MPRI – Cours 2.12.2

F. Morain

Lecture IV: Integer factorization

2010/09/28

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

I. Introduction.

II. Smoothness testing.

III. Pollard's rho.

IV. Pollard's $p-1$ method.

V. ECM.

I. Introduction

Input: an integer N;

Output: $N = \prod_{i=1}^{k} p_i^{\alpha_i}$ with p_i (proven) prime.

Major impact: estimate the security of RSA cryptosystems.

Also: primitive for a lot of number theory problems.

How do we test and compare algorithms? Cunningham project, RSA Security (partitions, RSA keys) – though abandoned?

What is the factorization of a random number?

$N = N_1 N_2 \cdots N_r$ with N_i prime, $N_i \geq N_{i+1}$.

Prop. $r \leq \log_2 N$; $\overline{r} = \log\log N$.

Size of the factors: $D_k = \lim_{N\to+\infty} \log N_k / \log N$ exists and

k	D_k
1	0.62433
2	0.20958
3	0.08832

"On average"

$$N_1 \approx N^{0.62}, \quad N_2 \approx N^{0.21}, \quad N_3 \approx N^{0.09}.$$

$\Rightarrow$ an integer has one "large" factor, a medium size one and a bunch of small ones.

II. Smoothness testing

Def. a B-smooth number has all its prime factors $\leq B$.

B-smooth numbers are the heart of all efficient factorization or discrete logarithm algorithms.

De Bruijn's function: $\psi(x, y) = \#\{z \leq x, z \text{ is } y-\text{smooth}\}$.

Thm. (Candfield, Erdős, Pomerance) $\forall\, \varepsilon > 0$, uniformly in $y \geq (\log x)^{1+\varepsilon}$, as $x \to \infty$

$$\psi(x, y) = \frac{x}{u^{u(1+o(1))}}$$

with $u = \log x / \log y$.

B-smooth numbers (cont'd)

Prop. Let $L(x) = \exp\left(\sqrt{\log x \log\log x}\right)$. For all real $\alpha > 0$, $\beta > 0$, as $x \to \infty$

$$\psi(x^\alpha, L(x)^\beta) = \frac{x^\alpha}{L(x)^{\frac{\alpha}{2\beta}+o(1)}}.$$

Ordinary interpretation:
a number $\leq x^\alpha$ is $L(x)^\beta$-smooth with probability

$$\frac{\psi(x^\alpha, L(x)^\beta)}{x^\alpha} = L(x)^{-\frac{\alpha}{2\beta}+o(1)}.$$

A) Trial division

Algorithm: divide $x \leq X$ by all $p \leq B$, say $\{p_1, p_2, \ldots, p_m\}$.

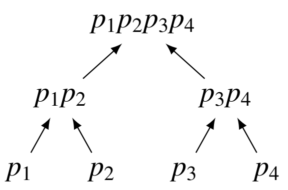
Cost: $\sum_{p \leq B} T(x,p) = O(m \lg X \lg B)$.

Implementation: use any method to compute and store all primes $\leq 2^{32}$ (one char per $(p_{i+1} - p_i)/2$; see Brent).

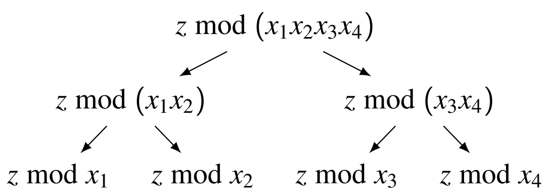
Useful generalization: given $x_1, x_2, \ldots, x_n \leq X$, can we find the B-smooth part of the x_i's more rapidly than repeating the above in $O(nm \lg B \lg X)$?

B) Product trees

Algorithm: Franke/Kleinjung/FM/Wirth improved by Bernstein
1. [Product tree] Compute $z = p_1 \cdots p_m$.



