Factorisation d’entier : état de l’art

Paul Zimmermann

CARAMEL

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche NANCY - GRAND-EST

ANSSI, 31 mai 2010


Ninth Algorithmic Number Theory Symposium
ANTS IX
Nancy, France
July 19-23, 2010
http://ants9.org/
LOCAL ORGANIZATION
Anne-Lise Charbonnier
Jérémie Detrey
Pierrick Gaudry
Emmanuel Thomé
Paul Zimmermann
INVITED SPEAKERS
Henri Darmon, McGill University
Fritz Grunewald, HHU Düsseldorf
Jean-François Mestre, Univ. Paris 7
Gabriele Nebe, RWTH Aachen
Carl Pomerance, Dartmouth College
Oded Regev, Tel-Aviv University
PROGRAM COMMITTEE
Nigel Boston
John Cremona
Claus Fieker
Guillaume Hanrot, chair
Kevin Hare
Thorsten Kleinjung
Kamal Khuri-Makdisi
François Morain, chair
Takakazu Satoh
Igor Shparlinski
Alice Silverberg
Frederik Vercauteren
Loria
INRIA
cnrs
Nancy-Université

Qui a dit en parlant de la taille des clés RSA ?

[...] s'il y a ne serait-ce qu'un bit de différence, c'est deux fois plus difficile en termes de calcul

Qui a dit en parlant de la taille des clés RSA ?

[...] s'il y a ne serait-ce qu'un bit de différence, c'est deux fois plus difficile en termes de calcul

[...] on s'appuie au sein de Cartes Bancaires sur les travaux de la recherche scientifique internationale [...]

Qui a dit en parlant de la taille des clés RSA ?

[...] s'il y a ne serait-ce qu'un bit de différence, c'est deux fois plus difficile en termes de calcul

[...] on s'appuie au sein de Cartes Bancaires sur les travaux de la recherche scientifique internationale [...]

Pierre Chassigneux, **directeur du Risk management** du GIE Cartes Bancaires, émission « Science Publique » du 7 mai 2010

Comment améliorer la sécurité des cartes bancaires ?

Plan de l'exposé

- courbes elliptiques (ECM)
- crible algébrique (NFS) et RSA768
- logiciels

Factorisation par ECM

ECM = *Elliptic Curve Method*

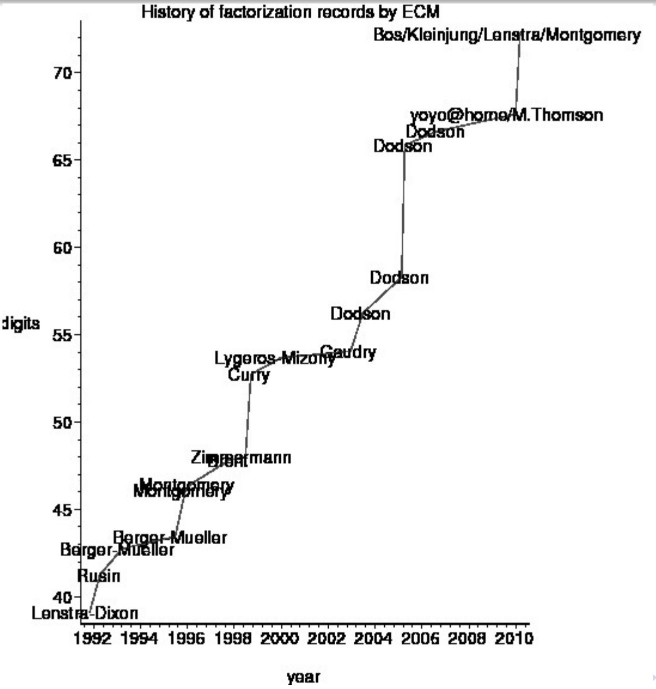
Inventé par H. W. Lenstra, Jr., en 1985.

