



PhD 2.0

Connect to the WiFi network¹²: **Luca's defense**

Open your browser, navigate to any page, and experience free access to:

- electronic manuscript,
- electronic presentation,
- live cast of the defense,
- many other goodies!

Also check us out on: <http://www.lix.polytechnique.fr/~defeo/>

¹This WiFi does not give you access to the WWW (not even DNS, sorry for IP-over-DNSers)

²No DOSes, please.



FAST ALGORITHMS FOR TOWERS OF FINITE FIELDS AND ISOGENIES

Luca De Feo

LIX, École Polytechnique & INRIA Saclay, Projet TANC

December 13, 2010
École Polytechnique, Palaiseau

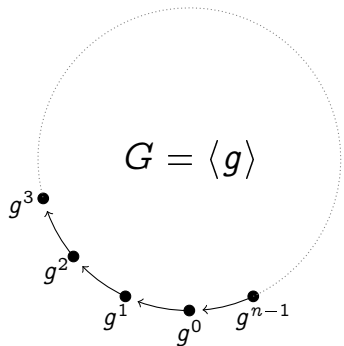


Outline

- 1 Introduction
- 2 Computing isogenies over finite fields
- 3 Artin-Schreier towers
- 4 Implementations
- 5 Conclusion



The Discrete Logarithm Problem

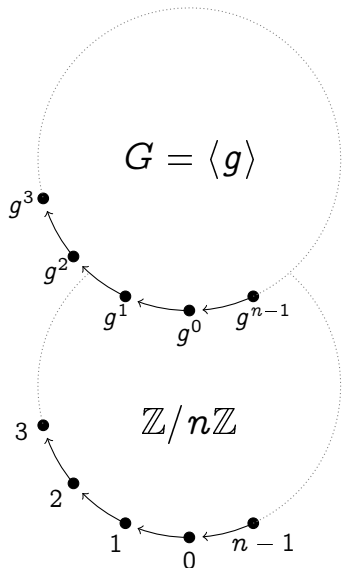


Exponentiation in a cyclic group

Let $G = \langle g \rangle$ be a cyclic group of order n . Define

$$\begin{aligned} \exp_g : \mathbb{Z}/n\mathbb{Z} &\rightarrow G, \\ x &\mapsto g^x. \end{aligned}$$

The Discrete Logarithm Problem



Exponentiation in a cyclic group

Let $G = \langle g \rangle$ be a cyclic group of order n . Define

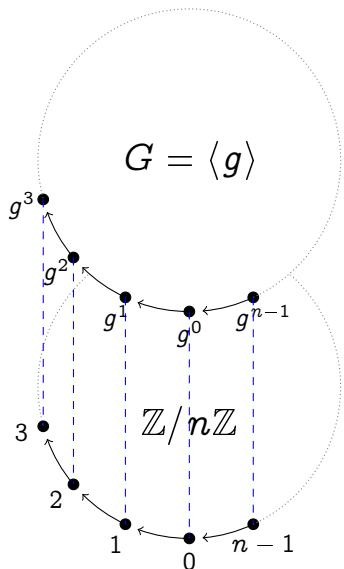
$$\begin{aligned} \exp_g : \mathbb{Z}/n\mathbb{Z} &\rightarrow G, \\ x &\mapsto g^x. \end{aligned}$$

Discrete logarithm

\exp_g is an isomorphism. Its inverse is called the **discrete logarithm**.

$$\begin{aligned} \log_g : G &\rightarrow \mathbb{Z}/n\mathbb{Z}, \\ g^x &\mapsto x. \end{aligned}$$

The Discrete Logarithm Problem



Exponentiation in a cyclic group

Let $G = \langle g \rangle$ be a cyclic group of order n . Define

$$\begin{aligned} \exp_g : \mathbb{Z}/n\mathbb{Z} &\rightarrow G, \\ x &\mapsto g^x. \end{aligned}$$

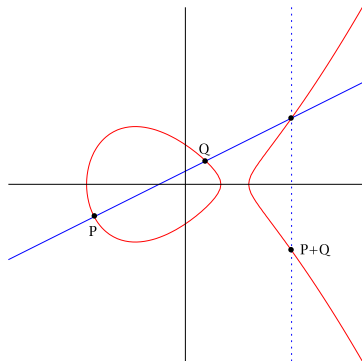
Discrete logarithm

\exp_g is an isomorphism. Its inverse is called the **discrete logarithm**.

$$\begin{aligned} \log_g : G &\rightarrow \mathbb{Z}/n\mathbb{Z}, \\ g^x &\mapsto x. \end{aligned}$$

Computing it is called the **Discrete Logarithm Problem** (DLP) of G .

Elliptic curve cryptography



Elliptic curve (Weierstrass form)

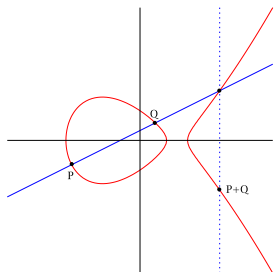
$$E : y^2 = x^3 + ax + b;$$

The set of points of an elliptic curve is endowed with a group law via the chord-and-tangent law.

Hasse bound: $\#E(\mathbb{F}_q) \sim q$.

	Finite field crypto	Elliptic curve crypto
Group	\mathbb{F}_q^*	$E(\mathbb{F}_q)$
Protocols	El Gamal, DSA, ...	ECDH, ECDSA, ECMQV, ...
Key sizes	1024, 2048, 3072	160, 225, 256

Group law and scalar multiplication



$$y^2 = x^3 + ax + b$$

$$P = (x_0, y_0), Q = (x_1, y_1)$$

$$\lambda = \frac{y_1 - y_0}{x_1 - x_0}$$

$$P + Q = (\lambda^2 - x_0 - x_1, (x_0 - x_2)\lambda - y_0)$$

Multiplication: $[m]P = \overbrace{P + P + \dots + P}^{m \text{ times}}$

m -torsion: $E[m] = \{P \in E(\bar{\mathbb{K}}) \mid [m]P = \mathcal{O}\} \cong (\mathbb{Z}/m\mathbb{Z})^2$

$$[m](x, y) = \left(\frac{\psi_m(x, y)}{\phi_m^2(x, y)}, \frac{\omega_m(x, y)}{\phi_m^3(x, y)} \right)$$

Division polynomials: ϕ_m can be computed with $O(\log m)$ polynomial multiplications, $\deg \phi_m = O(m^2)$.

