Technical history of discrete logarithms in small characteristic finite fields

Antoine Joux

Fondation UPMC, Sorbonne Universités/UPMC/LIP6/Almasty

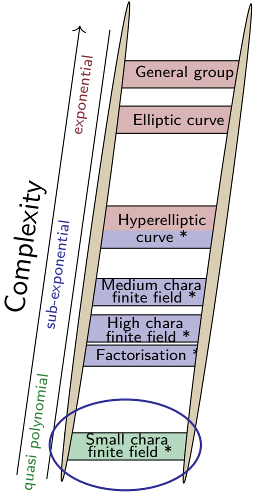
September 20th, 2016

Workshop on Maths of Information Theoretic Cryptography

Cécile Pierrot

The Discrete Logarithm Problem (DLP)

- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$
- A hard problem in general, and used as such in cryptography.
- Several groups in practice:
- Two algorithmic approaches:
  - Generic algorithms (Pollard's Rho, Pohlig-Hellman...)
  - Specific algorithms (Index Calculus *)



Generic algorithms

Generic algorithms: Pohlig-Hellman

- Given a multiplicative group G with generator g
- Given $|G| = \prod_{i=1}^{k} p_i^{e_i}$
- To compute dlogs in G, it suffices to compute dlogs in:

$$G_i = \langle g^{|G|/p_i} \rangle \quad \text{(Group of order } p_i\text{)}$$

Generic algorithms: $|G| = p$

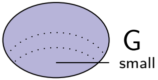
- There exist algorithms with complexity $O(\sqrt{p})$ to solve:

$$y = g^n$$

- Baby-step giant-step (let $R = \lceil \sqrt{p} \rceil$):
  - Create list $y, y/g, \cdots, y/g^{R-1}$
  - Create list $1, h, h^2, \cdots, h^{R-1}$, where $h = g^R$
  - Find collision
- Can be improved to memoryless algorithms using cycle finding techniques

Index Calculus Algorithms

To compute Discrete Logs in G:



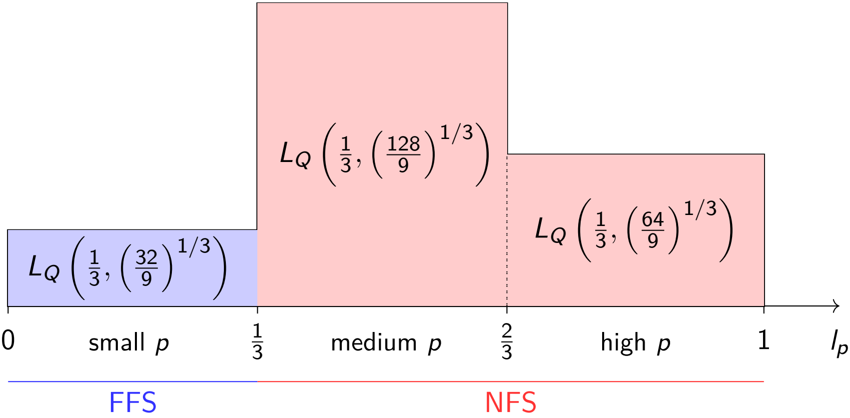
1. Collection of Relations
   → Create a lot of sparse multiplicative relations between some (small) specific elements = the factor base

   $$\prod g_i^{e_i} = \prod g_i^{e'_i} \quad \Rightarrow \quad \sum (e_i - e'_i) \log(g_i) = 0$$

   → So a lot of sparse linear equations
2. Linear Algebra
   → Recover the Discrete Logs of the factor base
3. Extension Phase (for small characteristic finite fields)
   → Recover the Discrete Logs of the extended factor base
4. Individual Logarithm Phase
   → Recover the Discrete Log of an arbitrary element

Complexity of Index calculus algorithms (before 2013)

$$L_Q(\beta, c) = \exp((c + o(1))(\log Q)^{\beta}(\log \log Q)^{1-\beta}).$$



Function field sieve (with polynomials)

Discrete Logarithms in the Medium prime case [JL06]

- Finite field of the form $\mathbb{F}_{p^k}$
- Choose two univariate polynomials f_1 and f_2
  - with degrees d_1 and d_2 and $d_1 d_2 \geq k$.
  - Such that $x - f_1(f_2(x))$ has:
    - an irreducible factor of degree k (modulo p).
- This defines the finite field by the relations:
  - $x = f_1(y)$ and $y = f_2(x)$

