Journées CRYPTIS

Limoges, 24 novembre 2011

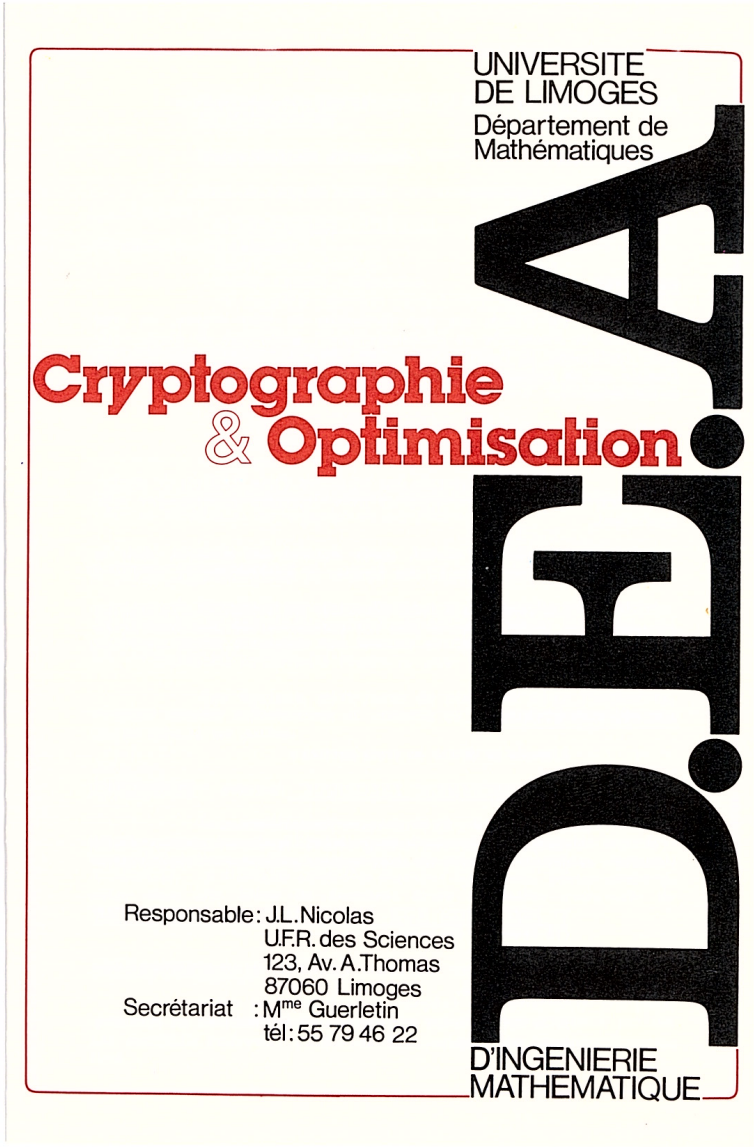
25 ans de cryptographie basée sur la théorie des nombres

F. Morain

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

(version amendée)

Nostalgie. . .


UNIVERSITE
DE LIMOGES
Département de
Mathématiques
Cryptographie
& Optimisation
D.E.A.
D'INGENIERIE
MATHEMATIQUE
Responsable: J.L.Nicolas
U.F.R. des Sciences
123, Av. A.Thomas
87060 Limoges
Secrétariat : Mme Guerletin
tél: 55 79 46 22

Nostalgie (suite): 1986

- Passage commencé des Main Frame vers les Mac (512Ke), Sun (3/50). Transpac plus utilisé qu'Internet (même sur minitel !).
- **Primalité :** sommes de Jacobi ; on commence à parler d'Atkin.
- **Factorisation :** crible quadratique ; ECM commence à faire parler de lui.
- **Logarithme discret :** Odlyzko (p grand) ; Coppersmith ($p = 2$).

Plan

I. Introduction.

II. Les paradoxes de la primalité.

III. Logarithme discret.

IV. Factorisation.

I. Introduction

Historiquement : longue tradition de calculs en théorie des nombres ($2^p - 1$, factorisation, zéros de la fonction de Riemann, tables, etc.).

Épanouissement : (depuis les années 1980)

- démocratisation des moyens de calcul ;
- problèmes d'énoncés faciles dans la théorie de la complexité (Eratosthene, etc.) ;
- invention de la cryptographie moderne (asymétrique).

Définition

Théorie algorithmique des nombres
$=$ théorie des nombres + algorithmique/complexité.

Évolution :
calculs faisables/infaisables $\Rightarrow$ faciles/difficiles.

But ultime : remplacer les cas particuliers par des algorithmes résolvant tous les cas.

Premiers exemples (1/3)

ENTRÉE : N

SORTIE : La factorisation de N

COMPLEXITÉ : Sous-exponentiel probabiliste ou exponentiel déterministe dans le modèle classique.

Rem. Preuve des calculs.

Premiers exemples (2/3)

ENTRÉE : N

SORTIE : La factorisation de N

COMPLEXITÉ : Temps polynomial dans le modèle quantique (Shor).

