Discrete logarithm in finite fields of small characteristic

Édouard Rousseau
Université de Versailles

June 1, 2017

Contents

CONTEXT

Let G be a cyclic group generated by an element g, we denote by N the cardinal of G. Then we have a *bijection*

$$\begin{array}{rccc} exp_g : & \mathbb{Z}/N\mathbb{Z} & \to & G \\ & \bar{n} & \mapsto & g^n \end{array} .$$

The inverse of exp_g will be denoted by log_g.

THE DISCRETE LOGARITHM PROBLEM

- In practice, given an integer n, we have efficient algorithms to compute g^n
- Given $x = g^n \in G$, we *do not* hace efficient algorithm to compute n.

This last problem is called the *discrete logarithm problem*.

DEFINITIONS

To express the hardness of a problem, we study its complexity and we use the notation

$$L_N(\alpha, c) = \exp((c + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}).$$

We also note $L_N(\alpha)$ when we do not want to deal with the constant.
There are two families of algorithms :

- The *generic* algorithms (with complexity $O(\sqrt{N})$)
- The *index calculus* algorithms, which use the structure of groups coming from finite fields : $\mathbb{F}_q^{\times}$

DEFINITIONS

- We say that a finite field $\mathbb{F}_q = \mathbb{F}_{p^k}$ is of *small characteristic* when p is small compared to q, it usually means that p is polynomial in $\log q$.
- Let $l = \log q$, the complexity is then said to be *quasi-polynomial* if it is $l^{O(\log l)}$. This complexity is smaller that $L(\varepsilon)$ for any $\varepsilon > 0$ but greater than any polynomial in l.

HISTORICAL BACKGROUND

- First appearance in [DH76]
- First sub-exponential algorithm [A79] : $L(1/2)$
- Between 1984 and 2006 : algorithms in $L(1/3)$

And more recently, in finite fields of small characteristic :

- New algorithm with $L(1/4)$ complexity [Joux13]
- *Quasi-polynomial* algorithm [BGJT14]
- Second quasi-polynomial algorithm [GKZ14]

Overview

Assume that we want to find $log_g(h)$ with $h \in G$. We first choose $F \subset G$ such that $\langle F \rangle = G$. Then

1. We find multiplicative relations between the elements in F
2. We solve the linear system arising from these relations
3. We express h as a product of elements in F

The steps 1 and 3 depends on the representation of the finite field, and give different complexities.

An example

Context :

- We considere $G = \mathbb{F}_p^\times$ for a prime integer p and we still have $N = |G|$
- We choose $F = \{f \mid f \leq B,\ f \text{ prime}\}$ for a chosen integer B
- We assume that $g \in F$, otherwise we add it to F.

AN EXAMPLE

Step 1 : relations generation

- We randomly choose $e \in \mathbb{Z}/N\mathbb{Z}$
- We compute g^e
- We test if g^e, seen as an integer, is B-smooth, *i.e.* has only prime factors $\leq B$
- If it is the case, it yields a relation in G :

$$g^e = \prod_{f \in F} f^{e_f}, \text{ où } e_f \in \mathbb{N}$$

that can be written

$$e = \sum_{f \in F} e_f \log_g(f).$$

AN EXAMPLE

Step 2 : linear algebra. Once we have enough relations, *i.e.* at least $|F|$, we solve the linear system in $\mathbb{Z}/N\mathbb{Z}$ in order to obtain the $\log_g(f)$ for all $f \in F$.
Step 3 : express h in function of the elements in F :

- We randomly choose $e \in \mathbb{Z}/N\mathbb{Z}$
- We compute hg^e
- We test if hg^e is B-smooth
- If it is the case, it yields a relation :

$$\log_g(h) = \sum_{f \in F} e_f \log_g(f) - e$$

An example

This algorithm depends on the choice of B :

- The larger B is, the larger $\langle F \rangle$ is, and the easier it is to find relations
- But when $|F|$ is large, we need to solve a large linear system

In the end, we can choose B to obtain a $L(1/2)$ complexity.

BARBULESCU, GAUDRY, JOUX AND THOMÉ ALGORITHM

Context : we denote by $\mathbb{K}$ the finite field where we want to compute discrete logarithms.

- We assume that $\mathbb{K} = \mathbb{F}_{q^{2k}}$ and we represent $\mathbb{K}$ by $\mathbb{F}_{q^2}[X]/(I_X)$ where I_X is an irreducible polynomial of degree k dividing $h_1 X^q - h_0$ and $\deg h_i \leq 2$.
  - The existence of suitable h_i is heuristic
- The set F is the set of the degree one polynomials.