Discrete Logarithms in the Medium prime case [JL06]

- Optimal for $p = L_{1/3}(p^k)$

- Choose smoothness basis $x - \alpha$ and $y - \alpha$
- Consider elements:

$$\begin{aligned} xy + ay + bx + c &= x\,f_2(x) + af_2(x) + bx + c \\ &= y\,f_1(y) + ay + bf_1(y) + c \end{aligned}$$

- When both sides split $\Rightarrow$ Relation
- Classical approach, get relations by sieving:
  - For each a, b and α, compute c such that $(x - \alpha) \mid x\,f_2(x) + ax + bf_2(x) + c$.
  - Idem for y
  - If c has enough hits $\Rightarrow$ Relation
- Cost of finding relation is $(d + 1)!\,(d' + 1)!$

Pinpointing

Linear change of variables [J13]

- Further restrict to $y = x^d$
- Then:

$$xy + ay + bx + c \quad = \quad x^{d+1} + ax^d + bx + c$$

- Perform change of variable: $x = aX$, we get:

$$a^{d+1}(X^{d+1} + X^d + b \cdot a^{-d}(X + c/(ab)).$$

- Change of variable does not affect splitting property
- One good left-hand side $\Rightarrow$ p good left-hand sides
- Amortized cost of relation reduced to

$$\left(\frac{(d+1)!}{p-1} + 1\right) \cdot (d'+1)!$$

Impact in the medium prime case

- In theory, complexity of function field sieve:
  - Reduce in the best case from $L_{1/3}(3^{1/3}) \approx L_{1/3}(1.44)$ to $L_{1/3}(2 \cdot 3^{-2/3}) \approx L_{1/3}(0.96)$
  - Regardless of Kummer extension or not
- In practice, new records:
  - First 1175-bit field $\mathbb{F}_{p^{47}}$ with p close to 2^{25}
  - Then 1425-bit field $\mathbb{F}_{p^{57}}$ with p close to 2^{25}
  - Previous record was 923 bits
  - Kummer extensions very useful for records

Contructed relations

Starting point

- We need a smooth polynomial to play with:

$$X^q - X = \prod_{\alpha \in \mathbb{F}_q} (X - \alpha).$$

- Linear transformations not enough, need more.
- Replace X by $A(X)/B(X)$:

$$\frac{A(X)^q}{B(X)^q} - \frac{A(X)}{B(X)} = \prod_{\alpha \in \mathbb{F}_q} \left(\frac{A(X)}{B(X)} - \alpha \right).$$

- Multiply by $B(X)^{q+1}$:

$$A(X)^q \, B(X) - A(X) \, B(X)^q = B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha \, B(X)).$$

- Rewrite as:

$$\tilde{A}(X^q) \, B(X) - A(X) \, \tilde{B}(X^q) = \prod_{\alpha \in \mathbb{P}_1(\mathbb{F}_q)} (A(X) - \alpha \, B(X)).$$

Dealing with the left side – Basic idea

- Consider $\tilde{A}(X^q)\,B(X) - A(X)\,\tilde{B}(X^q).$
- What can we do to make it smooth (w. h. p.) ?
- Ask for low degree !
- How can we replace X^q by a low degree thing ?
- By choosing a polynomial defining the extension field as:

$$X^q - h(X).$$

Example

- Kummer case $X^q - aX$
- If a is good $X^{q-1} - a$ is irreducible
- Twisted Kummer case $X^q - a/X$
- If a is good $X^{q+1} - a$ is irreducible
- More generally consider

$$X^q - \frac{h_0(X)}{h_1(X)} \quad \text{i.e.} \quad h_1(X)\,X^q - h_0(X).$$

- And let θ be a root of its large irred. factor I_k

What happens to the left side ?

- Now $\tilde{A}(X^q)\,B(X) - A(X)\,\tilde{B}(X^q)$ becomes:

$$\frac{[A,B]_D}{h_1(X)^D}.$$

- Where $[A,B]_D$ is defined as:

$$[A,B]_D = \left(\tilde{A}\left(\frac{h_0(X)}{h_1(X)}\right)\,B(X) - A(X)\,\tilde{B}\left(\frac{h_0(X)}{h_1(X)}\right)\right).$$

- $[A,B]_D$ is a polynomial of degree at most $D\,(H+1)$
- If A and B have degree at most D