Utile pour trouver un « *petit* » facteur p dans un *grand* entier n

Complexité $e^{c(\log p)^{1/2}(\log\log p)^{1/2}} M(\log n)$

Record actuel : p de 73 chiffres (n de 291 chiffres).

27 mars 2010 : Michael Vang a trouvé un $p54$ de F_{12} (n de 1187 chiffres) avec GMP-ECM.



History of factorization records by ECM
digits
year
Bos/Kleinjung/Lenstra/Montgomery
yoyo@home/M.Thomson
Dodson
Dodson
Dodson
Dodson
Gaudry
Lygeros-Mizony
Curry
Zimmermann
Brent
Montgomery
Montgomery
Berger-Mueller
Berger-Mueller
Rusin
Lenstra-Dixon
40
45
50
55
60
65
70
1992 1994 1996 1998 2000 2002 2004 2006 2008 2010

ECM : comment ça marche ?

Généralise les méthodes $p-1$ (Pollard, 1974) et $p+1$ (Williams, 1982).

ECM trouve p quand l'ordre de la courbe elliptique modulo p est friable :

$$E(p) = p_1 \cdot p_2 \cdots p_k \cdot q$$

où $p_1, p_2, \ldots, p_k \leq B_1$ et $q \leq B_2$.

Record ECM

Facteur de 73 chiffres de $2^{1181}-1$ trouvé par Bos, Kleinjung, Lenstra, Montgomery le 6 mars 2010 :

$$\begin{aligned} p &= 1808422353177349564546512035512530001\backslash \\ &\quad 279481259854248860454348989451026887 \end{aligned}$$

Étape 1 sur un cluster de PlayStation 3 ($B_1 = 3 \cdot 10^9$)
Étape 2 sur un cluster de processeurs « classiques »($B_2 \approx 10^{14}$)

$$E(p) = 2^4 \cdot 3^2 \cdot 13 \cdot 23 \cdot 61 \cdot 379 \cdot 13477 \cdot 272603 \cdot 12331747 \cdot \\ 19481797 \cdot 125550349 \cdot 789142847 \cdot 1923401731 \cdot 10801302048203$$

Factorisation par NFS

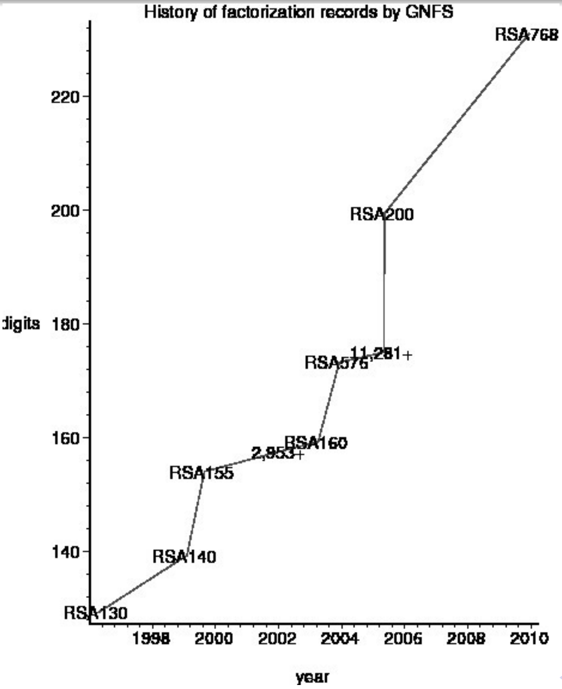
NFS = *Number Field Sieve* (crible algébrique en français)

Inventé par Pollard en 1988.

Utile pour factoriser un nombre RSA $n = pq$, produit de deux nombres premiers de même taille.

Complexité $e^{c'(\log n)^{1/3}(\log \log n)^{2/3}}$.

Record actuel : RSA-768, 232 chiffres (p et q de 116 chiffres).



