## WCC'01

January 11, 2001

# Cryptography and biodiversity

F. Morain

ÉCOLE POLYTECHNIQUE    LIX    CNRS

http://www.lix.polytechnique.fr/Labo/Francois.Morain/

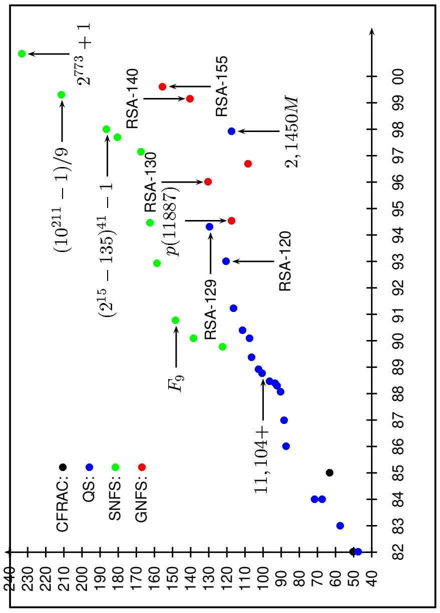
### Plan

I. The need for biodiversity.

II. The discrete logarithm problems.

III. Algebraic curves.

**Slide 1**




**Slide 2**



### I. The need for biodiversity

Threats:

- Ordinary: compromission of keys/systems; progress in computing power (beware of Moore's law; time is not money, memory is).
- Changes in life: new attack scenarios (find $K$ given $(M, C)$, adaptative chosen cipher text attacks).
- Extraordinary: discovery of weak instances, new technologies (quantum computers, $P = NP$?), new ideas in cryptanalysis.

Lesson from the old ages: be sure to have a replacement cipher in case one suspects a breaking of the preceding one.

$\Rightarrow$ we need a biodiversity of problems, but also a biodiversity of instances.

**Slide 3**

### What is modern cryptography?

Find a difficult problem and build a secure cryptosystem out of it.

Difficult? not solvable in reasonable time (during the lifetime of the information, of the programmer, polynomial time, etc.). Need an almost always difficult problem, not just a worst-case or average one.

Ex. integer factorization, discrete logarithms over groups, codes, lattice problems, polynomials, combinatorial group theory, etc.

Secure? Need to find a model of security and prove theorems within that model. Beware of Knudsen's motto:

*A provably secure system is probably not.*

More advanced scenarios: resistance to adaptative chosen ciphertext attacks, semantic security, etc.. See the recent OAEP flaw. Few known practical cryptosystems (Cramer-Shoup). This side of the force should strike everywhere.

**Slide 4**

## Basic security

- Breaking RSA $\Leftarrow$ factoring the modulus; $\Rightarrow$ may be false: Boneh and Venkatesan show that breaking LE-RSA cannot be equivalent to factoring (coherent with Coppersmith's attacks, etc.).
- Breaking DH $\Leftarrow$ solving the discrete log problem; $\Rightarrow$ almost true, see Maurer.
- Breaking McEliece, Niederreiter, Gabidulin $\Leftarrow$ bounded decoding $\Leftarrow$ complete decoding (NP-hard). Attacks can also be launched against the structure of the underlying code. What is a good code?
- Lattice problems: all practical instances broken, even those coming from Ajtai's work (see Nguyen's thesis).
- Polynomials: . . . , HFE, NTRU, . . .
- Combinatorial group theory: braid groups (CRYPTO'2000)? Other groups?
- . . .

Slide 5

## II. The discrete log problems

Idealized version of the Diffie-Hellman protocole: Alice and Bob choose a cyclic group $G =< g >$ and perform

$$A \xrightarrow{g^x} B$$
$$A \xleftarrow{g^y} B$$
$$A : K_{AB} = (g^y)^x = g^{xy}$$
$$B : K_{BA} = (g^x)^y = g^{xy}.$$

Diffie-Hellman problem: given $(g, g^x, g^y)$, compute $g^{xy}$.

Decisional Diffie-Hellman problem: given $(g, a = g^x, b = g^y, c)$, is it true that $c = g^{xy}$? (basis of the Cramer-Shoup cryptosystem).

Discrete log problem: given $g, g^x$, find $x$.

Thm. DL $\Rightarrow$ DH, converse almost true (Maurer).

Slide 6



## Can any group be used?

(we just need a semi-group)

- Abelian or not, but $< g >$ certainly is.
- $< g >$ finite, otherwise elements are too large.
- Need two representations of $G =< g >$: the obvious encoding is

$$\begin{array}{ccc} \mathbb{Z}/n\mathbb{Z} & \rightarrow & G \\ x & \mapsto & g^x \end{array}$$

  We must have a non obvious one: a presentation by generator is not enough for our group.
- computing $g^x$ should be fast, thus computing the group law should be.