Constructed relation

- In the field $\mathbb{F}_{q^k}$ (defined as $\mathbb{F}_q[\theta]$):

$$[A, B]_D(\theta) = h_1(\theta)^D \cdot \prod_{\alpha \in \mathbb{P}_1(\mathbb{F}_q)} (A(\theta) - \alpha\, B(\theta)).$$

- Also works directly in any extension $\mathbb{F}_{q^{tk}}$ with $\gcd(t, k) = 1$.
- Good equation if $[A, B]_D$ factors below degree D

A variant

- What happens with a finite field given by:

$$X - \frac{h_0(X^q)}{h_1(X^q)} \quad \text{i.e.} \quad h_1(X^q)\,X - h_0(X^q)?$$

- In particular, nothing changes for degree $H = 1$

Properties of $[A, B]_D$

- For A and B polynomials of degree D over $\mathbb{F}_{q^t}$.
- $[A, B]_D = -[B, A]_D$.
- $[A, A]_D = 0$.
- For $\lambda \in \mathbb{F}_q$: $[\lambda A, B]_D = [A, \lambda B]_D = \lambda [A, B]_D$.
- For $\Lambda \in \mathbb{F}_{q^t}$: $[\Lambda A, \Lambda B]_D = \Lambda^{q+1} [A, B]_D$.
- $[A, B_1 + B_2]_D = [A, B_1]_D + [A, B_2]_D$.

Counting the candidate equations over $\mathbb{F}_q$

- For A and B polynomials of degree D over $\mathbb{F}_q$?
- Tricky, because some equations are identicals (or even trivial).
- A and B may be supposed monic.
- $[A, B]_D = [A, B - A]_D$.
- Restrict to A of degree D, B of degree $D - 1$.
- $[A, B]_D = [A - \lambda B, B]_D$.
- Assume coeff of X^{D-1} in A is zero
- q^{2D-2} choices:

$$A = X^D + a_{D-2}(X) \quad \text{and} \quad B = X^{D-1} + b_{D-2}(X).$$

Counting the candidate equations over $\mathbb{F}_{q^t}$

- More complex !
- A may still be supposed monic.
- Only one dimension of coefficient in B is zero.
- Then one other dimension of coefficient in B is one.
- With a corresponding zero in A
- $q^{(2D+1)t-3}$ choices.

Analyzing the parameters

- Smoothness basis : pols of degree D over $\mathbb{F}_{q^t}$.
- Number of unknowns $\approx q^{tD}/D$.
- Number of candidate equations: $\approx q^{(2D+1)t-3}$
- If H is fixed, a constant fraction is kept.
- Asymptotically we want:

$$(2D+1)t - 3 > tD \quad \text{i.e.} \quad t(D+1) > 3.$$

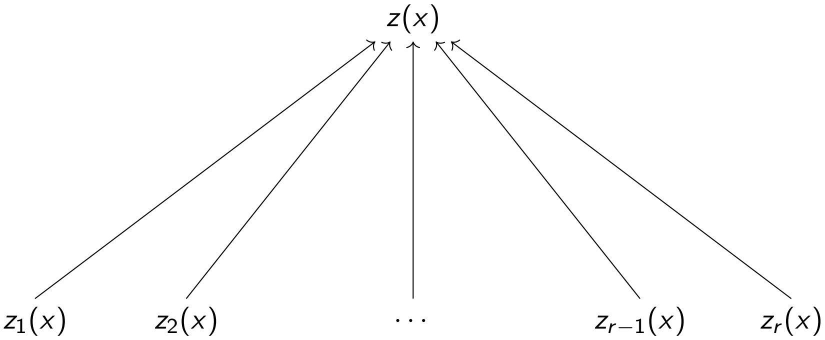
- Note, this only suffices for the initial computation.
- Smallest options: $D = 1, t = 2$ or $D = 3, t = 1$

The descent

General principle

- Given target $z(x)$ in finite field, write:

$$z(x) = \prod_i z_i(x)^{e_i}, \quad \text{with smaller } z_i\text{s}$$



Individual Logarithms (Descent strategies)

- Continued fractions (high degrees)
- Classical descent (for high to mid degrees, need subfield)
- Bilinear descent (for mid to low degrees)
- Quasi-polynomial descent (all degrees)
- ZigZag descent (all even degrees)

