Fast arithmetics for Artin-Schreier extensions

L. De Feo
joint work with Éric Schost

École Polytechnique, Paris, France

December 19, 2008
University of Western Ontario, London

Artin-Schreier

Definition (Artin-Schreier polynomial)

$\mathbb{K}$ a field of characteristic p, $\alpha \in \mathbb{K}$

$$X^p - X - \alpha$$

is an Artin-Schreier polynomial.

Theorem

$\mathbb{K}$ finite. $X^p - X - \alpha$ irreducible $\Leftrightarrow$ $\mathrm{Tr}_{\mathbb{K}/\mathbb{F}_p}(\alpha) \neq 0$.
If $\eta \in \mathbb{K}$ is a root, then $\eta + 1, \dots, \eta + (p-1)$ are roots.

Definition (Artin-Schreier extension)

$\mathcal{P}$ an irreducible Artin-Schreier polynomial.

$$\mathbb{L} = \mathbb{K}[X]/\mathcal{P}(X).$$

$\mathbb{L}/\mathbb{K}$ is called an Artin-Schreier extension.

Our context

$$\mathbb{U}_k = \frac{\mathbb{U}_{k-1}[X_k]}{P_{k-1}(X_k)}$$

$$\Big|\, p$$

$$\mathbb{U}_{k-1}$$

$$\vdots$$

$$\mathbb{U}_1 = \frac{\mathbb{U}_0[X_1]}{P_0(X_1)}$$

$$\Big|\, p$$

$$\mathbb{U}_0 = \mathbb{F}_{p^d} = \frac{\mathbb{F}_p[X_0]}{Q(X_0)}$$

Towers over finite fields

$$P_i = X^p - X - \alpha_i$$

We say that $(\mathbb{U}_0, \dots, \mathbb{U}_k)$ is defined by $(\alpha_0, \dots, \alpha_{k-1})$ over $\mathbb{U}_0$.

ANY extension of degree p can be expressed this way

Motivations

- p-torsion points of abelian varieties;
- Isogeny computation [Couveignes '96].

Size, complexities

$$\#\mathbb{U}_i \;=\; p^{p^i d}$$

$\mathbb{U}_k$ — $\mathbb{U}_{k-1}$ - - - $\mathbb{U}_1$ — $\mathbb{U}_0$

Optimal representation

All common representations achieve it: $O(p^i d \log p)$

Complexities in $\mathbb{F}_p$-operations

optimal:	$O(p^i d)$	addition
quasi-optimal:	$\tilde{O}(i^a p^i d)$	FFT multiplication
almost-optimal:	$\tilde{O}(i^a p^{i+b} d)$	
suboptimal:	$\tilde{O}(i^a p^{i+b} d^c)$	
too bad:	$\tilde{O}\left(i^a (p^{i+b})^e d^c\right)$	naive multiplication

Multiplication function $\mathsf{M}(n)$

FFT: $\mathsf{M}(n) = O(n \log n \log\log n)$, Naive: $\mathsf{M}(n) = O(n^2)$.

Plan

1. Representation
2. Arithmetics
3. Implementation
4. Applications and benchmarks

Representation matters!

$\mathbb{U}_k$

$\mathbb{U}_{k-1}$

$\mathbb{U}_1$

$\mathbb{U}_0$

Multivariate representation of $v \in \mathbb{U}_i$

$$v = X_0^{d-1} X_1^{p-1} \cdots X_i^{p-1} + 2X_0^{d-1} X_1^{p-1} \cdots X_i^{p-2} + \cdots$$

Univariate representation of $v \in \mathbb{U}_i$

- $\mathbb{U}_i = \mathbb{F}_p[x_i]$,
- $v = c_0 + c_1 x_i + c_2 x_i^2 + \cdots + c_{p^i d-1} x_i^{p^i d-1}$ with $c_i \in \mathbb{F}_p$.

How much does it cost to...

- Multiply?
- Express the embedding $\mathbb{U}_{i-1} \subset \mathbb{U}_i$?
- Express the vector space isomorphism $\mathbb{U}_i = \mathbb{U}_{i-1}^p$?
- Switch between the representations?