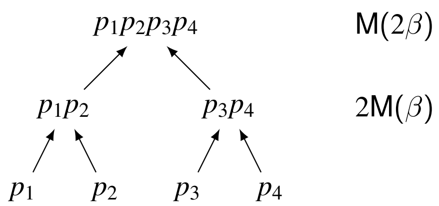
2. [Remainder tree] Compute $z \bmod x_1$, ..., $z \bmod x_n$.



3. [explode valuation] For each $k \in \{1, \ldots, n\}$, compute $y_k = z^{2^e} \bmod x_k$ with e s.t. $2^{2^e} \geq x_k$; print $\gcd(x_k, y_k)$.

Product trees (cont'd)

Imagine all p_i's have the same size β.



Product tree: $2\mathrm{M}(\beta) + \mathrm{M}(2\beta)$.

Naive case: $\underbrace{p_1p_2}_{\mathrm{M}(\beta)} + \underbrace{(p_1p_2)p_3}_{\mathrm{M}(2\beta,\beta)} + \underbrace{(p_1p_2p_3)p_4}_{\mathrm{M}(3\beta,\beta)} \approx 6\mathrm{M}(\beta)$.

Comparison: $4\mathrm{M}(\beta)$ vs. $\mathrm{M}(2\beta)$? Equal if $\mathrm{M}(\beta) = \beta^2$, product tree better if $\mathrm{M}(\beta) = \beta^a$, $a < 2$.

General principle: only the last step counts.

Validity and analysis

Validity: let $y_k = z^{2^e} \bmod x_k$. Suppose $p \mid x_k$. Then $\nu_p(x_k) \le 2^e$, since $2^\nu \le p^\nu \le 2^{2^e}$. Therefore $\nu_p(y_k) \ge 2^e \ge \nu$ and the $\gcd$ will contain the right valuation.

Division: If A has $r+s$ digits and B has s digits, then plain division requires $D(r+s, s) = O(rs)$ word operations.
In case $r \gg s$, break into r/s divisions of complexity $M(s)$.

Step 1: $M((m/2)\lg B)$.

Step 2: $D(m \lg B, n \lg X) \approx (m/n)M(n) \lg B \lg X$.

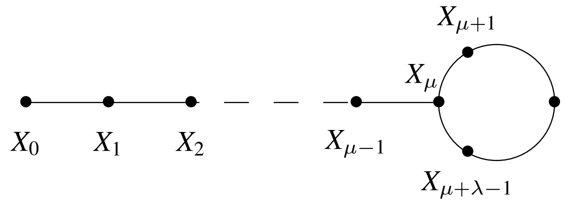
Ex. $B = 2^{32}$, $m \approx 1.9 \cdot 10^8$, $X = 2^{64}$.

Rem. If space is an issue, do this by blocks.

Rem. For more information see Bernstein's web page.

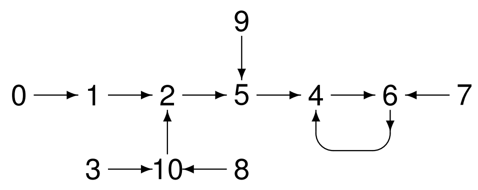
III. Pollard's ρ

Prop. Let $f : E \to E$, $\#E = m$; $X_{n+1} = f(X_n)$ with $X_0 \in E$. The functional digraph of X is:



Ex1. If $E_m = G$ is a finite group with m elements, and $a \in G$ of ordre N, $f(x) = ax$ and $x_0 = a$, (x_n) is purely periodic, i.e., $\mu = 0$, and $\lambda = N$.

Ex2. Soit $E_m = \mathbb{Z}/11\mathbb{Z}$, $f : x \mapsto x^2 + 1 \bmod 11$:



Epact

Thm. (Flajolet, Odlyzko, 1990) When $m \to \infty$

$$\overline{\lambda} \sim \overline{\mu} \sim \sqrt{\frac{\pi m}{8}} \approx 0.627\sqrt{m}.$$

Prop. There exists a unique $e > 0$ (epact) s.t. $\mu \le e < \lambda + \mu$ and $X_{2e} = X_e$. It is the smallest non-zero multiple of λ that is $\ge \mu$: if $\mu = 0$, $e = \lambda$ and if $\mu > 0$, $e = \lceil \frac{\mu}{\lambda} \rceil \lambda$.

