The discrete logarithm problem.
3 – Best complexities known in finite fields

Pierrick Gaudry

Caramel – LORIA
CNRS, Université de Lorraine, Inria

MPRI – 12.2 – 2013-2014

Plan

The general picture

The number field sieve for DL

The quasi-polynomial algorithm in small characteristic

Notations

Finite field $= \mathbb{F}_Q$, with $Q = p^n$ with:

- p is a prime $=$ the characteristic.
- n is integer (prime or not prime).

Main complexity is in $L_Q(\frac{1}{3})$.

Limits between algorithms:

- $p > L_Q(\frac{2}{3})$: NFS
- $L_Q(\frac{1}{3}) < p < L_Q(\frac{2}{3})$: NFS-HD
- $p < L_Q(\frac{1}{3})$: FFS – quasi-polynomial.

Relation between $\log p$ and n

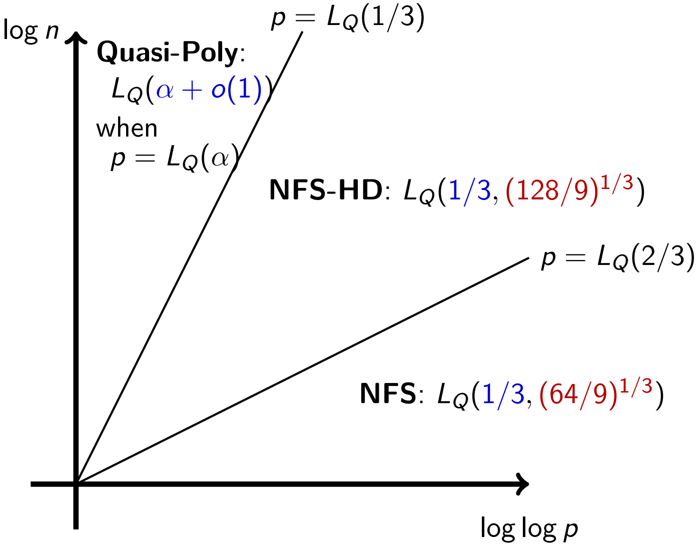
In terms of **size**: $\log Q = n \log p$.
If $p = L_Q(\alpha, c)$, then $n = \frac{1}{c}\left(\frac{\log Q}{\log\log Q}\right)^{1-\alpha}$.
Hence:

$$\begin{aligned} \log p &= c(\log Q)^{\alpha}(\log\log Q)^{1-\alpha} \\ n &= \tfrac{1}{c}(\log Q)^{1-\alpha}(\log\log Q)^{\alpha-1} \end{aligned}$$

The limits correspond to $\log p$ or n reaching $\approx (\log Q)^{2/3}$, thus creating **norms** that are **too big** for an $L_Q(1/3)$-algorithm.

Rem. To get straight lines on the next picture, we must add another level of log.

Complexities on a picture



Complexities: variants

Coppersmith's multiple-fields variant:

- Initially invented for factorization;
- Extended to discrete log over $\mathbb{F}_p$ by Matyukhin (see also Commeine–Semaev);
- Complexity exponent drops from $\left(\frac{64}{9}\right)^{1/3} \approx 1.923$ to ≈ 1.902.
- Uses a subexponential number of algebraic sides, with a common rational side.

SNFS variant:

- If N is of a **special form**, then its factorization by NFS can have a complexity exponent as low as $\left(\frac{32}{9}\right)^{1/3}$;
- Historically very important;
- Used for computing factors for the **Cunningham project**; in particular numbers $2^n \pm 1$;
- SNFS extended to DL in $\mathbb{F}_p$ by Semaev;
- Active research topic: recent paper by Joux–Pierrot.

DL records

We give records as of today (November 2013).
Most of them are very recent, or too old to hold long.

This might change quickly!

- $\mathbb{F}_p$, 160 digits. Kleinjung (2007).
- $\mathbb{F}_{33341353^{57}}$, 429 digits. Joux (2013).
- $\mathbb{F}_{3^{6\cdot 97}}$, 278 digits. Hayashi-Shimoyama-Shinohara-Takagi (2012).
- $\mathbb{F}_{2^{6168}}$, 1857 digits. Joux (2013) (see also records by Göloğlu-Granger-McGuire-Zumbrägel).
- $\mathbb{F}_{2^{809}}$, 244 digits. Nancy (2013).

Rem: Remember that the record for integer factorization is RSA-768, with 232 digits (2010).