A primitive tower

$$\mathbb{U}_k$$
$$|$$
$$\mathbb{U}_{k-1}$$
$$\vdots$$
$$\mathbb{U}_1$$
$$|$$
$$\mathbb{U}_0$$

Definition (Primitive tower)

A tower is primitive if $\mathbb{U}_i = \mathbb{F}_p[X_i]$.

In general this is not the case. Think of $P_0 = X^p - X - 1$.

Theorem (extends a result in [Cantor '89])

Let $x_0 = X_0$ *such that* $\mathrm{Tr}_{\mathbb{U}_0/\mathbb{F}_p}(x_0) \neq 0$ *, let*

$$\begin{aligned} P_0 &= X^p - X - x_0 \\ P_i &= X^p - X - x_i^{2p-1} \end{aligned}$$

with x_{i+1} *a root of* P_i *in* $\mathbb{U}_{i+1}$.
Then, the tower defined by $(P_0, \ldots, P_{k-1})$ *is primitive.*

Some tricks to play when $p = 2$.

Computing the minimal polynomials

We look for Q_i, the minimal polynomial of x_i over $\mathbb{F}_p$

$$\mathbb{U}_k \quad \mathbb{U}_{k-1} \quad \mathbb{U}_1 \quad \mathbb{U}_0$$

Algorithm [Cantor '89]

- $Q_0 = Q$ easy,
- $Q_1 = Q_0(X^p - X)$ easy,

Let ω be a $2p-1$-th root of unity,

- $q_{i+1} = \prod_{j=0}^{2p-2} Q_i(\omega^j X)$ not too hard[1],
- $Q_{i+1} = q_{i+1}(X^p - X)$ easy.

[1] No need to factor Φ_{2p-1}, one can simply work modulo it.

Complexity

$$O\left(\mathsf{M}(p^{i+2}d) \log p\right)$$

Plan

Level embedding

$\mathbb{U}_k$

$\mathbb{U}_{k-1}$

$\mathbb{U}_1$

$\mathbb{U}_0$

Push-down

Input $v \dashv \mathbb{U}_i$,

Output $v_0, \dots, v_{p-1} \dashv \mathbb{U}_{i-1}$ such that $v = v_0 + \cdots + v_{p-1}x_i^{p-1}$.

Lift-up

Input $v_0, \dots, v_{p-1} \dashv \mathbb{U}_{i-1}$,

Output $v \dashv \mathbb{U}_i$ such that $v = v_0 + \cdots + v_{p-1}x_i^{p-1}$.

Complexity function $\mathsf{L}(i)$

It turns out that the two operations lie in the same complexity class, we note $\mathsf{L}(i)$ for it:

$$\mathsf{L}(i) \;=\; O\left(p\mathsf{M}(p^i d) + p^{i+1} d \log_p(p^i d)^2\right)$$

Push-down

Input $v \dashv \mathbb{U}_i$,
Output $v_0, \ldots, v_{p-1} \dashv \mathbb{U}_{i-1}$ s.t. $v = v_0 + \cdots + v_{p-1} x_i^{p-1}$.

1. Reduce v modulo $x_i^p - x_i - T^{2p-1}$ by a divide-and-conquer approach,
2. each of the coefficients of x_i has degree in x_{i-1} less than $2\deg(v)$,
3. reduce each of the coefficients.

Duality I

Dual vector space

$\mathbb{U}_i^*$ the space of $\mathbb{F}_p$-linear forms over $\mathbb{U}_i$

$$\begin{aligned} B \text{ base of } \mathbb{U}_i &\rightarrow B^* \text{ base of } \mathbb{U}_i^* \\ \ell \in \mathbb{U}_i^* &\rightarrow (\ell(B_0), \dots, \ell(B_n)) \end{aligned}$$

Multiplication

Let $v \in \mathbb{U}_i$, multiplication by v is a linear application $\mathbb{U}_i \rightarrow \mathbb{U}_i$ with matrix M_v:

$$\left(\quad M_v \quad \right) \left(x \right) \mapsto \left(vx \right)$$

Transposed multiplication

Let $v \in \mathbb{U}_i$, $\ell \in \mathbb{U}_i^*$, transposed multiplication $v \cdot \ell$ is the linear form

$$\left(\quad v \cdot \ell \quad \right) \left(x \right) = \left(\quad \ell \quad \right) \left(\quad M_v \quad \right) \left(x \right) \mapsto \left(\quad \ell \quad \right) \left(vx \right) = \ell(vx)$$

hence M_v^T is the linear application computing $v \cdot \ell$ from ℓ.