Thm. $\overline{e} \sim \sqrt{\frac{\pi^5 m}{288}} \approx 1.03\sqrt{m}$.

Floyd's algorithm:

```
X <- X0; Y <- X0; e <- 0;
repeat
    X <- f(X); Y <- f(f(Y)); e <- e+1;
until X = Y;
```


Application to factorization of N

Idea: suppose $p \mid N$ and we have a random $f \bmod N$ s.t. $f \bmod p$ is "random".

```
function f(x, N) return (x^2 + 1) mod N; end.
function rho(N)
1. [initialization] x:=1; y:=1; g:=1;
2. [loop]
    while (g = 1) do
      x:=f(x, N); y:=f(f(y, N), N);
      g:=gcd(x-y, N);
    endwhile;
3. return g;
```

Conjecture. RHO finds $p \mid N$ using $O(\sqrt{p})$ iterations.

Thm. (Bach, 1991) Proba finding $p \mid N$ after k iterations is at least

$$\frac{\binom{k}{2}}{p} + O(p^{-3/2})$$

when p goes to infinity.

In practice:

- **Trick:** compute $\gcd(\prod_i (x_{2i} - x_i), N)$.
- **Choosing** f**:** some choices are bad, as $x \mapsto x^2$ et $x \mapsto x^2 - 2$. Tables exist for given f's.
- **Improvements:** reducing the number of evaluation of f, see Brent, Montgomery.

IV. Pollard's $p-1$ method

- Invented by Pollard in 1974.
- Williams: $p+1$.
- Bach and Shallit: Φ_k factoring methods.
- Shanks, Schnorr, Lenstra, etc.: quadratic forms.
- Lenstra (1985): ECM.

Rem. Almost all the ideas invented for the classical $p-1$ can be transposed to the other methods.

First phase

Idea: assume $p \mid N$ and a is prime to p. Then

$$(p \mid a^{p-1} - 1 \text{ and } p \mid N) \Rightarrow p \mid \gcd(a^{p-1} - 1, N).$$

Generalization: if R is known s.t. $p - 1 \mid R$,

$$\gcd((a^R \bmod N) - 1, N)$$

will yield a factor.

How do we find R**?** Only reasonable hope is that $p - 1 \mid B!$ for some (small) B. In other words, p is B-smooth.

Algorithm: $R = \prod_{p^\alpha \le B_1} p^\alpha = \mathrm{lcm}(2, \ldots, B_1)$.

Rem. (usual trick) we compute $\gcd(\prod_k ((a^{r_k} - 1) \bmod N), N)$.

Second phase: the classical one

Let $b = a^R \bmod N$ and $\gcd(b, N) = 1$.
Hyp. $p - 1 = Qs$ with $Q \mid R$ and s prime, $B_1 < s \le B_2$.

Test: is $\gcd(b^s - 1, N) > 1$ for some s.

$s_j = j$-th prime. In practice all $s_{j+1} - s_j$ are small (Cramer's conjecture implies $s_{j+1} - s_j \le (\log B_2)^2$).

- Precompute $c_\delta \equiv b^\delta \bmod N$ for all possible δ (small);
- Compute next value with one multiplication $b^{s_{j+1}} = b^{s_j} c_{s_{j+1}-s_j} \bmod N$.

Cost: $O((\log B_2)^2) + O(\log s_1) + (\pi(B_2) - \pi(B_1))$ multiplications $+(\pi(B_2) - \pi(B_1))$ gcd's. When $B_2 \gg B_1$, $\pi(B_2)$ dominates.

Rem. We need a table of all primes $< B_2$; memory is $O(B_2)$.

Record. Nohara (66 dd of $960^{119} - 1$, 2006; see http://www.loria.fr/%7Ezimmerma/records/Pminus1.html).