Rem. Preuve des calculs.

Premiers exemples (3/3)

ENTRÉE : N

SORTIE : N est-il premier ?

COMPLEXITÉ : Temps polynomial déterministe prouvé dans le modèle classique, $\tilde{O}((\log N)^6$ (AKS ; Lenstra/Pomerance).

Rem. Preuve des calculs.

Champs d'action

- Arithmétique (entiers, polynômes).
- Corps de nombres.
- Courbes algébriques.

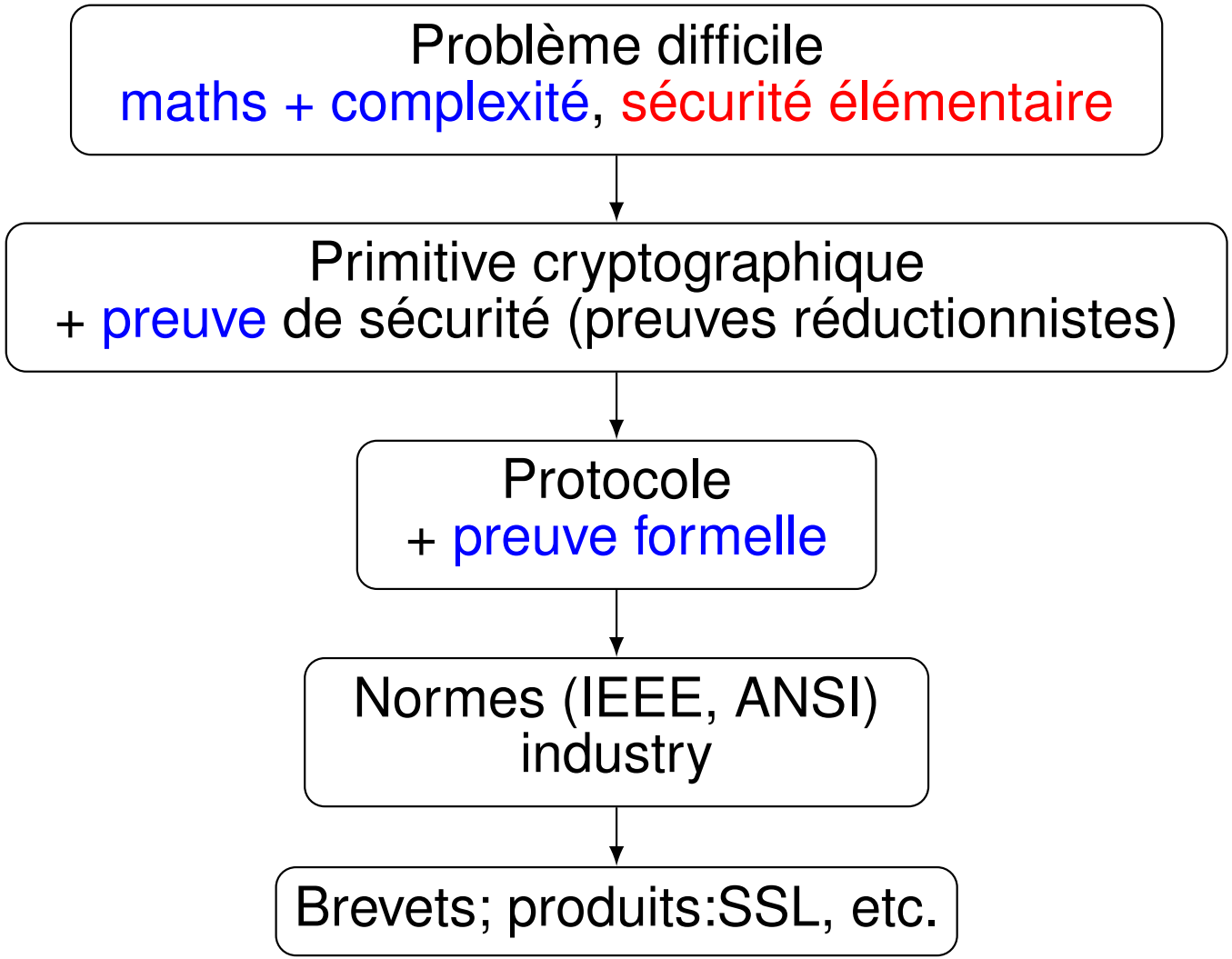
Recherches :

- étude des objets *per se* et intéractions entre domaines (anneaux d'endomorphismes des courbes, groupes de classes, etc.) ;
- applications *pro domo* : primalité, factorisation, calcul du logarithme discret (ECPP, NFS, . . .) ;
- applications extérieures : la cryptographie a besoin de réponses pour des paramètres petits ; la théorie algorithmique cherche à résoudre les problèmes quelle que soit leur taille.

Émergence d'une communauté

- Théories algébrique des nombres et des courbes sont utilisées couramment ; complexité intégrée.
- Outils puissants : GMP ($M(n)$ optimal dans la vraie vie), LLL, systèmes de calcul mathématique (Pari-gp, Magma, Sage, . . .).
- Knuth (vol. 2) a été rejoint par Cohen (1+2), Crandall & Pomerance ;
- *Math. Comp.*, ANTS.