Duality II

Change of basis

Vector spaces $V^B = V^D$ with bases B and D.

$$M \;:\; V_B \to V_D$$
$$M^T \;:\; V^{D^*} \to V^{B^*}$$

M^T is the dual change of basis.

Push-down

Push-down is a change of basis $P \;:\; \mathbb{U}_i^U \to \mathbb{U}_i^D$

$$U = \text{ polynomial basis in } x_i$$
$$D = \text{ bivariate basis in } x_i, x_{i-1}$$

hence $P^T \;:\; \mathbb{U}_i^{D^*} \to \mathbb{U}_i^{U^*}$.

Truncated power series

P^T sends linear forms $\ell \in \mathbb{U}_i^{D^*}$ onto the basis U^*:

$$\ell(1), \quad \ell(x_i), \quad \ell(x_i^2), \quad \ldots, \quad \ell(x_i^{p^i d-1})$$

These can be seen as the first coefficients of a formal power series ([Shoup '99]):

$$\sum_{j>0} \ell(x_i^j) Z^j$$

Lift-up

Trace formulas [Pascal, Schost '06, Rouillier '99]

Let $\ell \neq 0$ in $\mathbb{U}_i^{D^*}$, let $v_D \in \mathbb{U}_i^D$, then

$$\frac{\sum_{j>0} v_D \cdot \ell(x_i^j) Z^j}{\sum_{j>0} \ell(x_i^j) Z^j} = \frac{N(Z)}{D(Z)}$$

is in $\mathbb{F}_p(Z)$. Then the image of v_D in $\mathbb{U}_i^U$ is

$$v_U = \frac{\mathrm{rev}(N)(x_i)}{\mathrm{rev}(D)(x_i)}.$$

Transposition principle (see [Bürgisser, Clausen, Shokrollahi])

- We don't bother computing the matrices M_v and P,
- we use transposition principle instead.
- computing $v_D \cdot \ell$ is transposed multiplication in $\mathbb{U}_i^D$,
- computing the power series is transposed Push-down.

Lift-up

Input $v_0, \ldots, v_{p-1} \dashv \mathbb{U}_{i-1}$
Output $v \dashv \mathbb{U}_i$ s.t. $v = v_0 + \cdots + v_{p-1} x_i^{p-1}$

1. Chose a linear form $\ell \in \mathbb{U}_i^*$,
2. compute $\ell_v = v \cdot \ell$,
3. compute $P_1(Z) = \text{Push-down}^T(\ell)$,
4. compute $P_v(Z) = \text{Push-down}^T(\ell_v)$,
5. compute $V_1 = P_1(Z) \cdot \operatorname{rev}(Q_i)(Z) \mod Z^{p^i d - 1}$,
6. compute $V_v = P_v(Z) \cdot \operatorname{rev}(Q_i)(Z) \mod Z^{p^i d - 1}$,
7. return $\operatorname{rev}(V_v)(x_i) \;/\; \operatorname{rev}(V_1)(x_i)$.

Other operations, Isomorphism

$$\mathbb{U}_k \;-\; \mathbb{U}_{k-1} \;\cdots\; \mathbb{U}_1 \;-\; \mathbb{U}_0$$

Other operations

By divide and conquer, we give efficient routines for most operations in $\mathbb{U}_i$:

- push-down the operands;
- recursively solve p instances in $\mathbb{U}_{i-1}$;
- combine the results;
- lift-up.

It works fairly well for

- inversion,
- traces,
- iterated frobenius,
- p-th roots,
- ...

Isomorphism [Couveignes '00]

- Let $(\alpha_0, \dots, \alpha_{k-1})$ define another tower over $\mathbb{U}_0$,
- factoring $X^p - X - \alpha_i$ in $\mathbb{U}_{i+1}$ gives an isomorphism.
- Couveignes gives a fast factoring algorithm for this case,
- this way fast arithmetics can be brought to this new tower.

$$\mathbb{U}'_k \;-\; \mathbb{U}'_{k-1} \;\cdots\; \mathbb{U}'_1 \;-\; \mathbb{U}_0$$