Second phase: BSGS

Select $w \approx \sqrt{B_2}$, $v_1 = \lceil B_1/w \rceil$, $v_2 = \lceil B_2/w \rceil$.

Write our prime s as $s = vw - u$, with $0 \le u < w$, $v_1 \le v \le v_2$. One has $\gcd(b^s - 1, N) > 1$ iff $\gcd(b^{wv} - b^u, N) > 1$.

1. Precompute $b^u \bmod N$ for all $0 \le u < w$.
2. Precompute all$(b^w)^v$ for all $v_1 \le v \le v_2$.
3. For all u and all v evaluate $\gcd(b^{vw} - b^u, N)$.

Number of multiplications is $w + (v_2 - v_1) + O(\log_2 w) = O(\sqrt{B_2})$, memory is also $O(\sqrt{B_2})$.

Number of gcd is still $\pi(B_2) - \pi(B_1)$.

Second phase: using fast polynomial arithmetic

Algorithm:
1. Compute $h(X) = \prod_{0 \le u < w}(X - b^u) \in \mathbb{Z}/N\mathbb{Z}[X]$
2. Evaluate all $h((b^w)^v)$ for all $v_1 \le v \le v_2$.
3. Evaluate all $\gcd(h(b^{wv}), N)$.

Analysis:

Step 1: $O((\log w)\mathsf{M}_{\mathrm{pol}}(w))$ operations (using a product tree).

Step 2: $O((\log w)\mathsf{M}_{\mathrm{int}}(\log N))$ for b^w; $v_2 - v_1$ for $(b^w)^v$; multi-point evaluation on w points takes $O((\log w)\mathsf{M}_{\mathrm{pol}}(w))$.

Rem. Evaluating $h(X)$ along a geometric progression of length w takes $O(w \log w)$ operations (see Montgomery-Silverman).

Total cost: $O((\log w)\mathsf{M}_{\mathrm{pol}}(w)) = O\left(B_2^{0.5+o(1)}\right)$.

Trick: use $\gcd(u, w) = 1$ and $w = 2 \times 3 \times 5 \ldots$.

Second phase: using the birthday paradox

Consider $\mathcal{B} = \langle b \bmod p \rangle$. By hypothesis, $\#\mathcal{B} = s$.
If we draw $\approx \sqrt{s}$ elements at random in $\mathcal{B}$, then we have a collision (birthday paradox).

Algorithm: build (b_i) with $b_0 = b$, and

$$b_{i+1} = \begin{cases} b_i^2 \bmod N & \text{with proba } 1/2 \\ b_i^2\, b \bmod N & \text{with proba } 1/2. \end{cases}$$

We gather $r \approx \sqrt{s}$ values and compute

$$\prod_{i=1}^{r} \prod_{j \neq i} (b_i - b_j) = \mathrm{Disc}(P(X)) = \prod_i P'(b_i)$$

where

$$P(X) = \prod_{i=1}^{r} (X - b_i).$$

$\Rightarrow$ use fast polynomial operations again.

V. ECM

- Due to Lenstra in 1985.
- Improvements: Chudnovsky & Chudnovsky; Brent; Montgomery; Suyama; Atkin-FM; etc.
- Powerful method since complexity depends on $p \mid N$: 30dd factors easy; record 73dd (2010), see `http://wwwmaths.anu.edu.au/~brent/ftp/champs.txt`.
- Reference implementation: GMP-ECM (P. Zimmermann); see Zimmermann & Dodson.

Pseudo-addition

Let $\gcd(4a^3 + 27b^2, N) = 1$ and

$$E_N = \{ (x, y, z),\ y^2 z \equiv x^3 + axz^2 + bz^3 \bmod N \} \cup \{ O_N \},$$

Reduction for $p \mid N$

$$\begin{array}{rccl} \pi_p : & E_N & \rightarrow & E_p \\ & O_N & \mapsto & O_p \\ & (x, y, z) & \mapsto & (x \bmod p, y \bmod p, z \bmod p). \end{array}$$

It is possible to define properly a group law on E_N (Bosma & Lenstra).