Ex. $2\mathbb{Z}$ is bad, as is $((\mathbb{Z}/N\mathbb{Z})^*, +)$ (Euclid's algorithm), or the sandpile group; $G = (\mathbb{Z}/p\mathbb{Z})^*$ has a non-trivial encoding: any $z$ in $[1..p-1]$ is some $g^x$.

Slide 7

## What about DL?

Generically weak instance: $n = \mathrm{ord}(g)$ is smooth (Pohlig-Hellman). Hence prefer prime $n$'s.

Upper bound for DL: (Shanks) let $\# < g >= n$, solving $g^x = a$ can be done by writing $x = u + v\sqrt{n}$ and writing $a(g^{-\sqrt{n}})^v = g^u$. Also Pollard's $\rho$ with $O(1)$ space.

Csqce. for $n \approx 2^{200}$, DL is not solvable.

Lower bound for DL: (Nechaev, Shoup) any **generic algorithm** (i.e., that do not exploit any special properties of the encodings of the group elements) for solving DL (resp. DH) must perform at least $O(\sqrt{\#G})$ group operations.

Nechaev group: in which we could prove that the best algorithm is $O(\sqrt{\#G})$.

Do Nechaev groups exist at all???

Slide 8

## Algorithmic tasks

- Find a good representation of elements.
- Be able to draw elements at random.
- Compute the group law efficiently.
- Compute $\#G$ rapidly.

Proposed groups: multiplicative group of a finite field, class groups of number fields, (jacobians of) algebraic curves.

| Group | $\#G$ | DL |
|---|---|---|
| $\mathbb{F}_q^*$ | $q-1$ | $L_q[1/3]$ |
| class groups of number fields | subexp | subexp |
| jacobians | $g=1$: poly | $\sqrt{\#G}$ |
| | $g=2,3,4$: poly (?) | $\sqrt{\#G}$ |
| | $g\to\infty$: poly (?) | $L_{q^g}[1/2]$ |

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

Security: $1024$ bits for $\mathbb{F}_q^*$ or $200$ bits for curves.

**Slide 9**

## Index calculus methods

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

Methods in $L_N[\alpha, c]$ for $\alpha = 1/2, 1/3$.

Input: $G = \mathbb{F}_p^* =< g >, a = g^z$.

**Step 1:** find the logarithms of the primes of a factor basis $\mathcal{F} = \{p_1, p_2, \ldots, p_k\}$ by finding relations

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

for random $b_i$'s, i.e.,

$$b_i \equiv \sum_{i=1}^{k} \alpha_{i,j} \log_g p_j \bmod \#G.$$

Solve the system over $\mathbb{Z}/\#G\mathbb{Z}$.

**Step 2:** look for $b$ s.t. $ag^b$ factors over $\mathcal{F}$:

$$ag^b = \prod_{j=1}^{k} p_j^{\alpha_j} \Leftrightarrow z + b \equiv \sum_{j=1}^{k} \alpha_j \log_g p_j \bmod \#G.$$

**Slide 10**



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

Record: Joux & Lercier (100 digits, 1999, matrix size was $172,049 \times 171,061$, 3 weeks of Lanczos on a quadriproc DecAlpha 500 MHz).

General setting: Enge and Gaudry, following Knopfmacher's approach, give proven running times. Applicable to some algebraic curves. The cost is: $O(L_{\#G}[1/2, c])$ where $c$ depends on $G$ and/or $\#G$.

**Slide 11**

## III. Algebraic curves

Let $C$ be a plane smooth projective curve of genus $g$ with equation $F(X, Y) = 0$ with coefficients in $\mathbb{K}$, $\mathrm{char}(\mathbb{K}) = p$, $\#\mathbb{K} = q = p^n$.

Def. $C(\mathbb{K}) = \{P = (x, y) \in \mathbb{K}^2, F(x, y) = 0\} \cup \{$points at $\infty\}$.

Thm. When $g \in \{0, 1\}$, there is a group law on $C$, otherwise on the Jacobian (group of *divisors*, i.e., formal sums of points).

Thm. (Weil) let $N_r = \#C(\mathbb{K}_r)$ where $[\mathbb{K}_r : \mathbb{K}] = r$:

$$Z(T) = \exp\left(\sum_{r\geq 1} N_r T^r / r\right) = \frac{L(T)}{(1-T)(1-qT)}$$

Thm. $\#\mathrm{Jac}(C) = L(1)$.

Coro. $|\#C - (q+1)| \leq 2g\sqrt{q}$;
$(\sqrt{q}-1)^{2g} \leq \#\mathrm{Jac}(C) \leq (\sqrt{q}+1)^{2g}$.

Interest: many large objects of cardinality $\approx q^g$; paramaterized families easy to build.