Isogenies between elliptic curves

$$\begin{array}{ccc}
 E & \xrightarrow{\mathcal{I}} & E' \\
 [m] \downarrow & \swarrow \hat{\mathcal{I}} & \\
 E & &
 \end{array}$$

(Separable) isogeny: (separable) non-constant rational morphism preserving the identity.

Properties

- Isogeny = rational map + group morphism;
- Finite kernel, surjective (in $\bar{\mathbb{K}}$);
- **Dual isogeny theorem:** they factor the multiplication map into two pieces.

Multiplication

$$\begin{aligned}
 [m] : E(\bar{\mathbb{K}}) &\rightarrow E(\bar{\mathbb{K}}) \\
 P &\mapsto [m]P
 \end{aligned}$$

$$\ker \mathcal{I} = E[m].$$

Isogenies between elliptic curves

$$\begin{array}{ccc}
 E & \xrightarrow{\mathcal{I}} & E' \\
 [m] \downarrow & \swarrow \hat{\mathcal{I}} & \\
 E & &
 \end{array}$$

(Separable) isogeny: (separable) non-constant rational morphism preserving the identity.

Properties

- Isogeny = rational map + group morphism;
- Finite kernel, surjective (in $\bar{\mathbb{K}}$);
- **Dual isogeny theorem:** they factor the multiplication map into two pieces.

Frobenius endomorphism

$$\begin{aligned}
 \varphi : E(\bar{\mathbb{K}}) &\rightarrow E(\bar{\mathbb{K}}) \\
 (x, y) &\mapsto (x^q, y^q)
 \end{aligned}$$

$\ker \varphi = \{\mathcal{O}\}$ (inseparable).



Isogenies between elliptic curves

$$\begin{array}{ccc}
 E & \xrightarrow{\mathcal{I}} & E' \\
 [m] \downarrow & \swarrow \hat{\mathcal{I}} & \\
 E & &
 \end{array}$$

(Separable) isogeny: (separable) non-constant rational morphism preserving the identity.

Properties

- Isogeny = rational map + group morphism;
- Finite kernel, surjective (in $\bar{\mathbb{K}}$);
- **Dual isogeny theorem:** they factor the multiplication map into two pieces.

Separable isogeny (simplified Weierstrass model)

$$\mathcal{I}(x, y) = \left(\frac{g(x)}{h(x)}, cy \left(\frac{g(x)}{h(x)} \right)' \right)$$

h vanishes on the abscissas of $\ker \mathcal{I}$. $\deg \mathcal{I} = \# \ker \mathcal{I}$.

Why compute (large) isogenies over finite fields?

SEA algorithm (Schoof 1985; Elkies 1992; Atkin 1988)

Hasse bound $\#E(\mathbb{F}_q) = q - t + 1;$

Schoof Compute t modulo small primes $\ell \Leftrightarrow$ compute the action of φ_q on $E[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^2;$

Atkin Determine the order of the roots of $X^2 - tX + q$ by factoring the ℓ -th modular polynomial.

Elkies Compute an ℓ -isogeny \mathcal{I} and the action of φ_q on $\ker \mathcal{I} \cong \mathbb{Z}/\ell\mathbb{Z} \subset E[\ell];$

Other cryptographic applications

- Transfer DLPs between curves (Gaudry, Hess, and Smart 2002; Smith 2009);
- Construct new cryptosystems (Teske 2006; Rostovtsev and Stolbunov 2006);
- Construct hash functions (Charles, Lauter, and Goren 2009);
- Compute modular polynomials (Broker, Lauter, and Sutherland 2010);
- Compute the endomorphism ring (Kohel 1996; Sutherland 2010).



Outline

- 1 Introduction
- 2 Computing isogenies over finite fields**
- 3 Artin-Schreier towers
- 4 Implementations
- 5 Conclusion



Vélu's formulas

Compute an isogeny with given kernel (Vélu 1971)

Given the kernel H , computes $\mathcal{I} : E \rightarrow E/H$ given by

$$\mathcal{I}(\mathcal{O}_E) = \mathcal{I}(\mathcal{O}_{E/H}),$$

$$\mathcal{I}(P) = \left(x(P) + \sum_{Q \in H^*} x(P+Q) - x(Q), y(P) + \sum_{Q \in H^*} y(P+Q) - y(Q) \right).$$

In practice, given $h(x)$, of degree $\ell - 1$, vanishing on H

$$y^2 = f(x), \quad p_1 = \sum_{Q \in H^*} x(Q), \quad \frac{g(x)}{h(x)} = \ell x - p_1 - f'(x) \frac{h'(x)}{h(x)} - 2f(x) \left(\frac{h'(x)}{h(x)} \right)'$$

$$\mathcal{I}(x, y) = \left(\frac{g(x)}{h(x)}, y \left(\frac{g(x)}{h(x)} \right)' \right)$$

Isogeny computation

By Vélu's formulas: $\mathcal{I}(x, y) = \left(\frac{g(x)}{h(x)}, y \left(\frac{g(x)}{h(x)} \right)' \right)$, hence

$$(x^3 + ax + b) \left(\frac{g(x)}{h(x)} \right)'^2 = \left(\frac{g(x)}{h(x)} \right)^3 + a' \frac{g(x)}{h(x)} + b'$$

Algorithms: Given E, E', ℓ , compute $\mathcal{I} : E \rightarrow E'$

- **Stark 1973:** First algorithm in characteristic 0, using continued fractions.
- **Elkies 1992, 1998:** Find a power series solution to the differential equation.
- **Couveignes 1994:** (p small) Compute morphisms of formal groups, look for one that corresponds to an isogeny.
- **Lercier 1996:** (only for $p = 2$) solve a linear system.
- **Couveignes 1996:** (p small) Interpolate over the p -torsion, look for a polynomial that corresponds to an isogeny.
- **BMSS algorithm (Bostan, Morain, Salvy, and Schost 2008):** Improve Elkies 1998 to run in quasi-linear time.
- **Lercier and Sirvent 2008:** Lift E and E' in the p -adics, then apply BMSS.