La raison du succès en cryptologie moderne

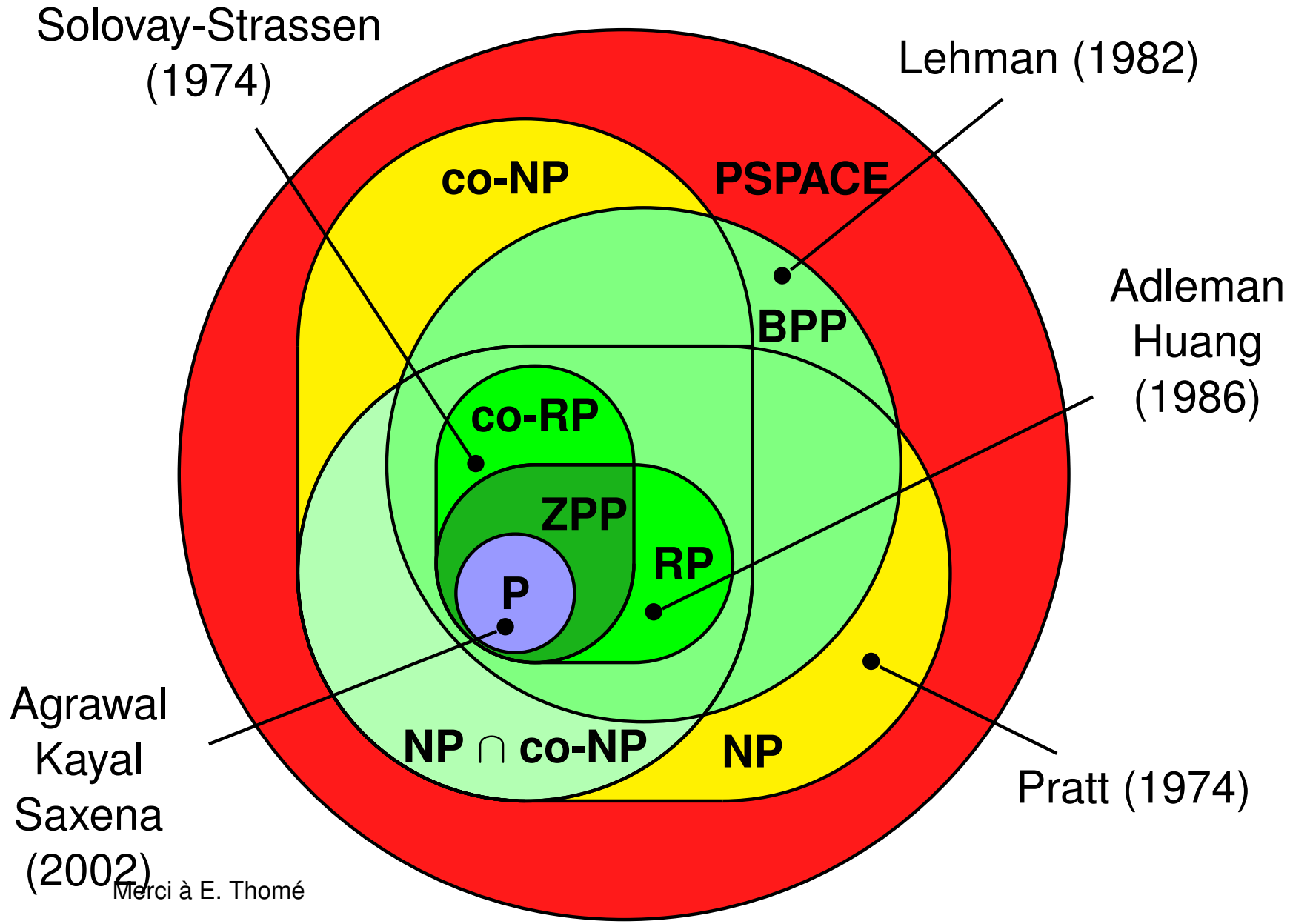


Uranium et plutonium

Deux systèmes emblématiques . . . et toujours là !

- Échange de clefs à la Diffie-Hellman : $g^x \leftrightarrow g^y$.
- Chiffrement RSA (Rivest-Shamir-Adleman) : $N = pq$.

II. Les paradoxes de la primalité (1/2)



Merci à E. Thomé

Les paradoxes de la primalité (2/2)

Les algorithmes décrits précédemment ne sont pas utilisés en pratique. . . !

- **Tests de composition :** Artjuhov-Miller-Rabin, Frobenius, Lucas, Gordon, etc. Très rapides, pas de preuve.
- **Tests de primalité :** sommes de Jacobi (super polynomial déterministe – Adleman/Pomerance/Rumely, Cohen/Lenstra, Cohen/Lenstra, Mihăilescu), courbes elliptiques (fastECPP – heuristiquement polynomial $\tilde{O}((\log N)^4)$, probabiliste, Atkin/M.).