Or: add M_1 and M_2 as if N were prime and wait for something to happen.

Factoring with elliptic curves

Ex. Let $N = 143$. Consider $P = (0, 1, 1)$ on

$$E_N : y^2 \equiv x^3 + x + 1 \bmod N.$$

Computing $[3!]P$:

	P	$Q = [2]P$	$[2]Q$	$[2]Q \oplus Q = [6]P$
N	$(0, 1, 1)$	$(36, 124, 1)$	$(127, 71, 1)$	
11	$(0, 1, 1)$	$(3, 3, 1)$	$(6, 5, 1)$	$(0, 10, 1)$
13	$(0, 1, 1)$	$(10, 7, 1)$	$(10, 6, 1)$	$(0, 1, 0)$

From the last line, we add two opposite points mod 13 and

$$\lambda = (124 - 71) \times (36 - 127)^{-1} \bmod 143.$$

but the inverse leads to

$$\gcd(36 - 127, 143) = \gcd(52, 143) = 13.$$

Verification: $\#E_{11} = 14$ (resp. $\#E_{13} = 18 = 2 \times 3^2$); $ord(P_{11}) = 7$ (resp. $ord(P_{13}) = 6$).

The algorithm

```
procedure ECM(N, J)
1. d:=1;
2. choose random x0,y0,a in [0..N-1];
3. b:=(y0^2-x0^3-a*x0) mod N;
4. Delta:=gcd(4*a^3+27*b^2, N);
5. if Delta=N then goto 2; // bad luck!
6. if 1 < Delta < N then
      return Delta; // incredible luck!
7. P:=(x0,y0);
// we operate on E_N : y^2 = x^3 + ax + b mod N containing P
8. for j:=2..J do
      P:=[j]P;
      if some factor d is found then return d;
9. if d=1 then goto 2; // same player try again
```

Rem. the easiest way to have (E, P) is the one given, since we cannot compute $\sqrt{z}$ modulo N.

Analysis

Conj. (H. W. Lenstra, Jr.) ECM finds $p \mid N$ in average time $K(p)(\log N)^2$ where $K(x)$ is s.t.

$$\log K(x) = \sqrt{(2 + o(1)) \log x \log \log x}$$

when $x \rightarrow +\infty$.
Proof: (sketch)
ECM succeeds whenever $\#E_p$ divise $J!$ for some J.
Heuristically: $\#E_p \approx p \Rightarrow \#E_p$ behaves like a random number close to $p \Rightarrow$ proba $\#E_p \mid J! \approx \frac{1}{p}\psi(p, J)$.
Choosing $J = L(p)^\beta$ yields

$$\frac{1}{p}\psi(p, J) = L(p)^{-1/(2\beta)+o(1)}$$

$\Rightarrow$ we need $L(p)^{1/(2\beta)}$ elliptic curves.
Running time: computing $[J!]P$ is $O(J \log J) = O(L(p)^{\beta+o(1)})$ so total time is

$$O(L(p)^{\beta+1/(2\beta)+o(1)})$$

minimized for $\beta = 1/\sqrt{2}$. □

Complementary remarks

Claim: (Lenstra; Howe) proba that $\ell \mid \#E(\mathbb{Z}/p\mathbb{Z})$ is $> 1/\ell$.

For instance, for $\ell = 2$, (x, y) is of order 2 iff $y = 0$, hence look at roots of $x^3 + ax + b$, that can be 0, 1 or 3, hence in 2 cases out of 3.

Rem. Several parametrizations have been proposed, see the references.

A special property of elliptic curves: one can build $E/\mathbb{Q}$ having a prescribed rational torsion group (Mazur's theorem). As a consequence, $\#E_p$ will be divisible by small primes for all p's. Explicit families are known for 12, 16 (Suyama, Atkin-M.).