History of factorization records by GNFS
digits
220
200
180
160
140
year
1998
2000
2002
2004
2006
2008
2010
RSA130
RSA140
RSA155
2,953+
RSA160
RSA576
11,281+
RSA200
RSA768

NFS par l'exemple

Soit à factoriser $n = 5105929$.

Sélection polynomiale.

$$F(x,y) = 173x^2 - 70xy - 63y^2, \quad G(x,y) = x - 172y$$

$f(x) = F(x,1)$ et $g(x) = G(x,1)$ ont une racine commune $\mu = 172$ modulo n

$$\mathrm{Res}(f(x), g(x)) = 5105929$$

Crible de 5105929

Trouver $F(a,b)$ et $G(a,b)$ simultanément friables pour a, b premiers entre eux.

$$F(x,y) = 173x^2 - 70xy - 63y^2, \quad G(x,y) = x - 172y$$

a,b	F(a,b)	G(a,b)
-573, 1213	$2^2 \cdot 3 \cdot 5^2 \cdot 23 \cdot 43^2$	$-1 \cdot 7 \cdot 11^2 \cdot 13 \cdot 19$
-108, 247	$3^2 \cdot 5^3 \cdot 37$	$-1 \cdot 2^5 \cdot 11^3$
-19, 39	$2^2 \cdot 5^3 \cdot 37$	$-1 \cdot 7 \cdot 31^2$
-9, 4	$3^3 \cdot 5^2 \cdot 23$	$-1 \cdot 17 \cdot 41$
7, 8	$3 \cdot 5^2 \cdot 7$	$-1 \cdot 37^2$
7, 12	$-1 \cdot 5^2 \cdot 7 \cdot 37$	$-1 \cdot 11^2 \cdot 17$
108, 127	$3^2 \cdot 5^3 \cdot 37$	$-1 \cdot 2^3 \cdot 11 \cdot 13 \cdot 19$
419, 529	$-1 \cdot 2^2 \cdot 3 \cdot 5^3 \cdot 43^2$	$-1 \cdot 41 \cdot 47^2$

Notion d'idéal

Côté rationnel : p divise $G(a,b) = bg(a/b)$ quand a/b est racine de $g(x)$ modulo p. Exactement une racine pour chaque p.

Côté algébrique : p divide $F(a,b) = b^d f(a/b)$ quand a/b est racine de $f(x)$ modulo p.

$f(x)$ peut avoir de 0 à d racines modulo p. Soit r une racine, on parle d'idéal (p,r) pour l'identifier de manière unique.

Filtrage

1. éliminer les relations trouvées plusieurs fois (a, b identiques)
2. supprimer les singletons (idéal (p, r) apparaissant dans une seule relation) et recommencer au besoin
3. si plus de relations que d'idéaux, supprimer les « cliques »
4. fusionner les relations ayant en commun un idéal (p, r) peu fréquent

Algèbre linéaire

On forme une matrice (creuse) contenant pour chaque relation, les exposants des idéaux modulo 2.
$m + e$ relations pour m idéaux $\longrightarrow$ une dépendance linéaire existe.
Algorithmes de Lanczos et Wiedemann : ne font que des multiplications matrice-vecteur Mx où M est la matrice de départ. Versions « block ».

Ici on multiplie simplement entre elles les huit relations :

$$\prod F(a,b) = (-1)^2 \cdot 2^6 \cdot 3^{10} \cdot 5^{20} \cdot 7^2 \cdot 23^2 \cdot 37^4 \cdot 43^4$$

$$\prod G(a,b) = (-1)^8 \cdot 2^8 \cdot 7^2 \cdot 11^8 \cdot 13^2 \cdot 17^2 \cdot 19^2 \cdot 31^2 \cdot 37^2 \cdot 41^2 \cdot 47^2$$