Plan

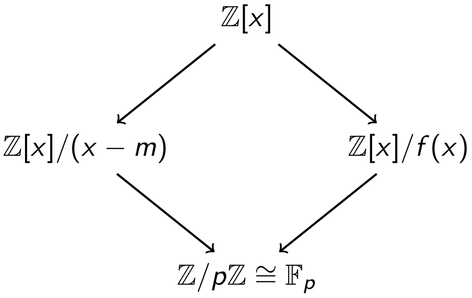
The general picture

The number field sieve for DL

The quasi-polynomial algorithm in small characteristic

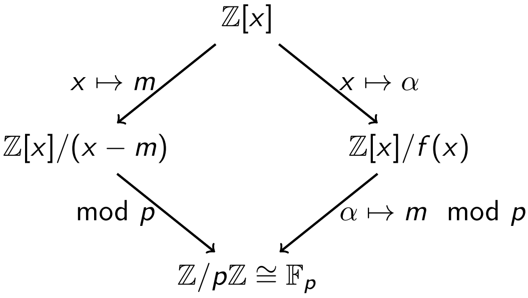
The magic diagram

Let $f(x)$ be a polynomial and m an integer such that $f(m) \equiv 0 \mod p$. We denote by α the algebraic number that is a root of f. The **diagram commutes** (the maps are **ring** homomorphisms).



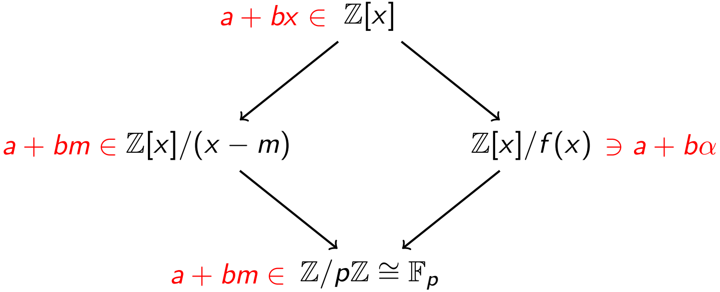
The magic diagram

Let $f(x)$ be a polynomial and m an integer such that $f(m) \equiv 0 \mod p$. We denote by α the algebraic number that is a root of f. The **diagram commutes** (the maps are **ring** homomorphisms).



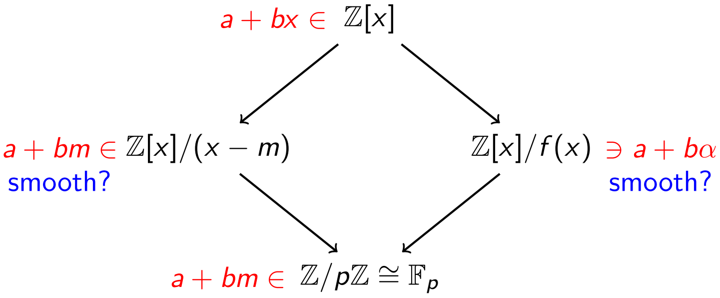
The magic diagram

Let $f(x)$ be a polynomial and m an integer such that $f(m) \equiv 0 \mod p$. We denote by α the algebraic number that is a root of f. The **diagram commutes** (the maps are **ring** homomorphisms).



The magic diagram

Let $f(x)$ be a polynomial and m an integer such that $f(m) \equiv 0 \mod p$. We denote by α the algebraic number that is a root of f. The **diagram commutes** (the maps are **ring** homomorphisms).



If smooth on both sides, then we get a **relation** in $\mathbb{F}_p$.

What does it mean to be smooth?

On the **rational side** (left): smoothness of integers. OK.

On the **algebraic side** (right):

- Smoothness in a number ring.
- In general, this is not a Unique Factorization Domain.
- Have to factor **ideals**.
- A lot of (theoretical and practical) technicalities to define the "log of an ideal mapped to $\mathbb{F}_p$."
  Work of Schirokauer.

Rem. Main mathematical notion: **Dedekind domain**. Algorithms for manipulating ideals have been developped in the late 80's (Cohen's school).

Practical considerations

Rem. For a fast implementation, have to write some two-dimensional Eratosthenes-like sieve.
A bit of **lattice theory**, here.

Rem. For finding relations, exactly the same code can be used as for integer factorization by NFS.

Warning. The linear algebra step is very different: over $\mathbb{F}_2$ for factoring; over $\mathbb{Z}/(p-1)\mathbb{Z}$ for DL.

Sizes, complexity

The degree d of $f(x)$ will be $\approx (\log p)^{1/3}$.

The size of the coefficients of f and g is around $p^{1/d} \approx L_p(2/3)$.

