Cours numéro 6 :

Arithmétique et cryptographie

1 Introduction

Si l'on m'avait demandé quand j'étais jeune chercheur, dans les années 1970, à quoi servaient les nombres premiers dans la vie courante, j'aurais répondu sans hésiter, à rien, et j'aurais peut-être ajouté comme un de mes vieux collègues, un peu bougon [1], qu'en tout cas ils ne servaient pas à faire la bombe atomique. En fait, j'aurais dit une bêtise, puisque les nombres premiers, avec le code RSA, jouent maintenant un rôle de premier plan dans tous les secteurs de la communication, de la finance, etc. et que parmi leurs principaux utilisateurs se trouvent justement ... les militaires.

2 La cryptographie

La cryptographie (du grec *crypto*, caché et *graphie*, écrire) est la science des codes secrets. Elle remonte à l'antiquité et Jules César l'a employée pour coder ses messages. Il utilisait le système le plus simple, celui des alphabets décalés d'un ou plusieurs crans (où l'on remplace, par exemple, A par B, B par C, etc). Ainsi peut-on penser qu'il envoya au sénat, après sa victoire sur Pharnace à la bataille de Zela, le message suivant : TCLG TGBG TGAG.

Bien entendu des méthodes beaucoup plus sophistiquées ont été inventées depuis. Le plus souvent ces méthodes utilisent le principe suivant. On code les lettres de l'alphabet de A à Z par les nombres [2] de 1 à 26. On traduit le message en chiffres. Par exemple si le message est A L'AIDE il devient 1 12 1 9 4 5. Ensuite on permute les nombres de 1 à 26 selon une certaine règle. On obtient par exemple ici 25 14 25 17 22 21 avec une règle très simple que je vous laisse deviner [3]. On retraduit alors le message en lettres et on a YNYQVU. On notera que dans ce message on voit tout de suite qu'une

1. Un indice : lui aussi a écrit un cours d'algèbre.
2. Dans la réalité on utilise plus de symboles, par exemple ceux du code ASCII.
3. Une méthode très simple de codage consiste à transformer l'entier z variant entre 1 et 26 en $az + b$ avec a, b entiers et a premier à 26, et à réduire ce nombre modulo 26, voir ci-dessous.

lettre intervient deux fois (le Y, traduction de A). Le défaut de ce genre de méthodes est dans cette remarque : elles ne résistent pas au décryptage par analyse de fréquences qui consiste à identifier quelles sont les lettres qui interviennent le plus.

C'est d'ailleurs ainsi que Marie Stuart, princesse écossaise, reine de France (1559-1560) puis d'Ecosse, a péri. En effet, elle était l'ennemie de la reine d'Angleterre Elisabeth première et elle fut capturée par elle en 1568. En 1586 elle participe de sa prison à un complot contre Elisabeth et communique avec ses partisans au moyen de messages codés. Mais ceux-ci sont interceptés par les anglais et son code est décrypté par Thomas Phelippes. Marie est accusée de complot, condamnée et décapitée en 1587.

Ce procédé est expliqué dans la nouvelle d'Edgar Poë : *Le scarabée d'or*. Il s'agit de déchiffrer un grimoire écrit par le capitaine Kidd, qui indique où se trouve le trésor caché par les pirates. Voici ce message :

53‡‡+305))6*;4826)4‡4‡);806*;48+8 960))85;1‡(;:+*8+83(88)5*+
;46(;88*96 *?;8)*‡(;485);5*+2:*‡(;4956*2(5*-4)8 98*;4069285);)6
+8)4‡‡;1(‡9;48081;8:8‡ 1;48+85;4)485+528806*81(‡9;48;(88;4 (‡?
34;48)4‡;161;:188;‡?;

Le héros de l'histoire, William Legrand, après avoir déterminé que le message est en anglais, part de la remarque que, dans cette langue, la lettre la plus fréquente est le E, ce qui lui donne $E = 8$. Il continue ainsi de proche en proche (notamment en identifiant la suite de caractères ; 48 comme l'article THE).

Par cette méthode, vous devez réussir à déchiffrer le message ci-dessous [4] :
SALCFCFVHLCNEANVHHPLGNZIPUUANAKNRNHHLBNCFVH
NYOANEGLYHKNZKVSOANHUNARNGNHZLHHNVAHGNZFGNH
HNZANOHUALYZLPHKNHNHMPFYHYFYOMKVHTVLSPNYHN
ONYPAUNKPZPOLOPFYH
en sachant qu'en français les lettres statistiquement les plus fréquentes sont, dans l'ordre, E, puis S et A, puis R, I, N et T, puis U, puis O et L, etc.

Bien entendu on peut parfois avoir quelques surprises comme avec le texte suivant :

Un voisin compatissant l'accompagna à la consultation à l'hôpital Cochin. Il donna son nom, son rang d'immatriculation à l'Association du travail. On l'invita à subir auscultation, palpation, puis radio. Il fut d'accord. On l'informa : souffrait–il ? Plus ou moins, dit–il. Qu'avait–il ? Il n'arrivait pas à dormir ? Avait–il pris un sirop ? Un cordial ? Oui, il avait, mais ça n'avait

4. Indication permettant de raccourcir notablement le travail : le codage est de la forme $x \mapsto ax + b$ modulo 26.

pas agi. Avait–il parfois mal à l'iris ? Plutôt pas. Au palais ? Ca pouvait ; Au front ? Oui. Aux conduits auditifs ? Non, mais il y avait, la nuit, un bourdon qui bourdonnait. On voulut savoir : un bourdon ou un faux-bourdon ? Il l'ignorait.

Il fut bon pour l'oto-rhino, un gars jovial, au poil ras, aux longs favoris roux, portant lorgnons, papillon gris à pois blancs, fumant un cigarillo qui puait l'alcool. L'oto-rhino prit son pouls, l'ausculta, introduisit un miroir rond sous son palais, tripota son pavillon, farfouilla son tympan, malaxa son larynx, son naso–pharynx, son sinus droit, sa cloison. L'oto-rhino faisait du bon travail, mais il sifflotait durant l'auscultation ; ça finit par aigrir Anton.

(Il s'agit d'un extrait du livre de 319 pages de Georges Perec (1969), intitulé *La disparition* et qui ne comporte pas la lettre E.)

3 Le code RSA

3.1 Le principe

La méthode RSA dont nous allons parler a été inventée en 1978 par Rivest, Shamir et Adleman (RSA) et repose sur les nombres premiers. La problématique de cette méthode est la suivante.

Imaginons un espion E (Ernesto), loin de son pays et de son chef C (Carlos). Il doit transmettre des messages secrets à C. Pour cela, il a besoin d'une clé pour coder ses messages. Cette clé doit lui être transmise par son chef. Le problème, de nos jours, avec Internet et tous les satellites qui nous tournent autour, c'est qu'on n'est pas sûr du tout que les ennemis n'écoutent pas les messages transmis. Avec la plupart des systèmes de codage, si l'on connaît la clé de codage, on sait aussi décoder les messages. Par exemple, imaginons que la clé soit l'opération qui à une lettre, représentée par un nombre x modulo 26, associe $11x - 8$ (toujours modulo 26), ce qui associe par exemple à la lettre E la lettre U. On calcule alors facilement l'opération inverse [5], ce qui permet de décoder les messages.

L'intérêt du code RSA, au contraire, c'est qu'il est à sens unique : la clé de codage n'est pas une clé de décodage ! Voici le principe de cette méthode.

