift curve to local field, use Cartier-Manin modulo small powers of p.
- Genus 2 version of Elkies-Atkin to reduce degree of polys.
- Extend Satoh's algorithm to genus 2.