MPRI – Cours 2.12.2

F. Morain

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA centre de recherche SACLAY - ÎLE-DE-FRANCE

Lecture V: solving DLP over finite fields

2010/10/05

The slides are available on `http://www.lix.polytechnique.fr/Labo/Francois.Morain/MPRI/2010`

I. Index-calculus: a general framework

(Western and Miller; Pollard, Adleman, Merkle, etc.)

Input: G in which primes exist ($\mathbb{F}_q^*$, hyperelliptic curves of large genus, etc.), $G = \langle g \rangle$, $n = \#G$, $h \in G$.

Output: x s.t. $h = g^x$.

Step 1: find the logarithms of the primes in $\mathcal{B} = \{p_1, p_2, \ldots, p_k\}$.

Step 2: look for b s.t. hg^b factors over $\mathcal{B}$:

$$hg^b = \prod_{j=1}^{k} p_j^{\alpha_j} \Leftrightarrow x + b \equiv \sum_{j=1}^{k} \alpha_j \log_g p_j \bmod n.$$

How do we find the $\log_g p_j$?

Choose random integers b_i for which

$$g^{b_i} = \prod_{j=1}^{k} p_j^{\alpha_{i,j}} \Leftrightarrow b_i \equiv \sum_{i=1}^{k} \alpha_{i,j} \log_g p_j \bmod n.$$

When enough relations have been gathered, solve the system.

Cost: $O(L_{\#G}[1/2, c])$ where c depends on G and/or $\#G$.

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

Adaptive and non-adaptive versions?

More adaptive:

$$g^{b_0} \prod_{i=1}^{r} h_i^{b_i} = \prod_{j=1}^{k} p_j^{\alpha_{i,j}}$$

and use linear algebra on the logs.
Depends on the value of r, etc.

Non-adaptive: perform Step 1 only and bet on $O((\log q)^c)$ queries?

Two strategies:

- non-adaptive versions;
- adaptive: precomputation phase only + improved Step 2.

The case of $\mathbb{F}_p$: Adleman's algorithm

This is the case $G = \mathbb{F}_p$ in the preceding:

Step 1: find the logarithms of the primes in $\mathcal{B} = \{p_1, p_2, \ldots, p_k\}$.

Step 2: look for b s.t. hg^b factors over $\mathcal{B}$:

$$hg^b = \prod_{j=1}^{k} p_j^{\alpha_j} \bmod p \Leftrightarrow x + b \equiv \sum_{j=1}^{k} \alpha_j \log_g p_j \bmod (p-1).$$

At the heart of the analyses of both steps is the probability

$$\frac{\psi(p, p_k)}{p}$$

Analysis

Prop. STEP1 costs $O(L(p)^{2+o(1)})$; STEP2 costs $O(L(p)^{3/2+o(1)})$.
Proof.
Proba(g^{b_i} is p_k-smooth)= $\dfrac{\psi(p,p_k)}{p}$ $\Rightarrow$ we need $k\frac{p}{\psi(p,p_k)}$ relations.
Trial division $\Rightarrow$ testing p_k-smoothness costs k divisions.
Linear algebra costs $O(k^r)$ with $2 \leq r \leq 3$ (see later).
Total cost:

$$O\left(k \cdot k\frac{p}{\psi(p,p_k)}\right) + O(k^r).$$

$k = L(p)^b \Rightarrow p_k \approx k \log k = O(L(p)^{b+o(1)})$

$$\Rightarrow O(L^{2b}L^{1/(2b)}) + O(L^{rb}) = O(L^{\max(2b+1/(2b), rb)}).$$

$2b + 1/(2b)$ minimal for $b = 1/2$ and has value $2 \geq rb$.

STEP2: $O(k\frac{p}{\psi(p,p_k)}) = O(L^{b+1/(2b)}) = O(L^{3/2})$. $\square$

II. Sieving techniques

Basic problem: find all $\mathcal{B}$-smooth numbers in $I = [1, X]$.

Naive approach: divide all numbers $x \in I$ by all primes in $\mathcal{B}$, with cost[1]

$$\sum_{x=1}^{X}\sum_{i=1}^{k} \mathrm{T}(x \text{ div } p_i) \approx Xk\log X.$$

Bernstein: general technique for any set, cost is $\approx X \log\log k$.

Using a sieve: on I, cost drops to $\approx X \log\log k$.

[1] assuming p_i fits in a word

Eratosthenes and beyond

Eratosthenes: how do we enumerate prime numbers in $[2, X]$?

It is much easier to enumerate composite numbers and then deduce the primes as being not composite.

	2**2**	3**3**	4	5**5**	6	7**7**	8	9	10
			2		**2**		**2**	**3**	**2**
					3				**5**
11**11**	12	13**13**	14	15	16	17**17**	18	19**19**	20
	2		**2**	**3**	**2**		**2**		**2**
	3		**7**	**5**			**3**		**5**

- empty sets denote primes;
- non-empty sets contain the prime factors of the composite number.