Le chef C calcule deux grands nombres premiers p et q (disons de l'ordre de 200 chiffres, on verra plus loin qu'on sait faire bien mieux), il calcule ensuite le produit pq (cela ne représente qu'une fraction de seconde pour une machine). Il choisit aussi un nombre e premier avec $p - 1$ et $q - 1$ (il y en a beaucoup, par exemple un nombre premier qui ne divise ni $p - 1$ ni $q - 1$). Il transmet à E la clé de codage, qui est constituée du nombre pq

5. C'est $x \mapsto -7x - 4$, voir ci-dessous.

et du nombre e (mais il garde jalousement secrets les deux nombres p et q). La clé est **publique** : peu importe si l'ennemi l'intercepte. Pour coder le message, E n'a besoin que pq et de e, en revanche, pour le décoder, le chef C a besoin des deux nombres p et q. Le principe qui fonde le code RSA c'est qu'il est beaucoup plus facile de fabriquer de grands nombres premiers p et q (et de calculer pq) que de faire l'opération inverse qui consiste à décomposer le nombre pq en le produit de ses facteurs premiers. Pour illustrer ce point, un bon exemple, avec *xcas* est le nombre de 65 chiffres suivant :

$$c = 33263290819929542686848148817697305155927928386133083389000759097.$$

La machine répond instantanément qu'il n'est pas premier, mais met environ 30 secondes pour le factoriser.

Voici précisément la méthode de codage. Le message est un nombre $a < pq$ et premier [6] avec p et q. Pour le coder, E calcule a^e modulo pq (le reste r de a^e dans la division par pq). Là encore, une machine fait cela instantanément, voir ci-dessous. C'est ce nombre r qu'il envoie à son chef.

Comment faire pour retrouver a à partir de r ? Nous l'expliquons en détail au paragraphe suivant. L'idée est la suivante : comme e est premier avec pq, le théorème de Bézout montre qu'il existe un nombre d tel que $de \equiv 1 \pmod{(p-1)(q-1)}$. On montre que grâce à ce d on peut calculer a en faisant l'opération à l'envers : $a = r^d \pmod{pq}$. Il suffit donc de calculer d. Quand on connaît $(p-1)(q-1)$, trouver d est facile (c'est l'algorithme d'Euclide). Mais voilà : on a $(p-1)(q-1) = pq - p - q + 1$ et pour connaître ce nombre il nous faut $p+q$, donc p et q et ça, on ne sait pas faire et c'est ce qui assure la sécurité du code RSA.

3.2 Quelques résultats arithmétiques

Rappelons d'abord le petit théorème de Fermat (voir par exemple *Mathématiques d'École*) :

3.1 Théorème. *Soient p un nombre premier et $a \in \mathbf{Z}$. Alors p divise $a^p - a$ donc on a $a^p \equiv a$ modulo p. Si de plus a est premier avec p, on a $a^{p-1} \equiv 1 \pmod{p}$.*

On a un corollaire de ce théorème :

6. Pour être sûr de réaliser cela on prendra des messages plus petits que p et q. Par exemple si pq a 200 chiffres, on prendra des messages de (nettement) moins de 100 chiffres. Ce seront des messages élémentaires, il en faudra sans doute plusieurs pour faire un message réel.

3.2 Corollaire. *Soient p et q deux nombres premiers distincts et soit a premier avec pq. Alors on a $a^{(p-1)(q-1)} \equiv 1 \pmod{pq}$.*

Démonstration. Il suffit de montrer que la congruence est vraie modulo p et modulo q. Pour cela on note que, comme a^{p-1} est congru à 1 modulo p, on a aussi $a^{(p-1)(q-1)} = (a^{p-1})^{q-1} \equiv 1^{q-1} = 1 \pmod{p}$. On procède de même pour q.

Le résultat suivant concerne encore les congruences (et c'est aussi la recette pour résoudre des équations du genre $ax \equiv b \pmod{s}$) :

3.3 Proposition. *Soit s un entier > 0 et soit e un entier > 0 premier avec s. Alors il existe un entier $d > 0$ tel que $de \equiv 1 \pmod{s}$.*

Démonstration. On applique le théorème de Bézout à s et e : il existe des entiers λ et μ avec $\lambda s + \mu e = 1$. Si μ est > 0 il suffit de poser $d = \mu$. Sinon, on remplace μ par $\mu + sk$ et λ par $\lambda - ek$ avec k assez grand.

Enfin, le dernier résultat est la base de la méthode RSA :

3.4 Proposition. *Soient p et q deux nombres premiers distincts et soit $a > 0$ premier avec pq. Soit e un entier > 0 premier avec $(p-1)(q-1)$ et soit $d > 0$ tel que de soit congru à 1 modulo $(p-1)(q-1)$ (un tel entier existe par 5.3). Alors, on a $a^{de} \equiv a \pmod{pq}$.*

Démonstration. On a $de = 1 + m(p-1)(q-1)$, avec $m > 0$, donc, en vertu de 3.2 :

$$a^{de} = a \times a^{(p-1)(q-1)m} \equiv a \times 1^m = a \pmod{pq}.$$

3.3 Méthodes de calcul : puissances

3.3.1 Un exemple

Considérons l'exemple suivant [7] : $n = pq = 11639$, $e = 3361$ et supposons que le message a est égal à 2511. Il s'agit de calculer a^e modulo pq. Dans ce qui suit on effectuera les calculs avec l'ordinateur et le logiciel *xcas*.

3.3.2 Avec l'ordinateur, sans programme

Attention, on ne peut pas calculer directement 2511^{3361} sinon la machine répond *integer_too_large_for_display* car on a dépassé sa capacité. Le principe est de réduire à chaque pas modulo n. Une méthode élémentaire, mais

7. En fait, le choix de e est très mauvais. En effet, avec cette valeur il y a de nombreux a dont le codage est égal à a, voir ci-dessous Annexe 5.

déjà efficace, est la suivante. On calcule $irem(2511^{10}, 11639)$ (le reste de la puissance dans la division par 11639). On trouve 5868. On recommence en calculant 5868^{10} modulo n, soit 9609, puis 9609^{10} modulo n, soit 2083. Ce nombre n'est autre que 2511^{1000} modulo n. On élève ce nombre au cube, ce qui donne 1146 (on a ainsi la puissance 3000), on trouve de même la puissance 300 : 11138 et on calcule directement $2511^{61} \equiv 2990 \pmod{n}$. En multipliant les trois on a $2511^{3361} \equiv 9404 \pmod{n}$.

3.3.3 Avec l'ordinateur, la voie des puissances de 2

C'est une méthode plus astucieuse qui va donner un algorithme très rapide. Elle combine deux types d'opérations simples :

1) l'élévation au carré,

2) la multiplication par $a = 2511$.

La méthode est la suivante : on part de e, si e est pair on le divise par 2, sinon on lui retranche 1, il est alors pair, on le divise par 2 et on recommence avec le quotient. On finit par aboutir à 1. (Si on écrit e en base 2, les opérations consistent à supprimer le dernier chiffre si c'est un 0 ou à le changer en 0 si c'est un 1.)

Avec $e = 3361$ on obtient successivement les nombres 3360, 1680, 840, 420, 210, 105, 104, 52, 26, 13, 12, 6, 3, 2, 1, autrement dit, on a écrit, en base 2 :

$$3361 = 1010010001000001.$$

On obtient alors le résultat en partant de la gauche de ce nombre et en multipliant par 2511 chaque fois qu'on rencontre un 1 et en élevant au carré pour chaque 0 (et bien entendu en réduisant modulo n à chaque pas). Voici les intermédiaires : 8422, 11218, 2656, 1102, 8679, 9072, 1815, 388, 8231, 10381, 11299, 10849, 7233, 10623, et enfin 9404. À faire à la main c'est pénible, mais on va écrire un programme qui fait le même travail.

