Comment fabriquer de grands nombres premiers ?

Michel LAFOND
mlafond001@yahoo.fr

Résumé : Une méthode simple pour obtenir des nombres premiers aussi grands que la calculette le permet, avec une preuve "rapide" de leur primalité. Cet article donne aussi des applications numériques et, pour l'anecdote, exhibe un nombre premier de 100 chiffres commençant par 2012... et se terminant par 2013.

Mots clés : Pocklington, nombres premiers.

1. Pourquoi vouloir des grands nombres premiers ?

Grand signifie ici "ayant plusieurs centaines de chiffres".
Ma réponse à la question est "pour le plaisir". Mais dans le domaine de la cryptographie, ces nombres ont une utilité certaine. Certains professionnels utilisent pour crypter leurs messages un codage basé sur un grand entier N qui est le produit de deux grands nombres premiers p et q (De tels nombres N sont appelés semi-premiers ou nombres RSA). Rivest, Shamir et Adleman ont développé un algorithme qui porte leur nom (algorithme RSA). Il permet un codage sûr tant que le nombre N qui est public reste indécomposable en temps raisonnable. L'astuce est que N suffit pour coder, mais que les deux facteurs p et q sont nécessaires pour décoder. La société "RSA data security" suit de très près l'évolution de ceux qui cherchent à factoriser les grands nombres RSA.
En 2005, l'Université de Bonn a réussi à factoriser un nombre RSA de 200 chiffres, et en 2009 Thorsten Kleinjung a réussi à factoriser un nombre RSA de 232 chiffres ce qui lui a valu un prix de 50 000 $, mais ces réussites sont exceptionnelles car on est très loin aujourd'hui de savoir factoriser à coup sûr un entier $N = pq$ de 200 chiffres.
Tout cela pour dire que des gens ont besoin d'avoir à leur disposition de grands nombres premiers, avec **une certitude** quant à leur primalité.

2. Comment être sûr qu'un nombre déclaré premier l'est vraiment ?

Pour tester la primalité de N, tout le monde connaît la méthode des divisions successives consistant à tester la divisibilité de N par tous les nombres premiers 2, 3, 5, 7, 11 … jusqu'à la racine carrée de N.

Si N vaut environ 10^6 la méthode précédente nécessite l'essai des diviseurs premiers jusqu'à 1000. Il y en a 168, ce n'est pas beaucoup.
Si N vaut environ 10^{12} la méthode précédente nécessite l'essai des diviseurs premiers jusqu'à un million. Il y en a environ 72000, ça ne va plus du tout. D'autant plus qu'il faudrait disposer d'une table FIABLE des nombres premiers jusqu'à un million.
Évaluons le temps de calcul en supposant connue la table des nombres premiers jusqu'à 10^6 :
À raison de 5 minutes par division, cela représenterait deux bonnes années de travail (à 8 heures par jour) sans vacances.
Ne parlons pas des entiers de l'ordre de 10^{100} ou 10^{500} qui nous intéressent ici.
Les logiciels comme MAPLE qui vous disent par exemple : 1 000 000 000 000 000 000 001 est premier, ne sont pas exempts de bugs. De plus certains algorithmes prouvant la primalité sont très compliqués (celui de LENSTRA fait appel aux courbes elliptiques voir QUADRATURE N° 34), d'autres utilisent beaucoup de mémoire, autant de sources potentielles d'erreurs.
Pire pour un mathématicien, certains algorithmes dits heuristiques donnent un résultat "presque certain" c'est-à-dire que si N est déclaré premier par ces algorithmes, il y a une probabilité (extrêmement faible) que N soit en fait composé !
Le théorème qu'on va voir ci-dessous permet d'obtenir des nombres premiers aussi grands que l'on veut. Pour l'ordre de grandeur 10^{100}, on a en plus la possibilité de vérifier humainement (à la main) la primalité en un temps raisonnable. De plus aucune table de nombres premiers (au-delà de 100) n'est requise. Que pourrait-on exiger de plus ?

