Algorithms for counting points on curves.

Pierrick Gaudry

gaudry@lix.polytechnique.fr

http://www.lix.polytechnique.fr/Labo/Pierrick.Gaudry/

Waterloo, 31 October 2001.

Context

Point counting is a tricky problem when designing a cryptosystem based on curves.

I concentrate on random curves (the ideal case) of genus ≤ 3.

The breakthrough of Satoh, 2 years ago, really changed the situation:

- before: SEA algo was the only one
- now: wide choice of algorithms that have to be combined / extended / implemented / improved.

Examples of new facts:

- Point counting on elliptic curves in char 2 is almost as fast as an RSA key generation.
- It is possible to count point on genus 2 curves in char 2 quite quickly.

Goal of this talk: take a snapshot of the situation of the current research:

- What has been done ?
- What is currently (probably) feasible ?
- What is still completely open ?

(Disclaimer: as far as I know at this time!!!)

Generalities and notations

Let $\mathcal{C}$ be a curve over $\mathbb{F}_q$ of genus g. q can be a prime p, or a power of a small prime p: $q = p^n$ with n prime.

Goal: compute $\#\text{Jac}(\mathcal{C})/\mathbb{F}_q$.

Let $\Sigma : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q$ be the q-th power Frobenius that we extend to an endomorphism of Jac$(\mathcal{C})$. Its characteristic polynomial is of the form $\chi(t) = t^{2g} - s_1 t^{2g-1} + s_2 t^{2g-2} + \cdots + q^g$, and $\chi(1) = \#\text{Jac}(\mathcal{C})/\mathbb{F}_q$.

More precisely, $\chi(t)$ gives all the $\#\mathcal{C}/\mathbb{F}_{q^k}$.

Thm: (Hasse-Weil) Complex roots of $\chi(t)$ have abs value $\sqrt{q}$.

$\Longrightarrow$ bounds on coeffs of $\chi(t)$ and on cardinalities.

Ex: Elliptic curve $\mathcal{E}$, $g = 1$, Jac$(\mathcal{E}) \cong \mathcal{E}$.

$\chi(t) = t^2 - ct + q, \quad |c| \leq 2\sqrt{q}$

$\#\mathcal{E}/\mathbb{F}_q = q + 1 - c, \quad c$ is called the trace of $\mathcal{E}$.

Two classes of algorithms

ℓ-adic methods (Schoof and variants)

Compute $\chi(t)$ modulo several small primes ℓ using ℓ-torsion elements of Jac$(\mathcal{C})$.

(NB: provides an early-abort strategy.)

p-adic methods

Lift the curve to characteristic zero (p-adics) and study the action of Frobenius on the lifted objects.

Works
- only for small p
- faster and/or for a larger class of curves.

Two approaches:

- canonical lifting (Satoh and variants)
- Monsky-Washnitzer cohomology (Kedlaya)

ℓ-adic methods (Schoof-like)

Polynomial time algorithm, possible to deal with crypto sizes

Polynomial time algorithm, impossible to reach crypto sizes

Theoretical polynomial time algorithm, not implemented

	$g = 1$	$g = 2$	$g = 3$		
			hyper	super	general
$p = 2$					
p small		?			
p large					

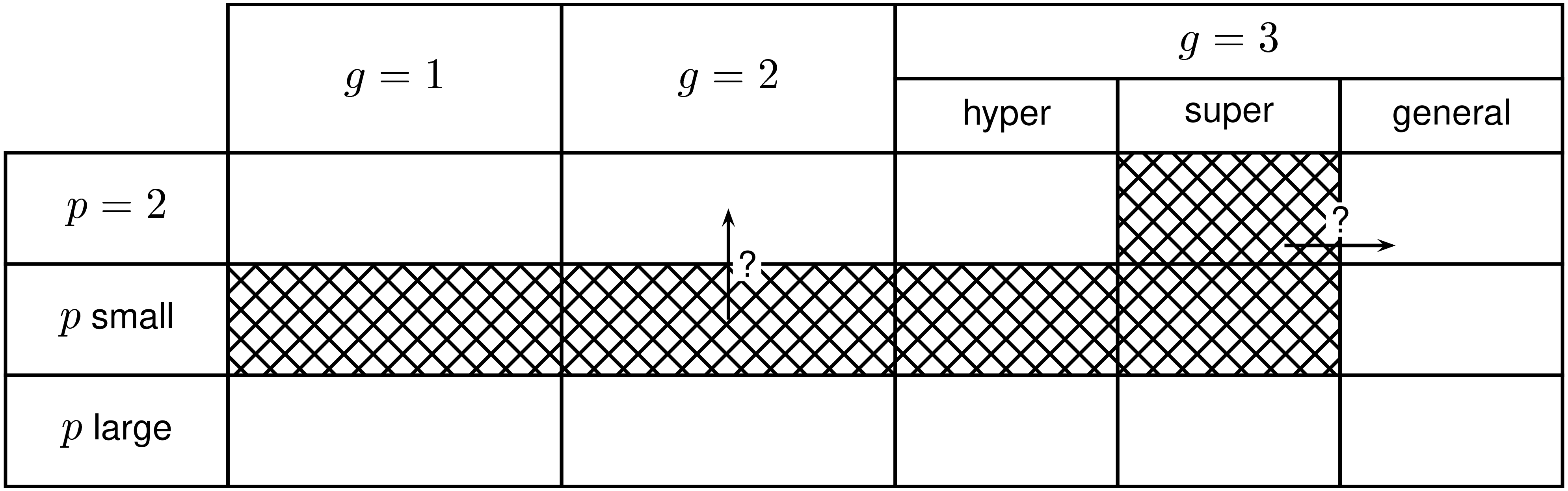
Names: Schoof, Elkies, Atkin, Couveignes, Lercier, Morain, Müller, Dewaghe, Vercauteren, Pila, Cantor, Kampkötter, Huang, Ierardi, Adleman, Harley, G.