3.3.4 Programmes

Voici deux programmes sur *xcas* pour faire automatiquement ces calculs de puissances. Le premier est plus simple [8], mais le second bien plus rapide (pour le calcul ci-dessus les deux donnent la réponse instantanément, mais pour calculer $12345678987654321^{135792468}$ modulo 98765432123456789 le premier met 17 minutes et 34 secondes, tandis que le second est instantané). Chacun de ces programmes calcule la puissance r-ième de a modulo p. Le premier consiste à multiplier a^k par a et à réduire modulo p à chaque pas :

8. Et facile à retrouver.

```
power(a,r,p):={
Local z;
z:=1;
pour k de 1 jusque r faire
z:=irem(a*z,p);
fpour
retourne z;
} :;
```

Le second programme utilise les puissances de 2, comme expliqué plus haut, pour grimper plus vite [9] :

```
powerv(a,r,p):={
local z;
z:=1;
tantque r>0 faire
si floor(r/2)=r/2 alors
r:=r/2;
a:=irem(a^ 2,p);
sinon
r:=(r-1)/2;
z:=irem(z*a,p);
a:=irem(a^ 2,p);
fsi
ftantque
retourne z;
} :;
```

Ce programme mérite un mot d'explication. On cherche $b := a^r \pmod p$. On entre a, r, p et on pose $z = 1$. On a donc $a^r z = a^r = b$. Dans le programme, les quantités a, r, z évoluent de telle sorte que $a^r z$ reste constant. En effet, si r est pair, z ne change pas, a devient a^2 et r devient $r/2$, donc $a^r z$ est invariant. Si r est impair, $r = 2k + 1$, a devient a^2, r devient k et z devient az, donc $a^r z$ devient $(a^2)^k \times az = a^{2k+1} z = a^r z$.

À la dernière étape on a $r = 0$, donc $a^r z = z$ et cette quantité est bien le b cherché.

3.4 Méthodes de calcul : les coefficients de Bézout

Reprenons notre exemple : $n = pq = 11639$, $e = 3361$ et $a = 2511$. On a trouvé $a^e \equiv 9404 \pmod n$.

9. La commande `floor` est la partie entière et elle sert à tester la parité de r.

Pour inverser le processus, il y a besoin de connaître $(p-1)(q-1)$, donc p et q. Ici, ce n'est pas trop compliqué : on a $p = 103$ et $q = 113$, d'où $(p-1)(q-1) = 11424 = 2^5 \times 3 \times 7 \times 17$. Comme e est premier, il est bien premier avec ce nombre. Il s'agit maintenant de trouver d tel que $de \equiv 1 \pmod{11424}$. Pour cela, on utilise l'algorithme d'Euclide pour trouver les coefficients de Bézout.

3.4.1 L'algorithme d'Euclide

On considère $a, b \in \mathbf{N}$ avec $b \neq 0$. On pose $a = r_0$, $b = r_1$. On effectue la division euclidienne de a par b : $a = bq + r$ avec $0 \leq r < b$. On pose $q = q_1$, $r = r_2$. On a donc $r_0 = r_1 q_1 + r_2$ avec $0 \leq r_2 < r_1$ et $d = pgcd\,(a, b) = pgcd\,(r_0, r_1) = pgcd\,(r_1, r_2)$.

On construit ainsi par récurrence des entiers $r_0, r_1, \cdots, r_k, r_{k+1}$ et $q_1, \cdots, q_k$ avec $r_{k-1} = q_k r_k + r_{k+1}$, $0 \leq r_{k+1} < r_k$ et $d = pgcd\,(r_k, r_{k+1})$. Si on a $r_{k+1} = 0$ on a $d = r_k$ et on s'arrête, sinon on continue l'algorithme en divisant r_k par r_{k+1}.

Comme on a $0 \leq r_{k+1} < r_k < \cdots < r_1$ on voit que l'on obtient nécessairement un reste nul au bout d'au plus r_1 opérations. Si on désigne par r_n le dernier reste non nul on a donc $r_{n-1} = q_n r_n$ d'où, $d = pgcd\,(r_{n-1}, r_n) = r_n$.

3.4.2 Le théorème de Bézout

Pour montrer Bézout, on utilise l'algorithme d'Euclide. On va montrer, par récurrence sur k que, pour tout k avec $0 \leq k \leq n$, il existe des entiers $u_k, v_k \in \mathbf{Z}$ vérifiant $r_k = u_k a + v_k b$.

L'assertion est vraie pour $k = 0$ puisqu'on a $r_0 = a = 1 \times a + 0 \times b$ et pour $k = 1$ puisqu'on a $r_1 = b = 0 \times a + 1 \times b$. Supposons l'assertion prouvée pour tout entier $\leq k$, avec k fixé vérifiant $1 \leq k < n$, et montrons la pour $k+1$. On a $r_{k+1} = r_{k-1} - q_k r_k = u_{k-1} a + v_{k-1} b - q_k (u_k a + v_k b)$ d'où la relation cherchée en posant $u_{k+1} = u_{k-1} - q_k u_k$ et $v_{k+1} = v_{k-1} - q_k v_k$.

Si on applique l'assertion au cas $k = n$, comme on a $r_n = d = pgcd\,(a, b)$, on obtient bien la relation de Bézout cherchée.

3.4.3 Calcul avec *xcas*

Le logiciel *xcas* a une commande qui donne directement les coefficients de Bézout : `iegcd(a,b)` renvoie dans l'ordre u, v, δ tels que $ua + vb = \delta = pgcd(a, b)$. Dans le cas présent on trouve $u = -198$ et $v = 673$. Ce dernier nombre est l'exposant d cherché. On vérifie qu'on a bien $9404^{673} \equiv 2511 \pmod{n}$.

Si l'on y tient [10] on peut aussi écrire un programme. Le suivant utilise des listes de trois termes et il est facile de comprendre son fonctionnement en le testant sur un exemple, à condition de savoir trois choses :

• La condition `b!=0` signifie $b \neq 0$.

• La commande `iquo (a,b)` est le quotient euclidien de a par b.

• L'écriture `la[2]` désigne le terme d'indice 2 de la liste `la` (qui commence à 0). Ici, c'est donc a.

Avec ces précautions, voici le programme :

```
Bezout(a,b):={
local la, lb, lr, q;
la:=[1,0,a];
lb:=[0,1,b];
tantque b!=0 faire
q:=iquo(la[2],b);
lr:=la+(-q)*lb;
la:=lb;
lb:=lr;
b:=lb[2];
ftantque
retourne la;
}:;
```

3.5 Trouver de grands nombres premiers

On sait depuis Euclide qu'il y a une infinité de nombres premiers mais il n'est pas si facile d'en donner explicitement de très grands. Pierre de Fermat (1601-1665) avait cru trouver une formule donnant à coup sûr des nombres premiers. Il prétendait que, pour tout entier n, le nombre [11] $F_n = 2^{2^n} + 1$ était premier. C'est effectivement le cas pour $n = 0, 1, 2, 3, 4$ qui correspondent respectivement aux nombres premiers $3, 5, 17, 257, 65537$, mais ce n'est pas vrai pour F_5 comme l'a montré Euler [12].