Couveignes' algorithm (Couveignes 1996)

Given E, E', ℓ , compute $\mathcal{I} : E \rightarrow E'$

Idea: Map $E[p^k]$ onto $E'[p^k]$

- Compute the extensions $\mathbb{U}_i/\mathbb{F}_q$ such that $E[p^i]$ is defined over \mathbb{U}_i ; $\tilde{O}(\ell^2)$
 - Pick k **large enough** ($k \sim \log_p 4\ell$);
 - Compute P , a generator of $E[p^k]$; $\tilde{O}(\ell^2)$
 - Compute P' , a generator of $E'[p^k]$; $\tilde{O}(\ell^2)$
 - Compute the polynomial T vanishing on $E[p^k]$; $\tilde{O}(\ell^2)$
 - For $m \in (\mathbb{Z}/p^k\mathbb{Z})^*$
 - Interpolate $A : x(P) \mapsto x([m]P')$; $\tilde{O}(\ell^2)$
 - Reconstruct a rational fraction $\frac{g}{h} \equiv A \pmod{T}$; $\tilde{O}(M(\ell))$
- Stop when $\frac{g}{h}$ is an isogeny. ℓ times on average

Couveignes' algorithm (Couveignes 1996)

Given E, E', ℓ , compute $\mathcal{I} : E \rightarrow E'$

Idea: Map $E[p^k]$ onto $E'[p^k]$

- Compute the extensions $\mathbb{U}_i/\mathbb{F}_q$ such that $E[p^i]$ is defined over \mathbb{U}_i ; $\tilde{O}(M(\ell))$
 - Pick k **large enough** ($k \sim \log_p 4\ell$);
 - Compute P , a generator of $E[p^k]$; $\tilde{O}(M(\ell))$
 - Compute P' , a generator of $E'[p^k]$; $\tilde{O}(M(\ell))$
 - Compute the polynomial T vanishing on $E[p^k]$; $\tilde{O}(M(\ell))$
 - For $m \in (\mathbb{Z}/p^k\mathbb{Z})^*$
 - Interpolate $A : x(P) \mapsto x([m]P')$; $\tilde{O}(M(\ell))$
 - Reconstruct a rational fraction $\frac{g}{h} \equiv A \pmod{T}$; $\tilde{O}(M(\ell))$
- Stop when $\frac{g}{h}$ is an isogeny. ℓ times on average

Couveignes' algorithm (Couveignes 1996)

Given E, E', ℓ , compute $\mathcal{I} : E \rightarrow E'$

Idea: Map $E[p^k]$ onto $E'[p^k]$

- Compute the extensions $\mathbb{U}_i/\mathbb{F}_q$ such that $E[p^i]$ is defined over \mathbb{U}_i ;
 - Pick k **large enough** ($k \sim \log_p 4\ell$);
 - Compute P , a generator of $E[p^k]$;
 - Compute P' , a generator of $E'[p^k]$;
 - Compute the polynomial T vanishing on $E[p^k]$;
 - For $m \in (\mathbb{Z}/p^k\mathbb{Z})^*$
 - Interpolate $A : x(P) \mapsto x([m]P')$;
 - Reconstruct a rational fraction $\frac{g}{h} \equiv A \pmod{T}$;
- Stop when $\frac{g}{h}$ is an isogeny.



How to recognize an isogeny?

- **Degree:** $\frac{g}{h}$ with $\deg g = \ell$, $\deg h = \ell - 1$; $O(1)$
- **Square factor:** $h = \prod_{Q \in H^*} (X - x(Q)) = f^2$ if ℓ odd; $\tilde{O}(M(\ell))$
- **Group action:** Test on random points: $\mathcal{I}(P + Q) = \mathcal{I}(P) + \mathcal{I}(Q)$; $O(\ell)$
- **Factor of the ℓ -division polynomial:** Check $\phi_\ell = 0 \pmod{h}$. $\tilde{O}(M(\ell))$



How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$
$$\ell = 11$$



How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$

$$\ell = 11$$

deg R_i	deg U_i
3141592653589793238462643	0



How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$

$$\ell = 11$$

deg R_i	deg U_i
3141592653589793238462643	0
3141592653589793238462642	1



How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$

$$\ell = 11$$

deg R_i	deg U_i
3141592653589793238462643	0
3141592653589793238462642	1
3141592653589793238462641	2



How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$

$$\ell = 11$$

deg R_i	deg U_i
3141592653589793238462643	0
3141592653589793238462642	1
3141592653589793238462641	2
⋮	⋮
3141592653589793238462634	9

How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$

$$\ell = 11$$

deg R_i	deg U_i
3141592653589793238462643	0
3141592653589793238462642	1
3141592653589793238462641	2
⋮	⋮
3141592653589793238462634	9
11	10

How to recognize an isogeny?

T vanishes on $E[p^k]$, A interpolates $x(P) \mapsto x(P')$

$$AU_i + TV_i = R_i \quad \Leftrightarrow \quad A \equiv \frac{R_i}{U_i} \pmod{T}$$

$$\ell = 11$$

deg R_i	deg U_i
3141592653589793238462643	0
3141592653589793238462642	1
3141592653589793238462641	2
⋮	⋮
3141592653589793238462634	9
11	10
10	3141592653589793238462633
⋮	⋮



Isogenies of unknown degree

It costs *only* $\tilde{O}(\ell)$

- This pattern is extremely rare.
- It can be detected in quasi-linear time using a fast XGCD algorithm (Khodadad and Monagan 2006).
- This is the only phase of Couveignes' algorithm that depends on ℓ .





Isogenies of unknown degree

It costs *only* $\tilde{O}(\ell)$

- This pattern is extremely rare.
- It can be detected in quasi-linear time using a fast XGCD algorithm (Khodadad and Monagan 2006).
- This is the only phase of Couveignes' algorithm that depends on ℓ .

Actually, it does not even depend on ℓ

- This just depends on the bound p^k , not on the exact degree ℓ .
- If ℓ is not known in advance, it is enough to look for a **gap**.
- Thus, any isogeny of degree $\ll p^k$ can be computed with one single run of Couveignes' algorithm.

