Advances in Discrete Logarithm Computations

Antoine Joux

CryptoExperts
INRIA/Ouragan
Chaire de Cryptologie de la Fondation de l’UPMC — LIP6

School on Cryptographic Attacks, Porto, October 13–16, 2014
Discrete logarithms
Discrete logarithms

- Given a multiplicative group $G$ with generator $g$
- Computing discrete logarithms is inversing $n \rightarrow g^n$
- Hard in general and used as a hard problem in cryptography
- Algorithmic viewpoint
  - Generic algorithms (for any $G$)
  - Specific algorithms (make use of group representation)
Classical groups for Dlog in Cryptography

- Integers modulo $p$
- More general finite fields $\mathbb{F}_{p^k}$
- Elliptic curves over finite fields
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:

  $$G_i = \langle g^{G/p_i} \rangle \quad \text{(Group of order } p_i)$$
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 $y, y/g, \ldots, y/g^{R-1}$
  - Create list $1, h, h^2, \ldots, h^{R-1}$, where $h = g^R$
  - Find collision

- Can be improved to memoryless algorithms using cycle finding techniques
Index calculus algorithms

- Relation generation phase
  - Choose small subset $S \subset G$ of “small elements”
  - Sparse multiplicative relation: Sequence $(s_1, e_1), \cdots, (s_k, e_k)$ such that:
    \[
    \prod_{i=1}^{k} s_i^{e_i} = 1
    \]
  - Each gives a sparse linear equation:
    \[
    \sum_{i=1}^{k} e_i \log(s_i) = 0
    \]
  - Modulo group order for discrete log
Index calculus algorithms

• Linear algebra phase
  • Large sparse system
  • Numbers of unknowns in range up to dozens of millions
  • Number of equations potentially very large
  • Need to use large computers to solve such systems
  • Often the limiting phase for practical computations

• Individual logarithm phase
Complexity of Index calculus algorithms (before 2013)

\[ L_Q(\beta, c) = \exp((c + o(1))(\log Q)^\beta (\log \log Q)^{1-\beta}). \]
To represent the finite field $\mathbb{F}_{p^k}$

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 $y = f_2(x)$
Here, $S$ contains low degree polynomials in $X$ and $Y$. 
Discrete Logarithms, simplified FFS [JL06]

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

- Choose smoothness basis $S = \{x - \alpha, y - \alpha \mid \alpha \in \mathbb{F}_p\}$

- Consider elements:
  
  $xy + ay + bx + c$

  $xf_2(x) + af_2(x) + bx + c$

  $yf_1(y) + ay + bf_1(y) + c$

- When both sides split $\Rightarrow$ Relation

- Heuristic cost of finding relation (sieving):
  
  $(d_1 + 1)! \cdot (d_2 + 1)!$

- Individual log. descent negligible compared to initial phase
Linear change of variables [J13]

- Further restrict to \( y = x^{d_1} \)
- Then:

\[
xy + ay + bx + c = x^{d_1+1} + ax^{d_1} + bx + c
\]

- Perform change of variable: \( x = aX \), we get:

\[
a^{d_1+1}(X^{d_1+1} + X^{d_1} + b \cdot a^{-d_1}(X + c/(ab))).
\]

- Change of variable does not affect splitting property
- One good left-hand side \( \Rightarrow p - 1 \) good left-hand sides
- Amortized cost of relation reduced to

\[
\left( \frac{(d_1 + 1)!}{p - 1} + 1 \right) \cdot (d_2 + 1)!
\]
Linear change of variables [J13]

\[
xy + ay + bx + c
\]

\[
\begin{align*}
X^{d_1+1} + ax^{d_1} + bx + c \\
y f_1(y) + ay + bf_1(y) + c
\end{align*}
\]

\[
X^{d_1+1} + X^{d_1} + b \cdot a^{-d_1}(X + c/(ab))
\]
Small characteristic
Small characteristic – Setting

- Use basefield $\mathbb{F}_q$
- Define extension field by a relation:

$$x^q = \frac{h_0(x)}{h_1(x)} \quad \text{or} \quad x = \frac{h_0(x^q)}{h_1(x^q)},$$

$$\mathbb{F}_{q^k} = \mathbb{F}_q[\theta]$$

with $k = \deg(I(x))$ extension, where $I(x)$ is a divisor of $h_1(x)x^q - h_0(x)$ or $h_1(x^q)x - h_0(x^q)$. Let $\theta$ be a root of $I$.