  200 chiffres en 1980 $\rightarrow > 26,000$ en 2010

III. Le problème du logarithme discret

Version idéalisée du protocole de Diffie-Hellman : Alice et Bob choisissent un groupe $G = \langle g \rangle$ et effectuent :

$$A \xrightarrow{g^x} B$$

$$A \xleftarrow{g^y} B$$

$$A : K_{AB} = (g^y)^x = g^{xy}$$

$$B : K_{BA} = (g^x)^y = g^{xy}.$$

Problème DH : étant donnés (g, g^x, g^y), calculer g^{xy}.

Problème LD : étant donnés (g, g^x), trouver x.

Thm. LD $\Rightarrow$ DH, réciproque presque vraie, (Maurer & Wolf).

$\Rightarrow$ trouver des groupes utilisables et résistants.

Que dire de LD en général ?

Instance faible générique : $n = \#G$ est friable (Pohlig-Hellman) $\Rightarrow$ n premier.

Borne supérieure : (Shanks) pour résoudre $g^x = a$, on écrit $x = u + v\sqrt{n}$ et $a(g^{-\sqrt{n}})^v = g^u$.

Borne inférieure : (Nechaev, Shoup) tout **algorithme générique** résolvant LD (resp. DH) doit effectuer au moins $O(\sqrt{\#G})$ opérations de groupe.

Groupe de Nechaev : meilleur algorithme possible est en $O(\sqrt{\#G})$.

Est-ce que les groupes de Nechaev existent ? ? ?

Csqce. Si $\#G \approx 2^{200}$, LD n'est pas résoluble.

Quelques groupes utilisés

Groupe	$\#G$	LD
$\mathbb{F}_q^*$	$q-1$	$L_q[1/3]$
groupe de classes	subexp	subexp
jacobienne	$g=1$: poly $g=2,3,4$: poly (?) $g \to \infty$: poly (?)	$\sqrt{\#G}$ $\sqrt{\#G}$ $L_{q^g}[1/2]$

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

$$L_N[0,c] = (\log N)^c, \quad L_N[1,c] = N^c$$

N.B. La théorie algébrique algorithmique a également énormément progressé depuis 25 ans.

Comparaisons rapides

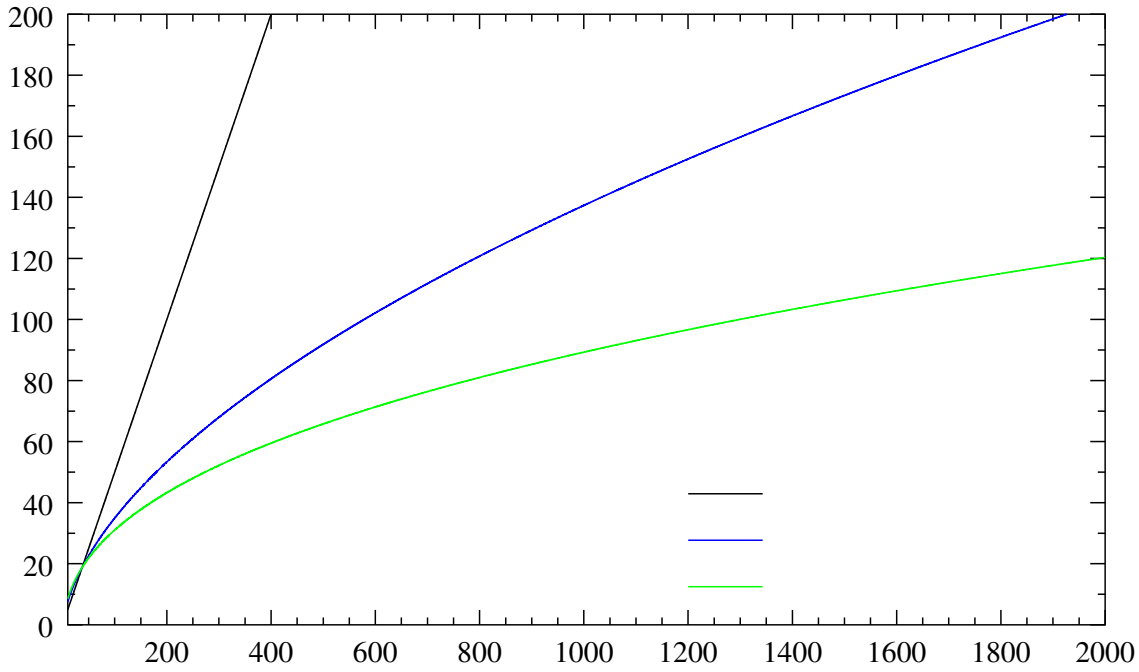


Figure: (Log de la) sécurité vs. taille de la clef en bits (exponentielle, $L(1/2)$, $L(1/3)$)

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

Sécurité : 1024 bits pour $\mathbb{F}_q^* = 200$ bits pour les courbes.