p-adic methods (Satoh-like)

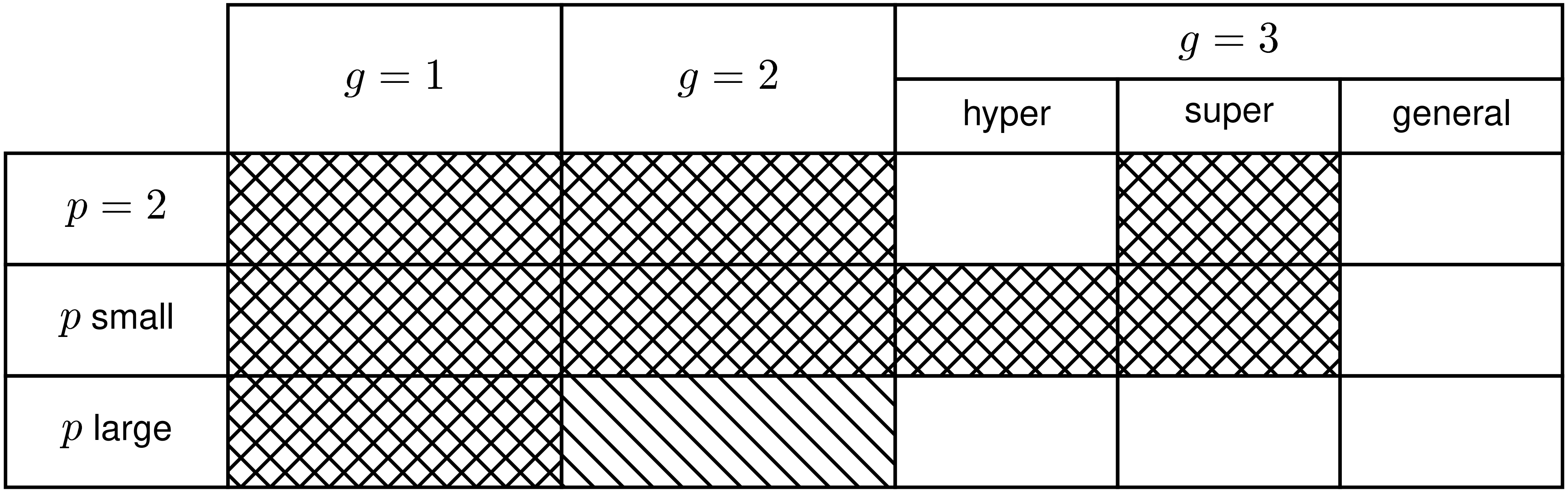
	$g=1$	$g=2$	$g=3$		
			hyper	super	general
$p=2$			?		
p small		?			
p large					

Names: Satoh, Skjernaa, Fouquet, Harley, G., Vercauteren, Mestre, Taguchi.

p-adic methods (Kedlaya)



Names: Kedlaya, Gürel, G.



$g = 1$
$g = 2$
$g = 3$
hyper
super
general
$p = 2$
p small
p large

Schoof algorithm and variants

Elliptic curves

- For ℓ prime
  - Compute an ℓ-torsion point $P \in E[\ell]$.
  - Find $c_\ell \in [0, \ell - 1]$ such that $P^{\Sigma^2} - [c_\ell]P^{\Sigma} + [q]P = 0$.
- Reconstruct c by Chinese remaindering.

Runs in polynomial time $O(\log^{5+\varepsilon} q)$.

Strategies to compute an ℓ-torsion point:

- Division polynomials (degree $O(\ell^2)$).
- Use modular equations and explicit isogenies (degree $O(\ell)$).

A lot of theoretical and practical work has been done (Schoof Elkies Atkin Couveignes Lercier Dewaghe Morain Müller ...)

