Grands nombres premiers
Cryptographie RSA

François DE MARÇAY

Département de Mathématiques d'Orsay
Université Paris-Sud, France

1. Limitations physiques

Soit un nombre donné quelconque $n \geqslant 1$, représenté par exemple en base 10 comme suite finie de chiffres appartenant à $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Le *Crible d'Eratosthène* est la méthode la plus simple et la plus directe pour déterminer si un tel nombre entier n donné est un nombre premier, ou un nombre composé.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

L'algorithme procède par élimination : il s'agit de supprimer de la table complète de tous les entiers allant de 2 jusqu'à n tous les entiers qui sont multiples d'un entier inférieur à n. À la fin il ne restera donc que les entiers qui ne sont multiples d'aucun entier, c'est-à-dire, il ne restera plus que les nombres premiers.

On commence tout simplement par rayer tous les multiples de 2, puis tous les multiples de 3, puis tous les multiples de 5, puis tous les multiples de 7, et ainsi de suite, jusqu'aux multiples du dernier entier premier k tel que $k^2 \leqslant n$.

~~1~~	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	~~29~~	~~30~~
31	~~32~~	~~33~~	~~34~~	~~35~~	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	~~49~~	~~50~~
~~51~~	~~52~~	53	~~54~~	~~55~~	~~56~~	~~57~~	~~58~~	59	~~60~~
61	~~62~~	~~63~~	~~64~~	~~65~~	~~66~~	67	~~68~~	~~69~~	~~70~~
71	~~72~~	73	~~74~~	~~75~~	~~76~~	~~77~~	~~78~~	79	~~80~~
~~81~~	~~82~~	83	~~84~~	~~85~~	~~86~~	~~87~~	~~88~~	89	~~90~~
~~91~~	~~92~~	~~93~~	~~94~~	~~95~~	~~96~~	97	~~98~~	~~99~~	~~100~~

On peut en effet s'arrêter à $k \approx \sqrt{n}$, car tous les entiers non premiers ont déjà été rayés précédemment !

Cependant, ce procédé direct est limité dans la pratique à cause du très grand nombre d'opérations qu'il exige.

> In a very real sense, there are no large numbers : Any explicit integer can be said to be 'small'. Indeed, no matter how many digits or towers of exponents you may write down, there are only finitely many natural numbers smaller than your candidate, and finitely many that are larger. Though condemned always to deal with small numbers, we can at least strive to handle numbers that are larger than those that could be handled before.
>
> Richard Crandall, Carl Pomerance

Affirmation de philosophie des mathématiques. *L'ensemble complet de tous les nombres premiers n'est pas connaissable comme totalité donnée de manière actuelle, c'est-à-dire de manière accessible, effective, concrète, visible, « vraiment présente ».* □

La raison métaphysique simplette de cette limitation est claire : *l'infini actuel n'existe pas*.

Afin de mieux comprendre en quoi il y a réellement *limitation*, spéculons quelque peu.

Supposons par exemple que l'on rêve d'imprimer la liste complète de tous les nombres entiers premiers qui sont inférieurs à un certain entier nombre assez grand, disons pour se satisfaire de possèder un petit « *trésor mathématique* ». Va-t-on y parvenir ?

Définition 1.1. Pour $x \geqslant 1$ entier, on note classiquement $\pi(x)$ le nombre d'entiers premiers inférieurs à x :

$$\pi(x) := \mathsf{Card}\left\{p \in \mathscr{P} \text{ premiers avec } p \leqslant x\right\}.$$

On connaît la valeur exacte de $\pi(x)$ pour des x contenant une vingtaine de chiffres en base 10.

x	$\pi(x)$
10^2	25
10^3	168
10^4	1229
10^6	78498
10^8	5761455
10^{12}	37607912018
10^{16}	279238341033925
10^{17}	2623557157654233
10^{18}	24739954287740860
10^{19}	234057667276344607
10^{20}	2220819602560918840
10^{21}	21127269486018731928
10^{22}	201467286689315906290
$4 \cdot 10^{22}$	783964159847056303858

Prenons par exemple $x = 10^{20}$. Il se trouve qu'à notre époque, on connaît la valeur exacte :

$$\begin{aligned}\pi\big(10^{20}\big) &= 2\,220\,819\,602\,560\,918\,840 \\ &\approx 2 \cdot 10^{18}.\end{aligned}$$

De tels nombres n'ont « *qu'une vingtaine de chiffres* », ils peuvent donc sembler « *petits* » — mais attention aux illusions !

