CSE291-14: The Number Field Sieve

https://cseweb.ucsd.edu/classes/wi22/cse291-14

Emmanuel Thomé

March 1st, 2022

Part 8a

Discrete logarithms in finite fields

Black-box algorithms for discrete logarithm

Index Calculus in $\mathbb{F}_p$

Building the diagram

What we would like to do with relations

Plan

Black-box algorithms for discrete logarithm

Index Calculus in $\mathbb{F}_p$

Building the diagram

What we would like to do with relations

The discrete logarithm problem

Definition of DLP in a cyclic group

In a cyclic group G, the Discrete Logarithm Problem (DLP) is, given a generating element g and a target h, to find x such that

$$g^x = h.$$

The solution $x \overset{\text{def}}{=} \log_g h$ is defined modulo the order of the group.

Many cryptographic primitives can be based on a cyclic group (DSA, ElGamal, ...). Their security relies on the hardness of DLP in the underlying group.

- Some groups have trivially easy DLP, and must not be used for cryptography.
- DLP is often harder. (FF, EC: no poly-time algorithm known).

Pohlig–Hellman

Let G be a cyclic group of size $n = \prod_{i=1}^{k} \ell_i^{e_i}$.

Theorem (Pohlig–Hellman): It is possible to compute a discrete logarithm in G by computing e_i discrete logarithms in groups of size ℓ_i, for $1 \leq i \leq k$.

Tools: Elementary group theory (to deal with prime powers) and Chinese Remainder Theorem (to combine results from different primes).

Asymptotically, computing a discrete logarithm in a cyclic group of order n is (up to a polynomial factor) as hard as computing a discrete logarithm with a black box algorithm in a cyclic group of order ℓ, where ℓ is the largest prime factor of n.

$\Rightarrow$ For cryptography, do not use groups with smooth order.

Pollard ρ for discrete logarithm

Let $h = g^x$. Computing the discrete logarithm x can be done with Pollard ρ algorithm by looking for an equality of the form

$$g^{\alpha_1} h^{\beta_1} = g^{\alpha_2} h^{\beta_2}.$$

Then, x is a solution of the equation $(\alpha_2 - \alpha_1)x = \beta_1 - \beta_2$ modulo the group order.

The "random" function must be defined so that it is possible to keep track of the exponent of g and h.

Example of a "random" function:

$$f(t) = \begin{cases} ht, & \text{for } t \in G_0 \\ t^2, & \text{for } t \in G_1 \\ gt, & \text{for } t \in G_2 \end{cases},$$

where $G_0 \cup G_1 \cup G_2$ is a partition of the group G.

Complexity: $O(\sqrt{\#G})$.

Multiple variants of Pollard ρ

Following the same basic idea, Pollard ρ has several known variations, notably “Parallel collision search”.

This is part of the old algorithmic number theory folklore, and is still the state of the art for problems such as ECDLP.

Most of the improvements on this in the last 2 decades or so have been about the possibility of winning a constant factor below 2, or about usefully employing platform X for this problem (X=FPGA, GPU, . . .).

Baby-step Giant-step algorithm

Let $h = g^x$. Write $x = iM + j$ for a chosen integer M, with $0 \leq i \leq \#G/M$ and $0 \leq j \leq M$.

Goal: find i and j such that $h(g^{-M})^i = g^j$.

Algorithm:

- compute $\gamma = g^{-M}$.
- Baby steps: compute $S = \{g^j \mid 0 \leq j \leq M\}$.
- Giant steps: for $0 \leq i \leq \#G/M$, compute $h\gamma^i$ and stop if it is in S.

Complexity: $O(\sqrt{\#G})$ (deterministic, proven).

- if M is chosen to be $\lceil \sqrt{\#G} \rceil$
- if the test "is in S" is done in $O(1)$ (e.g., with hash tables)

Shoup's theorem

Let G be a cyclic group of prime order ℓ.

Shoup's theorem: any "generic" algorithm that solves the discrete logarithm problem in G must perform at least $\Omega(\sqrt{\ell})$ group operations.