**Slide 12**

Representation of elements/group law: use divisors; in theory, the group law can be computed using algorithms by Coates, Huang & Ierardi; Volcheck; general plane curves by Hess.

Cardinality: use generalizations of Schoof's algorithm (Pila; Huang & Ierardi) that work in polynomial time.

### The curves used in practice

Most used case: hyperelliptic curves $Y^2 = X^{2g+1} + \cdots$ (Miller, Koblitz for $g = 1$, Koblitz for $g > 1$, etc.).

Other curves:

- superelliptic $Y^n = f(X)$ (Galbraith, Paulus, Smart);
- $C_{ab}$ (Arita), e.g. $Y^3 + 5X^2Y - 14XY + X^4 - 123 = 0$;
- modular curves (Frey, Müller).

Slide 13

Hyperelliptic curves:

- Representation of elements: for $g = 1$, use points; for $g > 1$, use Mumford's representation $< u(z), v(z) >$ where $u$ and $v \in \mathbb{F}_q[z]$ and $\deg(v) < \deg(u) \leq g$.
- Group law: for $g = 1$, chord-and-tangent; for $g > 1$, use Cantor's algorithm.

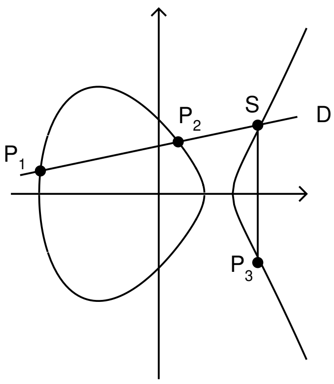
$C_{ab}$ curves: several possibilities for the representation (Gröbner basis, HNF, or divisors). For instance:

- Representation of elements: a divisor is represented by an element of the class group of the function field (HNF of a polynomial matrix).
- Group law: multiplication/reduction of ideals, equivalently HNF+ reduction of the product of two matrices; can be done in polynomial time.

Slide 14



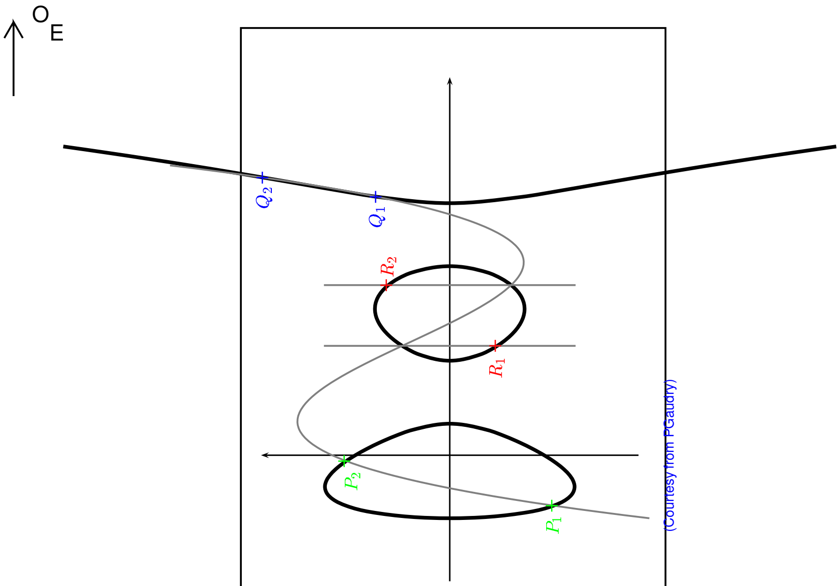
### Group law for $g = 1$




$$P_3 = P_1 \oplus P_2$$

$$[k]P = \underbrace{P \oplus \cdots \oplus P}_{k \text{ times}}$$

Slide 15




Slide 16

## Cardinality: practice

- $g = 1$: Schoof-Elkies-Atkin algorithm (SEA) efficient for large characteristic; Satoh's algorithm (1999) for small $p \geq 5$, extension by Fouquet-Gaudry-Harley for $p = 2$ (5 minutes on an Alpha 750 MHz to build a good *random* curve over $\mathbb{F}_{2^{233}}$); probably not the end of the story.
- $g = 2$: Gaudry & Harley (ANTS IV) mimic SEA whenever possible
  - find $\#\mathrm{Jac} \bmod \ell$ using Cartier-Manin ($\ell = p$) or Cantor's division polynomials, find $\#\mathrm{Jac} \bmod 2^k$ by looking for solutions of $[2]D = D_0$;
  - combine all this to reduce the birthday paradox.
  - record: $p = 10^{19} + 51$;

  $$y^2 = x^5 + 3141592653589793238\, x^4 + \cdots$$

  $$\#\mathrm{Jac} = 99999999982871020671452277000281660080,$$

  $> 50$ days on a 500 MHz Alpha.
- $g > 2$: Stein-Teske (1999): $q = 10^8 + 7$, $g = 3$, $\#\mathrm{Jac} \approx 10^{24}$ in 10 hours on a Pentium Pro/200 with SIMATH.