$\Longrightarrow$conjectural runtime $O(\log^{4+\varepsilon} q)$.

Ref: Book of Blake–Seroussi–Smart, thesis of Lercier, Magma package source code.

Records: Morain 95 (10^{500}+...), Vercauteren 99 (2^{1999})

Extension to other curves

Pila: Schoof's algorithm for every abelian variety given explicitly by equations and collection of local maps for the the group law, in a projective space.

Runtime is polynomial.

Impossible to be implemented as described.

Hyperelliptic curves: definitions

A Hyperelliptic curve of genus g is a curve $\mathcal{C}$ of equation $y^2 + h(x)y = f(x)$, with $\deg h \le g$ and $\deg f = 2g + 1$.

Note: if char $\neq 2$, one can assume $h(x) = 0$.

A divisor on $\mathcal{C}$ is a formal sum of points of $\mathcal{C}$: $D = \sum n_i P_i$, with $P_i = (x_i, y_i)$ is on the curve. Its degree is $\deg D = \sum n_i$.

A divisor D is principal if there exists a function φ on $\mathcal{C}$ such that $D = \operatorname{div}(\varphi) = \sum \operatorname{ord}_{P_i}(\varphi) P_i$.

Prop: $\deg \operatorname{div}(\varphi) = 0$.

The Jacobian of $\mathcal{C}$ is the group of degree 0 divisors modulo the principal divisors.

Representation of Mumford-Cantor: In each class there is a divisor of the form $\sum_{1 \le i \le r} P_i - r(\infty)$, with $r \le g$ and $P_i \neq \overline{P_j}$. Let $u(x) = \prod(x - x_i)$ and $v(x)$ of degree $< r$ such that $v(x_i) = y_i$. Then represent D by $\langle u(x), v(x) \rangle$.

Schoof algorithm in genus 2 (joint work with Harley)

A subgroup of ℓ-torsion is given by an ideal. (dimension zero, degree $O(\ell^4)$.)

The problem is to find a nice representation of this ideal. Two choices in the literature:

- Kampkötter
- Cantor.

Let $D \in$ Jac$(\mathcal{C})$, write $D = P + Q \ \ (-2\infty)$, where P and Q are points of $\mathcal{C}$.

$$[\ell]D = 0 \Longleftrightarrow [\ell]P = -[\ell]Q$$

Apply Cantor's formulae, clear denominators and get an "easy to solve" system.

$$[\ell]P = \left\langle x^2 + \frac{d_1(x_P)}{d_0(x_P)}x + \frac{d_2(x_P)}{d_0(x_P)},\ y_P\left(\frac{e_1(x_P)}{e_0(x_P)}x + \frac{e_2(x_P)}{e_0(x_P)}\right)\right\rangle,$$

where $d_0, d_1, d_2, e_0, e_1, e_2$ polynomials, degree $O(\ell^2)$, computed by recursive formulae.

$$\begin{cases} E_1(x_1,x_2) & = & d_1(x_1)d_2(x_2) - d_1(x_2)d_2(x_1) & = & 0, \\ E_2(x_1,x_2) & = & d_0(x_1)d_2(x_2) - d_0(x_2)d_2(x_1) & = & 0, \\ F_1(x_1,x_2,y_1,y_2) & = & y_1e_1(x_1)e_0(x_2) + y_2e_1(x_2)e_0(x_1) & = & 0, \\ F_2(x_1,x_2,y_1,y_2) & = & y_1e_2(x_1)e_0(x_2) + y_2e_2(x_2)e_0(x_1) & = & 0, \end{cases}$$

Advantage: Two eq. involve only x_1 and x_2

$\Longrightarrow$ Eliminate x_2 by a resultant computation and get $R(x_1) = 0$ of degree $O(\ell^4)$.

Rem: Computed by evaluation / interpolation, without expanding the formulae: requires only univariate (fast) polynomial arithmetic.

It is now possible to compute a ℓ-torsion divisor D, find the pairs $(s_1, s_2) \bmod \ell$ s.t. $\Sigma^4(D) - s_1\Sigma^3(D) + s_2\Sigma^2(D) - s_1 q\Sigma(D) + q^2 D = 0$,
(iterate if necessary)

$\Longrightarrow$ Get $\chi(1) \bmod \ell$.

Practical results

- Implemented in Magma
- Finish the computation by a $\sqrt{\ }$ algorithm implemented in C (Birthday paradox method).

Exple: Let $\mathcal{C}$ defined by $y^2 = x^5 + 314159\ldots x^4 + \cdots$ over $\mathbb{F}_p$ with $p = 10^{19} + 51$.

Then $\quad$ #Jac$(\mathcal{C}) = 99999999982871020671452277000281660080,$

$$s_1 = 1712898036 \text{ and } s_2 = 11452277089352355350.$$

Rem: Result computed modulo $2^8 \times 3 \times 5 \times 7 \times 11 \times 13 = 3843840$ in about 11 days. Finished by R. Harley with a parallel implementation: ≈ 50 days on a single station.