LD dans les corps finis

Principe de base : trouver $a^u g^v = 1$.

- $\mathbb{F}_p$:
  - Meilleur algorithme connu : *à la* NFS $O(L_p[1/3, c'])$ (Gordon, Schirokauer).
  - record avec $160dd$: T. Kleinjung (2007); 3.3 years of PC 3.2 GHz Xeon64; matrix $2,177,226 \times 2,177,026$ avec $289,976,350$ coefficients $\neq 0$, inversé en 14 ans de CPU.
- $\mathbb{F}_{p^n}$: Adleman-DeMarrais, function field sieve + optimisations.
  - $p = 2$: Coppersmith; record avec $\mathbb{F}_{2^{613}}$: Joux/Lercier (2005).
  - record $\mathbb{F}_{36\times 71}$: Hayashi *et al.* (2010), `http://eprint.iacr.org/2010/090`.
  - Cas p moyen : Joux/Lercier $\mathbb{F}_{370801^{30}}$ (168dd – 556b).

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

L'essor de la théorie algorithmique des courbes

Pourquoi des courbes ?

- Accès rapide à des groupes très variés (cardinal avec Hasse-Weil). Par exemple, explique le succès d'ECM (H. W. Lenstra, Jr.).
- Problèmes plus difficiles (?)
- Objets plus riches que ceux des corps finis : couplages (pour casser, pour construire, nouveaux protocoles) ; isogénies (transport de LD, etc.).

Quelques tâches :

- Classification des courbes résistantes (resp. faibles).
- Recherche d'algorithmes de calcul des lois de groupe.
- Algorithmes de calcul de cardinalité.

Courbe elliptique : $y^2 = x^3 + ax + b$

- **Courbes aléatoires :**
  - **Schoof (1985) :** calculer $\#E \bmod \ell$; $O(\log^{5+\epsilon} q)$.
  - **Améliorations** théoriques et pratiques (1986ff): Atkin, Elkies, Couveignes, Lercier, Morain, Müller, Dewaghe, . . . , Enge, Sutherland $O(\log^{4+\epsilon} q)$.
  - **Satoh (1999) :** remonter p-adiquement la cardinalité (Lubin-Serre-Tate) ; $O(\log^{3+\epsilon} q)$.
  - **Améliorations** théoriques et pratiques: Fouquet–Gaudry–Harley, Skjernaa, Vercauteren et al., re-Satoh, Mestre... (généralisable au genre 2). Construction d'une bonne courbe en quelques dizaines de secondes sur un PC.
- **Multiplication complexe :** (1985ff) très efficace, utilisable en primalité et cryptographie (Chudnovsky/Chudnosky, Atkin/Morain, Gee/Stevenhagen, Enge/Schertz, Bröker/Stevenhagen, Enge/Morain, . . . , Bröker/Lauter/Sutherland).

ECDLP

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

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

- 3.5 mois sur 200 PS3; 8.5×10^{16} additions ($\approx$ 14 cassages de DES 56-bit); commencé le 13/01/2009, et terminé le 08/07/2009.
- 1/2 milliard de points distingués occupant 0.6 Terabytes.

Joux/Vitse (2011) : cas particulier de $\mathbb{F}_{p^6}$ en exploitant plusieurs techniques (décomposition, revêtement de genre 3) ; calculs réels avec $p = 2^{25} + 35$ en 1 mois de calculs.

Autres courbes

- $g = 2 : y^2 = x^5 + \cdots$
  exponentiellement plus difficile que $g = 1$
  - **Courbes quelconques :** Gaudry/Schost, etc. (e.g., $p = 2^{127} - 1$: 1000 heures CPU en 2010 !) ; encore très difficile pour le moment.
  - **Multiplication complexe :** commence à devenir compétitif comme $g = 1$ (Spallek, Weng, Dupont, Gaudry/Houtmann/Kohel/Ritzenthaler, Streng, Carls, Lubicz, Eisentrager, Lauter/Bröker, Grünewald, Robert, Thomé).
  - **Multiplication réelle :** Gaudry/Kohel/Smith (2011), idem.
  - **Familles particulières :** Satoh, Furukawa/Kawazoe/Takahashi, Haneda/Kawazoe/Takahashi, Anurhada, Guillevic/Vergnaud.

- **Kedlaya (2000) :** cohomologie de Monsky-Washnitzer ($y^r = f(x)$, $(r, p) = 1$).

- **Lauder & Wan (2001) :** à la Dwork ($p = 2$, hyperelliptique).

LD sur les courbes hyperelliptiques

Genre fixé: une courbe est cassée s'il existe un algorithme en $O((q^g)^{1/2-\delta})$ pour un $\delta > 0$.

Adleman, DeMarrais, Huang : $L_{p^{2g+1}}[1/2, c]$ avec $c \leqslant 2.181$ si $\log p \leqslant (2g+1)^{0.98}$ (heuristique utilisant un théorème de Lovorn).