BARBULESCU, GAUDRY, JOUX AND THOMÉ ALGORITHM

The algorithm is based on a descent process: we express the logarithm of a polynomial as a linear combination of $O(q^2 k)$ logarithms of polynomials of degree two times smaller, until we have only polynomials in F.

- Complexity : $(q^2 k)^{O(\log k)}$.

The descent process is based on the equation :

$$X^q Y - XY^q = Y \prod_{a \in \mathbb{F}_q} (X - aY) = \prod_{\alpha \in \mathbb{P}^1(\mathbb{F}_q)} (X - \alpha Y) \qquad (1)$$

THE DESCENT

Assume that we want to find the logarithm of an element P. We will create relations by substituting X by $aP+b$ and Y by $cP+d$ in Equation (1), with $a,b,c,d \in \mathbb{F}_{q^2}$. We obtain a new Equation $(E_{a,b,c,d})$. It follows that

$$\frac{1}{h_1^D}\mathcal{L}_{a,b,c,d} \quad = \quad \lambda \prod_{\alpha \in \mathbb{P}^1(\mathbb{F}_q)} (P - \mu_\alpha)$$

where $\lambda, \mu_\alpha \in \mathbb{F}_{q^2}$ and $\mathcal{L}_{a,b,c,d}$ is a polynomial of degree $D \leq 3 \deg P$, obtained using the equality $X^q = \frac{h_0}{h_1} \mod I_X$.

THE DESCENT

We keep only the equations $(E_{a,b,c,d})$ where $\mathcal{L}_{a,b,c,d}$ is $\left\lceil \frac{\deg P}{2} \right\rceil$-smooth and we combine these equations in order to keep only P in the right hand side.
The left hand side is then composed of irreducible polynomials of degree at most $\left\lceil \frac{\deg P}{2} \right\rceil$.

- There are also heuristics
  - The existence of the combination
  - The smoothness of the polynomials $\mathcal{L}_{a,b,c,d}$

GRANGER, KLEINJUNG AND ZUMBRÄGEL ALGORITHM

Context : here $\mathbb{K} = \mathbb{F}_{q^k}$ and we see $\mathbb{K}$ as $\mathbb{F}_q[X]/(I_X)$ where I_X is a polynomial of degree k dividing $h_1 X^q - h_0$. The algorithm follows the same steps as the latter, but the descent is different.

"On the fly" elimination

Assume that $Q \in \mathbb{F}_{q^m}[X]$ and $\deg Q = 2$. This elimination is based on the fact that the polynomial $P = X^{q+1} + aX^q + bX + c$ splits completly in $\mathbb{F}_{q^m}[X]$ for approximately q^{m-3} triples (a, b, c). But

$$P = \frac{1}{h_1}((X+a)h_0 + (bX+c)h_1) \mod I_X$$

So if $Q | (X+a)h_0 + (bX+c)h_1$ (polynomial of degree 3), we have

$$h_1 P = QL \mod I_X$$

where L is of degree 1 and P splits completly.

THE DESCENT

Assume now that $Q \in \mathbb{F}_q[X]$ is irreducible of degree $2d$. Then we have

$$Q = \prod_{i=0}^{d-1} Q_i = \prod_{i=0}^{d-1} Q_0^{[i]}$$

where the Q_i's are irreducible polynomials of degree 2 in $\mathbb{F}_{q^d}[X]$ and are all conjugates, in the sense that $Q_i = Q_0^{[i]}$ denotes the polynomial obtained by raising all coefficients to the power q^i.

THE DESCENT

- We then apply the “on the fly” elimination to $Q_0^{[i]}$ to obtain $Q_0^{[i]}$ as a product of $O(q)$ $P_j^{[i]}$ where the $P_j \in \mathbb{F}_{q^d}[X]$ are all of degree 1
- Hence, Q is expressed as a product of $O(q)$ norms of linear polynomial $R_j = \prod_{i=0}^{d-1} P_j^{[i]}$

Recall that the norm of a linear polynomial in $\mathbb{F}_{q^d}[X]$ is an irreducible polynomial of degree d_1 to the power d_2, with $d_1 d_2 = d$.

- *Thus, we expressed* $\log Q$ *as a linear combination of* $O(q)$ $\log R_j$ *where* $\deg R_j | d$. *The complexity obtained is* $q^{O(\log q)}$.