Comparison of isogeny algorithms

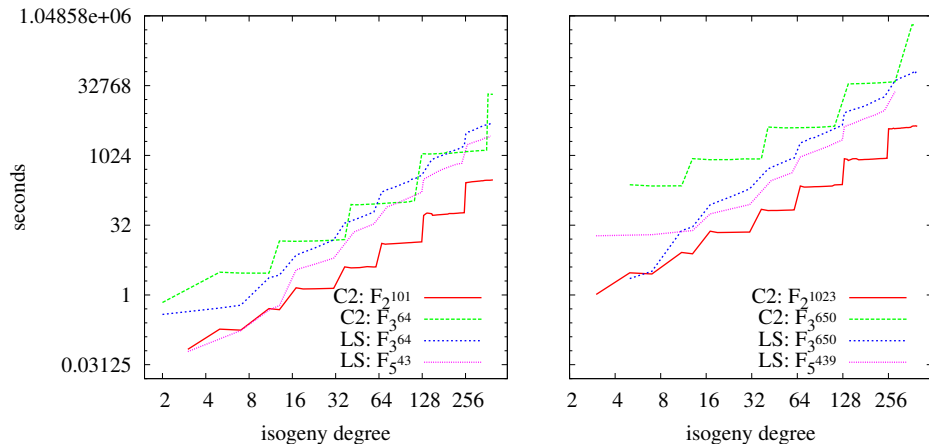


Figure: Comparative timings for Couveignes 1996 (C2) and Lercier and Sirvent 2008 (LS) over various curves. Plot in logarithmic scale.



Isogenies of unknown degree

- This variant of Couveignes 1996 is currently the fastest algorithm for this task (both in theory and in practice).
- We tested two curves over $\mathbb{F}_{2^{161}}$, isogenous of unknown degree, taken from Teske 2006: proven in 694 cpu-hours that no isogeny of degree less than 2^{13} exists.

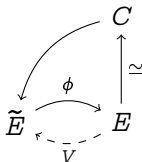
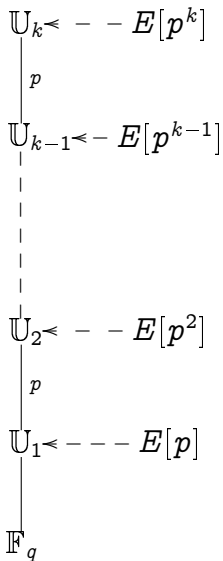




Outline

- 1 Introduction
- 2 Computing isogenies over finite fields
- 3 Artin-Schreier towers**
- 4 Implementations
- 5 Conclusion

The field of definition of $E[p^k]$



p -descent (Voloch 1990)

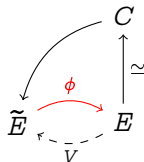
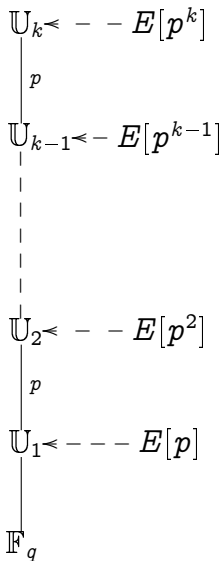
If $P_i = (x_i, y_i)$ generates $E[p^i]$, the solution to

$$\begin{cases}
 Z^p - Z & = \frac{\sqrt[p]{y_i \beta_E(x_i)}}{\sqrt[p-1]{H_E}} \\
 X^p & = x_i \\
 Y^p & = y_i
 \end{cases}$$

generates $C[p^{i+1}]$.

Then apply the isomorphism $C \cong E$.

The field of definition of $E[p^k]$



p -descent (Voloch 1990)

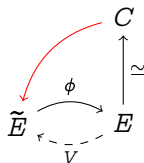
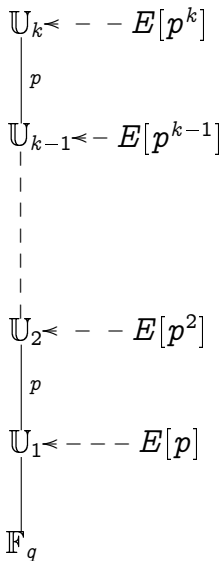
If $P_i = (x_i, y_i)$ generates $E[p^i]$, the solution to

$$\begin{cases}
 Z^p - Z & = \frac{\sqrt[p]{y_i \beta_E(x_i)}}{\sqrt[p-1]{H_E}} \\
 X^p & = x_i \\
 Y^p & = y_i
 \end{cases}$$

generates $C[p^{i+1}]$.

Then apply the isomorphism $C \cong E$.

The field of definition of $E[p^k]$



p -descent (Voloch 1990)

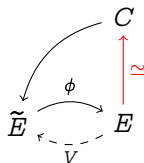
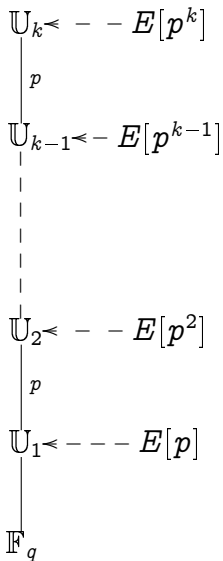
If $P_i = (x_i, y_i)$ generates $E[p^i]$, the solution to

$$\begin{cases}
 Z^p - Z & = \frac{\sqrt[p]{y_i \beta_E(x_i)}}{\sqrt[p-1]{H_E}} \\
 X^p & = x_i \\
 Y^p & = y_i
 \end{cases}$$

generates $C[p^{i+1}]$.

Then apply the isomorphism $C \cong E$.

The field of definition of $E[p^k]$



p -descent (Voloch 1990)

If $P_i = (x_i, y_i)$ generates $E[p^i]$, the solution to

$$\begin{cases}
 Z^p - Z & = \frac{\sqrt[p]{y_i \beta_E(x_i)}}{\sqrt[p-1]{H_E}} \\
 X^p & = x_i \\
 Y^p & = y_i
 \end{cases}$$

generates $C[p^{i+1}]$.

Then apply the isomorphism $C \cong E$.

Artin-Schreier towers

$$\begin{array}{c}
 \mathbb{U}_k = \frac{\mathbb{U}_{k-1}[X_k]}{X_k^p - X_k - \alpha_{k-1}} \\
 \left| \begin{array}{c} p \\ \hline \end{array} \right. \\
 \mathbb{U}_{k-1} \\
 \vdots \\
 \mathbb{U}_1 = \frac{\mathbb{U}_0[X_1]}{X_1^p - X_1 - \alpha_0} \\
 \left| \begin{array}{c} p \\ \hline \end{array} \right. \\
 \mathbb{U}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)}
 \end{array}$$

Artin-Schreier polynomials

Artin-Schreier polynomial:

$$X^p - X - \alpha \quad \text{with } \alpha \in \mathbb{K}$$

Proposition

$X^p - X - \alpha$ is either irreducible or split in \mathbb{K} .