"generic" means that the algorithm has access to the group structure only *via* to two oracles: one for performing group operations and one for testing for equality in the group.

In practice, we always have access to much more information on the group !

So there is still hope to find better algorithms than Pollard ρ and BSGS for specific groups.

Similarity with factoring

Note that we have analogies:

- Pollard ρ for DL $\leftrightarrow$ Pollard ρ for factoring.
- BS/GS for DL $\leftrightarrow$ Pollard-Strassen.

As we will see, this carries over to the index calculus setting.

Plan

Black-box algorithms for discrete logarithm

Index Calculus in $\mathbb{F}_p$

Building the diagram

What we would like to do with relations

Plan

Index Calculus in $\mathbb{F}_p$

- $L(1/2)$ algorithm
- $L(1/3)$ algorithm: NFS-DL

Index Calculus: algorithm for $\mathbb{F}_p$

Lots of common points with Dixon's algorithm.

Based on Kraitchik's idea of combination of congruences. Formalized in the 1970's. Proven $L_p(1/2)$.

Let $h = g^x$, where g is a generator of (a subgroup of) $\mathbb{F}_p^\times$.

- We are interested in the factorization of $h \times g^i \bmod p$ (seen as an integer) only if it is smooth.
- We fix a smoothness bound B.

A relation is interpreted as an equality between the unknown x and the (unknown) logarithms of the elements of the factor base:

$$\begin{aligned} h \times g^i &\equiv p_1^{e_{i,1}} \times \cdots \times p_k^{e_{i,k}} \pmod p \\ x + i \log_g(g) &\equiv e_{i,1} \log_g p_1 + \cdots + e_{i,k} \log_g p_k \pmod{p-1} \end{aligned}$$

Index Calculus: algorithm for $\mathbb{F}_p$

Algorithm:

- Pick i at random. Test divisibility by all primes below B.
  If $h \times g^i \bmod p$ is B-smooth, keep the relation:

$$h \times g^i \equiv p_1^{e_{i,1}} \times \cdots \times p_k^{e_{i,k}} \pmod{p}.$$

- Solve the linear system to find x (and the logarithms of all elements of the factor base that appear in at least one relation).
  This is a linear algebra problem modulo (a factor of) $p - 1$.

Index Calculus: example

Example with $p = 107$, $\ell = 53$, $g = 2$ and $h = 43$.

$$h \times g^{12} \bmod 107 = 6 = 2 \times 3$$
$$h \times g^{22} \bmod 107 = 45 = 3^2 \times 5$$
$$h \times g^{36} \bmod 107 = 50 = 2 \times 5^2$$
$$h \times g^{46} \bmod 107 = 54 = 2 \times 3^3.$$

$$\Longrightarrow \begin{pmatrix} -1 & 1 & 1 & 0 \\ -1 & 0 & 2 & 1 \\ -1 & 1 & 0 & 2 \\ -1 & 1 & 3 & 0 \end{pmatrix} \begin{pmatrix} x \\ \log_g(2) \\ \log_g(3) \\ \log_g(5) \end{pmatrix} = \begin{pmatrix} 12 \\ 22 \\ 36 \\ 46 \end{pmatrix}$$

Solving the linear system modulo $\ell = 53$ gives $x \equiv 6 \mod \ell$.
Indeed, $2^6 \equiv \pm 43 \mod p$.

We don't care about the 2-part!

Note that the previous example (on purpose) is not helpful to tell apart the cases $\log_g h = 6$ and $\log_g h = 6 + 53$.

- This is expected, as we're only trying to illustrate our capacity to solve the problem modulo large (prime) factors of $p - 1$.
- There are always easy ways to find out the discrete logarithm modulo other small factors.
  Extreme case modulo 2: the Legendre symbol.
- Pohlig-Hellman/CRT can be used to combine the information (possibly from various sources) and obtain "the" logarithm.

Main steps of index calculus

This (very basic) index calculus method goes through the following steps.

- Collect relations.
- Solve linear system modulo ℓ.