Question 1.2. Si on connaît la valeur exacte de $\pi(10^{20})$, cela semble vouloir dire que l'on connaît effectivement tous les nombres premiers $p \leqslant 10^{20}$, *mais cela est-il bien vrai ?*

En fait, non ! Même si chacun de ces $\approx 2 \cdot 10^{18}$ nombres pouvait être représenté par un seul caractère d'imprimerie ou un seul bit informatique, nous affirmons qu'il serait néanmoins totalement impossible de les voir tous, ou de les stocker tous dans des ordinateurs.

En effet, considérons par analogie le nombre record de décimales du nombre d'Archimède :

$$\pi = \mathbf{3,14159265358979323846264338327950288419716939937510}\cdots,$$

qui, depuis octobre 2011, va jusqu'à $10\,000$ milliards, ce qui nous fait donc :

$$10^{13}$$

décimales, c'est-à-dire 10^{13} caractères d'imprimerie, ou bits sur un ordinateur, un nombre vraiment inférieur à $2 \cdot 10^{18}$. Mais même ce nombre se situe à la limite du visible.

Question 1.3. Combien de livres de 500 pages comportant 40 lignes chacune avec 80 caractères seraient nécessaires pour montrer à l'humanité éclairée les 10^{13} décimales connues du nombre π ?

Dans un seul livre, on peut imprimer :

$$500 \times 80 \times 40 = 1\,600\,000$$

décimales, et donc il faudrait environ :

$$\frac{10\,000\,000\,000\,000}{1\,600\,000} = 6\,250\,000,$$

livres, environ la moitié des quatorze millions de livres que possède et conserve la *Bibliothèque Nationale de France*.

Pour faire voir les $\approx\ 2 \cdot 10^{18}$ nombres premiers inférieurs à 10^{20}, il faudrait alors au moins :

$$10^{18-13}\ =\ 100\,000$$

bibliothèques de cette taille de par le monde. En poussant jusqu'à 5 chiffres de plus :

Conclusion 1.4. *Pour des raisons purement physiques, on ne pourra jamais « voir »* tous *les nombres premiers ayant un nombre de chiffres* $\leqslant 25$ *en base* 10. □

> Rien n'est logique et rien ne semble absurde comme l'océan. Cette dispersion de soi-même est inhérente à sa souveraineté et est un des éléments de son ampleur. Victor Hugo

> Ainsi dans cette immensité, se noie ma pensée : et le naufrage m'est doux dans cette mer. Leopardi

Le paradoxe contre-paradoxal [1], c'est que les théoriciens des nombres réussissent quand même à capturer et à manipuler quelques gros poissons nombres premiers ayant jusqu'à 100, 200, 300 chiffres ! Les mathématiques évoluent alors comme dans une mer immense de nombres gigantesques, en connaissant très peu de nombres premiers parmi ceux qui comportent jusqu'à des dizaines de millions de chiffres en base 10.

Question 1.5. Comment engendrer des nombres premiers qui possèdent un très grand nombre de chiffres ?

2. Grands nombres premiers : pêche à la ligne

Fermat avait proposé, nous l'avons vu, une formule simple qui lui semblait produire des nombres premiers de plus en plus grands. Le *hic*, avec ces nombres de Fermat :

$$F_n\ :=\ 2^{2^n}+1,$$

c'est qu'ils ne sont presque jamais premiers, en tout cas, pour ce qui est de ceux que nous connaissons actuellement.

En 1732, Euler a découvert la factorisation :

$$\begin{aligned} F_5\ &=\ 4294967297 \\ &=\ 641 \cdot 6700417. \end{aligned}$$

En 1882, Landry a montré que le nombre de Fermat suivant est aussi composé :

$$\begin{aligned} F_6\ &=\ 2^{2^6}+1 \\ &=\ 18446744073709551617 \\ &=\ 274177 \cdot 67280421310721. \end{aligned}$$

> No prime F_n has ever been found beyond F_4, so Fermat's conjecture has not proved a very happy one. G.H. Hardy, E.M. Wright

Aux alentours de l'année 2005, l'état des connaissances concernant les nombres de Fermat jusqu'à F_{24} est résumé au moyen du tableau suivant.