Eratosthenes: algorithm

$\mathcal{B} = \{p \leq \sqrt{X}\}$

Sieve:
1. Set $T[x] = \emptyset$ for $x \in [2, X]$; $p := 1$;
2. **repeat**
    2.1 detect next prime $> p$.
    2.2 $x := 2p$;
    2.3 **while** $x \leq X$ **do**
        2.3.1 $T[x] := T[x] \cup \{p\}$;
        2.3.2 $x := x + p$;
   until $p > \sqrt{X}$.

Postsieve: find all x s.t. $T[x] = \emptyset$.

Rem. replace 2.2 by $x := p^2$; 2.3.2 by $x := x + 2p$ as soon as $p > 2$. Some tricks are available (Brent, etc.).

Finding smooth numbers using a sieve (1/2)

Minor variant of Eratosthenes:

1. Set $T[x] = \emptyset$ for $x \in [2, X]$;
2. **for all** $p \in \mathcal{B}$ and $p^e \leq X$ **do**
    2.2 $x := p^e$;
    2.3 **while** $x \leq X$ **do**
        2.3.1 $T[x] := T[x] \cup \{p\}$; {*trick!*}
        2.3.2 $x := x + p^e$;

Postsieve: find all x s.t. $x = \prod_{p \in T[x]} p$.

Cost: (forgetting the p^e for $e > 1$)

$$\sum_{i=1}^{k} \frac{X}{p} \approx X \sum_{i=1}^{k} \frac{1}{p} \approx X \int_2^k \frac{dt}{t \log t} \approx X \log\log k \approx X \log\log p_k$$

Moreover: no division!

Finding smooth numbers using a sieve (2/2)

A numerical variant of Eratosthenes:

1. Set $T[x] = \emptyset T[x] = 0$ for $x \in [2, X]$;
2. **for all** $p \in \mathcal{B}$ and $p^e \leq X$ **do**
    2.2 $x := p^e$;
    2.3 **while** $x \leq X$ **do**
        2.3.1 $T[x] := T[x] \cup \{p\} T[x] := T[x] + \log p$; {*trick!*}
        2.3.2 $x := x + p^e$;

Postsieve: find all x s.t. $x = \prod_{p \in T[x]} p \log x \approx T[x]$.

Rem. Some tunning is necessary in practice and depending on applications.

Large primes

Classical postsieving: $T[x]$ contains the smooth part of x.

What next? We can try to factor $x/F(x)$ as $q_1 \cdot q_2 \cdots q_k$ with $B < q_i < B'$ and for some $k \leq K$ (1, 2, 3, 4, 5?).
We end up with relations

$$(x, F(x), q_1, \ldots, q_k)$$

and we wait for collisions. For instance:

$$(x_1, F(x_1), q_1) : (x_2, F(x_2), q_1) = (x_1/x_2, F(x_1/x_2)).$$

- $K = 1$: hashing is enough;
- $K = 2$: reduce to finding cycles in a graph.
- larger K: filtering.

Comment: using large primes do not change complexity analyses; but this is dramatic in practice.

III. Coppersmith/Odlyzko/Schroeppel (COS)

$$H = \lfloor \sqrt{p} \rfloor + 1, \quad J = H^2 - p$$

$$\mathcal{B} = \{q | q \text{ prime } < q_{\max}\} \cup \{H + c, 0 < c < c_{\max}\} \quad \ni g$$

Let $H + c_1$, $H + c_2$ in $\mathcal{B}$:

$$(H + c_1)(H + c_2) \equiv F(c_1, c_2) = J + (c_1 + c_2)H + c_1 c_2 \bmod p$$

or

$$\log_g(H + c_1) + \log_g(H + c_2) \equiv \sum_i f_i \log q_i \bmod (p - 1).$$

Goal: find enough equations to get $\log$ of elements in $\mathcal{B}$.

COS: algorithm

If we fix c_1, then

$$F(c_1, c_2) \equiv 0 \Longleftrightarrow c_2 \equiv -(J + c_1 H)(H + c_1)^{-1} \bmod q.$$

$\Rightarrow$ sieve!

For given $0 < c_1 < c_{\max}$:

1. Set $T[c_2] = \emptyset$ for $0 < c_2 < c_{\max}$;
2. **for all** $q \in \mathcal{B}$ and $q^e \le X$ **do**
   - 2.2 $c_2 = -(J + c_1 H)(H + c_1)^{-1} \bmod q^e$;
   - 2.3 **while** $c_2 \le c_{\max}$ **do**
     - 2.3.1 $T[c_2] := T[c_2] \cup \{q\}$; {*trick!*}
     - 2.3.2 $c_2 := c_2 + q^e$;

Postsieve: find all c_2 s.t. $c_2 = \prod_{q \in T[c_2]} q$ and store $(c_1, c_2, \{q\})$.