Flassenberg & Paulus : cribles ; expériences avec $y^2 = x^{2g+1} + 2x + 1$, plus rapide que Shanks pour $g \geqslant 6$.

Müller-Stein-Thiel : $L_{p^{2g+2}}[1/2, 1.44]$ pour $y^2 = x^{2g+2} + \cdots$, $g/\log q \to +\infty$.

Enge : extensions, analyses prouvées et optimisations $L_{q^g}[1/2, c(\theta)]$ si $\theta \log q \leqslant g$, avec $\lim_{\theta \to 0} c(\theta) = +\infty$; $c = \sqrt{2}$ par Enge et Gaudry.

Autres courbes

Historiquement:

- Gaudry (2000): $O(q^2) \Rightarrow$ casse $g \geqslant 5$.
- Gaudry/Harley : $O(q^{2-2/(g+1)}) \Rightarrow$ casse $g = 4$.
- Gaudry/Thériault/Thomé (2005): $O(q^{2-2/g}) \Rightarrow$ casse $g = 3$.
- Smith (2008) : quartiques planes (genre 3 non hyperelliptique).

Rem. Gaudry fournit un cadre de travail très général $\Rightarrow$ permet de casser de nombreux systèmes très efficace comme outil dans la descente de Weil (Frey et al., Smart et al., etc.).

Enge/Gaudry/Thomé (2006): pour $Y^{g^{1-\alpha}} + \cdots = X^{g^{\alpha}} + \cdots$, si $g \in \Omega((\log q)^2)$, $\frac{1}{3} \leqslant \alpha \leqslant \frac{1}{2}$, $L_{q^g}[1/3, c]$.

III. La factorisation d'entiers

Rivest–**S**hamir–**A**dleman, 1976.

Fabrication des clefs : p et q premiers, $p \neq q$, $N = pq$, $\lambda(N) = \mathrm{ppcm}(p-1, q-1)$, e tq $\mathrm{pgcd}(e, \lambda(N)) = 1$, $d \equiv 1/e \bmod \lambda(N)$.

La clef publique est (N, e), la clef privée est d.

Chiffrement :

- Bob récupère la clef publique authentique d'Alice.
- Bob calcule $y = x^e \bmod N$ et l'envoie à Alice.

Déchiffrement : Alice calcule $y^d \bmod N \equiv x$.

Signature : $S_A(m) = m^d \bmod N$.

La sécurité

Problème RSA : étant donnés (N, e, y) trouver x tq $x^e \equiv y \bmod N$.

Factorisation : étant donné N, retrouver p et q (ou bien d).

Thm. casser RSA $\Leftarrow$ factoriser N ; $\Rightarrow$ est sans doute fausse : Boneh et Venkatesan ont montré que casser LE-RSA ne peut être équivalent à factoriser N.

Mais factoriser N casse complètement le système (on peut chiffrer, signer).

Il n'y a pas que la factorisation

Attaques sur la datation : (Franklin-Reiter) $C_1 \equiv M^3 \bmod N$, $C_2 \equiv (M+1)^3 \bmod N$; alors :

$$\begin{cases} C_2 + 2C_1 - 1 &= 3M^3 + 3M^2 + 3M \\ C_2 - C_1 + 2 &= 3M^2 + 3M + 3 \end{cases}$$