Complexity: $L_p(1/2)$, proven (like Dixon's algorithm).

Next: how do we use a Number Field Sieve-like setup?

Plan

Index Calculus in $\mathbb{F}_p$

- $L(1/2)$ algorithm
- $L(1/3)$ algorithm: NFS-DL

Setting from here on

- a finite field $\mathbb{F}_q$,
  - Our main case of interest is $q = p$ prime.
  - But $q = p^k$ a prime power sometimes.
- a prime factor ℓ of $q - 1$, without multiplicity.
  This technicality implies that the ℓ-th roots of unity in $\mathbb{F}_q^\times$ can be used as representatives for the quotient group $(\mathbb{F}_q^\times)/(\mathbb{F}_q^\times)^\ell$.

Goal

We want to find a non-trivial discrete logarithm map

$$\mathbf{L} : \begin{cases} \mathbb{F}_q^\times & \to & \mathbb{Z}/\ell\mathbb{Z}, \\ x & \mapsto & \mathbf{L}(x). \end{cases}$$

with $\operatorname{Ker} \mathbf{L} = (\mathbb{F}_q^\times)^\ell$.

Setting from here on

Goal

We want to find a non-trivial discrete logarithm map

$$\mathbf{L} : \begin{cases} \mathbb{F}_q^\times & \rightarrow & \mathbb{Z}/\ell\mathbb{Z}, \\ x & \mapsto & \mathbf{L}(x). \end{cases}$$

with $\operatorname{Ker}\mathbf{L} = (\mathbb{F}_q^\times)^\ell$.

There is more than one solution to this problem.

- Any two solutions are proportional, since $(\mathbb{F}_q^\times)/(\mathbb{F}_q^\times)^\ell$ is a 1-dimensional vector space.
- Each can be linked to the logarithm in some base, but we don't really have to care, as $\log_a b \equiv \mathbf{L}(b)/\mathbf{L}(a) \mod \ell$.
- Given $\mathbf{L}$, we can find out the corresponding base easily.

The diagram

The diagram for factoring looked like:

$$\begin{tikzcd}
 & & \mathbb{Z}[x] \arrow[dl, "x \to m"'] \arrow[dr, "x \to \alpha"] & & \\
\text{subring of } \mathbb{Q} & \mathbb{Z}[m] \subset \mathbb{Q} \arrow[dr, "\bmod N"'] & & \mathbb{Z}[\alpha] \arrow[dl, "\alpha \to m"] & \text{subring of } \mathbb{Q}(\alpha) \\
 & & \mathbb{Z}/N\mathbb{Z} & &
\end{tikzcd}$$

Some adaptation work ahead

Several differences.

- The structure below should be $\mathbb{F}_q$, or even better, a subgroup of $\mathbb{F}_q^\times$.
- It might be useful to write down something with only group morphisms throughout the diagram. (not rings)
- We know that $\mathbb{F}_q$ is field. We can do more things in a field than in $\mathbb{Z}/N\mathbb{Z}$.

Write something multiplicative

$$\begin{tikzcd}
 & \begin{array}{c}\text{group generated by}\\ \mathbb{Z}[x]-\langle p, x-m\rangle\end{array} \arrow[dl, "x \to m"'] \arrow[dr, "x \to \alpha"] & \\
\text{subgroup of } \mathbb{Q}^{\times} \arrow[dr, "\text{mod } p"'] & & \text{subgroup of } \mathbb{Q}(\alpha)^{\times} \arrow[dl, "\alpha \to m"] \\
 & \mathbb{F}_p^{\times} &
\end{tikzcd}$$

Write something multiplicative

$$\begin{tikzcd}
 & \begin{array}{c}\text{group generated by}\\ \mathbb{Z}[x]-\langle p, x-m\rangle\end{array} \arrow[dl, "x \to m"'] \arrow[dr, "x \to \alpha"] & \\
\begin{array}{c}\text{subgroup of } \mathbb{Q}^\times\\ \text{mod } \ell\text{-th powers}\end{array} \arrow[dr, "\text{mod } p"'] & & \begin{array}{c}\text{subgroup of } \mathbb{Q}(\alpha)^\times\\ \text{mod } \ell\text{-th powers}\end{array} \arrow[dl, "\alpha \to m"] \\
 & (\mathbb{F}_p^\times)/(\mathbb{F}_p^\times)^\ell &
\end{tikzcd}$$

