MPRI – cours 2.12.2

In order of apparition:

F. Morain, R. Barbulescu, B. Smith

morain@lix.polytechnique.fr

```
http://www.lix.polytechnique.fr/Labo/...
      .../Francois.Morain/MPRI/2016
```


I. Administrative details

Schedule, etc.

16 $\times$ 1.5 hour lectures:

- F. Morain (2 lectures): groups for cryptology.
- R. Barbulescu (4 lectures): factorization and discrete logarithms.
- B. Smith (10 lectures):(hyper)elliptic curves and pairings.

See official MPRI page for more details, including dates, labs, etc.

Internships:

- F. MORAIN: primality proving with polynomials (AKS, Jacobi Sums, etc.);
- B. SMITH: algebraic curves, point counting algorithms.

Expectations

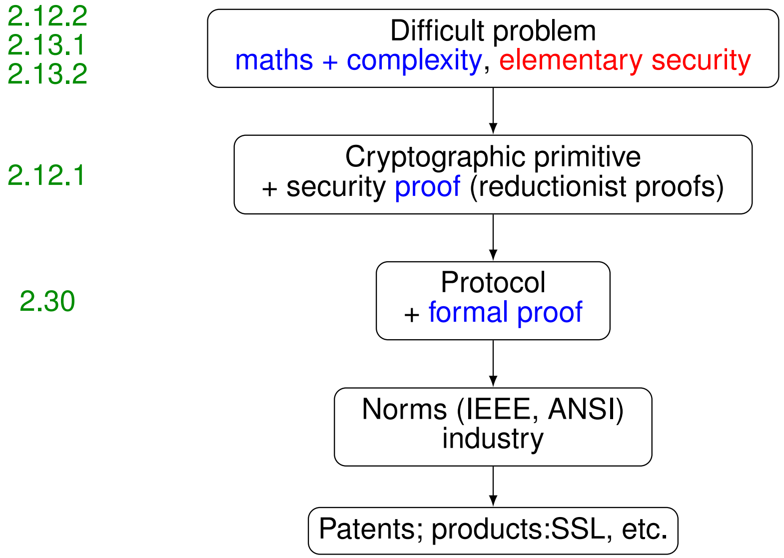
- Algorithmic number theory is about algorithms of number theory and they need to be practiced (python/SAGE, Maple, Magma, pari-gp, etc.).
  Best way to realize that real computations take time and must be carefully implemented.
- Of course, we expect students to study and work/read/do exercises between lectures.

Good reading

- G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. Clarendon Press, 5th edition, 1985.
- D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*. Addison-Wesley, 2nd edition, 1981.
- H. Cohen. *A course in algorithmic algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer–Verlag, 4th printing, 2000.
- P. Ribenboim. *The new book of prime number records*. Springer-Verlag, 1996.
- R. Crandall and C. Pomerance. *Primes – A Computational Perspective*. Springer Verlag, 2nd edition, 2005.
- FM. La primalité en temps polynomial [d'après Adleman, Huang; Agrawal, Kayal, Saxena]. Séminaire Bourbaki, Mars 2003.

II. Overview of the lectures

Goals



Cryptographic motivations: two algorithms

A) Diffie-Hellman

Public parameters: p prime number, g generator of $\mathbb{F}_p^*$.
Protocol:

$$A \overset{g^a \bmod p}{\longrightarrow} B$$

$$A \overset{g^b \bmod p}{\longleftarrow} B$$

$$A : K_{AB} = (g^b)^a \equiv g^{ab} \bmod p$$

$$B : K_{BA} = (g^a)^b \equiv g^{ab} \bmod p$$

DH problem: given (p, g, g^a, g^b), compute g^{ab}.

DL problem: given (p, g, g^a), find a.

Thm. DLP $\Rightarrow$ DHP; converse true for a large class of groups (Maurer & Wolf).

$\Rightarrow$ **Goal for us:** find a good resistant group.

The difficulty of discrete logarithm computations

Over $\mathbb{F}_p$: Best algorithm so far: *à la* NFS $O(L_p[1/3, c'])$ (Gordon, Schirokauer).