Artin-Schreier extensions

Defined by an irreducible Artin-Schreier polynomial

$$\mathbb{L} = \mathbb{K}[X]/(X^p - X - \alpha)$$

ANY separable extension of degree p can be expressed this way.

Arithmetic in towers of extensions

$$\begin{array}{c}
 \mathbb{U}_k = \frac{\mathbb{U}_{k-1}[X_k]}{X_k^p - X_k - \alpha_{k-1}} \\
 \left| \begin{array}{c} p \\ \vdots \\ p \end{array} \right. \\
 \mathbb{U}_{k-1} \\
 \vdots \\
 \mathbb{U}_1 = \frac{\mathbb{U}_0[X_1]}{X_1^p - X_1 - \alpha_0} \\
 \left| \begin{array}{c} p \end{array} \right. \\
 \mathbb{U}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)}
 \end{array}$$

Multiplication

$$M(\mathbb{U}_0) = O(M(d))$$

$$M(\mathbb{U}_1) = O(p^2 M(d))$$

$$M(\mathbb{U}_2) = O(p^4 M(d))$$

$$\vdots$$

$$M(\mathbb{U}_k) = O(p^{2k} M(d)) \not\subseteq \tilde{O}(p^k d)$$

Even using Karatsuba or FFT multiplication, $\tilde{O}(p^k d)$ can't be attained.

Other operations

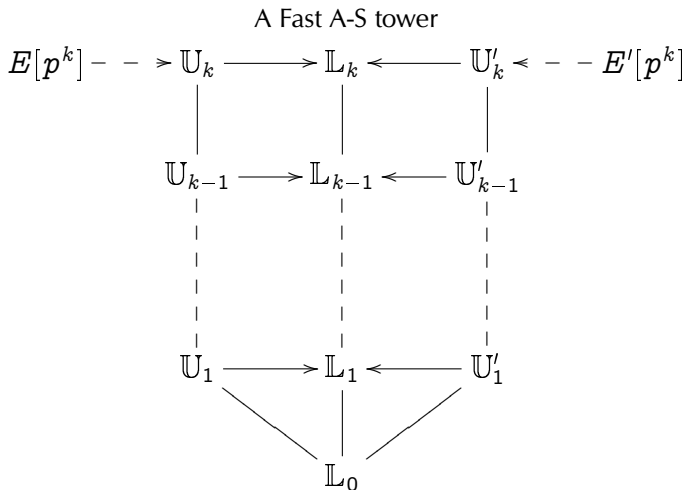
- Exponentiation, Inversion, GCD, all depend upon $M(\mathbb{U}_k)$,
- The canonical injection $\mathbb{U}_{i-1} \rightarrow \mathbb{U}_i$ can be computed in $O(p^i)$.

One fast tower, many fast towers!

A Fast A-S tower



One fast tower, many fast towers!



Theorem (Couveignes 2000): There exist an isomorphism algorithm that runs in $O(k^3 M(\mathbb{L}_k))$ operations in \mathbb{L}_0 . (Remember that $k = \log_p \#\mathbb{L}_k$)

A fast tower

$$\begin{array}{c}
 \mathbb{L}_k = \frac{\mathbb{L}_{k-1}[X_k]}{X_k^p - X_k - \alpha_{k-1}} \\
 \left| \begin{array}{c} p \\ \mathbb{L}_{k-1} \\ \vdots \\ \mathbb{L}_1 = \frac{\mathbb{L}_0[X_1]}{X_1^p - X_1 - \alpha_0} \\ \left| \begin{array}{c} p \\ \mathbb{L}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)} \\ \left| \begin{array}{c} d \\ \mathbb{F}_p \end{array} \end{array} \right. \end{array} \right.
 \end{array}$$

Idea: Convert to a univariate basis over \mathbb{F}_p

A fast tower

$$\begin{array}{c}
 \mathbb{L}_k = \frac{\mathbb{L}_{k-1}[X_k]}{X_k^p - X_k - \alpha_{k-1}} \\
 \left| \begin{array}{c} p \\ \mathbb{L}_{k-1} \\ \vdots \\ \mathbb{L}_1 \\ \mathbb{L}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)} \\ \left| \begin{array}{c} d \\ \mathbb{F}_p \end{array} \right. \end{array} \right.
 \end{array}$$

Idea: Convert to a univariate basis over \mathbb{F}_p



$$\begin{array}{c}
 \mathbb{L}_k = \frac{\mathbb{F}_p[Y_k]}{Q_k(Y_k)} \\
 \left| \begin{array}{c} p \\ \mathbb{L}_{k-1} \\ \vdots \\ \mathbb{L}_1 \\ \mathbb{L}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)} \\ \left| \begin{array}{c} d \\ \mathbb{F}_p \end{array} \right. \end{array} \right.
 \end{array}$$

A fast tower: the multivariate side

$$\mathbb{L}_k = \frac{\mathbb{L}_{k-1}[X_k]}{X_k^p - X_k - x_{k-1}^{2p-1}}$$

 $|$
 p

\mathbb{L}_{k-1}

 \vdots

$$\mathbb{L}_1 = \frac{\mathbb{L}_0[X_1]}{X_1^p - X_1 - x_0}$$

 $|$
 p

$$\mathbb{L}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)}$$

 $|$
 d

\mathbb{F}_p

Our construction (inspired by Cantor 1989)

$$\left\{ \begin{array}{l} X_i^p - X_i - X_{i-1}^{2p-1} \\ \vdots \\ X_2^p - X_2 - X_1^{2p-1} \\ X_1^p - X_1 - X_0 \\ Q(X_0) \end{array} \right.$$

Theorem: Let x_i be the class of X_i in \mathbb{L}_i (and thus in \mathbb{L}_{i+1}, \dots). If $\text{Tr}_{\mathbb{L}_0/\mathbb{F}_p}(x_0) \neq 0$, each line contains an irreducible polynomial and x_i generates \mathbb{L}_i over \mathbb{F}_p .

A fast tower: the univariate side

$$\begin{array}{c}
 \mathbb{L}_k = \frac{\mathbb{F}_p[Y_k]}{Q_k(Y_k)} \\
 \left| \begin{array}{c} p \\ \vdots \\ p \end{array} \right. \\
 \mathbb{L}_{k-1} \\
 \left| \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \right. \\
 \mathbb{L}_1 = \frac{\mathbb{F}_p[Y_1]}{Q_1(Y_1)} \\
 \left| \begin{array}{c} p \\ \vdots \\ p \end{array} \right. \\
 \mathbb{L}_0 = \mathbb{F}_q = \frac{\mathbb{F}_p[X_0]}{Q(X_0)} \\
 \left| \begin{array}{c} d \\ \vdots \\ d \end{array} \right. \\
 \mathbb{F}_p
 \end{array}$$

Univariate representation

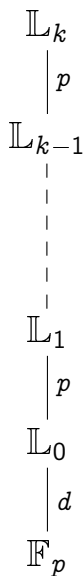
The minimal polynomials Q_i of x_i over \mathbb{F}_p can be computed as follows.