- Goal 1: build the diagram.
  - How do we create the two sides?
  - Generalization to two number fields.
- Goal 2: understand it and use it.
  - How do we make sense of relations?
  - How do we formulate the linear algebra problem?
- Goal 3: how do we compute discrete logarithms in $\mathbb{F}_p$?

Plan

Black-box algorithms for discrete logarithm

Index Calculus in $\mathbb{F}_p$

Building the diagram

What we would like to do with relations

Polynomial selection methods

Easy starting point: any polynomial selection methods that works for N (for factoring) will work for p (for DLP in $\mathbb{F}_p$).
Several computations actually used that in the past.

Rephrasing of the polynomial selection problem:

Find two polynomials f_0, f_1 with a common known root in the target ring (here, $\mathbb{F}_p$).

- HUGE difference with the $\mathbb{Z}/N\mathbb{Z}$ case: we can find roots of polynomials in $\mathbb{F}_p$ in polynomial time.
- Can we use that to find a better pair of polynomials?

Joux-Lercier polynomial selection (2003)

Let d be a target degree.

Algorithm:

- Pick any irreducible polynomial f_0 of degree $d+1$ and very small coefficients (good for NFS).
- Test if f_0 has a root m modulo p (and find it).
- There are polynomials f_1 of degree $\leq d$ that also have m as a root modulo p.
  - f_0 is NOT one of them. (degree too large!)
  - The set of such polynomials f_1 is a lattice. Find a small (irreducible) one.
  - Any solution will be coprime with f_0 over $\mathbb{Z}$.

Joux-Lercier polynomial selection (2003)

```
Input: $p$ prime, degree $d$
Output: $f_0, f_1, m$ with $f_0, f_1 \in \mathbb{Z}[x]$ irreducible of degrees
        $d+1, d,\ f_0(m) = f_1(m) = 0 \mod p$
repeat
    Choose $f_0$ of degree $d+1$ and tiny coefficients, irreducible
     in $\mathbb{Z}[x]$ and having a root $m$ modulo $p$
    LLL $\left.\begin{bmatrix} p & & & \\ -m & 1 & & \\ & \ddots & \ddots & \\ & & -m & 1 \end{bmatrix}\right\}_{d+1} = \begin{bmatrix} c_0 & c_1 & \cdots & c_d \\ & & * & \\ & & & \end{bmatrix}$
    $f_1 \leftarrow \sum_{i=0}^{d} c_i x^i$
until $f_1$ is irreducible in $\mathbb{Z}[x]$
return $(f_0, f_1, m)$
```

Joux-Lercier polynomial selection (2003)

Joux-Lercier polynomial selection creates two number fields of degrees $d+1$ and d.

- It is certainly not a big deal as far as building the diagram goes.
  - Both f_0 and f_1 have the root m modulo p.
  - Number field $K_0 = \mathbb{Q}(\alpha_0)$. Map to $\mathbb{F}_p$ sends α_0 to m mod p.
  - Number field $K_1 = \mathbb{Q}(\alpha_1)$. Map to $\mathbb{F}_p$ sends α_1 to m mod p.
- Defining relations as simultaneous occurrences of $\operatorname{Res}(\cdot, f_0)$ and $\operatorname{Res}(\cdot, f_1)$ being smooth is easy as well.
- We'll need need to be somewhat more formal in order to properly make sense of relations.

Joux-Lercier polynomial selection

Search is pretty simple.

- Pick good-looking f_0
  (upper bound on coefficients, upper bound on $\alpha(f_0)$, ...).
- Run LLL, see the quality of the resulting f_1
  (metrics: size of coefficients, and $\alpha(f_1)$, ...).
- Break ties with some sample sieving.