Continued fractions

- Given target $Z(x)$ find matrix:

$$\begin{pmatrix} A_1(x) & A_2(x) \\ B_1(x) & B_2(x) \end{pmatrix}, \text{such that}$$

$$Z(x) \equiv \frac{A_1(x)}{B_1(x)} \equiv \frac{A_2(x)}{B_2(x)} \pmod{I(x)}.$$

- With continued fraction or half-Gcd algorithms.
- Reduce degree by factor ≈ 2. Many representations:

$$Z(x) \equiv \frac{c_1(x)A_1(x) + c_2(x)A_2(x)}{c_1(x)B_1(x) + c_2(x)B_2(x)} \pmod{I(x)}.$$

Classical descent

- Need two variables x and y
- If $q = p^\ell$, let:

$$\begin{aligned} y &= x^{p^{\ell_1}} \qquad \text{then} \\ y^{p^{\ell_2}} &= x^{p^\ell} = \frac{h_0(x)}{h_1(x)}. \end{aligned}$$

- Let $F(x, y)$ be a (low degree) bivariate polynomial in $\mathbb{F}_q[x, y]$, then:

$$F(x, x^{p^{\ell_1}})^{p^{\ell_2}} = F(x^{p^{\ell_2}}, h_0(x)/h_1(x)) \quad \text{in } \mathbb{F}_{q^k}.$$

- Force $z(x)$ as divisor of $F(x, x^{p^{\ell_1}})$ or $F(x^{p^{\ell_2}}, h_0(x)/h_1(x))$ (linear algebra)
- Low arity in descent but can't go very low

Modern descent strategies

- Remember basic Equation:

$$[A, B]_D(\theta) = h_1(\theta)^D \cdot \prod_{\alpha \in \mathbb{P}_1(\mathbb{F}_q)} (A(\theta) - \alpha\, B(\theta)).$$

- Make $z(\theta)$ appear on the right or left
  - On the left: bilinear descent
  - On the right: quasi-polynomial
  - On the left (powers of two): ZigZag descent [GKZ14]

Bilinear descent

- Search for A and B of degree $\mathcal{D}$ such that:

$$z(x) | [A, B]_{\mathcal{D}}.$$

- Then $z(\theta)$ appears on the left.
- Arity $\approx q$ in descent

How to find A and B ?

- Algebraic approach : divisibility condition as a bilinear system
  - In general, use Groebner bases
  - For low-degree, it goes well.
- **Open problem:**
  Is there a more direct/efficient general approach ?
  Partial answer: Degree $2\mathcal{D}$ to degree $\mathcal{D}$ a.k.a ZigZag [GKZ14]

Quasi-polynomial descent

- Make $z(x)$ appear on the right in the term:

$$\prod_{\alpha\in\mathbb{P}_1(\mathbb{F}_q)} (A(\theta) - \alpha\, B(\theta))$$

- Choose $A(x) = z(x) + \alpha$ and $B(x) = x + \beta$
- Gives $\approx q^2$ equations.
- Simultaneous descent of all $z(x) + \lambda_1\, x + \lambda_0$
- Requires extra linear algebra step
- Arity q^2 in descent

Descent Tree

- Continued fractions, **at most one application**
- Classical descent, **many levels possible**
- Bilinear descent (or [GKZ14]), **in practice 4-5 levels max.**
- Quasi-polynomial descent **in practice 2 levels max.**

Practical bottleneck

- Is it possible to go all the way down to degree $\leq D$?
- In particular, descent should work for polys of deg $D+1$
- On the left. As before degree D over $\mathbb{F}_{q^t}$.
- Now we want:

$$(2D+1)t - 3 > t(D+1) \quad \text{i.e.} \quad tD > 3.$$

- Small options become: $D = 2, t = 2$ or $D = 4, t = 1$
- Polynomial time part can be lowered to $O(q^6)$.

Main theoretic difficulty

- Heuristics can be removed
(Granger, Kleinjung, Zumbragel, arxiv 2015)

- Except one, the existence of h_0 and h_1

Optimizing the polynomial time part

Systematic factors of $[A, B]_D$ over $\mathbb{F}_q$

- Definition polynomial
  $h_1(X)\,X^q - h_0(X)$ with $h_0 = rX + s$ and $h_1 = X(X + t)$.
- For $D = 2$, see that $[A, B]_2$ is a degree 6 polynomial.
- But systematic factor $Xh_1(X) - h_0(X)$.
- Indeed:

$$\begin{aligned}
[X^2, 1]_2 &= & h_0(X)^2 - X^2 h_1(X)^2 \\
[X, 1]_2 &= & h_0(X)h_1(X) - Xh_1(X)^2 \\
[X^2, X]_2 &= & Xh_0(X)^2 - X^2 h_0(X)h_1(X)
\end{aligned}$$

- Remaining degree $= 3$.
- Cost of linear algebra $O(q^5)$.