Racine carrée

Côté rationnel : on multiplie les $a - rb$, où μ est la racine commune de $f(x)$ et $g(x)$ modulo n :

$$\prod_{(a,b)\in S} a - \mu b = u^2 \quad \text{avec } u = 15218777599577552.$$

Côté algébrique : on multiplie $a - xb$ modulo $f(x)$:

$$\prod_{(a,b)\in S} a - xb = \frac{20009231592883459891453125000}{173^8}x$$

$$+ \frac{66414502509679015109578125000}{173^8} \bmod f,$$

dont la racine carrée modulo f est :

$$v(x) = \frac{1}{173^3}(-759208295625x + 109567198125).$$

Racine carrée (suite et fin)

$$n = 5105929$$

$$u = 15218777599577552 \equiv 701937 \bmod n$$

$$v(x) = \frac{1}{1733^3}(-759208295625x + 109567198125).$$

ce qui conduit à :

$$v(\mu) = 4220991 \bmod n.$$

$$\gcd(u + v(\mu), n) = \gcd(701937 + 4220991, 5105929) = 2011$$

Factorisation de RSA-768

NTT : Kazumaro Aoki

EPFL : Joppe Bos, Thorsten Kleinjung, Arjen Lenstra, Dag Arne Osvik

Bonn : Jens Franke

CWI : Peter Montgomery, Andrey Timofeev

INRIA/LORIA/CARAMEL : Pierrick Gaudry, Alexander Kruppa, Emmanuel Thomé, PZ

Sélection polynomiale

Temps utilisé : 40 années cpu (environ 2% du temps total).

$$\begin{aligned}
f(x) &= 265482057982680\,x^6\\
&+ 1276509360768321888\,x^5\\
&- 5006815697800138351796828\,x^4\\
&- 46477854471727854271772677450\,x^3\\
&+ 6525437261935989397109667371894785\,x^2\\
&- 18185779352088594356726018862434803054\,x\\
&- 277565266791543881995216199713801103343120,
\end{aligned}$$

$$\begin{aligned}
g(x) &= 34661003550492501851445829\,x\\
&- 1291187456580021223163547791574810881.
\end{aligned}$$

$$\mathrm{Res}(f(x), g(x)) = \mathrm{RSA768}$$

Crible

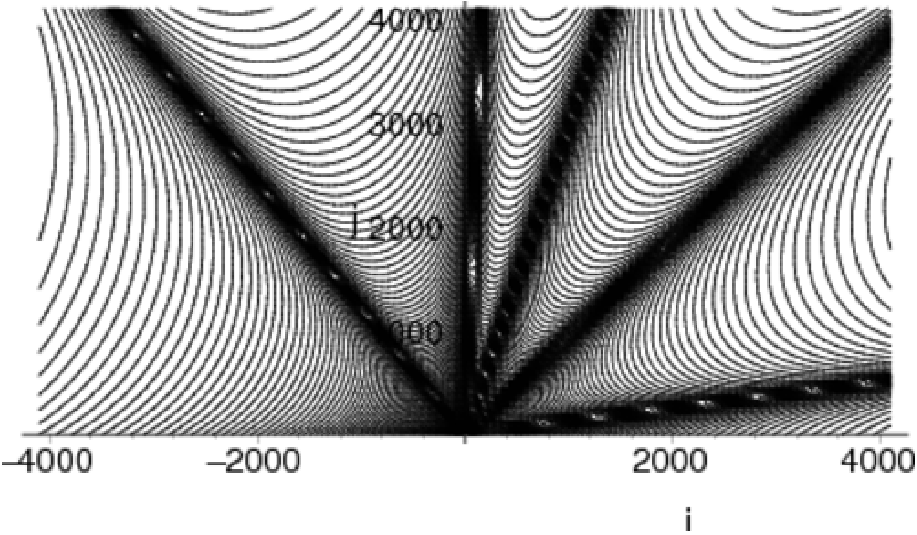
lattice sieving uniquement, *special q* entre 110M et 11100M.