On peut faire le calcul à la main jusqu'à 257. Pour voir que 65537 est premier, mais que $2^{32} + 1$, $2^{64} + 1$ et $2^{128} + 1$ ne le sont pas on peut utiliser la fonction `isprime` de *xcas* qui répond presque instantanément. (Jusqu'à

10. Ou si le jury de CAPES y tient ...

11. Seuls les $2^r + 1$ où r est une puissance de 2 ont une chance d'être premiers à cause de la formule $a^m + 1 = (a + 1)(a^{m-1} - a^{m-2} + a^{m-3} - \cdots - a + 1)$ lorsque m est impair qui montre que $a + 1$ divise $a^m + 1$ (ce qu'on retrouve encore plus simplement grâce aux congruences).

12. Pour comprendre pourquoi 641 divise F_5 et d'où il sort, voir Annexe 1 ci-dessous.

$2^{4096}+1$ il donne une réponse négative en moins d'une seconde, pour $2^{16384}+1$ il met seize secondes.) L'ordinateur factorise instantanément :

$$2^{32}+1 = 641 \times 6700417$$

$$2^{64}+1 = 274177 \times 67280421310721$$

$$2^{128}+1 = 59649589127497217 \times 5704689200685129054721$$

et il met moins de 5 secondes pour $2^{256}+1$. En revanche, il cale sur le suivant [13], à savoir $2^{512}+1$ (qui a quand même 150 chiffres, c'était il n'y a pas si longtemps le record du monde de factorisation). En tous cas, on constate sur cet exemple que la primalité est plus facile que la factorisation !

On notera qu'à l'heure actuelle on ne sait pas exactement lesquels parmi les F_n sont premiers ou non. La réponse est seulement connue pour un nombre fini de n et, sauf pour les 5 premiers, tous les F_n en question sont composés. Cet exemple montre déjà deux choses, d'abord qu'un grand mathématicien peut dire des bêtises, et ensuite qu'il y a des questions, somme toute assez simples, pour lesquelles on n'a pas de réponse. J'y reviens plus loin.

Il y a donc des records du plus grand nombre premier connu qui sont détenus par d'énormes ordinateurs [14] (en général il s'agit de certains nombres de Mersenne (1588-1648) : $M_n = 2^n - 1$). Le plus ancien record est celui de Cataldi en 1588 avec $M_{19} = 524287$. Il y eut ensuite Lucas (1876) avec M_{127} qui a 39 chiffres. Le record, en 1999, était le nombre de Mersenne $M_{6972593}$ qui a tout de même plus de 2 millions de chiffres ! Je ne vais pas l'écrire [15], mais je peux tout de même dire qu'il commence par 437075 et finit par 193791. Je vous laisse montrer cela à titre d'exercice (pas si facile, voir Annexe 2).

En 2008, le record est $M_{43112609}$ qui a 12 millions de chiffres.

3.6 Factoriser des grands nombres ?

Ce qu'il faut comprendre, c'est que les ordres de grandeur des nombres premiers que l'on sait exhiber, d'une part, et des nombres que l'on sait factoriser, d'autre part, ne sont pas du tout les mêmes, comme on l'a déjà senti à propos des nombres de Fermat. Pendant longtemps, factoriser un nombre de l'ordre d'un milliard était considéré comme à peu près impossible. Ainsi Mersenne, en 1643, avait donné à Fermat, comme un défi, de factoriser le

13. J'ai laissé tourner la machine toute une nuit sans succès. Il faut noter qu'il y a juste un an, *xcas* mettait 2 heures 7 minutes et 40 secondes pour factoriser $2^{128}+1$: on voit les progrès des machines et des algorithmes !

14. Ce n'est pas seulement la puissance des ordinateurs qui est en jeu, mais surtout la qualité des algorithmes qu'ils utilisent (donc des mathématiques qui sont derrière).

15. Il y faudrait un livre de 500 pages !

nombre [16] 100895598169 et le même défi avait été présenté comme impossible par Stanley Jevons en 1874 avec le nombre 8616460799. Pourtant, aujourd'hui, une calculatrice un peu perfectionnée factorise ces deux nombres sans difficulté.

Cependant, le record absolu de factorisation (daté du 12 décembre 2009) est bien loin de celui de primalité, c'est un nombre n de 232 chiffres, produit de deux nombres p et q de 116 chiffres, et encore a-t-il fallu pour cela faire travailler plusieurs centaines d'ordinateurs en parallèle pendant 2 ans sur un algorithme très complexe, ce qui représente environ 1500 années de temps de calcul pour une machine seule.

Voilà ces nombres :

$$\begin{aligned}
&1230186684530117755130494958384962720772853569\ 5953347921973224\\
&5215172640050726365751874520219978646938995647494277406384592519\\
&2555732630345373154826850791702612214291346167042921431160222124\\
&0479274737794080665351419597459856902143413\\
&= 3347807169895689878604416984821269081770479498371376856891\\
&2431388982883793878002287614711652531743087737814467999489\\
&\times\\
&3674604366679959042824463379962795263227915816434308764267603228381573966651127923333417143396810270092798736308917
\end{aligned}$$

On notera tout de même qu'il y a seulement 30 ans, on estimait qu'il faudrait 50 milliards d'années pour factoriser un nombre de 150 chiffres. Les progrès accomplis par les mathématiciens et les ordinateurs sont donc considérables. Bien entendu, cela ne remet pas en cause la fiabilité du code RSA : si on sait factoriser un nombre $n = pq$ de 250 chiffres il suffit de choisir des nombres p et q plus grands. On a vu qu'il y a de la marge puisqu'on sait expliciter des nombres premiers avec des millions [17] de chiffres. Les banques travaillent déjà avec des clés n de l'ordre de 300 chiffres et les militaires avec des clés de 600 chiffres.

Et si un mathématicien améliorait fondamentalement les algorithmes de factorisation et leur permettait de rattraper les tests de primalité ? Alors, pour un temps au moins, il ne serait pas loin d'être le maître du monde [18] !

16. Fermat avait répondu au défi, et semble-t-il très rapidement. On ignore comment il a fait. On trouvera en annexe une hypothèse que je soumets au lecteur, sans la moindre garantie.

17. En fait, les nombres de Mersenne sont proscrits comme clés RSA car ils sont trop particuliers, mais les logiciels comme Pari fournissent sans problème des nombres premiers de 5000 chiffres et *xcas* en donne de plus de 1500 chiffres en 40 secondes.

18. N'ayez pas trop d'espoir tout de même. On pense qu'il y a vraiment une raison profonde qui fait que la factorisation est beaucoup plus difficile que la primalité.

Si vous pensez détenir une méthode, voici un nombre à factoriser, qui vous rapportera la modique somme de 200 000 dollars :

25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748808428401797429100642458691817195118746121515172654632282216869987549182422433637259085141865462043576798423387184774447920739934236584823824281198163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386356373289912154831438167899885040445364023527381951378636564391212010397122822120720357

3.7 Combien de nombres premiers dans une dizaine ?

Ce texte est inspiré d'une conférence que je donne régulièrement devant des collégiens ou des lycéens. Mon but est de leur expliquer deux choses :

1) les mathématiques c'est utile,

2) il y a beaucoup de choses qui ne sont pas encore connues en mathématiques.