Limitations

Problem: Have to compute with polynomials of degree $O(\ell^4)$.

Question: Can we adapt Elkies–Atkin techniques for genus 2 in order to reduce the degree $O(\ell^4) \to O(\ell^3)$?

Elkies–Atkin for elliptic curves

Let E elliptic curve over $\mathbb{F}_q$, $(\ell, q) = 1$, $\chi(t) = t^2 - ct + q$.

Factorization pattern of $\Phi_\ell(X, j_E)$ (degree $\ell + 1$):

- $(1, \ell)$ or $(1, \ldots, 1)$ if $c^2 - 4q \equiv 0 \bmod \ell$
- $(1, 1, r, \ldots, r)$ if $c^2 - 4q \equiv \square \bmod \ell$
- $(r, \ldots, r)$ if $c^2 - 4q \not\equiv \square \bmod \ell$

The pattern gives information about $c \bmod \ell$.

If there is a root in $\mathbb{F}_{q^k}$ one can construct a factor of ψ_ℓ of degree $k(\ell - 1)$.

Rem:

Roots of $\Phi_\ell(X, j_E)$ $\leftrightarrow$ ℓ-isogenies of E $\leftrightarrow$ subgroups of E of order ℓ.

Current work for genus 2 (joint work with Schost)

Theoretical and practical construction of a modular polynomial $\Xi_\ell(T)$ having same kind of properties as Φ_ℓ for elliptic curves.

Let $\mathcal{C}$ be a generic curve of equation $y^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$.

Thm: There exists a polynomial $\Xi_\ell(T, f_0, f_1, f_2, f_3)$ with the following prop:

- Ξ_ℓ is homogeneous with weights 1 for T and $5 - i$ for f_i.
- The degree of Ξ_ℓ in T is $\ell^3 + \ell^2 + \ell + 1$.
- Its factorization pattern is linked to cardinalities modulo ℓ.
- Ξ_ℓ is effectively computable.

Rem: This is the minimal polynomial of a parameter describing the subgroups of order ℓ in Jac$[\ell]$. (cf Charlap–Coley–Robbins, 1991).

Example: $\Xi_3(T)$

$$
\begin{aligned}
&\Xi_3(T, f_0, f_1, f_2, f_3) = \\
&T^{40} + 312 f_3 T^{38} - 7904 f_2 T^{37} + (-16344 f_3^2 + 183488 f_1) T^{36} + (337536 f_3 f_2 - 4179456 f_0) T^{35} + \\
&(345456 f_3^3 + 464064 f_3 f_1 - 2381088 f_2^2) T^{34} + (-6044256 f_3^2 f_2 - 196322688 f_3 f_0 + 33608832 f_2 f_1) T^{33} + \\
&(-3966084 f_3^4 - 58451616 f_3^2 f_1 + 52440768 f_3 f_2^2 + 1324005696 f_2 f_0 - 364387392 f_1^2) T^{32} + \\
&(57911232 f_3^3 f_2 + 280858752 f_3^2 f_0 + 591085824 f_3 f_2 f_1 - 241107200 f_2^3 - 1357926912 f_1 f_0) T^{31} + \\
&(26487216 f_3^5 + 388684800 f_3^3 f_1 - 378838944 f_3^2 f_2^2 + 8135750592 f_3 f_2 f_0 - 8711658240 f_3 f_1^2 + \\
&1407612480 f_2^2 f_1 - 30436734528 f_0^2) T^{30} + (-308962080 f_3^4 f_2 + 15709314816 f_3^3 f_0 - 7039348992 f_3^2 f_2 f_1 + \\
&1847884416 f_3 f_2^3 - 21546021888 f_3 f_1 f_0 - 4295003904 f_2^2 f_0 + 7496418816 f_2 f_1^2) T^{29} + (-97855128 f_3^6 + \\
&717110304 f_3^4 f_1 + 731068032 f_3^3 f_2^2 - 75564061056 f_3^2 f_2 f_0 - 47621349504 f_3^2 f_1^2 + 55550744448 f_3 f_2^2 f_1 + \\
&141450046848 f_3 f_0^2 - 8868753152 f_2^4 - 240969913344 f_2 f_1 f_0 + 76579431936 f_1^3) T^{28} + (815181696 f_3^5 f_2 - \\
&7457389056 f_3^4 f_0 - 11717124096 f_3^3 f_2 f_1 + 4346795520 f_3^2 f_2^3 + 144606394368 f_3^2 f_1 f_0 - \\
&71162772480 f_3 f_2^2 f_0 + 16981051392 f_3 f_2 f_1^2 - 11416397824 f_2^3 f_1 + 131172009984 f_2 f_0^2 - \\
&150350708736 f_1^2 f_0) T^{27} + (140091120 f_3^7 - 9497570112 f_3^5 f_1 + 1617994080 f_3^4 f_2^2 - \\
&202079060352 f_3^3 f_2 f_0 - 85220137728 f_3^3 f_1^2 + 146302604928 f_3^2 f_2^2 f_1 + 357349905792 f_3^2 f_0^2 - \\
&35760872448 f_3 f_2^4 - 857345121792 f_3 f_2 f_1 f_0 + 254363526144 f_3 f_1^3 + 16342396416 f_2^3 f_0 + \\
&4166138880 f_2^2 f_1^2 + 1294107121152 f_1 f_0^2) T^{26} + \cdots \quad \textit{(30 more slides)}
\end{aligned}
$$