The size of the candidates (a, b) for getting a relation is $\approx L_p(1/3)$.

The integers that we have to test for smoothness (the "norms") have size $L_p(2/3)$.

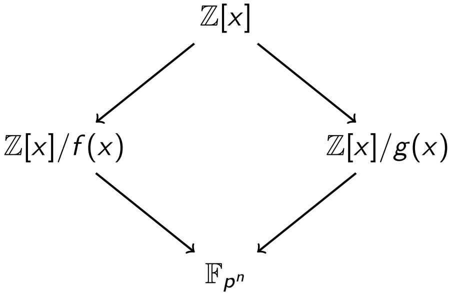
We set the smoothness bound to $\approx L_p(1/3)$.

$$\text{The overall complexity is } L_p\left(1/3, \left(\frac{64}{9}\right)^{1/3}\right).$$

Rem. Understanding the $L_p(1/3)$ nature of the complexity is ok. Getting the right exponent is very much error-prone.

DL in $\mathbb{F}_{p^n}$, with small n

Need to find f and g, such that we get a similar commutative diagram:



This imposes that both f and g have a degree at least n, in order to have a **common irreducible factor** of degere n modulo p.

Game: Find such polynomials with coefficents and degree as small as possible.

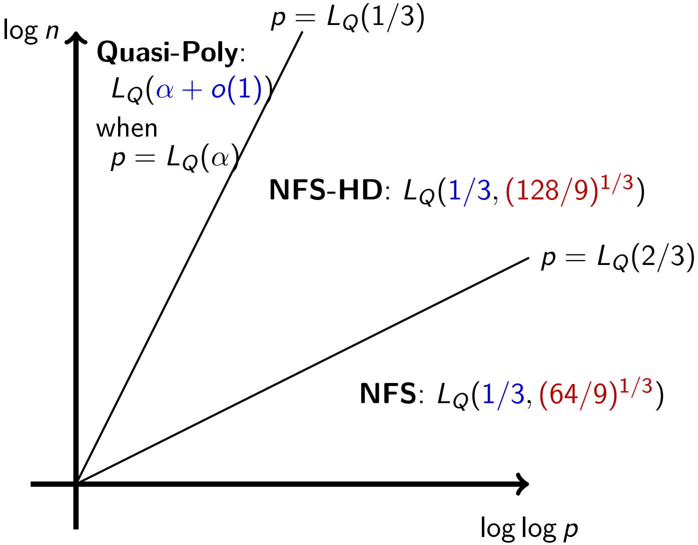
Plan

The general picture

The number field sieve for DL

The quasi-polynomial algorithm in small characteristic

Complexities on a picture



Quasi-polynomial algorithm

Recent result by Barbulescu, Gaudry, Joux, Thomé (2013, still under review).

Theorem (based on heuristics)

Let K be a finite field of the form $\mathbb{F}_{q^k}$. A discrete logarithm in K can be computed in heuristic time

$$\max(q,k)^{O(\log k)}.$$

Cases:

- $K = \mathbb{F}_{2^n}$, with prime n. Complexity is $n^{O(\log n)}$. Much better than $L_{2^n}(1/3 + o(1)) \approx 2^{\sqrt[3]{n}}$.
- $K = \mathbb{F}_{q^k}$, with $q \approx k$. Complexity is $\log Q^{O(\log\log Q)}$, where $Q = \#K$. Again, this is $L_Q(o(1))$.
- $K = \mathbb{F}_{q^k}$, with $q \approx L_{q^k}(\alpha)$. Complexity is $L_{q^k}(\alpha + o(1))$, i.e. better than Joux-Lercier or FFS for $\alpha < 1/3$.

Setting

The setting of the algorithm is the following:

$K = \mathbb{F}_{q^{2k}}$, with $k \approx q$.
The field $\mathbb{F}_{q^2}$ is represented in any usual way.

The **extension** of degree k is constructed as follows:

- Take h_0 and h_1 two polynomials over $\mathbb{F}_{q^2}$, of small degree (2 should be ok).
- Let $\Phi(X) = h_1(X)X^q - h_0(X)$.
- Until there is an irreducible factor $I(X)$ of $\Phi(X)$ of degree k.

Rem. This works only if $k \leq q + 2$.

How to fit in this setting?

If the given field $\mathbb{F}_{p^n}$ is such that $n > p + 2$, we **embed** the DL in $\mathbb{F}_{p^n}$ into a **larger field**:
Let q be the smallest power of p such that $q + 2 \geq n$ and set $k = n$.