Trivialement parallélisable (plages de *special q*).

Total 64G relations (5To compressées), 1500 années cpu.

INRIA 38%, EPFL 30%, NTT 15%, Bonn 8%, CWI 3%.

En moyenne 4 relations en 3 secondes.


4000
3000
j 2000
1000
−4000
−2000
2000
4000
i

Crible : détails techniques

$I = 2^{16}, J = 2^{15}$: espace de crible de 2^{31}.

Factor base bounds : 200M (rationnel) et 1100M (algébrique) sur machine avec 2Go de mémoire, sinon 100M et 450M.

Large prime bounds : 2^{40} des deux côtés.

Cofactor bounds : 100-110 bits du côté rationnel, 130-140 bits du côté algébrique.

$\Longrightarrow$ jusque 4 *large primes* en plus du *special q*.

Exemple de relation

$F(104262663807, 271220)$ a 81 chiffres :

301114673492631466171967912486669486315616012885653409138028100146264068435983640

$2^3 \cdot 3^2 \cdot 5 \cdot 1429 \cdot 51827 \cdot 211373 \cdot 46625959 \cdot 51507481$
$\cdot 3418293469 \cdot 4159253327 \cdot 10999998887 \cdot 11744488037 \cdot 12112730947$

$G(104262663807, 271220)$ (42 chiffres) :

$$-350192248125072957913347620409394307733817$$

$-1 \cdot 11 \cdot 1109 \cdot 93893 \cdot 787123 \cdot 9478097 \cdot 2934172201 \cdot 13966890601$

Filtrage

Doublons : 27.4% (environ 10 jours de calcul).

Reste 48G relations pour 35G idéaux.

Après une passe d'élimination des singletons : 29G relations pour 14G idéaux.

Au final : 25G relations pour 10G idéaux.

Clique removal : 2.5G relations pour 1.7G idéaux.

Total 10 jours pour singletons et clique.

Merge

Début d'une élimination de Gauss.

Idée : on fusionne les 2 relations contenant un idéal (p, r) n'apparaissant que 2 fois, les 3 relations ...

$\Longrightarrow$ matrice de 193M lignes/colonnes avec 144 éléments non nuls par ligne (105Go).

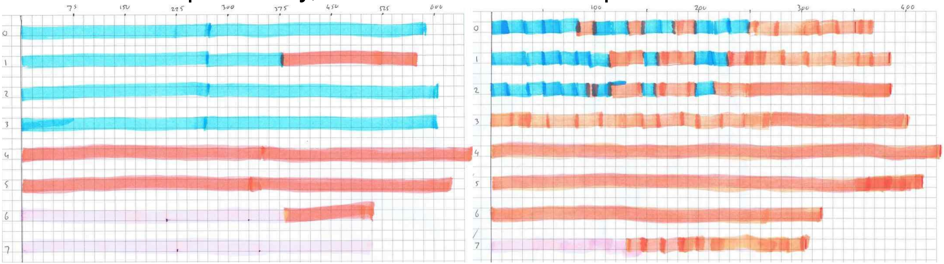
En se limitant aux relations contenant des idéaux $< 2^{34}$, on aurait pu finir la factorisation.

Cela concerne 2% des relations, soit 100 Go seulement.

$\Longrightarrow$ matrice de $253M$ lignes/colonnes avec 147 éléments non nuls par ligne.

Mais plus de travail pour l'algèbre linéaire.

Block Wiedemann avec 8 séquences en parallèle.
Calcul distribué entre INRIA (bleu), EPFL (orange) et NTT (violet).
Temps total (*wall clock*) de 119 jours, dont 17h pour Berlekamp-Massey, environ 155 années cpu.



(Voir appendices B et E de l'article.)