**Slide 17**

## Weak instances

- Reduction $\mathrm{Jac}(C)/\mathbb{F}_q \hookrightarrow \mathbb{F}_{q^k}$ with $k$ small:
  - Supersingular curves: MOV (Menezes, Okamoto, Vanstone using the Weil pairing); Frey & Rück (using the Tate pairing); Galbraith.
  - other cases: elliptic curves with $\#E = p - 1$ with the Tate pairing.
- Discrete logs in subgroups of order $p^e$ of $\mathrm{Jac}(C)/\mathbb{F}_{p^r}$ can be found in polynomial time: $g = 1$ (anomalous curves, $\#E = p$) done by Satoh-Araki, Semaev, Smart; $g > 1$ by Rück.
- Weil-Koblitz: curves over $\mathbb{F}_q$ for $q$ small and used $\mathrm{Jac}(C)/\mathbb{F}_{q^k}$. ECDL is a little easier.

**Slide 18**



## ECC2K-108

Harley *et al.*, taken from `http://cristal.inria.fr/~harley/` using Pollard's rho in parallel.

- 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 110000 days.
- $2.8 \times 10^{15}$ elliptic-curve operations of which $2.3 \times 10^{15}$ led to distinguished points recorded at INRIA; 2.05 million distinguished points in 1.3 Gigabytes of email.

**Slide 19**

## Computing the discrete log on hyperelliptic curves

- Algorithm ADH from Adleman, DeMarrais, Huang (ANTS I): $L_{p^{2g+1}}[1/2, c]$ with $c \leq 2.181$ if $\log p \leq (2g + 1)^{0.98}$ (heuristic using Lovorn's theorem on smooth polynomials); SNF.
- Flassenberg & Paulus: experiments with $Y^2 = X^{2g+1} + 2X + 1$ using sieves, faster than Shanks for $g \geq 6$.
- $Y^2 = X^{2g+2} + \cdots$ (Müller-Stein-Thiel): proved $L_{p^{2g+2}}[1/2, 1.44]$.
- Extensions, proved analysis and optimizations by Enge: $L_{q^g}[1/2, c(\theta)]$ if $\theta \log q \leq g$, with $\lim_{\theta \to 0} c(\theta) = +\infty$; easier SNF. Smaller $c = \sqrt{2}$ by Enge and Gaudry.
- Gaudry's variant: fast, practical, implemented, used in the final stage of the Weil descent approach to break some ECDL (Frey; Gaudry, Hess, Smart; Menezes & Wu; Smart). Works also on $C_{ab}$ curves (Arita), and supposedly on more general curves.

**Slide 20**

## Gaudry's variant

Main features:

- index calculus using a basis of $S$-smooth divisors, i.e. $\mathcal{F} = \{< u(z), v(z) > \in \mathrm{Jac}(C), u(z)$ is irreducible of degree $\leq S\}$, with $S = 1$ for $g \leq 8$.
- Finding relations using a random walk.
- Relatively easy linear algebra (at most $g$ entries per row).
- exponential asymptotic complexity but very practical and breaks a lot of systems proposed by various authors.

  Thm. Gaudry's variant takes time $O(q^2 \log^c q)$ for fixed $g$, hence better than Pollard's rho for $g > 4$.

Size of Jac breakable: $\#\mathrm{Jac} = q^g = (2\#\mathcal{F})^g$:

| $\#\mathcal{F} \backslash g$ | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|
| $10^5$ | $2^{70}$ | $2^{88}$ | $2^{105}$ | $2^{123}$ | $2^{140}$ |
| $10^6$ | $2^{83}$ | $2^{104}$ | $2^{125}$ | $2^{146}$ | $2^{167}$ |

**Slide 21**

## An example proposed by Buhler and Koblitz

**Magma** 2.3 on Pentium II 450 MHz, 128 Mb.

| | |
|---|---|
| field | $\mathbb{F}_{5026243}$ |
| equation | $y^2 + y = x^{13}$ |
| genus | $6$ |
| auto | $13$ |
| $\#\mathrm{Jac}$ | $\approx 10^{40}$ |
| basis | $193,485$ |
| matrix | $165,778 \times 165,779$ |

| | |
|---|---|
| construct. basis | 1638 s |
| relations | 6 days |
| SGE | 1218 |
| Lanczos | 9 days |

**Slide 22**



## Conclusions

- Security proofs will soon become available and destroy more systems.
- We still do not have that many difficult problems.
- Nor do we have enough difficult instances.
- Very few groups actually: we need to investigate more complex mathematical objects, but are there many useful ones?
- As for curves: the world ends at $g = 3$! Will elliptic curves resist?

**Slide 23**