COS: analysis

Step 1:
With $k = L(p)^\beta$, $q_{\max} = L^\beta$, $c_{\max} = L^{\beta+\varepsilon}$, $|F(c_1, c_2)| = O(p^{1/2})$.

$$O\left(k \cdot k^0 \cdot \frac{p^{1/2}}{\psi(p^{1/2}, p_k)}\right) + O(k^r).$$

$$O(L^\beta L^{1/(4\beta)}) + O(L^{r\beta}) = O(L^{\max(\beta + 1/(4\beta), r\beta)}).$$

Minimum for $\beta = 1/2$ (giving L^1); linear algebra dominates anyway, so cost in $O(L^{r/2})$, better than L^2.

Step 2: if we use the plain version, we still have $L^{3/2}$, which we can improve.

COS: smoothing

Step 1: express $g^w h$ as a product of small and medium primes $u < L[2]$ in time $L[1/2]$

$$g^w h \equiv \left(\prod_i q_i^{e_i}\right) \times \left(\prod_i u_i^{f_i}\right);$$

Step 2: find $\log u$ for $u \in \{u_i\}$:

- find $y = \prod_i q_i^{m_i} \approx \sqrt{p}/u$ which is $L[1/2]$ smooth in an interval of length $L[1/2]$;
- find $v = H + c$ s.t. $vyu - p = \prod_i q_i^{n_i}$ is $L[1/2]$-smooth.
- compute

$$\log u \equiv \sum_i (n_i - m_i) \log q_i - \log(H + c).$$

Step 3: combine to find

$$\log h = -w + \sum_i e_i \log q_i + \sum_i f_i \log u_i.$$

Improved smoothing

Classical technique: write $g^i h = u/v$ where $u, v = O(\sqrt{p})$ using Euclid's algorithm; use medium primes + q-descent; use (1 or 2) large primes.

Joux/Lercier: see this as reducing the lattice

$$\begin{pmatrix} z & p \\ 1 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} A_1 & A_2 \\ B_1 & B_2 \end{pmatrix}$$

and remark that

$$g^i h \equiv z \equiv \frac{A_1}{B_1} \equiv \frac{A_2}{B_2} \equiv \frac{k_1 A_1 + k_2 A_2}{k_1 B_1 + k_2 B_2}$$

$\Rightarrow$ sieve on (k_1, k_2) and use 2 large primes.

See later specific NFS smoothing.

IV. The case of $\mathbb{F}_{2^n}$

Thm. $I(n, q) = \#\{f \in \mathbb{F}_q[X], f \text{ irreducible}\} \approx \frac{q^n}{n}$

Smoothness for polynomials:

$$N_q(n, m) = \#\{f \in \mathbb{F}_q[X], \deg(f) \leq n, g \mid f \Rightarrow \deg(g) \leq m\}$$

Thm. (Soundararajan) Let $u = n/m$ and assume $1 \leq m \leq n$. Uniformly for all prime powers $q \geq (n \log^2 n)^{1/m}$, we have

$$N_q(n, m) = \frac{q^n}{u^{(1+o(1))u}}$$

as $m, u \rightarrow \infty$.

Detecting smooth polynomials: (see Coppersmith) compute

$$g(X) = f'(X) \prod_{k \leq m} (X^{q^k} - X) \bmod f(X);$$

f is m-smooth iff $g \equiv 0$.

Coppersmith's algorithm (1/2)

$$\mathbb{F}_{2^{127}} = \mathbb{F}_2[X]/(f(X)) = \mathbb{F}_2[X]/(X^{127} + X + 1)$$

$\mathcal{B} = \{P_i(X), \text{irreducible}, \deg(P_i) \leq 17\}$. Consider

$$C(X) = X^{32} A(X) + B(X)$$

with A, B of degrees ≤ 10 (there are 2^{21} of them).

$$D(X) \equiv C(X)^4 \bmod f(X) \equiv (X^2 + X) A(X)^4 + B(X)^4 \bmod f(X),$$

where r.h.s. has degree $\leqslant 42$.
If $C(X)$ and $D(X)$ are smooth

$$D(X) = \prod_{i=1}^{\ell} P_i(X)^{e_i}, C(X) = \prod_{i=1}^{\ell} P_i(X)^{f_i}$$

then

$$\sum_{i=1}^{\ell} e_i \log P_i \equiv 4 \sum_{i=1}^{\ell} f_i \log P_i \mod (2^{127} - 1).$$

Coppersmith (2/2)

Thm. For $\mathbb{F}_{2^m}$, Coppersmith's algorithm is in $O(\exp(c m^{1/3} (\log m)^{2/3}))$.

Rem. Gordon & McCurley: smoothness testing of $(C(X), D(X))$ can be done using a sieve.

Current record: Joux & Lercier with $m = 613$ (in 2005).