- $Q_0 = Q$;
- $Q_1 = Q_0(Z^p - Z)$.

Let ω be a $(2p - 1)$ -th root of unity,

- $q_{i+1}(Z^{2p-1}) = \prod_{j=0}^{2p-2} Q_i(\omega^j Z)$,
- $Q_{i+1} = q_{i+1}(Z^p - Z)$.

A fast tower: change of basis



Two bases to represent \mathbb{L}_k

- Univariate:

$$\mathbb{L}_k = \mathbb{F}_p[Y_k] / Q_k(Y_k)$$

- Multivariate:

$$\mathbb{L}_k = \mathbb{F}_p[X_0, X_1, \dots, X_k] / (Q, \dots)$$

A fast change of basis is the key to fast arithmetic:

- Multiplication is faster in **univariate**;
- Field embeddings are faster in **multivariate**.

Change of basis

Univariate \rightarrow multivariate

- Recursively split input in p slices, recombine by Horner's rule;
- Simple and fast: $O(M(\ell) + \ell \log^2 \ell)$

Multivariate \rightarrow univariate

- Trace formulas + duality + transposed algorithms;
- Same complexity: $O(M(\ell) + \ell \log^2 \ell)$

Example: univariate \rightarrow multivariate

$$\mathbb{F}_{3^9} = \mathbb{F}_3[X_1, X_2]/I = \mathbb{F}_3[Y_2]/Q_2(Y_2)$$

$$I = \begin{cases} X_2^3 - X_2 - X_1^5, \\ X_1^3 - X_1 - 1, \end{cases}$$

$$Q_2(Z) = Z^9 + Z^6 + Z^4 + Z^2 - Z - 1, \quad Q_2(x_2) = 0,$$

Let a be given in the univariate basis:

$$a = \underbrace{X_2^8 + X_2^7 + X_2^6}_{\text{group 1}} + \underbrace{X_2^5}_{\text{group 2}} + \underbrace{X_2}_{\text{group 3}} =$$

Example: univariate \rightarrow multivariate

$$\mathbb{F}_{3^9} = \mathbb{F}_3[X_1, X_2]/I = \mathbb{F}_3[Y_2]/Q_2(Y_2)$$

$$I = \begin{cases} X_2^3 - X_2 - X_1^5, \\ X_1^3 - X_1 - 1, \end{cases}$$

$$Q_2(Z) = Z^9 + Z^6 + Z^4 + Z^2 - Z - 1, \quad Q_2(x_2) = 0,$$

Let a be given in the univariate basis:

$$a = \underbrace{X_2^8 + X_2^7 + X_2^6}_{a_2} + \underbrace{X_2^5}_{a_1} + \underbrace{X_2}_{a_0} = X_2^6 \underbrace{(X_2^2 + X_2 + 1)}_{a_2} + X_2^3 \underbrace{(X_2^2)}_{a_1} + \underbrace{(X_2)}_{a_0}$$

Example: univariate \rightarrow multivariate

$$\mathbb{F}_{3^9} = \mathbb{F}_3[X_1, X_2]/I = \mathbb{F}_3[Y_2]/Q_2(Y_2)$$

$$I = \begin{cases} X_2^3 - X_2 - X_1^5, \\ X_1^3 - X_1 - 1, \end{cases}$$

$$Q_2(Z) = Z^9 + Z^6 + Z^4 + Z^2 - Z - 1, \quad Q_2(x_2) = 0,$$

Let a be given in the univariate basis:

$$a = \underbrace{X_2^8 + X_2^7 + X_2^6}_{a_2} + \underbrace{X_2^5}_{a_1} + \underbrace{X_2}_{a_0} = X_2^6 \underbrace{(X_2^2 + X_2 + 1)}_{a_2} + X_2^3 \underbrace{(X_2^2)}_{a_1} + \underbrace{(X_2)}_{a_0}$$

$$\text{Then, } a = (X_2 + X_1^5) ((X_2 + X_1^5)a_2 + a_1) + a_0$$

Example: univariate \rightarrow multivariate

$$\mathbb{F}_{3^9} = \mathbb{F}_3[X_1, X_2]/I = \mathbb{F}_3[Y_2]/Q_2(Y_2)$$

$$I = \begin{cases} X_2^3 - X_2 - X_1^5, \\ X_1^3 - X_1 - 1, \end{cases}$$

$$Q_2(Z) = Z^9 + Z^6 + Z^4 + Z^2 - Z - 1, \quad Q_2(x_2) = 0,$$

Let a be given in the univariate basis:

$$a = \underbrace{X_2^8 + X_2^7 + X_2^6}_{a_2} + \underbrace{X_2^5}_{a_1} + \underbrace{X_2}_{a_0} = X_2^6 \underbrace{(X_2^2 + X_2 + 1)}_{a_2} + X_2^3 \underbrace{(X_2^2)}_{a_1} + \underbrace{(X_2)}_{a_0}$$

$$\text{Then, } a = (X_2 + X_1^5) \left((X_2 + X_1^5) a_2 + a_1 \right) + a_0 \pmod{(X_2^3 - X_2 - X_1^5)} =$$

$$(X_1^2 - X_1 - 1)X_2^2 + (X_1 - 1)X_2 - X_1^2 - X_1 - 1$$

Multivariate \rightarrow univariate

Trace formulas (Rouillier 1999)

$$a(Z) = \frac{A(Z)}{Q'_k(Z)} \bmod Q_k(Z) \quad \text{where} \quad A(T) = Q_k(T) \sum_{i \geq 0} \frac{\text{Tr } ax_k^i}{T^{i+1}}.$$

- Compute the *trace form* Tr ,
- compute the form $a \cdot \text{Tr} : x \mapsto \text{Tr } ax$,
- compute $\sum_{i \geq 0} \frac{a \cdot \text{Tr } x_k^i}{T^{i+1}}$,
- deduce $a(Z)$.