Note: This construction is purely algebraic (does not involve modular forms).

Patterns of Ξ_3

$q \equiv 2 \bmod 3$	
pattern	$\#\mathsf{Jac}(\mathcal{C}) \bmod 3$
$(10)^4$	2
$(2)^2(6)^6$	1
$(4)(12)^3$	1
$(2)^{20}$	1
$(1)^2(2)(4)(8)^4$	0
$(1)^2(2)(3)^2(6)^5$	0
$(1)^8(2)^{16}$	0
$(1)^5(2)^4(3)(6)^4$	0
$(4)^{10}$	1, 2

$q \equiv 1 \bmod 3$	
pattern	$\#\mathsf{Jac}(\mathcal{C}) \bmod 3$
$(5)^8$	1, 2
$(1)(2)^2(3)(4)^2(12)^2$	0, 2
$(1)^4(2)^2(4)^8$	0, 2
$(1)(3)^4(9)^3$	0, 1
$(1)^4(3)^{12}$	0, 1
$(1)^{40}$	0, 1
$(4)^{10}$	2
$(2)^2(6)^6$	1
$(2)^{20}$	1
$(1)^4(3)^{12}$	0
$(1)^2(2)(3)^2(6)^5$	0
$(1)^5(2)^4(3)(6)^4$	0
$(1)^8(2)^{16}$	0

Atkin's algorithm in genus 2

Assume that all the (generic) Ξ_ℓ have been precomputed.

Algorithm:

1. While not enough information
   (a) Choose a new prime ℓ;
   (b) Compute the pattern of $\Xi_\ell(T)$;
   (c) Find the compatible values of $\chi(1) \bmod \ell$;
2. Recover $\chi(1)$ compatible with modular information.

Cost of one loop: Pattern costs a factorization of poly of deg $O(\ell^3)$; (c) is negligible.

Problem: Evaluating Ξ_ℓ at the coeff of the curve costs as many mult. as the number of monomials: $O(\ell^{12})$.

Question: Is it possible to compute $\Xi_\ell(T)$ for a given curve, with a cost lower than the subsequent factorization ?

p-adic methods using canonical lifting

Elliptic case (Satoh's algorithm)

Input: E curve over $\mathbb{F}_q$ with $q = p^n$, p small, $n > 2$. E not defined over $\mathbb{F}_{p^2}$

- Compute $\mathcal{E}$:= canonical lift of E to $\mathbb{Z}_q$ up to prec $\approx \frac{n}{2}$.
- Compute the trace of the Frobenius endomorphism of $\mathcal{E}$.

Thm (Serre–Tate): There exists a unique $\mathcal{E}$ (the canonical lift) such that

- $\mathcal{E} \equiv E \mod p$.
- $\mathrm{End}(\mathcal{E}) \cong \mathrm{End}(E)$.

Cor: $\mathrm{Tr}(\Sigma$ on $\mathcal{E}) = \mathrm{Tr}(\Sigma$ on $E)$.

Rem: $\mathcal{E}$ is characterized by $\Phi_p(j(\mathcal{E}), j(\mathcal{E})^\sigma) = 0$ and $j(\mathcal{E}) \cong j(E) \mod p$, where σ is the p-th power Frobenius.

Example

$\mathbb{F}_q = \mathbb{F}_{2^5} = \mathbb{F}_2[t]/(t^5 + t^2 + 1)$.

$\mathbb{Z}_q = \mathbb{Z}_{2^5} := \mathbb{Z}_2[t]/(t^5 + t^2 + 1)$, where $\mathbb{Z}_2$ is the set of integers taken modulo 2^k for increasing k.

