Computing square roots in nice field extensions

Javad Doliskani, Éric Schost

ORCCA, UWO

Context: genus 2 point-counting

With P. Gaudry: Schoof algorithm in $\mathbb{F}_p$, $p = 2^{127} - 1$.

To find the characteristic polynomial ϕ of the Frobenius:

- find ϕ modulo large primes
  bivariate resultants
- find ϕ modulo powers of 2
  square roots
- find ϕ modulo powers of 3
  homotopy techniques, root finding
- find ϕ modulo powers of 5, 7
  bivariate resultants
- random walk
  cockroaches

Schoof's algorithm modulo powers of 2

Main task: lifting the 2^k-torsion

- division by two on the Kummer surface
- invert doubling formulas
  Chudnovky^2, Gaudry
- each step requires 4 square roots, over increasing extensions of $\mathbb{F}_p$
  $2^4 = 16 = 2^{2\times 2}$
- each step, 1 out of 4 requires to extend the base field
  after k steps, we are in $\mathbb{F}_{p^{2^k}}$
- the cost of each step is $d^{O(1)}$, $d = 2^k$.

Remark: would work the same using Jacobian coordinates

This talk

Computing square roots in $\mathbb{F}_{p^{2^k}}$

- $O(\mathsf{M}(d)\log(d))$ (expected) operations in $\mathbb{F}_p$, with $d = 2^k$, if we are allowed to build $\mathbb{F}_{p^{2^k}}$ as we want

Factoring over $\mathbb{F}_{p^{2^k}}$

- factor f in $\mathbb{F}_{p^k}[x]$ with $\deg(f) = n$ and $d = 2^k$
- n, p fixed: $O(\mathsf{M}(d)\log(d))$ operations in $\mathbb{F}_p$
- cost quadratic in n, linear in $\log(p)$

$\mathsf{M}(d)$: cost of multiplying polynomials in degree d

- $\mathsf{M}(d) = O(d\log(d))$ or $\mathsf{M}(d) = O(d\log(d)\log\log(d))$

Previous work: taking square roots

To compute a square root of α in $\mathbb{F}_q$: compute something to a power something, somewhere.

Examples

- if $q \bmod 4 = 3$, compute $\alpha^{(q+1)/4}$ in $\mathbb{F}_q$
  $O(\log(q))$ products in $\mathbb{F}_q$
- otherwise, compute $(x+\alpha)^{(q-1)/2}$ in $\mathbb{A} = \mathbb{F}_q[x]/(x^2-\alpha)$
  $O(\log(q))$ products in $\mathbb{A} \to O(\log(q))$ products in $\mathbb{F}_q$
- Cipolla-Lehmer, Atkin, Tonelli-Shanks, Müller, Han *et al.*, …

Cost

- $d\mathsf{M}(d)$, with $q = p^d$

Previous work: factoring

Kaltofen, Shoup, 1997: factoring in high-degree field extensions

- factor $f \in \mathbb{F}_q[x]$ of degree n
- uses Cantor-Zassenhaus' approach (DDF / EDF)

Main contribution: Frobenius and trace

- $\alpha \mapsto \alpha^{p^i}$ in $\mathbb{F}_q[x]/f$
- $\alpha \mapsto \alpha + \alpha^p + \cdots + \alpha^{p^{k-1}}$ in $\mathbb{F}_q[x]/f$

Cost

- n, p fixed: $O(\mathsf{C}(d)\log(d))$
- $\mathsf{C}(d)$ = cost of modular composition $f, g, h \mapsto f(g) \bmod h$
- $\mathsf{C}(d) \in O(\sqrt{d}\,\mathsf{M}(d) + \sqrt{d}\,\sqrt{d}^{\,\omega})$
- Kedlaya-Umans: $\mathsf{C}(d)$ quasi-linear in d

Previous work: taking square roots

Wang, Nogami, Morikawa, 2005

- dedicated to $q = p^d$, with $d = 2^k$
- tests quadratic residuosity and computes square roots in $\mathbb{F}_{p^{2d}}$
- Tonelli-Shanks: computes α^s, such that $p^{2^k} - 1 = 2^t s$ and s odd
- reduces to computations in $\mathbb{F}_{p^{2^i}}$, $i = 0, \ldots, k$. Dominant factors
  - $O(\mathsf{M}(d) \log(d)^2)$
  - $O(\log(d)^2)$ Frobeniuses

Kato, Nogami, Morikawa, 2009

- extensions to $q = p^d$, with $d = r_1 \cdots r_\ell 2^k$

Defining $\mathbb{F}_{p^{2^k}}$

Assumptions

- $p = 1 \bmod 4$.
- we know a non-quadratic residue $r \in \mathbb{F}_p$

Consequence

- $x^{2^k} - r$ is irreducible for all k

Remark

- if $p = 3 \bmod 4$, replace the base field $\mathbb{F}_p$ by $\mathbb{F}_p[z]/(z^2 + 1)$
- most likely, not too many differences