Ce qui suit vise à étayer ce second point. Ce n'est pas si facile car les questions actuellement ouvertes nécessitent le plus souvent, simplement pour comprendre la question, des connaissances qui sont à cent lieues au-dessus de celles des lycéens (on pensera à la conjecture de Poincaré récemment prouvée par Perelman, par exemple). Il n'y a guère qu'en arithmétique où l'on peut trouver des poblèmes sérieux et faciles à présenter.

Si on regarde combien il y a de nombres premiers dans une dizaine, on peut éliminer les multiples de 2 et ceux de 5. Il reste donc à regarder les nombres se terminant par $1, 3, 7, 9$. Il se peut qu'ils soient tous premiers, c'est le cas de $11, 13, 17, 19$, mais c'est rare. Si l'on cherche ensuite, cela n'arrive plus jusqu'à 100 (sont non premiers : 21, 33, 49, 51, 63, 77, 81, 91, par exemple). En revanche, $101, 103, 107$ et 109 sont tous premiers (il suffit de voir qu'ils ne sont pas multiples de 3 ni de 7). La question est donc : peut-on trouver une infinité de dizaines riches contenant 4 nombres premiers ? La calculatrice (et l'ordinateur) permettent d'explorer le problème (jusqu'à 10000 il y a 11 dizaines riches), mais pas de le résoudre et, à l'heure actuelle, on ne sait pas s'il y a une infinité de telles dizaines. Pire, on ne sait même pas s'il y a une infinité de nombres premiers jumeaux (c'est-à-dire avec 2 d'écart comme 11 et 13, ou 59 et 61).

Ce dernier problème date des Grecs, il est très facile à exprimer, mais très difficile, puisque personne n'a su le résoudre encore. Bien entendu, ce problème a été exploré avec l'ordinateur (jusqu'à 10^{15} on a trouvé environ 1177 milliards de paires de jumeaux), mais cela ne permet pas de répondre à la question : les capacités des ordinateurs, même immenses, sont finies.

À propos de la répartition des nombres premiers, si l'on regarde le début des tables on peut avoir l'impression qu'il y a des nombres premiers dans toutes les dizaines. Eh bien, ce n'est pas vrai et il n'y a pas besoin d'aller chercher très loin (il n'y en a pas entre 200 et 210). En fait, même si on prend un nombre même très grand (disons par exemple 1000, voire un milliard), on peut toujours trouver 1000 nombres (ou un milliard) de suite sans aucun nombre premier. Cette affirmation semble ambitieuse ? Elle est pourtant bien facile à prouver si l'on pense aux factorielles.

Sur ces deux exemples, on voit combien il peut être délicat de prévoir, face à un problème de mathématiques inconnu, quelle va être sa difficulté.

4 Annexe 0, codage par application affine

4.1 Le codage

4.1 Proposition. *Soient a, b deux entiers compris entre 0 et 25. L'application $\Phi : \mathbf{Z}/26\mathbf{Z} \to \mathbf{Z}/26\mathbf{Z}$ qui à x associe $ax + b$ (modulo 26) est bijective si et seulement si a est premier à 26 (c'est-à-dire impair et différent de 13).*

Démonstration. Comme l'application est entre deux ensembles finis de même cardinal, elle est bijective si et seulement si elle est injective. La condition $ax + b = ay + b$ signifie $a(x - y) \equiv 0 \pmod{26}$, donc $a(x - y)$ multiple de 26. Si a est premier avec 26, cela donne $x \equiv y \pmod{26}$ en vertu du théorème de Gauss. Si a n'est pas premier à 26, il existe x, non nul modulo 26, tel que $ax \equiv 0 \pmod{26}$ et on a alors $\Phi(x) = \Phi(0)$.

4.2 Le décodage

Le décodage se fait en utilisant Bézout. On a y et on cherche x vérifiant $y = ax + b$. Si l'on connaît l'inverse de a modulo 26 on a $x = a^{-1}(y - b)$. Pour cela on écrit l'égalité de Bézout : $\lambda a + 26\mu = 1$ et λ est un inverse de a.

4.2 *Exemple.* Si l'on pose $\Phi(x) = 11x - 8 = 11x + 18$, on peut calculer l'inverse de 11 par la méthode artisanale : on se récite les multiples de 26 jusqu'à ce qu'on voie un multiple de 11 qui le jouxte : 26, 52, 78, stop. On a

donc $3 \times 26 - 7 \times 11 = 1$, de sorte que l'inverse de 11 est -7 (ou 19 si l'on préfère) et on a alors $\Phi^{-1}(y) = -7(y+8) = -7y - 4$.

5 Annexe 1, Euler et les nombres de Fermat

On a vu, grâce à l'ordinateur, que 641 divise $F_5 = 2^{32} + 1$. La question est de savoir comment on peut trouver ce facteur et comment montrer directement qu'il divise F_5.

5.1 Montrer que 641 divise F_5

On pose $p = 641$ (on vérifie que c'est bien un nombre premier). On note les deux formules : $641 = 625 + 16 = 5^4 + 2^4$ et $641 = 640 + 1 = 5 \times 2^7 + 1$. On calcule 2^{32} modulo p. On a $5 \times 2^7 \equiv -1 \pmod{p}$. En élevant cette relation à la puissance 4 on a $5^4 \times 2^{28} \equiv 1 \pmod{p}$. Mais, on a $5^4 \equiv -2^4 \pmod{p}$ et donc $2^{32} \equiv -1 \pmod{p}$. Cela signifie exactement que p divise $2^{32} + 1$.

5.2 D'où sort le 641 ?

On suppose que F_5 admet un facteur premier p et on travaille dans le groupe multiplicatif $G = (\mathbf{Z}/p\mathbf{Z})^*$. Dans ce groupe on a donc $2^{32} = -1$, donc $2^{64} = 1$. On voit que 2 est un élément d'ordre 64 de G. Comme l'ordre d'un élément divise l'ordre du groupe, c'est que 64 divise $p - 1$. Cela signifie que p est congru à 1 modulo [19] 64. On examine les nombres premiers possibles : 193, 257, 449, 577, 641, le cinquième est le bon.

5.3 Éliminer les autres possibles

Pour $p = 257$ c'est évident car on a $2^8 \equiv -1 \pmod{p}$ donc $2^{16} \equiv 1$ et $2^{32} \equiv 1$.

Pour 193 on peut par exemple raisonner ainsi. On a $192 = 3 \times 64$ et donc $3 \times 2^6 \equiv -1 \pmod{p}$. On en déduit $3^4 \times 2^{24} \equiv 1$ et, si l'on suppose $2^{32} = 2^{24} \times 2^8 \equiv -1$ on trouve $2^8 + 3^4 = 337 \equiv 0 \pmod{p}$, ce qui est clairement faux.

19. En fait, quand on est plus instruit, on sait même que p est congru à 1 modulo 128, voir ci-dessous.

5.4 Amélioration

Je montre que si p divise $2^{32}+1$, $p-1$ est multiple de 128. Pour cela, il suffit de montrer que si on a $p \equiv 1 \pmod 8$, 2 est un carré modulo p. En effet, si 2 est le carré de a, on a $2^{32} = a^{64} = -1$ et a est d'ordre [20] 128.

Pour cela, deux voies. Soit on sait que $\mathbf{F}_p^*$ est cyclique d'ordre $p-1$, donc contient un élément ζ d'ordre 8, qui vérifie donc $\zeta^4+1=0$, donc $\zeta^2+\zeta^{-2}=0$, et on voit que $a=\zeta+\zeta^{-1}$ vérifie $a^2=2$.