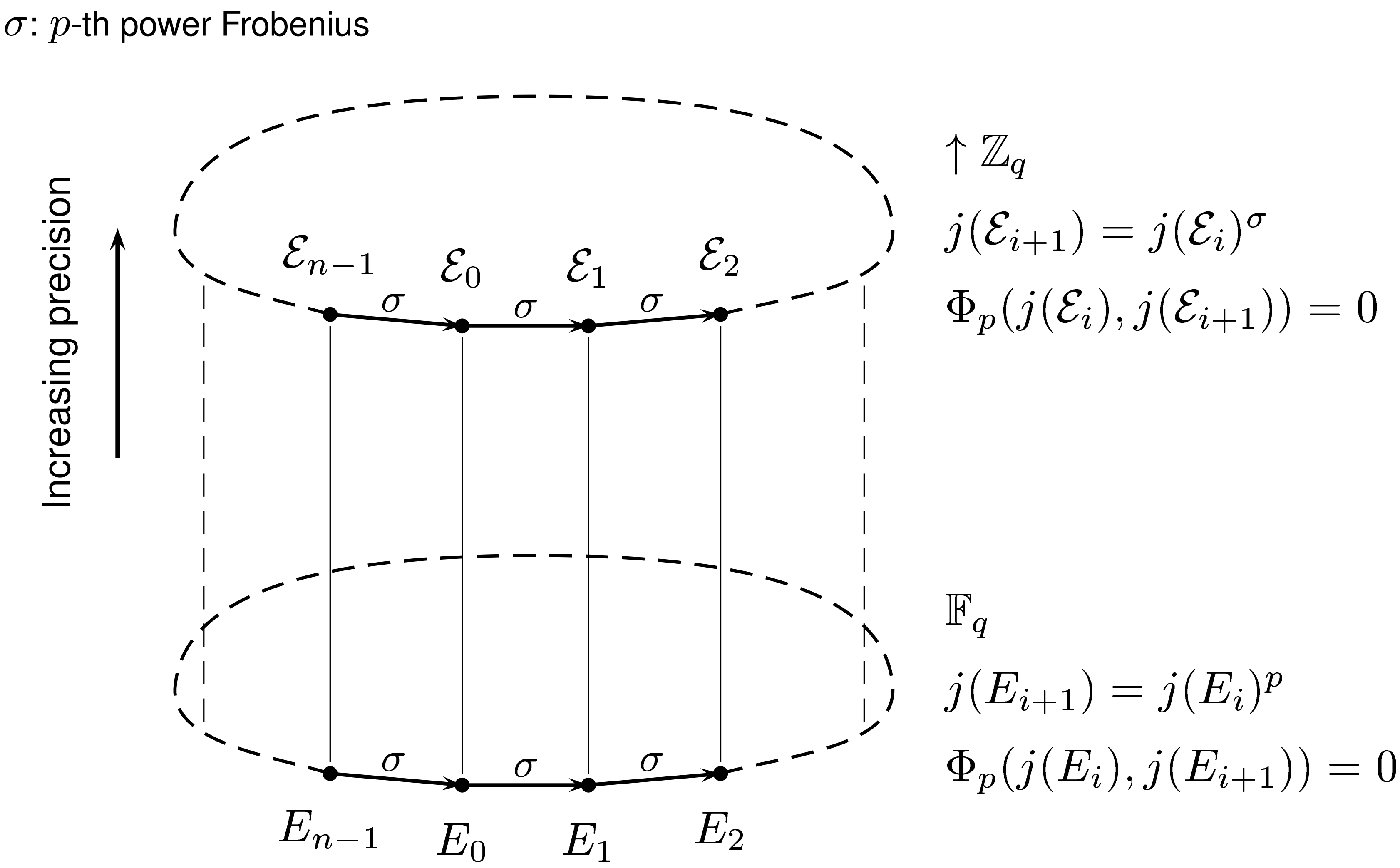
Let E over $\mathbb{F}_q$ of equation $y^2 + xy = x^3 + t$.

Then $j(E) = t^4 + t$. One can compute
$j(\mathcal{E}) \equiv 433t^4 + 184t^3 + 276t^2 + 47t + 466 \mod 2^{10}$.

Rem: Frobenius in $\mathbb{Z}_q$ is not as simple as in $\mathbb{F}_q$ (squaring).

$j(E)^\sigma = t^3 + 1 = j(E)^2$ in $\mathbb{F}_q$. One has $\Phi_2(j(E), j(E)^\sigma) = 0$ in $\mathbb{F}_q$.

$j(\mathcal{E})^\sigma \equiv -494t^4 - 271t^3 + 240t^2 + 50t - 349 \mod 2^{10} \neq j(\mathcal{E})^2$. One has $\Phi_2(j(\mathcal{E}), j(\mathcal{E})^\sigma) \equiv 0 \mod 2^{10}$ in $\mathbb{Z}_q$.