Records:

- $160dd$ (2007): T. Kleinjung, 3.3 years of PC 3.2 GHz Xeon64; matrix $2,177,226 \times 2,177,026$ with $289,976,350$ non-zero coefficients, inverted in 14 years CPU.
- $180dd = 596b$ (2014): Bouvier/Gaudry/Imbert/Jeljeli/Thomé (CADO-NFS), matrix $7.28 \cdot 10^6$ rows and columns.
- $768b$ (2016): Kleinjung *et al.*, $24 \cdot 10^6$ rows and columns.

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

As a quick comparison

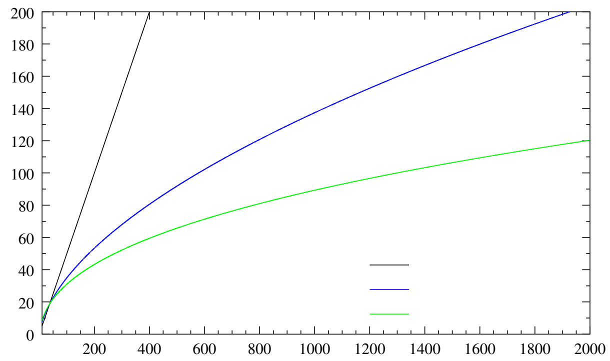


Figure: (Log of) Security vs. bit size of key (exponential, $L(1/2)$, $L(1/3)$)

$$L_x[\alpha, c] = \exp\left((c + o(1))(\log x)^{\alpha}(\log\log x)^{1-\alpha}\right).$$

DLP over $\mathbb{F}_{p^n}$

Adleman-DeMarrais, function field sieve + optimizations.

Records:

- $p = 2$: (Coppersmith)
  - $\mathbb{F}_{2^{809}}$: Gaudry *et al.* (2013).
  - $\mathbb{F}_{2^{1279}}$: Kleinjung (2014).
- $p = 3$:
  - $\mathbb{F}_{3^{6\times 71}}$: Hayashi *et al.* (2010).
  - $\mathbb{F}_{3^{6\times 509}}$: Adj *et al.* (2016).
- Medium p case: Joux+Lercier; etc.; **lots of results in 2012-2013**; **Barbulescu/Gaudry/Thomé/Joux (2013): doable in quasipolynomial time** $\Rightarrow$ see Barbulescu's part.
- $\mathbb{F}_{p^2}$: p with 90dd, Barbulescu/Gaudry/Guillevic/M. (2014).
- $\mathbb{F}_{p^3}$: p with 60dd, Gaudry/Guillevic/M. (2016), matrix $48 \cdot 10^6$ rows and columns.

ECDLP

ECC2K-108: (Harley *et al.*, taken from `http://cristal.inria.fr/~harley/`)

- 1300 individuals, 9500 machines, dec 1999 until april 2000.
- 200,000 days on a 450 MHz PC with MMX, i.e. more than 500 years. For comparison, cracking a 56-bit DES key by exhaustive search would take about 110,000 days.
- 2.8×10^{15} elliptic-curve operations of which 2.3×10^{15} led to distinguished points recorded at INRIA; 2.05 million distinguished points in 1.3 Gigabytes of email.

ECC112b: taken from `http://lacal.epfl.ch/page81774.html`, Bos/Kaihara/Kleinjung/Lenstra/Montgomery (EPFL/Alcatel-Lucent Bell Laboratories/MSR)

$p = (2^{128} - 3)/(11 \cdot 6949)$, curve secp112r1

- 3.5 months on 200 PS3; 8.5×10^{16} ec additions ($\approx$ 14 full 56-bit DES key searches); started on January 13, 2009, and finished on July 8, 2009.
- half a billion distinguished points using 0.6 Terabyte of disk space.

ECDLP – cont'd

ECC2K-113: *Solving the discrete logarithm of a 113-bit Koblitz curve with an FPGA cluster*, E. Wenger & P. Wolfger, 2014.

24 days on an 18-core Virtex-6 FPGA cluster.

Hardware is fun:

- 165 MHz instead of maximum 275 MHz.
- (more or less related) one ECC-breaker per FPGA.

B) RSA

Key generation: Alice chooses two primes p and q, $p \neq q$, $N = pq$, e s.t. $\gcd(e, \lambda(N)) = 1$, $d \equiv 1/e \bmod \lambda(N)$.