Soit on sait ça, mais on fait semblant de ne pas le savoir et on regarde une racine huitième explicite, à savoir 2^8 et son inverse $2^{56} = -2^{24}$ et on montre que $a = 2^8 - 2^{24}$ a pour carré 2. En effet, on a $(2^8-2^{24})^2 = 2^{16}+2^{48}-2.2^{32}$ et cela résulte de $2^{32}=-1$.

6 Annexe 2, le grand nombre de Mersenne

Je montre que $M := M_{6972593} = 2^N - 1$ commence par 437075 et finit par 193791.

6.1 Il finit par 193791

Il s'agit de calculer M modulo 10^6. Le problème a été étudié ci-dessus, le programme *powerv* donne instantanément le résultat : $2^{6972593} \equiv 193792 \pmod{10^6}$, d'où le résultat. Bien entendu, on peut aussi le faire à la main, de proche en proche.

6.2 Il commence par 437075

On commence par calculer le nombre m de chiffres de M, ou de 2^N. On a l'encadrement : $10^{m-1} \leq 2^N < 10^m$ et, en passant au logarithme, on trouve $(m-1)\ln 10 \leq N \ln 2 < m \ln 10$, d'où $m = 2098960$.

Si l'on veut calculer les 6 premiers chiffres de M, qui forment un nombre p, on écrit alors l'encadrement :

$$p \times 10^{2098954} \leq M < (p+1) \times 10^{2098954}$$

ce qui donne, en passant au logarithme, $\ln p = 12,987862$ et $p = 437075$.

20. Avec cette ruse, il y a seulement à éliminer 257 et c'est évident car c'est lui-même un nombre de Fermat.

7 Annexe 3, le nombre de Ramanujan

La problématique de ce paragraphe est celle des tests de primalité : comment décider si un nombre p est premier ou non ? Beaucoup des méthodes utilisées pour traiter cette question tournent autour du petit théorème de Fermat. En effet, si p est premier, on a $a^p \equiv a \pmod{p}$, de sorte que si a^p n'est pas congru à a c'est que p n'est pas premier. On peut déjà appliquer ce critère avec $a = 2$. Par exemple, on a $2^{11639} \equiv 194 \pmod{11639}$, ce qui prouve que 11639 n'est pas premier. Malheureusement, on peut avoir $2^p \equiv 2 \pmod{p}$ sans que p soit premier (par exemple avec $p = 341 = 11 \times 31$). Bien entendu, on peut ensuite essayer avec 3. On montre par exemple ainsi que les nombres de Fermat $2^{2^n} + 1$ ne sont pas premiers pour $n = 5, 6, 7, 8, 9$ etc. Le problème c'est qu'il y a des nombres p non premiers qui vérifient $a^p \equiv a \pmod{p}$ pour tout a. Ce sont les nombres de Carmichaël, dont le plus petit est $561 = 3 \times 11 \times 17$. On a montré récemment qu'il y avait une infinité de tels nombres et parmi eux, le fameux nombre de Ramanujan.

C'est le nombre [21] $1729 = 12^3 + 1^3 = 10^3 + 9^3$. Ce nombre est un nombre de Carmichaël, c'est-à-dire un nombre non premier (on a $n = 1729 = 7 \times 13 \times 19 = pqr$) et qui pourtant vérifie le petit théorème de Fermat pour tous les entiers : on a $a^n \equiv a \pmod{n}$ pour tout a.

La preuve de ce résultat est facile avec le lemme chinois. En effet, on a un isomorphisme :

$$\mathbf{Z}/1729\mathbf{Z} \simeq \mathbf{Z}/7\mathbf{Z} \times \mathbf{Z}/13\mathbf{Z} \times \mathbf{Z}/19\mathbf{Z}.$$

Si maintenant on prend $x \in \mathbf{Z}/1729\mathbf{Z}$, on le décompose en $x = (a, b, c)$ dans

21. Ce nombre est au cœur d'une belle histoire qui met en scène deux grands mathématiciens : Hardy et Ramanujan. Srinivasa Ramanujan était un jeune indien, de modeste origine, qui vivait en Inde au début du vingtième siècle (1887-1920). Il apprit les mathématiques en autodidacte à partir de deux livres élémentaires et se passionna notamment pour l'arithmétique, obtenant des résultats qu'il jugea assez intéressants pour les envoyer en 1913 au ponte des mathématiques britanniques de l'époque : Godfrey-H. Hardy. Celui-ci, l'archétype du britannique de cette époque, considéra d'abord avec condescendance ce que lui envoyait ce jeune indien inconnu, mais il s'aperçut très vite qu'à côté de choses bien connues, il y avait des formules nouvelles, qui semblaient exactes et intéressantes, et que pourtant lui, Hardy, ne savait pas prouver. Bref, il se rendit compte qu'il avait affaire à un vrai génie et s'empressa de le faire venir en Angleterre où ils collaborèrent pendant six ans. Malheureusement, le climat de l'Angleterre ne valut rien à Ramanujan qui contracta la tuberculose et en mourut à 32 ans. Un jour qu'il lui rendait visite à l'hôpital, Hardy, qui ne savait pas bien quoi raconter et qui connaissait sa passion pour les nombres, lui dit :

– *Je suis venu en taxi, mais le numéro n'avait rien d'extraordinaire, c'était* 1729.

– *Détrompez-vous, répliqua Ramanujan, ce nombre est remarquable car c'est le plus petit entier qui s'écrit de deux manières différentes comme somme de deux cubes.*

le produit et on a $x^n = (a^n, b^n, c^n)$. Il suffit de montrer qu'on a $z^n = z$ dans les trois facteurs. C'est le fait que $p-1 = 6$, $q-1 = 12$ et $r-1 = 18$ divisent $n-1 = 12^3$. En effet, si a, b, c sont respectivement premiers à p, q, r on a $a^{p-1} \equiv 1 \pmod{p}$, $b^{q-1} \equiv 1 \pmod{q}$ et $c^{r-1} \equiv 1 \pmod{r}$, donc $a^{n-1} \equiv 1 \pmod{p}$ et de même pour les autres. On en déduit $x^{n-1} \equiv 1 \pmod{n}$ et le résultat. Si l'un des a, b, c n'est pas premier modulo p, q, r, il devient 0 dans le quotient et vérifie évidemment, par exemple, $a^n \equiv a \pmod{p}$ et la conclusion reste valable.

Le lecteur prouvera, à titre d'exercice :

7.1 Proposition. *Soit n un entier composé. Alors n est de Carmichaël si et seulement si n n'a pas de facteur carré et si, pour tout diviseur premier p de n, $p-1$ divise $n-1$.*

7.2 *Exemple.* Si m est tel que $6m+1$, $12m+1$ et $18m+1$ soient premiers, $n = (6m+1)(12m+1)(18m+1)$ est de Carmichaël.

8 Annexe 4, la factorisation de Fermat

8.1 Le problème

Rappelons la question de Mersenne :

Le nombre 100895598169 *est-il premier ?*

Voici la réponse de Fermat :

À cette question je réponds que ce nombre est composé et se fait du produit des deux : 898423 *et* 112303 *qui sont premiers. Je suis toujours, mon révérend Père, votre très humble et très affectionné serviteur.*

La question est : comment a-t-il fait ?