RSA-768

Le 12 décembre 2009 :

RSA768 =
1230186684530117755130494958384962720772853569595334792197
3224521517264005072636575187452021997864693899564749427740
6384592519255732630345373154826850791702612214291346167042
9214311602221240479274737794080665351419597459856902143413
=
3347807169895689878604416984821269081770479498371376856891
2431388982883793878002287614711652531743087737814467999489
*
3674604366679959042824463379962795263227915816434308764267
6032283815739666511279233373417143396810270092798736308917

Logiciels (ECM)

- GMP-ECM : état de l'art pour la méthode ECM
- Prime95/mprime : optimisé pour $2^n \pm 1$

Logiciels (NFS)

- GGNFS (Chris Monico) : comprend le *lattice siever* de Franke et Kleinjung
- msieve (Jason Papadopoulos) : très efficace pour la sélection polynomiale et le filtrage
- CADO-NFS : très efficace pour la sélection polynomiale, la cofactorisation lors du crible (ECM), et l'algèbre linéaire (block Wiedemann)

CADO-NFS

Développé par CARAMEL et TANC (F. Morain) dans le cadre d'une ANR à partir de 2007.
Code sous LGPL, disponible via
`http://cado.gforge.inria.fr/`
Utilisé par Shi Bai (ANU, Canberra) pour (re)factoriser RSA-180.
Les plus de CADO-NFS :

- sélection polynomiale de Kleinjung 2006 et 2008 (en cours)
- implantation indépendante du *sieving by vectors*
- algèbre linéaire via block Wiedemann (MPI + threads)
- racine carrée *naïve* mais efficace

Sélection polynomiale : Kleinjung 2008

Polynôme pour RSA-768 trouvé avec CADO-NFS et msieve (J. Papadopoulos) :

```
# norm 2.945639e-17 alpha -8.954331 e 3.253e-17 rroots 4
skew: 18896073.64
c0 1480989204576286010643240604994098083906841260168220 8
c1 87045342272299622748191182646254248954869076 4
c2 -74531506007439821387629066108322855421 2
c3 -11979108212476678844571704907523
c4 646547959863963043020098 5
c5 266997560749144 63
c6 61633680
Y0 -16469945186136645086261941317897879755
Y1 279273739009 9
```

Sur Core 2 (2.83 Ghz) : 1.49 relation/seconde (contre 2.03 pour le polynôme utilisé pour la factorisation).

Racine carrée « naïve »

Côté rationnel : on accumule le produit des $a - b\mu$, et on prend la racine carrée.

339.965.199 paires (a, b)

Taille du produit : 47.966.524.207 bits (6Go)

Avec GMP 5.0.0 + FFT patch (Kruppa, Gaudry, PZ) sur machine 32Go :

Accumulation : 2 heures.

Racine carrée : 30 minutes.

Racine carrée « naïve » : côté algébrique

Méthode naïve :

1. on accumule le produit des $a - bx$ en réduisant modulo $f(x)$
2. on choisit un premier p inerte
3. on calcule la racine carrée modulo p et on lifte modulo p^k

Algorithme original implanté par E. Thomé dans CADO-NFS, où on réduit modulo plusieurs p_i via CRT.

RSA-768 : 6h30 de *wall clock time* sur 18 nœuds 32G (tout premier essai, plein d'optimisations possibles).

Coppersmith « factorization factory »

Idée : pour factoriser plusieurs nombres de même taille, on utilise le même polynôme linéaire $g(x) = \ell x - m$, et on sauvegarde les paires (a, b) telles que $G(a, b)$ est friable.
Peut-on réutiliser les relations de RSA-768 ?

$$g(x) = 34661003550492501851445829x - 1291187456580021223163547791574810881$$

Pas évident à cause du coefficient dominant ℓ de $g(x)$.
Condition nécessaire : $n \equiv a_d m^d \bmod \ell$ pour un polynôme algébrique $f(x) = a_d x^d + \cdots$
A priori a_d sera de même taille que ℓ.
La *factorization factory* reste-t-elle intéressante avec les progrès en sélection polynomiale ?