Le théorème magique qui va permettre ce miracle est :

3. Le théorème de POCKLINGTON. (1916)

Toutes les lettres désignent des entiers naturels.
Si $N - 1 = q^n R$ avec q premier et $n \geq 1$
et s'il existe $a > 0$ tel que :

1) $a^{N-1} \equiv 1 \mod N$

2) PGCD $(a^{\frac{N-1}{q}} - 1, N) = 1$

alors : tout facteur premier p de N est de la forme $k\,q^n + 1$

On voit bien l'intérêt d'un tel théorème : on aura à tester q^n fois moins de diviseurs que par la méthode naïve. Pour peu que q^n soit assez grand le gain sera donc considérable.
Et si q^n est supérieur à la racine carrée de *N*, il n'y a même pas de facteur à tester !

Démonstration du théorème :

Sous les hypothèses du théorème, soit p un facteur premier de N.

$a^{N-1} \equiv 1 \mod N \quad \Rightarrow \quad a^{N-1} \equiv 1 \mod p$ (1)

donc a est premier avec p. Cela entraîne $a^{p-1} \equiv 1 \mod p$ (2)

Soit e l'ordre de a modulo p, c'est-à-dire le plus petit entier positif t tel que $a^t \equiv 1 \mod$ p.
On sait que $a^t \equiv 1 \mod p$ si et seulement si t est multiple de e.

D'après (1) et (2) e divise $N-1$ et e divise $p-1$ (3)

Or par hypothèse $\frac{N-1}{q}$ est entier et $a^{\frac{N-1}{q}} \not\equiv 1 \mod$ p

(Sinon p serait facteur commun à $a^{\frac{N-1}{q}} - 1$ et N ce qui est contraire à l'hypothèse 2).
Donc e ne divise pas $\frac{N-1}{q}$ donc q ne divise pas $\frac{N-1}{e}$ (qui est entier d'après (3)).

Si on remplace N – 1 par $q^n R$, on a donc : q ne divise pas $\frac{q^n R}{e}$.

Posons $\frac{q^n R}{e} = u$ c'est à dire $q^n R = u\, e$.
q premier, ne divise pas u donc q est premier avec u ainsi que q^n.
Mais q^n, premier avec u, divise $u\, e$ donc q^n divise e.

Ainsi q^n divise e et e divise $p-1$ d'où l'on déduit q^n divise $p-1$ autrement dit $p - 1 = k\, q^n$ ou encore $p = k\, q^n + 1$. C Q F D.

4. Applications.

Commençons doucement.
On souhaite par exemple un nombre premier de l'ordre de 10^{12}.
Puisque N sera choisi tel que $N - 1 = q^n R$, nous prendrons q^n et R de l'ordre de $\sqrt{N}$.

Choisissons d'abord q premier dans la table des nombres premiers inférieurs à 100.
Allons-y pour $q = 5$.

Choisissons n pour que q^n soit de l'ordre de $\sqrt{N}$ soit ici :

$$5^n \approx 10^6 \text{ ou } n \approx 8{,}58\ldots\ldots$$

$n = 9$ fera l'affaire.
$N - 1 = q^n R$ devra être pair, donc R aussi.
On va donc imposer à R d'être pair et légèrement inférieur à $q^n = 5^9 = 1\ 953\ 125$.
Pourquoi prendre $R < q^n$?
Parce que dans ce cas, si on trouve un nombre a qui vérifie les deux hypothèses du théorème, on aura comme conclusion que tout facteur premier p de N est de la forme $k\,q^n + 1$.
Mieux, les hypothèses et le fait que $R < q^n$ suffisent en fait pour avoir N premier, sinon N aurait un facteur premier $p = k\,q^n + 1$ inférieur ou égal à $\sqrt{N}$ et cela entraînerait :

$$q^n < k\,q^n + 1 = p \le \sqrt{N} = \sqrt{q^n R + 1} \qquad (4)$$

Divisons les deux membres de (4) par $\sqrt{q^n}$. On obtiendrait : $\sqrt{q^n} < \sqrt{R + 1/q^n}$ soit en élevant au carré : $q^n < R + 1/q^n$ donc $q^n \le R$ ce qui est en contradiction avec $R < q^n$.