Example: Let $a = x_1$, then

$$A(T) = Q_k(T) (T^{-6} - T^{-8} - T^{-9} + O(T^{-10})) = T^3 - T,$$

$$a = \frac{x_2^3 - x_2}{x_2^3 - x_2 - 1} = x_2^6 + x_2^4 - x_2^3 + x_2^2 + x_2 + 1.$$

Multivariate \rightarrow univariate

Trace formulas (Rouillier 1999)

$$a(Z) = \frac{A(Z)}{Q'_k(Z)} \bmod Q_k(Z) \quad \text{where} \quad A(T) = Q_k(T) \sum_{i \geq 0} \frac{\text{Tr } ax_k^i}{T^{i+1}}.$$

- Compute the *trace form* Tr ,
- compute the form $a \cdot \text{Tr} : x \mapsto \text{Tr } ax$,
- **compute** $\sum_{i \geq 0} \frac{a \cdot \text{Tr } x_k^i}{T^{i+1}}$,
- deduce $a(Z)$.

Example: Let $a = x_1$, then

$$A(T) = Q_k(T) (T^{-6} - T^{-8} - T^{-9} + O(T^{-10})) = T^3 - T,$$

$$a = \frac{x_2^3 - x_2}{x_2^3 - x_2 - 1} = x_2^6 + x_2^4 - x_2^3 + x_2^2 + x_2 + 1.$$

Duality (Shoup 1995, 1999; Bostan, Salvy, and Schost 2003)

Polynomial

evaluation: $k[Z] \rightarrow \mathbb{K}/k$

$$g \mapsto g(\sigma)$$

Power projection: $(\mathbb{K}/k)^* \rightarrow k[T]^* \simeq k[[1/T]]$

$$\ell \mapsto \sum_{i>0} \frac{\ell(\sigma^i)}{T^i}$$

Univariate \rightarrow multivariate = Polynomial evaluation

If $Z \mapsto x_k$ (with $x_k \in \mathbb{U}_k$ written in the multivariate basis):

$$a(Z) = a_d Z^d + \cdots + a_1 Z + a_0 \quad \mapsto \quad a_d x_k^d + \cdots + a_1 x_k + a_0 = a(x_k) = a.$$

Transposed univariate \rightarrow multivariate = Power projection

$$\begin{aligned} \text{Tr} &\mapsto \sum_{i \geq 0} \frac{\text{Tr } x_k^i}{T^i}, \\ a \cdot \text{Tr} &\mapsto \sum_{i \geq 0} \frac{\text{Tr } a x_k^i}{T^i}. \end{aligned}$$

The transposition principle

Theorem

Let \mathcal{P} be an arbitrary set. To any R -algebraic algorithm A computing a family of linear functions $(f_p : M \rightarrow N)_{p \in \mathcal{P}}$ corresponds an R -algebraic algorithm A^* computing the **dual family** $(f_p^* : N^* \rightarrow M^*)_{p \in \mathcal{P}}$. The algebraic time and space complexities of A^* are bounded by the time complexity of A .

Change of basis

Univariate \rightarrow multivariate

- Recursively split input in p slices, recombine by Horner's rule;
- Simple and fast: $O(M(\ell) + \ell \log^2 \ell)$

Multivariate \rightarrow univariate

$$O(M(\ell) + \ell \log^2 \ell)$$

By *trace formulas* (Rouillier 1999):

$$a(Z) = \frac{A(Z)}{Q'_k(Z)} \bmod Q_k(Z) \quad \text{where} \quad A(T) = Q_k(T) \sum_{i \geq 0} \frac{\text{Tr } ax_k^i}{T^{i+1}}.$$

- Compute the *trace form* Tr and the form $a \cdot \text{Tr} : x \mapsto \text{Tr } ax$;
- compute $\sum_{i \geq 0} \frac{a \cdot \text{Tr } x_k^i}{T^{i+1}}$ using a **transposed univariate \rightarrow multivariate**;
- deduce $a(Z)$.



Outline

- 1 Introduction
- 2 Computing isogenies over finite fields
- 3 Artin-Schreier towers
- 4 Implementations**
- 5 Conclusion



Implementations

Automatic transposition

- Algorithms are hard to transpose, transposed algorithms are hard or impossible to understand;
- How can we be confident that a transposed algorithm is well implemented if no one understands it?
- When proving programs with a proof assistant, why should we do the work twice?

<http://transalpyne.gforge.inria.fr/>

A Python-like ad-hoc language for automatic transposition, compiled/interpreted in Python.

Artin-Schreier towers and isogenies

- **FAAST** (Fast Arithmetic in Artin-Schreier towers): C++ with NTL implementation released under GPL:

<http://www.lix.polytechnique.fr/~defeo/FAAST/>

- With F. Morain and É. Schost, writing a C++ library for isogenies over finite fields.





Outline

- 1 Introduction
- 2 Computing isogenies over finite fields
- 3 Artin-Schreier towers
- 4 Implementations
- 5 Conclusion**



Perspectives

The future of transalpyne

- Finish the implementation, add more output languages.
- Integration in a Computer Algebra System (Sage? Mathemagix?).
- Development in a dependently typed system (Coq? Agda?).

Perspectives on isogenies

- All known algorithms to compute isogenies have suboptimal complexity $\Omega(\ell^2)$.
- Are there models, where algorithms *à la* BMSS and Lercier and Sirvent 2008 may be faster?
- Couveignes' first isogeny algorithm (Couveignes 1994) is very similar to Couveignes 1996 and may be modified in the same way. There is some hope that it may be faster in practice, because of the simpler arithmetic operations needed.
- Generalize Couveignes' algorithms to genus 2?

Thank you all for coming. . .



References I



Kohel, David (1996).

“Endomorphism rings of elliptic curves over finite fields.”

PhD thesis. University of California at Berkley.



Couveignes, Jean-Marc (1994).

“Quelques calculs en théorie des nombres.”

PhD thesis. Université de Bordeaux.



Schoof, René (1985).

“Elliptic Curves Over Finite Fields and the Computation of Square Roots mod p .”

In: Mathematics of Computation 44.170 ,

Pp. 483–494.

URL: <http://dx.doi.org/10.2307/2007968>.



Elkies, Noam D. (1992).

“Explicit isogenies.”

manuscript, Boston MA.

References II



Atkin, A. O. L. (1988).

“The number of points on an elliptic curve modulo a prime.”
manuscript, Chicago IL.