Plan

1. Representation
2. Arithmetics
3. Implementation
4. Applications and benchmarks

Engineering issues

Field Lattices

$$\begin{tikzcd}
 & & \mathbb{F}_{p^{12}} \arrow[dl, dash] \arrow[ddr, dash] & & \\
\dfrac{\mathbb{F}_p[X]}{P(X)} \arrow[r, dash] \arrow[dddrr, dash] & \mathbb{F}_{p^6} \arrow[ddr, dash] & & & \\
 & & & \mathbb{F}_{p^4} \arrow[r, dash] \arrow[dl, dash] & \dfrac{\mathbb{F}_{p^2}[Y]}{Q(Y)} \arrow[dll, dash] \\
 & & \mathbb{F}_{p^2} \arrow[d, dash] & & \\
 & & \mathbb{F}_p & &
\end{tikzcd}$$

Stem lattice

$$\begin{tikzcd}
 & \mathbb{U}_k \arrow[r, dash] \arrow[d, dash] & \mathbb{L}_k \\
\mathbb{K}_{k-1} \arrow[r, dash] & \mathbb{U}_{k-1} \arrow[r, dash] \arrow[dd, dashed, dash] & \mathbb{L}_{k-1} \\
 & & \\
\mathbb{K}_1 \arrow[r, dash] & \mathbb{U}_1 \arrow[r, dash] \arrow[d, dash] & \mathbb{L}_1 \\
 & \mathbb{U}_0 &
\end{tikzcd}$$

Platform

Magma (native language, C)

- full support for field lattices,
- FFT multiplication,
- large library,
- transparent type-system,
- not open source.

Sage (Python, Cython, C++)

- Open source,
- future support for field lattices,
- transparent type-system,
- large library,
- large community,
- interfaces to NTL, Pari, and others.

NTL (C++)

- Open source,
- optimised library for $\mathbb{F}_2$,
- support for transposed operations,
- no support for field lattices,
- FFT for $p > 2$ (via gmp), Karatsuba for $p = 2$,
- non-transparent type system (three different types for finite fields),
- relatively restricted library (no integer factorisation),
- a “one man library”.
- “stubborn design”.

Implementation in NTL

Adding transparency

- Use templates instead of native types for finite fields,
- add wrappers when necessary for compatibility.
- Implement resource localisation to work in field lattices.

Adding functionalities

- Pollard rho for integer factorisation,
- folklore algorithm for cyclotomic polynomials,
- elliptic curve addition (classic and Montgomery).

Conclusions

- NTL lacks a type : Field !
- Our implementation is not maintainable : every function in NTL needs to be wrapped, every change needs to be reflected.
- Few chances for future improvements.

Plan

p^k-torsion

p-division

- In ordinary elliptic curves $E[p^k] \simeq \mathbb{Z}/p^k\mathbb{Z}$.
- Knowing a p^i-torsion point,
- factorise the p-division polynomial to find a p^{i+1}-torsion point.

Make it Artin-Schreier [Voloch '90]

- By a change of variables we can factor an Artin-Schreier polynomial instead,
- using Couveignes' algorithm for the isomorphism, we can do it efficiently.

Isogeny interpolation

Computing an isogeny of degree ℓ between two curves E and F

The idea [Couveignes '96, '00]

- Compute enough ($p^k \sim \ell$) torsion points in E and F,
- since the curves are isogenous, the towers are isomorphic,
- use the isomorphism algorithm to bring them to the same primitive tower,
- interpolate the isogeny over the points.

Fast interpolation [D.F. '07]

- Use the same divide-and-conquer approach as for the arithmetics in $\mathbb{U}_k$,
- throw some Galois-theory in,
- the interpolation step can be done in $\tilde{O}(\ell^2)$.