1. Ce n'est parce qu'on ne peut pas *tout* connaître qu'il est impossible de connaître *un tout petit peu* d'un trop grand tout.

$$
\begin{aligned}
F_7 &= 59649589127497217 \cdot 5704689200685129054721 \\
F_8 &= 1238926361552897 \cdot P \\
F_9 &= 2424833 \cdot 7455602825647884208337395736200454918783366342657 \cdot P \\
F_{10} &= 45592577 \cdot 6487031809 \cdot 4659775785220018543264560743076778192897 \cdot P \\
F_{11} &= 319489 \cdot 974849 \cdot 167988556341760475137 \cdot 3560841906445833920513 \cdot P \\
F_{12} &= 114689 \cdot 26017793 \cdot 63766529 \cdot 190274191361 \cdot 1256132134125569 \cdot C \\
F_{13} &= 2710954639361 \cdot 2663848877152141313 \cdot 3603109844542291969 \cdot \\
&\qquad 319546020820551643220672513 \cdot C \\
F_{14} &= C \\
F_{15} &= 1214251009 \cdot 2327042503868417 \cdot 1687688170295169723830241270169 61 \cdot C \\
F_{16} &= 825753601 \cdot 188981757975021318420037633 \cdot C \\
F_{17} &= 31065037602817 \cdot C \\
F_{18} &= 13631489 \cdot 81274690703860512587777 \cdot C \\
F_{19} &= 70525124609 \cdot 646730219521 \cdot C \\
F_{20} &= C \\
F_{21} &= 4485296422913 \cdot C \\
F_{22} &= C \\
F_{23} &= 167772161 \cdot C \\
F_{24} &= C
\end{aligned}
$$

Dans ce tableau, tous les nombres explicitement écrits sont des nombres *premiers*, et la lettre « P » désigne un grand facteur entier dont on a démontré qu'il est *premier*, tandis que la lettre « C » désigne un très grand facteur composé, dont on ne connaît pas nécessairement les facteurs premiers.

Néanmoins, comme les mathématiques savent produire un très grand nombre de formules très compliquées, on peut toujours espérer qu'une modification de la formule de Fermat pourrait engendrer une collection infinie de nombres premiers.

Question 2.1. Existe-t-il une formule générale simple $\mathscr{F}(k)$ dépendant d'un entier $k \geqslant 1$ dont toutes les valeurs :

$$\mathscr{F}(1) = 2, \qquad \mathscr{F}(2) = 3, \qquad \mathscr{F}(3) = 5, \qquad \mathscr{F}(4) = 7, \qquad \mathscr{F}(5) = 11, \qquad \ldots\ldots,$$

décrivent exactement la suite des nombres premiers ?

Aucune telle formule magique n'est connue sur Terre, et il est très probable qu'il n'en existe pas non plus sur les autres planètes habitées de l'Univers.

On peut alors abaisser les ambitions d'un cran.

Question 2.2. Existe-t-il une formule générale simple $\mathscr{G}(k)$ dépendant d'un entier $k \geqslant 1$ dont les valeurs contiennent au moins une infinité de nombres premiers ? (Sans demander, toutefois, que *toutes* les valeurs $\mathscr{G}(1), \mathscr{G}(2), \mathscr{G}(3), \ldots$, soient des nombres premiers.)

Par exemple, la formule :

$$\mathscr{G}(k) \;:=\; k^2 - k + 41$$

ne produit que des valeurs entières *premières* pour :

$$0 \;\leqslant\; k \;\leqslant\; 40,$$

à savoir les valeurs :

$$\begin{aligned}
&41,\quad 41,\quad 43,\quad 47,\quad 53,\quad 61,\quad 71,\quad 83,\quad 97,\quad 113,\quad 131,\\
&\quad 151,\quad 173,\quad 197,\quad 223,\quad 251,\quad 281,\quad 313,\quad 347,\quad 383,\quad 421,\\
&\quad 461,\quad 503,\quad 547,\quad 593,\quad 641,\quad 691,\quad 743,\quad 797,\quad 853,\quad 911,\\
&\quad 971,\quad 1033,\quad 1097,\quad 1163,\quad 1231,\quad 1301,\quad 1373,\quad 1447,\quad 1523,\quad 1601,
\end{aligned}$$

mais à partir de $k = 41$, ces nombres cessent d'être toujours premiers :

$$41\cdot 41,\quad 41\cdot 43,\quad 1847,\quad 1933,\quad 43\cdot 47,\quad 2111,\quad 2203,\quad 2297,\quad 2393,\quad 47\cdot 53,$$

quoiqu'il y en ait toujours pas mal qui soient premiers.