Cela se précise :
Pour le nombre a du théorème, nous prendrons $a = 2$ [Cela n'a pas d'importance].
Le seul degré de liberté qui reste est le choix de R. Le travail à faire est donc le suivant :

> Balayer les valeurs paires de R depuis $q^n - 1 = 1\ 953\ 124$ en décroissant.
> Pour chaque valeur de R, poser $N = 1 + q^n R = 1 + 5^9 R$.
> Si $2^{N-1} \equiv 1 \mod N$ et PGCD $(2^{\frac{N-1}{5}} - 1, N) = 1$ alors N est premier.

Le programme en annexe fonctionne une fraction de seconde et donne pour $R = 1\ 953\ 114$ le nombre premier $N = 1 + q^n R = 3\ 814\ 675\ 781\ 251$.

Vérifier à la main, par la méthode des divisions successives nécessiterait disons 4 ans, mais en annexe figure une preuve nécessitant moins de 120 multiplications et tout à fait exécutable à la main en une journée !

Voulez-vous un nombre premier de 200 chiffres ?
La même technique et le même programme exécuté avec $q = 83$ et $n = 52$ donne le nombre premier de 200 chiffres ci-dessous décomposé en 4 tranches de 50 chiffres :

38381470267922467264441665594985762925284275145909
88706567736712230410269996048771200754839846435950
00718909794532108160942122820849529565415233502809
35902334467934772554430386153929473554154963932089.

Voulez-vous un nombre premier de 100 chiffres commençant par 2012 et se terminant par 2013 ? Pas de problème. Deux petites modifications au programme (on se place au voisinage de 2012×10^{96} et on filtre pour ne garder que les nombres

premiers congrus à 2013 modulo 10 000) permettent de trouver en quelques secondes que :

2012573273162587865667865637253743574534625566376607583799726818493315009313799550180168722135402**013** est premier.

5. Annexe.

Ce qui est difficile, ce n'est pas d'obtenir de grands nombres premiers, c'est de les CERTIFIER premiers.

Détaillons avec le nombre $N = 3\,814\,675\,781\,251$ vu plus haut.
Ce que nous devons faire pour certifier la primalité de N est de s'assurer de deux choses :

$2^{N-1} \equiv 1 \mod N$ et PGCD $(2^{\frac{N-1}{5}} - 1, N) = 1$

c'est-à-dire $2^{3814675781250} \equiv 1 \mod 3814675781251$ et
PGCD $(2^{762935156250} - 1 ; 3814675781251) = 1$

Cela paraît insurmontable, mais une astuce nommée "exponentiation rapide" permet de le faire.
D'abord, tous les calculs étant faits modulo N, on n'aura jamais à manipuler de grands nombres. Tout au plus, lors de multiplications, on rencontrera des nombres de l'ordre de N^2 qui n'a guère que 26 chiffres.
Ce qui effraie, ce sont les exposants. Mais pour calculer disons $y = x^e$ il suffit de remarquer que :

Si e est pair alors $y = (x^{\frac{e}{2}})^2$ [Une multiplication]

Si e est impair alors $y = x \times (x^{\frac{e-1}{2}})^2$. [Deux multiplications]

Au prix d'une ou deux multiplications, on a divisé l'exposant par 2 !
Les choses ne vont donc par traîner.