Gaudry, Pierrick, Florian Hess, and Nigel Smart (2002).

“Constructive and destructive facets of Weil descent on elliptic curves.”
In: *Journal of Cryptology* 15.1 ,
Pp. 19–46–46.

URL: <http://dx.doi.org/10.1007/s00145-001-0011-x>.



Smith, Benjamin (2009).

“Isogenies and the Discrete Logarithm Problem in Jacobians of Genus 3
Hyperelliptic Curves.”

In: *Journal of Cryptology* 22.4 ,
Pp. 505–529–529.

URL: <http://dx.doi.org/10.1007/s00145-009-9038-1>.

References III



Teske, Edlyn (2006).
“An Elliptic Curve Trapdoor System.”

In: *Journal of Cryptology* 19.1 ,
Pp. 115–133.

URL: <http://dx.doi.org/10.1007/s00145-004-0328-3>.



Rostovtsev, Alexander and Anton Stolbunov (2006).
Public-key Cryptosystem Based On Isogenies .

URL: <http://eprint.iacr.org/2006/145>.



Charles, Denis, Kristin Lauter, and Eyal Goren (2009).
“Cryptographic Hash Functions from Expander Graphs.”

In: *Journal of Cryptology* 22.1 ,
Pp. 93–113.

URL: <http://dx.doi.org/10.1007/s00145-007-9002-x>.



Broker, Reinier, Kristin Lauter, and Andrew V. Sutherland (2010).
Modular polynomials via isogeny volcanoes .

URL: <http://arxiv.org/abs/1001.0402>.



References IV



Sutherland, Andrew V. (2010).

“Computing Hilbert class polynomials with the Chinese remainder theorem.”

In: *Mathematics of Computation* .



Vélu, Jean (1971).

“Isogénies entre courbes elliptiques.”

In: *Comptes Rendus de l'Académie des Sciences de Paris* 273 ,
Pp. 238–241.



References V



Stark, Harold M. (1973).

“Class-Numbers of Complex Quadratic Fields.”

In: *Modular Functions of One Variable I*.

Ed. by Willem Kuyk.

Vol. 320.

Lecture Notes in Mathematics.

Berlin, Heidelberg: Springer Berlin / Heidelberg.

Chap. 5, pp. 153–174.

URL: http://dx.doi.org/10.1007/978-3-540-38509-7_5.

References VI



Elkies, Noam D. (1998).

“Elliptic and modular curves over finite fields and related computational issues.”

In: Computational perspectives on number theory (Chicago, IL, 1995).
Vol. 7.

Studies in Advanced Mathematics.

Providence, RI: AMS International Press,

Pp. 21–76.

URL: <http://www.ams.org/mathscinet-getitem?mr=1486831>.



Couveignes, Jean-Marc (1996).

“Computing l-Isogenies Using the p-Torsion.”

In: ANTS-II: Proceedings of the Second International Symposium on
Algorithmic Number Theory .

London, UK: Springer-Verlag,

Pp. 59–65.

URL: <http://portal.acm.org/citation.cfm?id=749581>.



References VII



Lercier, Reynald (1996).

“Computing Isogenies in $\text{GF}(2,n)$.”

In: ANTS-II: Proceedings of the Second International Symposium on Algorithmic Number Theory .

London, UK: Springer-Verlag,

Pp. 197–212.

URL: <http://portal.acm.org/citation.cfm?id=749582>.



Bostan, Alin, François Morain, Bruno Salvy, and Éric Schost (2008).

“Fast algorithms for computing isogenies between elliptic curves.”

In: Mathematics of Computation 77 ,

Pp. 1755–1778.

URL: <http://dx.doi.org/10.1090/S0025-5718-08-02066-8>.

References VIII



Lercier, Reynald and Thomas Sirvent (2008).

“On Elkies subgroups of ℓ -torsion points in elliptic curves defined over a finite field.”

In: Journal de théorie des nombres de Bordeaux 20.3 ,
Pp. 783–797.

URL:

<http://perso.univ-rennes1.fr/reynald.lercier/file/LS08.pdf>.



Khodadad, Sara and Michael Monagan (2006).

“Fast rational function reconstruction.”

In: Proceedings of the 2006 international symposium on Symbolic and algebraic computation.

ISSAC '06.

Genoa, Italy: ACM,

Pp. 184–190.

URL: <http://www.cecm.sfu.ca/~monaganm/papers/fastRFR.pdf>.

References IX



Voloch, José F. (1990).

“Explicit p -descent for elliptic curves in characteristic p .”

In: *Compositio Mathematica* 74.3 ,

Pp. 247–258.



Couveignes, Jean-Marc (2000).

“Isomorphisms between Artin-Schreier towers.”

In: *Mathematics of Computation* 69.232 ,

Pp. 1625–1631.

URL: <http://dx.doi.org/10.1090/S0025-5718-00-01193-5>.



Cantor, David G. (1989).

“On arithmetical algorithms over finite fields.”

In: *Journal of Combinatorial Theory, Series A* 50.2 ,

Pp. 285–300.

URL: [http://dx.doi.org/10.1016/0097-3165\(89\)90020-4](http://dx.doi.org/10.1016/0097-3165(89)90020-4).



References X



Rouillier, Fabrice (1999).

“Solving Zero-Dimensional Systems Through the Rational Univariate Representation.”

In: *Applicable Algebra in Engineering, Communication and Computing* 9.5

,
Pp. 433–461.

URL: <http://dx.doi.org/10.1007/s002000050114>.



Shoup, Victor (1995).

“A new polynomial factorization algorithm and its implementation.”

In: *Journal of Symbolic Computation* 20.4 ,

Pp. 363–397.

URL: <http://dx.doi.org/10.1006/jsco.1995.1055>.

References XI



Shoup, Victor (1999).

“Efficient computation of minimal polynomials in algebraic extensions of finite fields.”

In: ISSAC '99: Proceedings of the 1999 international symposium on Symbolic and algebraic computation .

Vancouver, British Columbia, Canada: ACM,

Pp. 53–58.

URL: <http://dx.doi.org/10.1145/309831.309859>.



Bostan, Alin, Bruno Salvy, and Éric Schost (2003).

“Fast Algorithms for Zero-Dimensional Polynomial Systems using Duality.”

In: Applicable Algebra in Engineering, Communication and Computing 14.4 ,

Pp. 239–272.

URL: <http://dx.doi.org/10.1007/s00200-003-0133-5>.