Public key: (N, e).

Private key: d (or (p, q)).

Encryption: Bob recovers the authenticated public key of Alice; sends $y = x^e \bmod N$.

Decryption: Alice computes $y^d \bmod N \equiv x \bmod N$.

Rem. of course, in real life, more has to be done, but this has already been told somewhere else.

$\Rightarrow$ **Goal for us:** what size should N have, in order not to be factored?

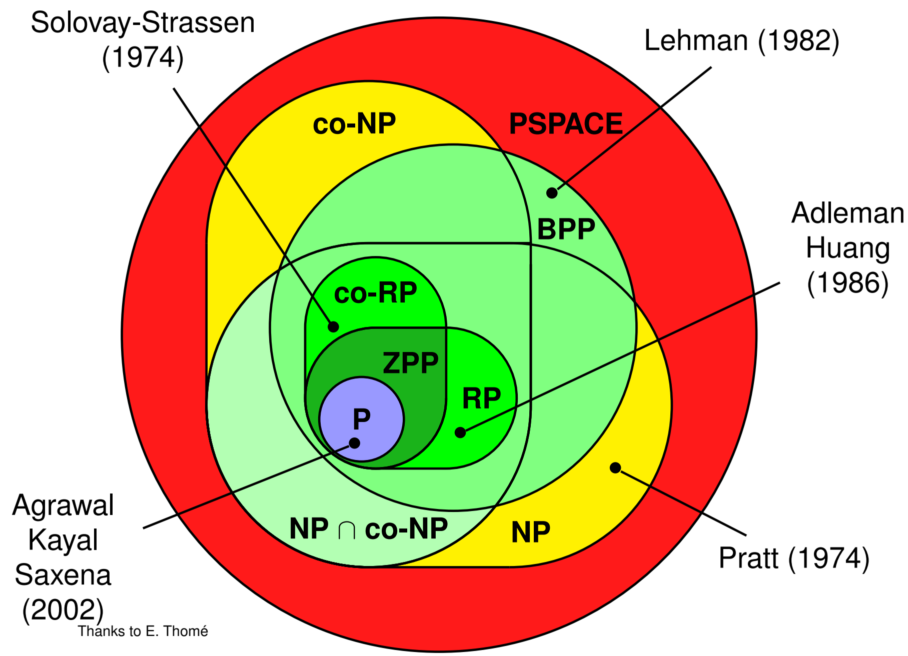
Rules of the game

$$N = \prod_{i=1}^{k} p_i^{\alpha_i}.$$

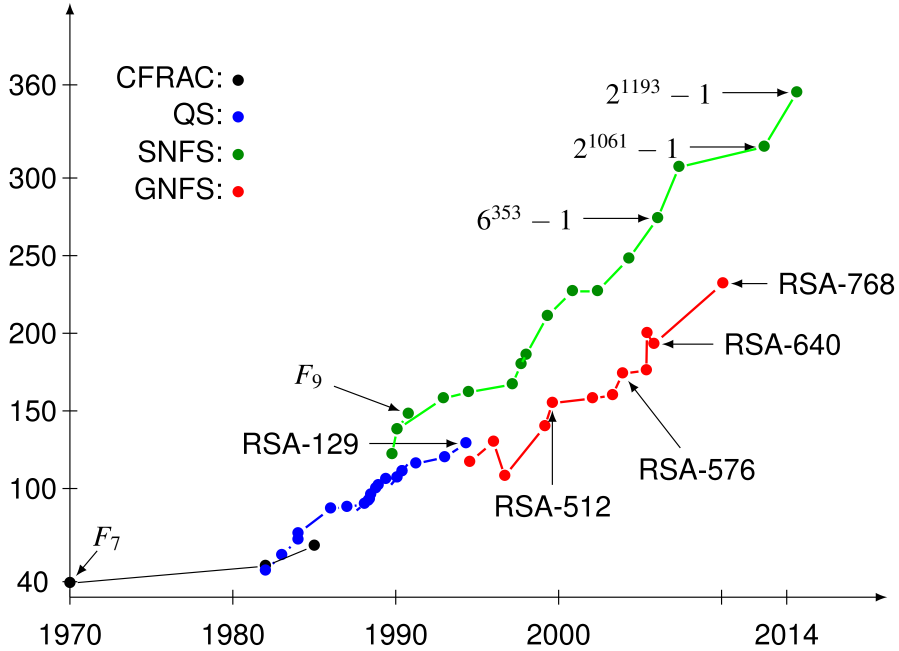
- What do we do in practice? Which size is doable?
  Factorization : number field sieve
  $O(\exp(c(\log N)^{1/3}(\log\log N)^{2/3}))$; 768 bits (a lot of people, 2010).
  Primality: hopefully without too much factoring, past some easy trial division; 25,000 decimal digits.
- Complexity question: to which class does **isPrime?** belong?