L'algorithme d'exponentiation rapide qui ne fait que traduire la remarque ci-dessus est le suivant :

Soit à calculer $Y = X^E$ X, E connus, E entier positif.

```
Soit Y = 1.
   Tant que E > 0 faire
      Si E impair faire Y = Y . X fin Si
      Faire X = X . X
      Faire E = partie entière (E / 2)
   fin Tant que
```

La validation de l'algorithme est dans la remarque que la quantité $Z = Y \cdot X^E$ est invariante lors de l'exécution de l'algorithme.
[Examiner les deux cas E pair, E impair].
Au début cette quantité vaut $Z = X^E$ (puisque $Y = 1$).
A la fin $E = 0$ donc $Z = Y$ et $X^E = Y$. D'où par transitivité $Y = Z = X^E$.
Ainsi pour calculer $Y = 3^{10}$ l'exécution pas à pas donne :

	$Y = 1$	$X = 3$	$E = 10$
E pair		$X = 3 \times 3 = 9$	$E = 5$
E impair	$Y = 1 \times 9 = 9$	$X = 9 \times 9 = 81$	$E = 2$
E pair		$X = 81 \times 81 = 6561$	$E = 1$
E impair	$Y = 9 \times 6561 = 59049$	$X = 6561 \times 6561$	$E = 0$

C'est fini et $Y = 3^{10} = 59049$.

Bien sûr, on peut faire toutes les multiplications modulo *N*.

Passons à la certification manuelle de la primalité de $N = 3814675781251$. Rappelons qu'on doit vérifier que :

$$2^{3814675781250} \equiv 1 \mod 3814675781251 \quad \text{et}$$
$$\text{PGCD}\,(2^{762935156250} - 1 ; 3814675781251) = 1$$

Puisque $\left(2^{\frac{N-1}{5}}\right)^5 = 2^{N-1}$ on va déjà calculer $y = 2^{\frac{N-1}{5}} = 2^{762935156250}$ modulo *N*.
On utilise l'algorithme d'exponentiation rapide avec $X = 2$ et $E = 762935156250$.
Tous les calculs sont faits modulo $N = 3814675781251$.
Le tableau ci-dessous montre tous les résultats intermédiaires de l'exécution.

Détail du calcul de $y = 2^{762935156250} \mod N$ (explications page suivante)

Y	**X**	**E**	***Facteur f*** YX – f *N*	***Facteur g*** XX – g *N*
1	2	762935156250		
1	4	381467578125		
4	16	190733789062		
4	256	95366894531		
1024	65536	47683447265		
67108864	4294967296	23841723632		4835730
67108864	1958040653386	11920861816		1005045623839
67108864	1337798422407	5960430908		469162969967

67108864	1752824984932	2980215454		805414563120
67108864	1147174981504	1490107727	20181429	344986183270
1380617863777	2414826231246	745053863	873980496366	1528671389525
2408512542276	1428753916741	372526931	902087602099	535127458180
2814294196667	1095029477901	186263465	807862917205	314335903293
3383920032512	2673088806258	93131732		1873135274368
3383920032512	3262824688196	46565866		2790807281244
3383920032512	889642578172	23282933	789183541324	207478685550
1293198611740	3146316238534	11641466		2595058254103
1293198611740	520785246303	5820733	176549409741	71098381178
561482831229	1007831474131	2910366		266267525340
561482831229	929582804821	1455183	136825464349	226526247726
2137567434410	840768656815	727591	471127771726	185308522878
2459473294924	395788983847	363795	255180909733	41064805691
730218276645	3404891019968	181897	651775876992	3039126657851
55967170368	223376969423	90948		13080343738
55967170368	817101296691	45474		175022614591
55967170368	2598010716140	22737	38116819542	1769392752682
2914833332478	1209880934418	11368		383731661459
2914833332478	1317745493515	5684		455203347611
2914833332478	3028957613864	2842		2405075752879
2914833332478	3160187538867	1421	2414732077741	2617990585175
2647116188435	60852089764	710		970718624
2647116188435	3263817857072	355	2264859605637	2792506523489
1367510850433	3147163108445	177	1128216379497	2596455426130
3092580795938	1338960829395	88		469978631333
3092580795938	2658813528442	44		1853182232097
3092580795938	1102459934017	22		318616306026
3092580795938	3728744637763	11	3022915896648	3644749218788
871459660046	2240258300381	5	511785233875	1315644510889
414637199901	1880973203022	2		927486474178
414637199901	2144441895806	1	233090682934	1205510325960
3806252644772	420693813676	0		

À la fin (en bas à gauche du tableau), on a : $y = 2^{\frac{N-1}{5}} \bmod N = 3806252644772$.