- We have a systematic relation:

$$\prod_{\alpha \in \mathbb{F}_q} (x - \alpha) = x^q - x. \quad (1)$$
Small characteristic – Basic idea (simplified)

- Substitute $x$ by $\frac{A(\theta)}{B(\theta)}$ in (1) and multiply by $B(\theta)^{q+1}$:

$$B(\theta) \cdot \prod_{\alpha \in \mathbb{F}_q} (A(\theta) - \alpha B(\theta)) = B(\theta) \cdot A(\theta)^q - A(\theta) \cdot B(\theta)^q$$

- Moreover, after expanding the right-hand size, we find:

$$B(\theta)A \left( \frac{h_0(\theta)}{h_1(\theta)} \right) - A(\theta)B \left( \frac{h_0(\theta)}{h_1(\theta)} \right).$$

- Let $D$ be the maximum degree of $A$ and $B$ and define:

$$[A, B]_D(X) = h_1(X)^D \left( B(X)A \left( \frac{h_0(X)}{h_1(X)} \right) - A(X)B \left( \frac{h_0(X)}{h_1(X)} \right) \right).$$
Equation (1) after substitution can be rewritten as

\[ \prod_{\alpha \in P_1(F_q)} (A(\theta) - \alpha B(\theta)) = \frac{[A, B]_D(\theta)}{h_1(\theta)^D}. \] (2)

The degree of \([A, B]_D\) is \(\leq D \cdot (H + 1)\), where \(H = \max(\deg(h_0), \deg(h_1))\).

Thus relations between degree \(D\) polynomials can be found easily.
Properties and simplification of $[A, B]_D(X)$

- $[A, B]_D$ is bilinear
- $[A, A]_D = 0$.
- In Equation (2), $A$ and $B$ can be assumed monic.
- Since $[A, B]_D = [A, B - A]_D$, we may also assume $\deg B < \deg A$.
- Assume $\deg A = D$ and $\deg B = D - 1$. Then, using bilinearity, one may reduce the coefficient of $X^{D-1}$ in $A$ to 0.
- In the sequel, we assume:

\[
A(X) = X^D + A_{D-2}(X) \quad \text{and} \quad B(X) = X^{D-1} + B_{D-2}(X).
\]
Small characteristic – Choice of $D$

- If $D = 0$ then $A$ and $B$ are constants, thus $[A, B]_0 = 0$.
- If $D = 1$ then $A = X$ and $B = 1$ is the only choice.
- If $D = 2$ then $A = X^2 + \alpha$ and $B = X + \beta$: $q^2$ candidates
- If $D = 3$ then $A = X^3 + \alpha_1 X + \alpha_0$ and $B = X^2 + \beta_1 X + \beta_0$: $q^3$ candidates
- Cost of Linear algebra is at least $O(q^{2D+1})$.

⇒ Logs of Degree $\leq D$ Polynomials in $\theta$
Individual Logarithms a.k.a. Descent
Descent strategies (Higher degree polynomial)

- 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)
**General principle**

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

\[
z(x) = \prod_{i} z_i(x)^{e_i}, \quad \text{with smaller } z_i \text{s}
\]
Continued fractions

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

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

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

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

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

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

\[
\begin{align*}
y &= x^{p^{\ell_1}} \quad \text{then} \\
y^{p^{\ell_2}} &= x^{p^\ell} = \frac{h_0(x)}{h_1(x)}.
\end{align*}
\]

- 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
New descents

- Remember Equation (2):

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

- Make \( z(x) \) appear on the right or left
Bilinear descent

- Search for $k_1$ and $k_2$ such that:

\[ z(x) \mid [k_1, k_2]_D(x) \]

- Then $z(x)$ appears on the right in Equation (2).
- Arity $\approx q$ in descent
How to find $k_1$ and $k_2$?

- Algebraic approach: divisibility condition as a bilinear system
  - In general, use Groebner bases
  - For low-degree, it degenerates into easy linear algebra

- **Open problem:**
  Is there a more direct/efficient general approach?
  
  *Partial answer:* Degree $2D$ to degree $D$ a.k.a ZigZag [GKZ14]
Quasi-polynomial descent

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

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

- 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.**
Complexities of Index Calculus Algorithms

\[ L_Q \left( \frac{1}{3}, \downarrow \right) \]

Quasi-Polynomial FFS

NFS and variants

Discrete logarithms
Small characteristic
Individual Logarithms a.k.a. Descent

Complexities of Index Calculus Algorithms

\[ L_Q (\alpha + o(1)) \]

when \( p = L_Q(\alpha) \)

\[ L_Q \left( \frac{1}{3} \right) \]

\[ L_Q \left( \frac{1}{3}, \downarrow \right) \]

\[ L_Q \left( \frac{1}{3}, \downarrow \right) \]
Conclusion

Questions ?
How to find $k_1$ and $k_2$?

- **Algebraic approach**: divisibility condition as a bilinear system
  - In general, use Groebner bases
  - For low-degree, it degenerates into easy linear algebra

- **Lattice reduction approach**:
  - Further assume that $k_1$ and $k_2$ split into linear term
  - Since $z(x)$ is irreducible, it encodes a finite field
  - Take logarithms of elements:
    \[
    \frac{x - \alpha}{h_0(x)/h_1(x) - \alpha}, \quad \text{for } \alpha \in \mathbb{F}_q.
    \]

- Find low weight sum of logarithms equal to 0

- **Open problem**:
  Is there a more direct/efficient general approach?
  Partial answer: Degree $2D$ to degree $D$ [GKZ14]