Problème mathématique ouvert. *Existe-t-il une infinité de nombre premiers dans la suite :*

$$\big(k^2+1\big)_{k=1}^{\infty}\,?$$

ou encore dans la suite :

$$\big(k^2-k+41\big)_{k=1}^{\infty}\,?$$

De même, on constate à la main ou sur un ordinateur que la suite :

$$\big(k^2-79\,k+1601\big)_{k=1}^{\infty}$$

ne prend que des valeurs entières *premières* pour :

$$0 \leqslant k \leqslant 79,$$

mais pour $k = 80$ on a :

$$80^2 - 79\cdot 80 + 1601 \;=\; 41\cdot 41.$$

Il n'est en fait pas difficile de constater que ce phénomène est général.

Théorème 2.3. *Aucun polynôme non constant à coefficients entiers :*

$$\mathscr{G}(x) \;\in\; \mathbb{Z}[x]$$

ne peut prendre une suite complète de valeurs :

$$\big(\mathscr{G}(k)\big)_{k=K}^{\infty}$$

qui sont toutes *des nombres premiers pour $k \geqslant K \gg 1$ assez grand.*

Démonstration. Quitte à remplacer $\mathscr{G}$ par $-\mathscr{G}$, on peut supposer que le coefficient de tête de $\mathscr{G}$ est positif :

$$\mathscr{G}(x) \;=\; g_0\,x^d + \cdots + g_{d-1}\,x + g_d \qquad\qquad (d = \deg \mathscr{G},\ g_0 > 0),$$

de telle sorte que $\mathscr{G}(k)$ tend vers $+\infty$ lorsque $k \to \infty$.

Ainsi, en un entier k assez grand, la valeur du polynôme :

$$\begin{aligned}
\mathscr{G}(k) &= a_0\,k^d + \cdots + a_{d-1}\,k + g_d\\
&=: \ell
\end{aligned}$$

est certainement $\geqslant 2$. L'astuce élémentaire, pour conclure, consiste alors à observer, grâce la formule du binôme de Newton (exercice), que pour tout entier $\lambda \geqslant 1$:

$$\begin{aligned}\mathscr{G}\big(\lambda\,\ell+k\big) &= g_0\,\big(\lambda\,\ell+k\big)^d+\cdots+g_{d-1}\,\big(\lambda\,\ell+k\big)+g_d\\ &= \mathscr{G}(k)+\text{multiple de }\ell\\ &= \text{multiple de }\ell\\ &= \text{nombre non premier},\end{aligned}$$

ce qui montre que $\mathscr{G}$ ne prend *pas* de valeurs entières premières sur la suite *infinie* $(\lambda\,\ell+k)_{\lambda=1}^{\infty}$. □

L'énoncé suivant, dont la démonstration peut faire l'objet d'un mémoire de Master 1 à l'université, utilise de magnifiques outils d'Analyse Complexe.

Théorème 2.4. [Dirichlet] *Pour tout couple d'entiers $a,b\geqslant 1$ premiers entre eux :*

$$a\wedge b=1,$$

la suite des valeurs :

$$\big(a+b\,k\big)_{k=1}^{\infty}$$

contient une infinité de nombres premiers. □

Pour formuler un énoncé plus fin, rappelons que le Théorème des nombres premiers stipule que la fonction :

$$\pi(x)\;:=\;\text{Card}\,\big\{2\leqslant p\leqslant x\colon\;p\in\mathscr{P}\text{ est premier}\big\}$$

se comporte asymptotiquement comme :

$$\pi(x)\;\sim\;\frac{x}{\log x}.$$

Théorème 2.5. [Dirichlet affiné] *Pour tout couple d'entiers $a,b\geqslant 1$ premiers entre eux, dans la progression arithmétique infinie :*

$$a,\quad a+2\,b,\quad a+3\,b,\quad a+4\,b,\quad a+5\,b,\quad \ldots\ldots,$$

le nombre de nombres qui sont premiers :