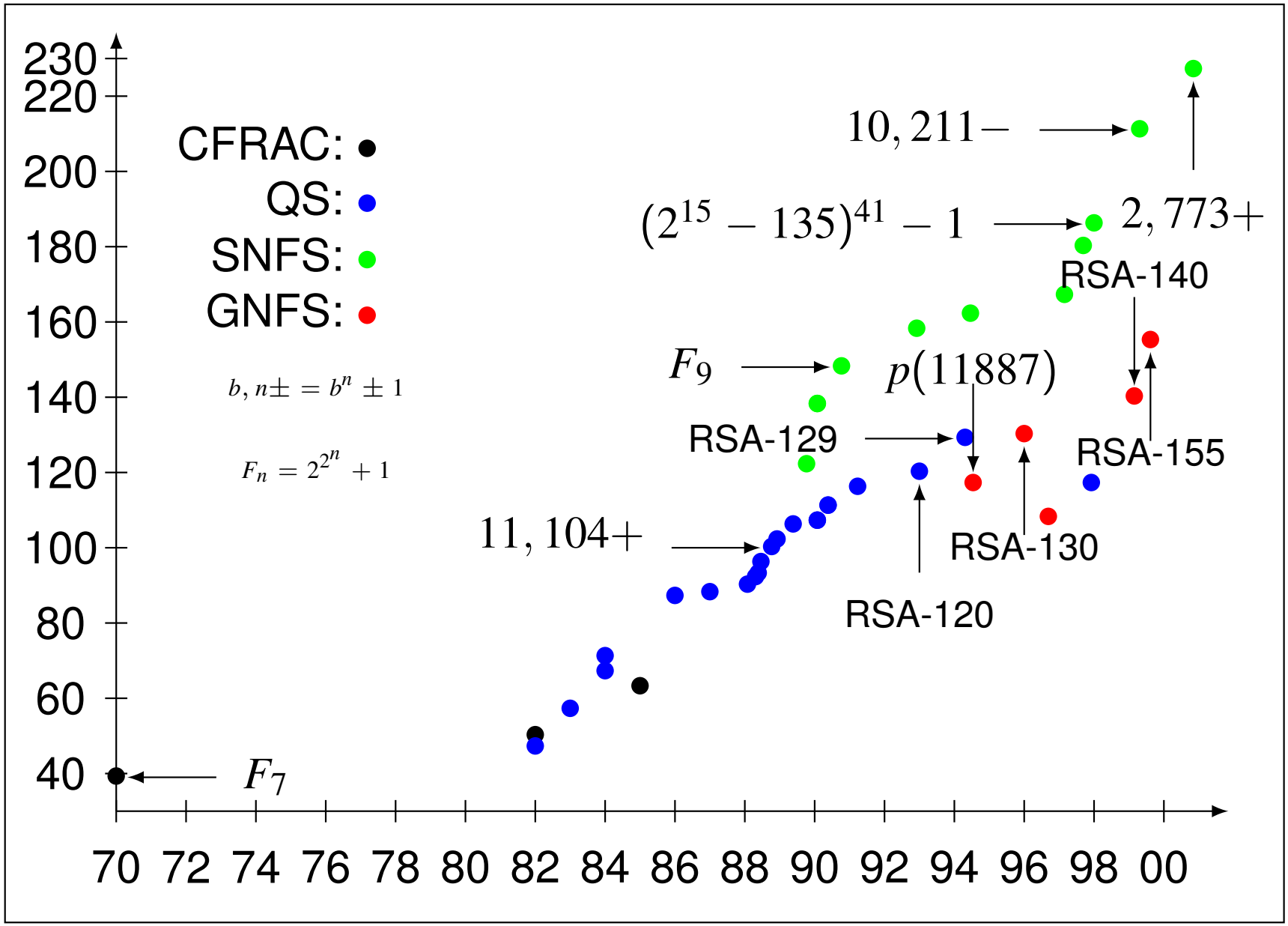
d'où $M = (C_2 + 2C_1 - 1)/(C_2 - C_1 + 2) \bmod N$.

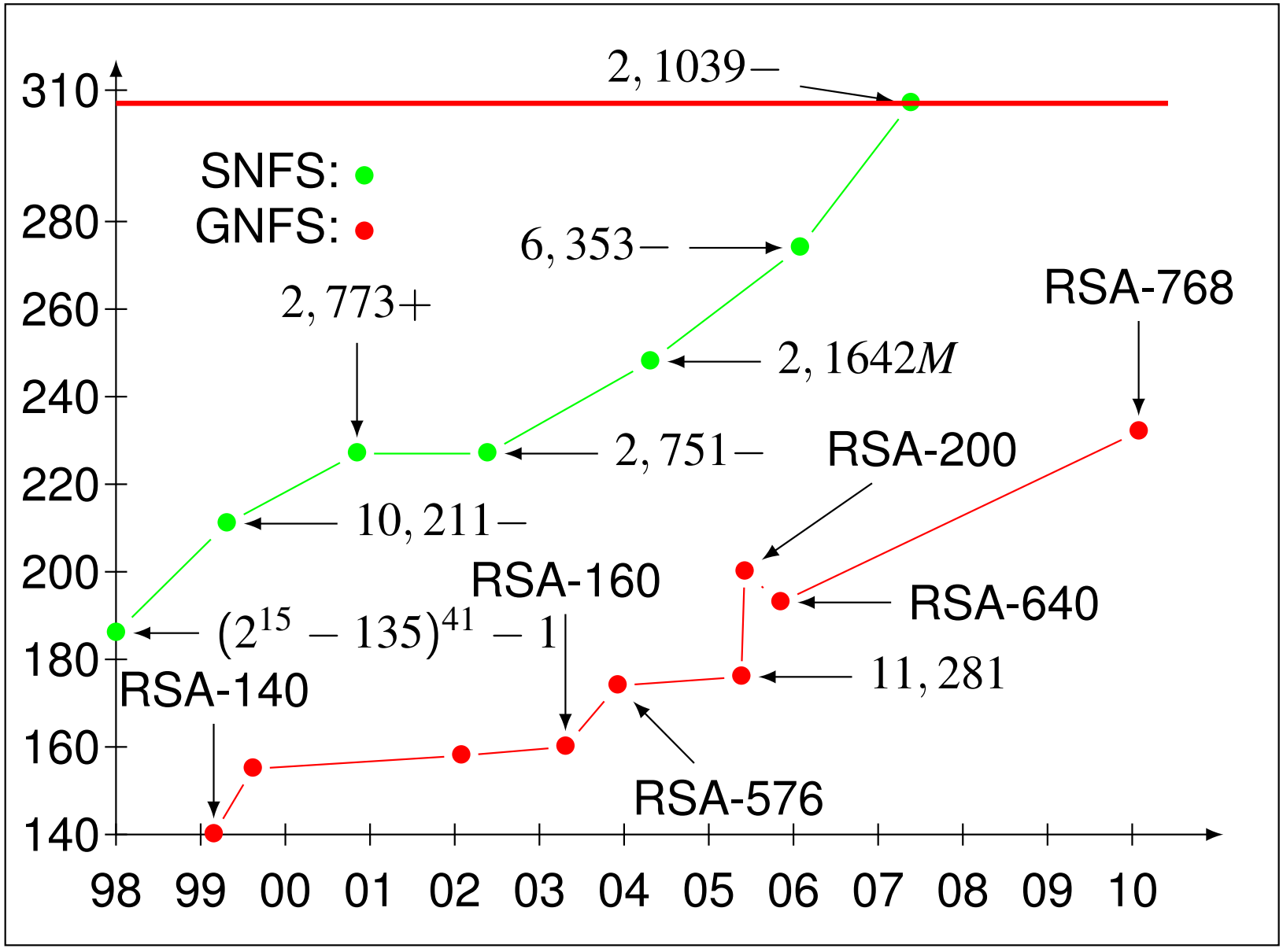
Solution : il faut rendre le chiffrement et la signature aléatoire. Par exemple : OAEP (Bellare & Rogaway ; Shoup, etc.)