Explication et évaluation du temps de calcul :
On ne compte pas les divisions par 2 qui permettent d'avoir la colonne E. Pour les colonnes X et Y, on a des multiplications à faire MODULO *N*.
Pour un humain, la division avec des grands nombres n'est pas folichonne. Regardons de près ce qui se passe :

La première multiplication à vérifier qui est véritablement modulo N (ligne 7 du tableau) est :
4294967296 × 4294967296 = 1958040653386 modulo 3814675781251.
Comme il ne s'agit que de vérifier, on fait faire le gros du calcul à une machine qui nous donne le quotient g = 4835730 et le reste 1958040653386 de la division de XX par N.
À la main il n'y a qu'à constater que :
X X – g N = 18446744073709551616 – 4835730 N = 1958040653386
Soit deux multiplications et une soustraction.
Pour la colonne Y c'est la même chose. Ainsi à la ligne 11 du tableau, 1380617863777 provient du calcul :
Y X – f N = 67108864 × 1147174981504 – 20181429 × N.
Soit encore deux multiplications et une soustraction.

La vérification de $y = 2^{\frac{N-1}{5}} \bmod N$ = 3806252644772 [le gros du travail] nécessite donc pour l'humain vérificateur moins de 110 multiplications (on néglige les soustractions et divisions par 2) et aucune autre opération.

Il faut encore vérifier que :
PGCD ($2^{\frac{N-1}{5}} - 1, N$) = PGCD (3806252644772, 3814675781251) = 1.
Ici, deux multiplications et une soustraction suffisent grâce à M. Bezout :

3806252644772 × 1278964482833 – 1276140417825 × 3814675781251 = 1

S'il y avait un facteur commun entre 3806252644772 et 3814675781251, on le retrouverait dans la combinaison précédente.

À ce stade, la deuxième hypothèse du théorème est vérifiée.
Il ne reste plus qu'à vérifier $2^N - 1 \equiv 1 \bmod N$.
Or $2^N - 1 = \left(2^{\frac{N-1}{5}}\right)^5 - 1 = y^5 - 1$ et
$y^2 = y \times y \bmod N$ = 809533324672.
Preuve : $y \times y$ – 3797848107312 N = 809533324672.
$y^4 = y^2 \times y^2 \bmod N$ = 1278964482833.
Preuve : 809533324672^2 – 171795518501 N = 1278964482833.
$y^5 = y^4 \times y \bmod N = 1$.
Preuve : 1278964482833 × y – 1276140417825 N = 1.
Six multiplications ont suffi.

C'est terminé.
Un humain peut faire cela en une journée (moins de 120 multiplications et aucune division !).

Ci-dessous, le programme (MAPLE) qui a calculé N. Il utilise la procédure nommée "puiss" dépendant de 3 paramètres (*x*, *e*, *m*) qui calcule $y = x^e \bmod m$ par la procédure dite d'exponentiation rapide.

puiss:=proc(x,e,m) local ee,xx,y: ee:=e: xx:=x: y:=1: while ee>0 do if ee mod 2=1 then y:=y*xx mod m: fi: xx:=xx*xx mod m: ee:=floor(ee/2): od: y: end:	*calcule y = x ^ e mod m*
q:=5: e:=9: qe:=q^e: for r from qe-1 to 2 by -2 do n:=1+qe*r: n1:=n-1: nq:=n1/q: p:=puiss(2,n1,n): if p<>1 then next: fi: p:=puiss(2,nq,n)-1: if gcd(p,n)<>1 then next: fi: print (`q =`,q,`exposant `,e,`r = `,r,n,`est premier`): r:=2: od:	*balayage des valeurs de r* *test première hypothèse* *test seconde hypothèse*

Impression après exécution :
`q =`, 5, `exposant `, 9, `r = `, 1953114, `n = `, 3814675781251, `est premier`

Sitographie :
WIKIPEDIA "NOMBRE RSA" consultée le 3 juillet 2012.