Best : **P** (e.g., integer multiplication).
At least : **RP**.
And: what about a proof?

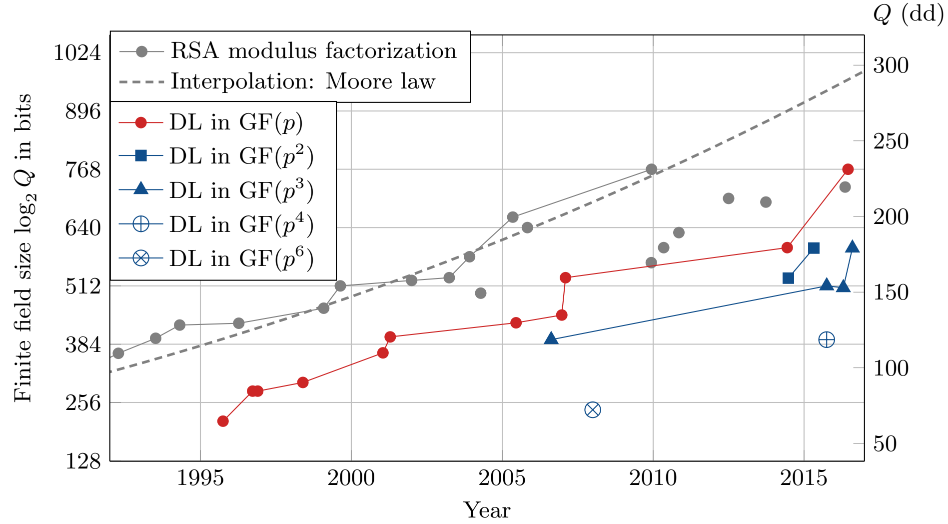
Complexity classes



How difficult is factoring?

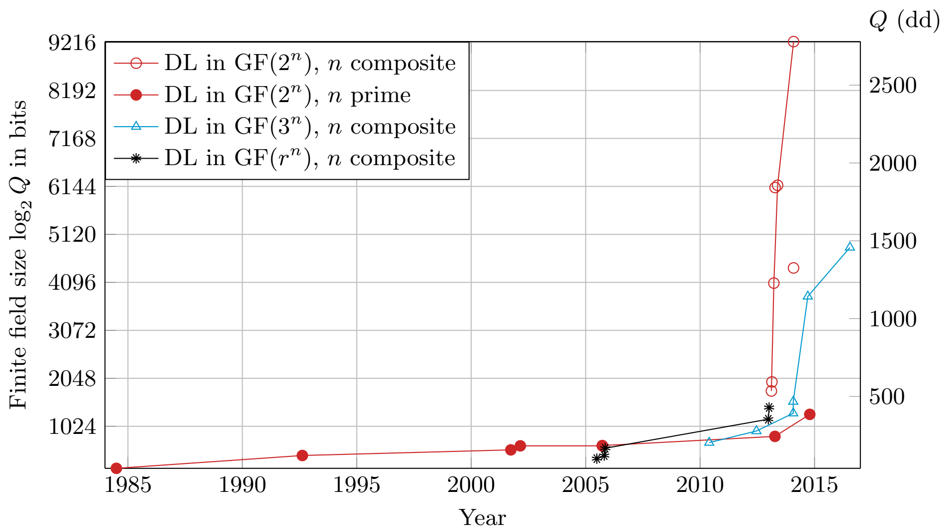


DL vs. IF



Courtesy A. Guillevic

DL over GF(p^n), p not large



Courtesy A. Guillevic