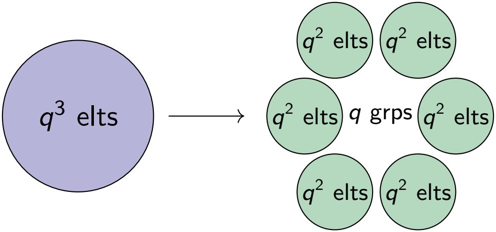
Descent bottleneck

- Can we get degree 3 polynomials ?
- From $[A, B]_2$, no ! At most $O(q^2)$ of $\approx q^3/3$.
- Direct approach would cost $O(q^7)$

Extend the Factor Base to Degree 3

Extend without performing linear algebra on a matrix of dim q^3 ?

1. Divide the deg. 3 monic polynomials into groups.

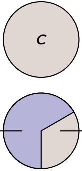


   How? Group polynomials by their constant coefficient.

2. Given q^2, generate equations involving only polys in q^2 and degree 1 and 2 polys (Logs are already known).

Extend the Factor Base to Degree 3

- An example: let (c) $= \{(X^3 + c) + \alpha X^2 + \beta X | (\alpha, \beta) \in {\mathbb{F}_q}^2\}$.



Reducible — — Irreducible $\Rightarrow$ new unknowns

As for degree 2: set $A(X) = (X^3 + c) + \alpha X^2$ and $B(X) = (X^3 + c) + \beta X$ and create relations of the form:

$$\underbrace{h_1(X)^3 B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{all belongs to (c) !!}} = \underbrace{[A, B]_3(X)}$$

deg 8 with these A and B
$+$ deg 3 systematic factor
$+$ divisible by X

Prob that $[A, B]_3$ factors into deg $\leqslant 2 \Rightarrow 41\%$. Enough !

- Complexity to recover the Dlogs of all degree 3 polynomials:

$O((\#\text{(c)})(\#\text{ factor base})^2(\#\text{ of entries})) \approx O(q^6)$ ops.

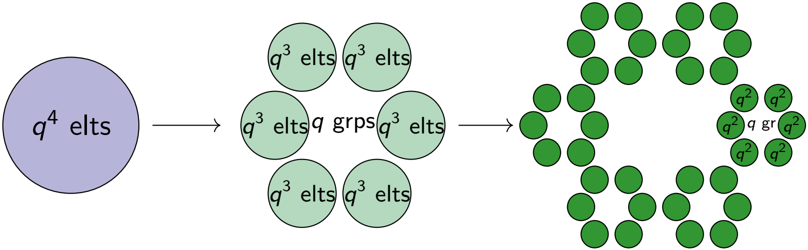
Descent bottleneck revisited

- Can we get degree 4 polynomials ?
- From $[A, B]_3$, may be ? At most $O(q^4)$ of $\approx q^4/4$.
- Unfortunately, only half of them are accessible.

Extend the Factor Base to Degree 4

Final goal: extend the factor base to degree 4

1. by performing smaller linear algebra steps.



What is simple ? To consider that:

2 poly belongs to the same (q^3) if same constant coefficient.

AND 2 poly belongs to the same (q^2) if same coeff before X.

2. Given (q^2), generate equations involving only poly in it and degree 1, 2 and 3 polynomials.

Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE $\textcircled{q^3}$:

$$O\big(\underbrace{(\#\ \textcircled{q^2}\ \text{in}\ \textcircled{q^3})}_{q} \cdot \underbrace{(\#\ \textcircled{q^2})}_{q^2}{}^2\ \underbrace{(\#\text{entries})}_{q}\big) = O(q^6)\ \text{ops.}$$

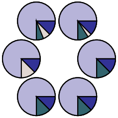
- Final complexity dominated by the first $\textcircled{q^3}$ computation:

Unknown

Reducible

Bili. desc. $4 \rightarrow 3$

Bili. desc. $4 \rightarrow 4$

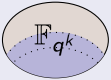


$\Rightarrow$ Final complexity of extension to deg 4 in $O(q^6)$ operations.

End Result

Final asymptotic complexity of the polynomial phase:

$O(q^6)$ operations – to be compared with previous $O(q^7)$.



Conclusion