σ: p-th power Frobenius
Increasing precision
$\mathcal{E}_{n-1}$ $\mathcal{E}_0$ $\mathcal{E}_1$ $\mathcal{E}_2$
σ σ σ
$\uparrow \mathbb{Z}_q$
$j(\mathcal{E}_{i+1}) = j(\mathcal{E}_i)^\sigma$
$\Phi_p(j(\mathcal{E}_i), j(\mathcal{E}_{i+1})) = 0$
$\mathbb{F}_q$
$j(E_{i+1}) = j(E_i)^p$
$\Phi_p(j(E_i), j(E_{i+1})) = 0$
σ σ σ
E_{n-1} E_0 E_1 E_2

Computing the canonical lift: Satoh's algorithm

Let $j_i = j(\mathcal{E})^{\sigma^i}$. Then $j_n = j_0 = j(\mathcal{E})$.

These n indeterminates are related by n polynomial equations, namely

$$\Phi_p(j_i, j_{i+1}) = 0, \quad \text{for } i \in [0, n-1].$$

One can use a Newton iteration to compute an approximation of the solutions of this system with a quadratic convergence.

(to find a solution of $F(X) = 0$, iterate $X \leftarrow X - \frac{F(X)}{F'(X)}$.)

Complexity: $O(n^3)$ in space and $O(n^{3+\varepsilon})$ in time.

Note: Satoh's algorithm was originally designed for $p > 3$ and was extended to $p = 2$ independently by Skjernaa and Fouquet–G.–Harley.

Records: Satoh: $\mathbb{F}_{5^{569}}$, FGH: $\mathbb{F}_{2^{8009}}$ (2 weeks, 17GB).

Computing the canonical lift: Vercauteren et al.'s and AGM algorithms (char 2)

Fact (first noticed by Vercauteren): if $j(\mathcal{E})$ is known to precision 2^k, then the equation $\Phi_2(j(\mathcal{E}), j(\mathcal{E})^\sigma) = 0$ determines $j(\mathcal{E})^\sigma$ to precision 2^{k+1}.

On the picture: replace the cylinder by a spiral!

Complexity in space reduced to $O(n^2)$.

- Vercauteren et al. : uses a Newton iteration.
- AGM (Mestre, Harley, G.) : uses compact formulae for computing a 2-isogeny.

Rem: in Vercauteren, only the j-invariants are lifted, whereas in AGM, we manipulate equations of curves. Still, at each step, the j-invariant computed in Vercauteren is the one of the curve whose equation is computed via AGM.

Details of the AGM algorithm

Fact 1: $\mathcal{E}_i$ and $\mathcal{E}_{i+1} = \mathcal{E}_i^\sigma$ are linked by a 2-isogeny.

Fact 2: Let $\mathcal{E}_{ab} \ : \ y^2 = x(x-a^2)(x-b^2)$ and
$\mathcal{E}'_{a'b'} \ : \ y^2 = x(x-a'^2)(x-b'^2)$, with $a' = (a+b)/2$ and $b' = \sqrt{ab}$.

Then $\mathcal{E}_{ab}$ and $\mathcal{E}'_{a'b'}$ are 2-isogenous.

Cor: By choosing the appropriate change of variables, if (a, b) corresponds to the curve $\mathcal{E}$ to precision 2^k, then $((a+b)/2, \sqrt{ab})$ corresponds to the curve $\mathcal{E}^\sigma$ to precision 2^{k+1}.

Algo:

- Put E in form $y^2 = x(x-a^2)(x-b^2)$.
- Iterate $(a, b) \leftarrow (\frac{a+b}{2}, \sqrt{ab})$.
- When enough precision, do one complete loop to compute the trace of Frobenius.

Pseudo-code of AGM

Let $\mathcal{E}$ be $y^2 + xy = x^3 + \alpha$, with $0 \neq \alpha \in \mathbb{F}_{2^n}$.

1) Let $a = 1 + 8\alpha$ and $b = 1$.

2) Repeat $\frac{n}{2} + O(1)$ times with increasing precision:

$$(a, b) \leftarrow \left(\frac{a+b}{2}, \sqrt{ab}\right)$$ *choose sqrt congruent to 1 modulo 4*

3) $(a_0, b_0) \leftarrow (a, b)$.

4) Repeat n times:

$$(a, b) \leftarrow \left(\frac{a+b}{2}, \sqrt{ab}\right)$$

5) Return $\frac{a_0}{a} \mod 2^n$ as a signed integer in $[-2\sqrt{q}, 2\sqrt{q}]$.

“Proof”

At step 3, (a_0, b_0) gives a curve $\mathcal{E}_0$ close enough to its canonical lift.

$$\begin{tikzcd} \mathcal{E}_0 \arrow[r, "\sigma_0"] & \mathcal{E}_1 \arrow[r, "\sigma_1"] & \dots \arrow[r, "\sigma_{n-2}"] & \mathcal{E}_{n-1} \arrow[r, "\sigma_{n-1}"] & \mathcal{E}_n \arrow[llll, bend left, "\cong \varphi"] \end{tikzcd}$$

The Frobenius on $\mathcal{E}_0$ is given by $\Sigma = \varphi \circ \sigma_{n-1} \circ \sigma_{n-2} \circ \cdots \circ \sigma_0$, where everything can be made explicit.