Implementation

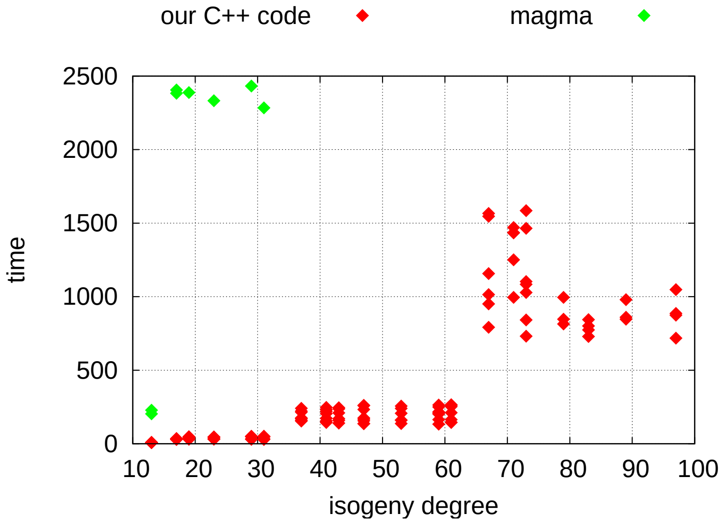
- Implementation in NTL for $p = 2$ (no FFT).
- Benchmarks on two fields: $\mathbb{F}_{2^{101}}$ and $\mathbb{F}_{2^{1999}}$.
- Up to 15 levels on a Intel Core 2 @2GHz, 4GB ram.

	$\mathbb{F}_{2^{101}}$	$\mathbb{F}_{2^{1999}}$	levels
Construction of Q_i	$0:42$	$42:00$	15
Push-down, lift-up	$0:30$	$20:00$	15
Couveignes '00	$3:40:00$		15
Couveignes '00	$1:30:00$	$76:40:00$	13

- We are working on a new, faster, NTL implementation for any p;
- porting to a computer algebra platform is in study.

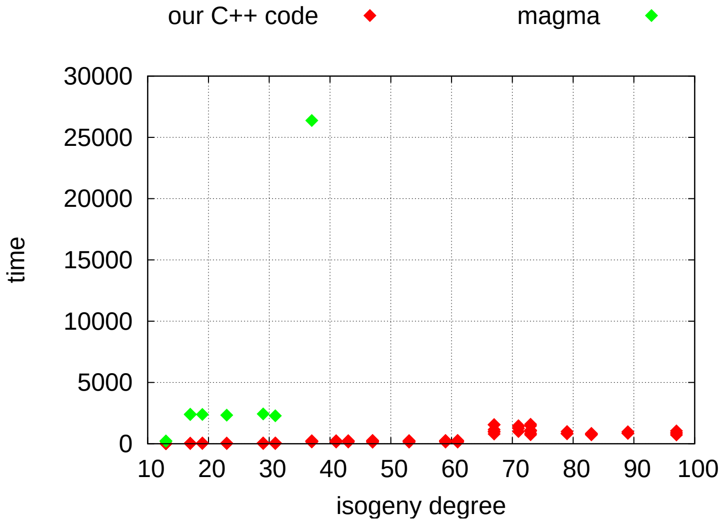
Benchmarks on isogenies

Over $\mathbb{F}_{2^{101}}$, on an AMD Athlon 64 X2 Dual Core Processor 4000+, 5GB ram



Benchmarks on isogenies

Over $\mathbb{F}_{2^{101}}$, on an AMD Athlon 64 X2 Dual Core Processor 4000+, 5GB ram



Bibliography

P. Bürgisser, M. Clausen, and A. Shokrollahi.
Algebraic complexity theory, volume 315 of *Grundlehren Math. Wiss.*
Springer–Verlag, 1997.

D. G. Cantor.
On arithmetical algorithms over finite fields.
Journal of Combinatorial Theory, Series A 50, 285-300, 1989.

J.-M. Couveignes.
Computing ℓ-isogenies with the p-torsion.
Lecture Notes in Computer Science vol. 1122, pages 59–65, Springer-Verlag, 1996.

J.-M. Couveignes.
Isomorphisms between Artin-Schreier tower.
Math. Comp. 69(232): 1625–1631, 2000.

L. De Feo.
Calcul d'isogénies.
Master thesis. http://www.lix.polytechnique.fr/∼defeo

Bibliography

C. Pascal and É. Schost.
Change of order for bivariate triangular sets.
In *ISSAC'06*, pages 277–284. ACM, 2006.

F. Rouillier.
Solving zero-dimensional systems through the Rational Univariate Representation.
Appl. Alg. in Eng. Comm. Comput., 9(5):433–461, 1999.

V. Shoup.
Efficient computation of minimal polynomials in algebraic extensions of finite fields.
In *ISSAC'99*, ACM Press, 1999.

J.F. Voloch.
Explicit p-descent for Elliptic Curves in Characteristic p.
Compositio Mathematica 74, pages 247–58, 1990.