$$((M \oplus G(r)) || r \oplus \mathcal{H}(M \oplus G(r)))^e \bmod N.$$

$\Rightarrow$ preuve de sécurité.

La factorisation à travers les âges




310
280
260
240
220
200
180
160
140
98
99
00
01
02
03
04
05
06
07
08
09
10
SNFS:
GNFS:
$2, 1039-$
$6, 353-$
$2, 773+$
$2, 1642M$
$2, 751-$
$10, 211-$
$(2^{15} - 135)^{41} - 1$
RSA-768
RSA-200
RSA-160
RSA-640
RSA-140
$11, 281$
RSA-576

RSA 768 (232dd)

Principe de base : résoudre $x^2 \equiv 1 \bmod N$.

- **Participants :** Thorsten Kleinjung ; Kazumaro Aoki ; Jens Franke ; Arjen Lenstra ; Emmanuel Thomé ; Joppe Bos ; Pierrick Gaudry ; Alexander Kruppa ; Peter Montgomery ; Dag Arne Osvik ; Herman Te Riele ; Andrey Timofeev ; Paul Zimmermann.
- **Crible :** 2007/08 $\rightarrow$ 2009/04 ; 1500 années AMD64 sur plusieurs sites de par le monde.
- **Algèbre linéaire :** après nettoyage, matrice $192,796,550 \times 192,795,550$ (poids : 27, 797, 115, 920) ; 155 années de CPU (Wiedemann par blocs).
- **Facteurs :** trouvés rapidement (2009/12/12).

Utilise une partie de CADO-NFS (512b : 2 à 3 jours sur 100 cœurs ; cf. `http://cado-nfs.gforge.inria.fr/`).

Prospective

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

$$L_x[0, c] = (\log x)^c, \quad L_x[1, c] = x^c$$

	$p \leqslant \sqrt{N}$	QS	NFS
N	$L_N[1, 1/2]$	$L_N[1/2, 1]$	$L_N[1/3, 1]$
$2^{128} \approx 10^{38}$	1.84×10^{19}	4.61×10^{8}	1.85×10^{5}
$2^{256} \approx 10^{77}$	3.40×10^{38}	1.46×10^{13}	2.02×10^{7}
$2^{512} \approx 10^{154}$	1.16×10^{77}	6.69×10^{19}	1.02×10^{10}
$2^{768} \approx 10^{231}$	3.94×10^{115}	1.27×10^{25}	9.49×10^{11}
$2^{1024} \approx 10^{308}$	1.34×10^{154}	4.42×10^{29}	3.82×10^{13}

Brent :

$$\text{année} = 2.15(\log N)^{1/3}(\log\log N)^{2/3} + 1951.45$$

$\Rightarrow 896b = 2014,\ 1024b = 2018,\ 1536b = 2031.$

Conclusions (1/2)

Et si on ne faisait pas de théorie des nombres ?

- décodage d'un code aléatoire (McEliece) ;
- problèmes dans les réseaux (Merkle, . . . , Ajtai, . . . NTRU, . . .) ;
- systèmes d'équations polynomiales (HFE, etc.) ;
- théorie des groupes (groupes de tresses ?) ;
- etc.

Conclusions (2/2)

- La cryptanalyse a au minimum bénéficié de la loi de Moore (attention aux systèmes trop vieux).
- La factorisation est toujours aussi difficile, la primalité est devenue très facile.
- Le logarithme discret est toujours aussi difficile, voire encore plus en changeant de groupe (corps finis $\rightarrow$ courbes de genre $\leqslant 2$).
- Systèmes ancestraux toujours debouts, malgré de nombreuses attaques, contrées en partie par la randomisation des messages (hachage, modèles de sécurité).