8.2 Une procédure bien connue de Fermat

8.2.1 Une citation

Je recopie ici un extrait d'une lettre du même Fermat au même Mersenne en 1664 :

Cela posé, qu'un nombre me soit donné, par exemple 2027651281*, on demande s'il est premier ou composé, et de quels nombres il est composé, au cas qu'il le soit. J'extrais la racine, pour connaître le moindre des dits nombres, et trouve* 45029 *avec* 40440 *de reste, lequel j'ôte du double plus* 1 *de la racine trouvée, savoir de* 90059 *: reste* 49619*, lequel n'est pas carré, parce qu'aucun carré ne finit par* 19*, et partant je lui ajoute* 90061*, savoir*

2 *plus* 90059 *qui est le double plus* 1 *de la racine* 45029. *Et parce que la somme* 139680 *n'est pas encore carrée, comme on le voit par les finales, je lui ajoute encore le même nombre augmenté de* 2, *savoir* 90063 *et je continue ainsi d'ajouter tant que la somme soit un carré, comme on peut voir ici. Ce qui n'arrive qu'à* 1040400 *; qui est carré de* 1020 *et partant le nombre donné est composé ; car il est aisé, par l'inspection des dites sommes, de voir qu'il n'y a aucune qui soit nombre carré que la dernière, car les carrés ne peuvent souffrir les finales qu'elles ont, si ce n'est* 499944 *qui néanmoins n'est pas carré. Pour savoir maintenant les nombres qui composent* 2027651281*, j'ôte le nombre que j'ai premièrement ajouté, savoir* 90061*, du dernier ajouté* 90081. *Il reste* 20*, à la moitié duquel plus* 2*, savoir à* 12*, j'ajoute la racine premièrement trouvée* 45029. *La somme est* 45041*, auquel nombre ajoutant et ôtant* 1020*, racine de la dernière somme* 1040400*, on aura* 46061 *et* 44021*, qui sont les deux nombres plus prochains qui composent* 2027651281. *Ce sont les seuls, parce que l'un et l'autre sont premiers.*

8.2.2 Traduction

La procédure, dite avec des symboles [22], est donc la suivante. On a à décomposer un nombre N. On en calcule la racine carrée et sa partie entière q, ici $q = 45029$. On a donc $q^2 \leq N < (q+1)^2$. Si on a $N = q^2$ on a fini. Ici, ce n'est pas le cas car on a $N = q^2 + 40440$. On pose $r = N - q^2$.

L'idée, ensuite, est d'écrire N sous la forme $N = (q+k)^2 - s^2 = (q + k - s)(q + k + s)$. On essaie successivement avec $k = 1, 2, ...$ et on s'arrête si $(q+k)^2 - N$ est un carré. Par exemple pour $k = 1$, on regarde $(q^2 + 2q + 1) - N = (2q + 1) - r$. On retranche donc $r = 40440$ de $2q + 1 = 90059$ comme le dit Fermat. Il reste 49619 qui n'est toujours pas un carré. Comme on a $2kq + k^2 = 2(k-1)q + (k-1)^2 + (2q + 2k - 1)$, on continue en ajoutant $2q + 3$, $2q + 5$, ..., $2q + 2k - 1$, jusqu'à ce qu'on trouve un carré. Ici, il faut aller jusqu'à $k = 12$:

$$2q + 1 - r + (2q + 3) + (2q + 5) + \cdots + (2q + 23) = 1040400 = (1020)^2 = s^2.$$

On a donc $N = (q+k)^2 - s^2 = (45029 + 12)^2 - 1020^2 = 44021 \times 46061$.

8.2.3 La méthode de Fermat dite à ma manière

Il s'agit de décomposer le nombre N en produit de facteurs premiers. On suppose N impair. On pose $q = [\sqrt{N}]$ et on suppose que N n'est pas un carré. On a donc $q^2 < N < q^2 + 2q + 1$. On cherche une décomposition de N sous la forme $N = (q + a)(q + b)$ avec $a, b \in \mathbf{Z}$.

22. Et l'on voit ici quelle économie de pensée ils procurent !

8.1 Lemme. *Si on a une décomposition comme ci-dessus :*

- *a et b sont de même parité,*
- *a et b sont non nuls et de signes contraires, sauf si l'on a $N = q(q+2)$,*
- *on a $ab \leq 0$ et $a + b > 0$.*

Démonstration. Comme N est impair il en est de même de $q + a$ et $q + b$ et on a donc $a \equiv q + 1$ et $b \equiv q + 1$ modulo 2.

Si a, b sont tous deux ≤ 0 on a $(q + a)(q + b) \leq q^2 < N$, s'ils sont tous deux > 0 on a $(q+a)(q+b) \geq q^2+2q+1 > N$. Ces cas sont donc impossibles. Il reste à examiner le cas où l'un des deux, disons a, est nul et l'autre > 0. Comme on a $q(q+b) \leq q^2+2q$ on a $b \leq 2$ et donc $b = 2$ à cause de la parité.

Pour le dernier point, il est clair que ab est ≤ 0. On a donc $q^2 < N = q^2 + (a + b)q + ab \leq q^2 + (a + b)q$, ce qui montre que $a + b$ est > 0.

On cherche donc a, b tels que $N - q^2 = (a + b)q + ab$ et on sait que $a + b$ est pair et $ab \geq 0$. Pour cela, on effectue une pseudo-division euclidienne de N par $2q$ en écrivant $N - q^2 = 2nq - r_n$, avec $n \in \mathbf{N}^*$ et $r_n < 0$, mais sans imposer la condition $|r_n| < q$. Autrement dit, on essaie successivement $n = 1, 2, 3$, etc.

On aura la factorisation cherchée si l'on peut résoudre en $a, b \in \mathbf{Z}$ les équations $a+b = 2n$ et $ab = -r_n$. Les nombres a, b sont racines de l'équation $X^2 - 2nX - r_n = 0$ et cette équation admet des solutions entières si et seulement si son discriminant (réduit) $\Delta_n = n^2 + r_n$ est un carré, c'est-à-dire si $(q + n)^2 - N$ est un carré, comme le fait Fermat.

Pour faire ce calcul de proche en proche, on peut, comme Fermat, utiliser une formule de récurrence : $\Delta_{n+1} = \Delta_n + 2q + 2n + 1$ (qui résulte de $r_{n+1} = r_n + 2q$).

La validité de la méthode est donnée par le lemme suivant :

8.2 Lemme. *On suppose que N est produit de deux nombres premiers $p_1 p_2$. Soit A la "patience" de l'utilisateur (c'est-à-dire le nombre d'essais qu'il est prêt à faire pour mettre en œuvre la méthode). La méthode de Fermat donne un résultat pourvu que la moyenne $m = \dfrac{p_1 + p_2}{2}$ diffère de $q = [\sqrt{N}]$ de moins de A.*

Démonstration. Si on a $N = (q + a)(q + b)$, la moyenne est $m = q + \dfrac{a + b}{2}$. Un succès de la tentative correspond à l'écriture $N - q^2 = 2nq - r_n$, avec $n = \dfrac{a + b}{2} = m - q$ et il doit être obtenu avec $n \leq A$, d'où le résultat.

8.3 *Remarques.* 1) Si $N = q(q + 2)$, on a $N - q^2 = 2q$ et la factorisation est obtenue dès la première opération.

2) Le cas le plus favorable après celui-ci est celui où le succès est obtenu avec $n = 1$. Dans ce cas, l'écriture $N - q^2 = 2q - r$ est la division euclidienne de $N - q^2$ par $2q$ (avec reste r négatif mais tel que $|r| < 2q$).

Attention, l'exemple de $N = q^2 + q - s$ avec $0 < s < q$ et $q + s + 1$ carré comme $N = 65$, $q = 8$, $r = 7$ montre que $N - q^2 = 2q - (q + s)$ n'est pas nécessairement la division euclidienne de $N - q^2$ par q.