On differential forms: $\Sigma^*(\frac{\mathrm{d}x}{y}) = \sigma_0^* \circ \sigma_1^* \circ \cdots \circ \sigma_{n-1}^* \circ \varphi^*(\frac{\mathrm{d}x}{y})$

But we can compute $\sigma_i^*(\frac{\mathrm{d}x}{y}) = 2\frac{\mathrm{d}x}{y}$ and $\varphi^*(\frac{\mathrm{d}x}{y}) = \frac{a_n}{a_0}\frac{\mathrm{d}x}{y}$.

Hence $\Sigma^*(\frac{\mathrm{d}x}{y}) = 2^n \frac{a_n}{a_0}\frac{\mathrm{d}x}{y}$.

Therefore $2^n \frac{a_n}{a_0}$ is a root of $\chi(t)$ and the other has to be $\frac{a_0}{a_n}$.

$\Longrightarrow \mathrm{Tr}(\Sigma) = \frac{a_0}{a_n} + 2^n \frac{a_n}{a_0}$

Results (implementation by R. Harley)

Times on Alpha processor, EV6 at 750 MHz (Spring 2001).

- Record:

elliptic curves: $n = 15,013$ bits in 24 days.

- Curve generation (crypto size):

Combination of AGM (without fast norm computation) and early-abort (Lercier).

	counting 1 curve	finding a good curve
160 bits	0.4 sec	4 sec
200 bits	0.8 sec	8 sec

AGM in genus 2

Key points:

- AGM algorithm does not require modular equations nor Vélu's formulae.
- Bost–Mestre (1988): based on work by Richelot extended the link between AGM and isogenies to the case of genus 2 curves.

Def: If S and T are monic polynomials of degree 2, define $[S,T](x) = S'(x)T(x) - S(x)T'(x)$.

Thm (AGM): Let $\mathcal{C}$ of equation $y^2 = P(x)Q(x)R(x)$, where P, Q, R are monic of degree 2. Let $\mathcal{C}'$ the curve given by

$$\Delta y^2 = [Q,R](x)[R,P](x)[P,Q](x),$$

where Δ is the determinant of P, Q, R in the basis $1, x, x^2$.

Then Jac$(\mathcal{C})$ and Jac$(\mathcal{C}')$ are $(2,2)$-isogenous abelian varieties. Moreover everything can be made explicit.

Rem: In the context of canonical lifting of a curve over $\mathbb{F}_{2^n}$, the curves $\mathcal{C}$ and $\mathcal{C}^\sigma$ have $(2,2)$-isogenous jacobians.

Details to be filled in to turn this into a point counting algorithm:

- Initialization: transform the given curve in char 2 into a curve in the form as in the theorem.
- Computation of Frobenius by doing one complete loop: work with differential forms, make all the maps explicit.

Results / Records (R. Harley)

Genus 2 curves: $n = 4000$ bits in 6 days.

(compare with elliptic curves over 8000 bits in 2.5 days)

Remarks about norm computation

The final loop in all the Satoh-like algorithms can be replaced by a norm computation. See recent work by Satoh–Skjernaa–Taguchi and Harley.

Ex: AGM revisited

Let $\mathcal{E}$ be $y^2 + xy = x^3 + \alpha$, with $0 \neq \alpha \in \mathbb{F}_{2^n}$.

1) Let $a = 1 + 8\alpha$ and $b = 1$.

2) Repeat $\frac{n}{2} + O(1)$ times with increasing precision:

$$(a, b) \to \left(\frac{a+b}{2}, \sqrt{ab}\right)$$

choose sqrt congruent to 1 modulo 4

3) Let $(a', b') = \left(\frac{a+b}{2}, \sqrt{ab}\right)$

4) Return $\mathrm{Norm}(\frac{a}{a'})$ in $[-2\sqrt{q}, 2\sqrt{q}]$.

Rem: This is also possible in genus 2.

Beyond the AGM

- Mix AGM and ideas of Satoh–Skjernaa–Taguchi. This should result in a smaller modular equation to solve and a gain by a constant factor.
- Extend AGM to $p > 2$:
  - For $g = 1$, and $p = 3, 5$: not to difficult, formulae are in Borwein and Borwein *Pi and the AGM*.
  - For $g = 2$: ???
- Extend Satoh or AGM (or any variant) to higher genus: ???

Conclusion

Still many developments:

- For elliptic curves in characteristic 2
  - Better asymptotic complexity (during the lift and during the norm computation)
  - Better implementations
- For other curves / other characteristics:
  - still almost no implementation (except for $g = p = 2$)
  - still extensions / improvements to do.

Other new ideas coming out:

Alan Lauder and Daqing Wan designed another p-adic algorithm based on Dwork's work (no canonical lifting).