Many unexplored things:

- There's no ultra-specialized algorithm like Kleinjung's algorithms.
- Our real focus of interest is the infinity (max-) norm, while lattice reduction gives a Euclidean short vector. Any possible improvement here, no idea.

Joux-Lercier polynomial selection in practice

As a rule of thumb, for identical bit size:

- If NFS-factoring would like a $(\deg f_0 = 1, \deg f_1 = d)$ pair, with even $d = 2d'$, then a JL pair with $(1 + d', d')$ would be a better choice at this size (for NFS-DL).
- For odd d, the JL construction puts us further from the optimum, so factoring-like polynomial selection can win.

Example for 240 decimal digits:

- Factoring (RSA-240): $(\deg f_0 = 1, \deg f_1 = 6)$.
- DLP (DLP-240, current record): $(\deg f_0 = 4, \deg f_1 = 3)$.

Plan

Black-box algorithms for discrete logarithm

Index Calculus in $\mathbb{F}_p$

Building the diagram

What we would like to do with relations

Fast forward to after relation collection

Relation collection works exactly the same way.

Of course, we want to keep track valuations in $\mathbb{Z}$, and no longer reduce them modulo 2.

A relation is. . .

- $a - b\alpha_0$ factors into small prime ideals in $\mathcal{O}_{K_0}$.

$$(a - b\alpha_0)\mathcal{O}_{K_0} = \prod_i \mathfrak{p}_i^{e_i}.$$

- $a - b\alpha_1$ factors into small prime ideals in $\mathcal{O}_{K_1}$.

$$(a - b\alpha_1)\mathcal{O}_{K_1} = \prod_j \mathfrak{q}_j^{g_j}.$$

- and both map to the finite field element $(a - bm) \in \mathbb{F}_p^\times$.

Why is it not a trivial question?

$$(a - b\alpha_0)\mathcal{O}_{K_0} = \prod_i \mathfrak{p}_i^{e_i} \qquad \text{and} \qquad (a - b\alpha_1)\mathcal{O}_{K_1} = \prod_j \mathfrak{q}_j^{g_j}.$$

- Can we turn this into an additive relation involving logarithms?

$$\sum_i e_i \mathbf{L}(\text{image of } \mathfrak{p}_i) \text{ ``}{=}\text{'' } \sum_j g_j \mathbf{L}(\text{image of } \mathfrak{q}_j).$$

- Main problem: ideals are not elements. No image in $\mathbb{F}_p$!!

An idealized approach

Assume that we know the class number and the unit group in both number fields. This is totally unrealistic!

Rosy setup

- Class number h_0 in K_0 is such that for any ideal $\mathfrak{p}$ in $\mathcal{O}_{K_0}$, the ideal $\mathfrak{p}^{h_0}$ is a principal ideal. Same on the other side.
  Small note: we may safely assume that ℓ and h_0 are coprime.
- Knowledge of the unit group: any unit can be rewritten as a combination of some known generators.

Approach:

- For all ideals, compute a generator $\gamma_\mathfrak{p}$ of the ideal $\mathfrak{p}^{h_0}$.
- Then all valuations of $(a - b\alpha_0)^{h_0} / \prod_i \gamma_{\mathfrak{p}_i}^{e_i}$ cancel. This is a unit!
- Decompose this unit w.r.t. the generating system.

An idealized approach

The idealized approach rewrites $(a - b\alpha_0)$ and $(a - b\alpha_1)$ as products of elements.

$$(a - b\alpha_0)^{h_0} = u_1^{\text{something}} \times \cdots \times u_{\text{something}}^{\text{something}} \times \prod_i \gamma_{\mathfrak{p}_i}^{e_i}.$$

- These elements form a generating system of the set of $(a - b\alpha_0)$ we are considering, and we have an explicit decomposition.
- We can map each of these elements (u_i and $\gamma_{\mathfrak{p}_i}$) to $\mathbb{F}_p$, and write a linear relation involving $\mathbf{L}$(only elements of $\mathbb{F}_p$).
- Alas, it is not practical.