3) Dans le cas du défi posé par Stanley Jevons en 1874 (factoriser $N = 8616460799$), la méthode fonctionne avec $n = 56$, donnant $q = 92824$ et $(q + n)^2 - N = 3199^2$, d'où $p = 89681$ et $q = 96079$.

8.3 Une hypothèse sur la décomposition de 100895598169 ?

8.3.1 Le principe

L'idée est très voisine de celle de la méthode précédente. On part d'un entier N et on suppose qu'il est impair et que ce n'est pas un carré. Au lieu de regarder seulement $q = [\sqrt{N}]$, on regarde tous les nombres $q = [\sqrt{N/k}]$ pour $k = 1, 2, \ldots$, jusqu'à la patience de l'utilisateur. On cherche ensuite, avec le nombre q en question, une décomposition de la forme $N = (kq + a)(q + b) = kq^2 + (kb + a)q + ab$. Précisément :

8.4 Proposition. *On suppose que N est produit de deux nombres premiers $p_1 p_2$. Soit A la "patience" de l'utilisateur (c'est-à-dire le nombre d'essais qu'il est prêt à faire pour mettre en œuvre la méthode). On suppose qu'il existe $k \leq A$ tel que, si $q = [\sqrt{N/k}]$, on ait $p_1 = kq + a$ et $p_2 = q + b$ avec $|a| < \sqrt{q}$ et $|b| < \sqrt{q}$. On trouve alors la décomposition en effectuant les divisions euclidiennes de $N - kq^2$ par q pour $k \leq A$.*

Démonstration. Si on a $N = (kq + a)(q + b)$, on a $N - kq^2 = (kb + a)q + ab$. Comme on a supposé $|ab| < q$, la division euclidienne de $N - kq^2$ par q donne $kb + a$ comme quotient et ab comme reste (éventuellement négatif) et, comme k est connu, on en déduit a, b donc p_1 et p_2.

8.5 *Remarque.* Si l'on est dans cette situation, on a $N - kq^2 = q(kb + a) + ab$ avec $|ab| < q$ et le quotient approché de $N - kq^2$ par q est proche de l'entier $kb + a$. C'est ainsi qu'on peut repérer les cas propices à un essai.

8.3.2 Application à Fermat et Mersenne

Pour $k = 1, 2, \ldots$ on calcule $q = [\sqrt{N/k}]$, puis le quotient approché de $N - kq^2$ par q et on regarde si ce quotient est proche d'un entier. On obtient successivement, pour $k = 1, \ldots, 7$ les quotients suivants : $1,34$; $3,5$; $5,57$; $2,69$; $2,28$; $3,14$; $12,46$, tous assez éloignés des entiers.

En revanche, pour $k = 8$, on a $q = 112302$, et $\dfrac{N - kq^2}{q} \simeq 15,0000623$. On écrit donc :

$$100895598169 - 8 \times 112302^2 = 15 \times 112302 + 7.$$

Il reste à résoudre les équations : $8b + a = 15$ et $ab = 7$, ce qui donne évidemment $a = 7$ et $b = 1$. On obtient la décomposition $N = p_1 p_2$ avec $p_1 = kq + a = 898423$ et $p_2 = q + b = 112303$.

9 Annexe 5 : le choix de l'exposant e dans le codage RSA

Reprenons l'exemple ci-dessus : $n = pq = 11639$, $e = 3361$. On vérifie, en écrivant quelques lignes de programme, qu'il y a 790 valeurs de a pour lesquelles on a $a^e \equiv a \pmod{11639}$, autrement dit où le message codé est identique au message initial, ce qui est gênant pour sa confidentialité !

C'est le cas, par exemple, de $a = 4589$ sur laquelle je suis tombé par hasard. L'explication est simple. Dire qu'on a $a^e \equiv a \pmod{pq}$ signifie que $a^e - a$ est multiple de p et de q et il y a quatre cas possibles en notant $\omega_p(a)$ l'ordre de a dans $(\mathbf{Z}/p\mathbf{Z})^*$:

C_1) p et q divisent a (et donc $a \equiv 0 \pmod{pq}$),

C_2) p divise a et $\omega_q(a)$ divise $e - 1$,

C_3) q divise a et $\omega_p(a)$ divise $e - 1$,

C_4) $\omega_p(a)$ et $\omega_q(a)$ divisent $e - 1$.

Dans le cas présent, on a $p-1 = 102 = 2 \times 3 \times 17$, $q-1 = 112 = 2^4 \times 7$ et $e-1 = 3360 = 2^5 \times 3 \times 5 \times 7$ donc beaucoup de possibilités d'ordres modulo p et q divisant $e - 1$ (par exemple $a = 4589$ est d'ordre 6 modulo 103 et 8 modulo 113 et ces nombres divisent $e - 1 = 3360$).

Il faut donc, dans le choix de e, essayer d'éviter d'avoir trop de facteurs communs avec $p - 1$ et $q - 1$. Par exemple, avec $e = 3623$, comme on a $3622 = 2 \times 1811$, il ne peut y avoir que le facteur [23] 2. Dans ce cas, on voit facilement que les seuls a à éviter sont $1, 3502, 3503, 4634, 7005, 8136, 8137$ et 11638. Ces nombres sont d'ailleurs inévitables quel que soit e car congrus à $0, 1$ ou -1 modulo p et q (par exemple, on a $4634 \equiv 1 \pmod{113}$ et $\equiv -1 \pmod{103}$). Ils représentent la tare congénitale du nombre 11639.

On notera d'ailleurs que, quel que soit le choix de p, q et e, il y a toujours de tels a invariants. Par exemple, avec $p < q$, si l'on a l'égalité de Bézout

23. Le facteur commun 2 est inévitable car e doit être premier avec $(p-1)(q-1)$, donc impair, donc $e - 1$ pair.

$\lambda p - \mu q = 1$ avec $0 < \lambda < q$, ce qui est toujours possible, $a = \lambda p$ est congru à 0 modulo p et à 1 modulo q donc invariant. La solution pour éviter les ennuis est d'imposer que les messages soient assez petits. En résumé, on appliquera les deux règles contenues dans la proposition suivante :

9.1 Proposition. *Soient p, q deux nombres premiers distincts et $e < pq$. On suppose :*

1) Que le nombre e est premier avec $(p-1)(q-1)$ et qu'on a :

$$\operatorname{pgcd}(e-1, p-1) = \operatorname{pgcd}(e-1, q-1) = 2.$$

2) Que le message a est plus grand que 1 et plus petit que p et q.

Alors, le message codé $a^e \pmod{pq}$ est distinct de a.

Démonstration. Les conditions de 2) assurent que p et q ne divisent pas a, ce qui écarte les cas $C_1), C_2), C_3)$. Si on est dans le cas $C_4)$, la condition 1) montre que $\omega_p(a)$ et $\omega_q(a)$ ne peuvent valoir que 1 ou 2, ce qui signifie que a est congru à ± 1 modulo p et q. Mais, avec 2), on voit que c'est impossible car $a \neq 1$ et $p-1$ et $q-1$ sont distincts.

9.2 *Remarque.* Pour appliquer la consigne 2), la personne qui code, et qui ne connaît pas p et q, devra se limiter à des messages de taille assez petite (si pq a, disons, 400 chiffres, le message a devra avoir nettement moins de 200 chiffres). Bien entendu, il faudra peut-être envoyer plusieurs messages.