Seen in Shoup, 1994

Bases for $\mathbb{F}_{p^{2^k}}$

Multivariate basis: $\{x_1^{e_1} \cdots x_k^{e_k}\}$, for $e_i \in \{0,1\}$, modulo the relations

$$\left| \begin{array}{l} x_k^2 - x_{k-1} \\ \quad \vdots \\ x_2^2 - x_1 \\ x_1^2 - r \end{array} \right.$$

Univariate basis: $\{x_k^i\}$, for $i \in \{0, \ldots, 2^k - 1\}$, modulo $x_k^{2^k} - r$

Change of basis

- no arithmetic operation
- shuffling coefficients (bit reversal)

Multiplication in $\mathbb{F}_{p^{2^k}}$

- $\mathsf{M}(d) + O(d)$, with $d = 2^k$

cf. work with De Feo on Artin-Schreier extensions

Inverse in $\mathbb{F}_{p^{2^k}}$

To invert $A(x_k)$ in $\mathbb{F}_{p^{2^k}}$, write

$$A = A_0(x_k^2) + x_k A_1(x_k^2) = A_0(x_{k-1}) + x_k A_1(x_{k-1}).$$

Then,

$$\frac{1}{A} = \frac{1}{A_0 - x_k A_1} = \frac{A_0 - x_k A_1}{A_0^2 - x_{k-1}^2 A_1^2}.$$

- shuffling
- multiplications in $\mathbb{F}_{p^{2^{k-1}}}$
- one inversion in $\mathbb{F}_{p^{2^{k-1}}}$

Total: $O(\mathsf{M}(d))$, instead of $O(\mathsf{M}(d)\log(d))$ for a general $\mathbb{F}_q$.

Inspired by Schönhage, 2000 (power series inverse)

Frobenius

To compute $\pi(A, i, k) = A(x_k)^{p^{2^i}}$ in $\mathbb{F}_{p^{2^k}}$:

- if $i \geq k$, do nothing
- else, write

$$A = A_0(x_k^2) + x_k A_1(x_k^2) = A_0(x_{k-1}) + x_k A_1(x_{k-1});$$

  then, $\pi(A, i, k) = \pi(A_0, i, k-1) + \pi(x_k, i, k)\pi(A_1, i, k-1)$.

- because $x_k^{2^k} = r$, $\pi(x_k, i, k) = x_k^{p^{2^i}} = r^{q_{i,k}} x_k^{s_{i,k}}$, with

$$p^{2^i} = q_{i,k} 2^k + s_{i,k}$$

Two recursive calls and $O(d)$ multiplications by constants.

- Total: $O(d \log(d))$
- Or maybe $O(d)$

Norm and QR test

Given α in $\mathbb{F}_{p^{2^k}}$, to compute

$$N(\alpha, k) = \alpha \cdot \alpha^p \cdots \alpha^{p^{2^k-1}} = \alpha^{1+p+\cdots+p^{2^k-1}}$$

(a simplified version of von zur Gathen, Shoup, 1992)

- compute $N(\alpha, k-1)$
- then $N(\alpha, k) = N(\alpha, k-1) N(\alpha, k-1)^{p^{2^{k-1}}}$

Total:

- k Frobenius and k products = $O(\mathsf{M}(d)\log(d))$, since $k = \log(d)$.

Quadratic residuosity test: compute $N(\alpha, k)^{\frac{p-1}{2}}$

Square root, factoring, etc

Given α in $\mathbb{F}_{p^{2^k}}^2$ and $\beta = \beta_0 + x\beta_1$ in $\mathbb{F}_{p^{2^k}}[x]$, to compute

$$T(\beta, k) = \beta + \beta^p + \cdots + \beta^{p^{2^k - 1}} \bmod (x^2 - \alpha)$$

(a simplified version of von zur Gathen, Shoup, 1992)

- $O(\mathsf{M}(d) \log(d))$

Square root: compute $\gcd(x^2 - \alpha, T(\beta, k)^{\frac{p-1}{2}} - 1)$

Isomorphisms between towers describing $\mathbb{F}_{p^{2^k}}[x]$

Factoring: same approach to factor $f \in \mathbb{F}_{p^{2^k}}[x]$

Timings for square root

Naive root-finding in $\mathbb{F}_q$ vs. Kaltofen, Shoup, 1997

n	α non quadratic residue		α quadratic residue	
	NTL	Kaltofen, Shoup	NTL	Kaltofen, Shoup
4	0.012	0.0008	0.05	0.0036
8	0.039	0.0052	0.22	0.009
16	0.23	0.026	1.6	0.037
32	1.5	0.078	9.4	0.12
64	6.3	0.18	51	0.36
128	32	0.64	155	0.9
256	124	1.7	823	3.3
512	512	4.9	3353	10

Timings to lift 2^k-torsion

Curve of genus 2, defined over $\mathbb{F}_p$, with $p \simeq 2^{128}$.

index	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}
degree d	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
square root	23	77	280	1100	5000	22000		
new square root	4.5	10	23	80	70	190	400	1200
s_1, s_2	15	36	90	290	900	3000	6000	18000

Bonus: memory

- the memory in Kaltofen-Shoup's algorithm is non-linear
- now, linear memory (up to logs)