Then, $\mathbb{F}_{q^{2k}}$ contains $\mathbb{F}_{p^n}$ and we are in the previous setting.
The cost of this embedding is reflected by the max() in the formula of the complexity.

Rem. If n is composite, it might not be necessary to pay as much for this extension.

General strategy

Given an element $P(x)$ in $\mathbb{F}_{q^{2k}}$ represented as a polynomial of degree $D \leq k-1$ over $\mathbb{F}_{q^2}$, we are going to **descend** it:

- Find a linear relation between $\log P$ and the logs of elements of degrees at most $D/2$;
- Do it **recursively**: each new log can be again expressed in terms of logs of polynomials of smaller degrees;
- Go down to degree 1;
- The logs of all linear polynomials can be found in polynomial-time in q.

One step of descent

Proposition (heuristic)

Let $P(X) \in \mathbb{F}_{q^2}$ of degree $D < k$. In time polynomial in D and q, we can express $\log P$ as a linear combination $\sum e_i \log P_i$, where $\deg P_i \leq D/2$, and the number of P_i is in $O(q^2 D)$.

Provided that the logs of linear polynomials can be computed in polynomial time in q, then the main result follows from the analysis of the size of the descent tree.

The descent tree

Each node of the descent tree corresponds to one application of the Proposition, hence its arity is in $q^2 D$.

level	$\deg P_i$	width of tree
0	k	1
1	$k/2$	$q^2 k$
2	$k/4$	$q^2 k \cdot q^2 \frac{k}{2}$
3	$k/8$	$q^2 k \cdot q^2 \frac{k}{2} \cdot q^2 \frac{k}{4}$
$\vdots$	$\vdots$	$\vdots$
$\log k$	1	$\leq q^{2 \log k} k^{\log k}$

Total number of nodes $= q^{O(\log k)}$.
Each node yields a cost that is polynomial in q, hence the result.

One step of descent: how?

Start from the field equation:

$$X^q - X = \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta X - \alpha),$$

Plug the input $P(X)$, twisted by an homography $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$:

$$\begin{aligned}
(aP(X)+b)^q(cP(X)+d) - (aP(X)+b)(cP(X)+d)^q \\
&= \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} \beta(aP(X)+b) - \alpha(cP(X)+d) \\
&= \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} (\beta a - \alpha c)P(X) + (\beta b - \alpha d) \\
&= \lambda \prod_{(\alpha:\beta)\in\mathbb{P}^1(\mathbb{F}_q)} P(X) - m^{-1}\cdot(\alpha:\beta).
\end{aligned}$$

One step of descent: how?

Left-hand side:
Let the q-power come inside the formulae, and use
$X^q \equiv h_0(X)/h_1(X)$.
Hence, modulo denominator cleaning, it is a polynomial of degree
$O(\deg P)$.
Probability that LHS splits in polys of degree $\leq \frac{1}{2} \deg P$ is
constant.

Right-hand side:
All factors are in $\left\{P(X) - \gamma \mid \gamma \in \mathbb{F}_{q^2}\right\}$.

One step of descent: how?

Now, we let the matrix $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ vary.

The RHS is the same as for $m = \mathrm{Id}$ if m is in $PGL_2(\mathbb{F}_q)$.
The appropriate set where to pick m is the set of cosets:

$$\mathcal{P}_q = PGL_2(\mathbb{F}_{q^2})/PGL_2(\mathbb{F}_q).$$

For any $\mathfrak{q}$, the order of $PGL_2(\mathbb{F}_\mathfrak{q}) = \mathfrak{q}^3 - \mathfrak{q}$, so

$$\#\mathcal{P}_q = q^3 + q.$$

Conclusion: Have $\Theta(q^3)$ relations; need q^2 to eliminate the right-hand sides. More than enough! (but heuristic)

Logarithms of linear polynomials

Strategy: set $P(X) = X$ in the same machinery as before.

The LHS have degree: the same as degrees of h_0 and h_1, say 2.
The probability that it splits into linear factors is $1/2$.

By construction, the RHS is a product of linear factors.

Conclusion: Have $\Theta(q^3)$ relations; expect to need $O(q^2)$ to get a full rank matrix. Again, more than enough! (but heuristic)

Rem: Here, this is a kernel computation, whereas inside the descent tree, we solve inhomogenous systems.

Final remarks

Today's situation:

- Very recent algorithm; DL is a hot topic these days.
- Many practical improvements yet to be discovered.
- It might be possible to prove some of the heuristics.

Big Open Questions:

- Can we get a (heuristic) polynomial-time algorithm in small characteristic ?
- Can we extend the range of applicability of the quasi-polynomial time algorithm.