$$\pi\big(x;a,b\big)\;:=\;\text{Card}\,\big\{k\in\mathbb{N}\text{ avec }a+bk\leqslant x\text{ tels que }a+b\,k\;\in\;\mathscr{P}\big\}$$

est asymptotiquement égal à :

$$\begin{aligned}\pi\big(x;a,b\big)\;&\sim\;\frac{1}{\varphi(b)}\,\pi(x)\\ &\sim\;\frac{1}{\varphi(b)}\,\frac{x}{\log x},\end{aligned}$$

où $\varphi(b)=\text{Card}\,\{1\leqslant b'\leqslant b\colon\;b'\wedge b=1\}$ est l'indicateur d'Euler. □

En tout cas, que ce soit au moyen de formules linéaires $k\longmapsto a\,k+b$ ou, conjecturalement, au moyen de formules quadratiques telles que $k\longmapsto k^2+1$, même s'il elles produisent une infinité de nombres premiers, il reste en général difficile, comme le dit l'Ancien Testament, de *séparer le bon grain de l'ivraie*, à savoir il reste difficile de déterminer si une valeur $\mathscr{G}(k)$ est un nombre premier ou un nombre composé.

Mais revenons à notre Question 1.5 qui demandait comment produire de *grands* nombres premiers. Il est clair qu'il vaut mieux utiliser des formules exponentielles, car les polynômes croissent moins vite que les exponentielles.

Il est intéressant de comparer le destin des nombres de Fermat à celui d'une autre conjecture célèbre, qui concerne les nombres premiers de la forme :

$$2^n - 1.$$

Théorème 2.6. *Si, pour un entier $n \geqslant 2$, le nombre :*

$$a^n - 1$$

est premier, alors $a = 2$ et n lui-même est premier.

Démonstration. En effet, si $a \geqslant 3$, alors on a la factorisation :

$$a^n - 1 = (a-1)\big(a^{n-1} + \cdots + a + 1\big).$$

Donc on a $a = 2$ nécessairement.

Ensuite, si $n = k\,l$ est composé, on peut à nouveau constater (exercice) que $2^k - 1$ et $2^l - 1$ divisent $2^n - 1$. □

Définition 2.7. Un *nombre de Mersenne*[2] est un nombre premier qui s'écrit sous la forme :

$$M_p \,:=\, 2^p - 1,$$

avec p lui-même entier premier.

En 1644, Mersenne affirmait que M_p est premier pour les valeurs :

$$p \,=\, 2,\quad 3,\quad 5,\quad 7,\quad 13,\quad 17,\quad 19,\quad 31,\quad 67,\quad 127,\quad 257,$$

et que M_p est composé pour les 44 autres valeurs de p premier inférieures à 257.

La première erreur[3] dans la liste de Mersenne n'a été trouvée qu'en 1886, lorsque Pervusin et Seelhoff découvrirent que M_{61} est en fait premier. Par la suite, d'autres erreurs furent trouvées, et la liste de Mersenne commença à ne plus être prise au sérieux.

En 1876, Lucas mis au point une méthode pour tester si M_p est premier, méthode qu'il utilisa pour montrer que M_{127} est premier.

Entre 1876 et 1951, c'est-à-dire pendant trois-quarts de siècle, le nombre de Lucas :

$$2^{127} - 1 \,=\, 170141183460469231731687303715884105727$$

demeura le plus grand nombre premier connu.

2. Marin Mersenne, religieux érudit et mathématicien français du XVII$^{\text{ème}}$ siècle.

3. En 1732, Euler affirmait que M_{41} et M_{47} sont premiers, ce qui n'était pas vrai (et n'est toujours pas vrai aujourd'hui).

En 2005, tous les nombres premiers de Mersenne connus étaient ceux qui apparaissent dans le tableau suivant.

2^2-1	2^3-1	2^5-1	2^7-1
$2^{13}-1$	$2^{17}-1$	$2^{19}-1$	$2^{31}-1$
$2^{61}-1$	$2^{89}-1$	$2^{107}-1$	$2^{127}-1$
$2^{521}-1$	$2^{607}-1$	$2^{1279}-1$	$2^{2203}-1$
$2^{2281}-1$	$2^{3217}-1$	$2^{4253}-1$	$2^{4423}-1$
$2^{9869}-1$	$2^{9941}-1$	$2^{11213}-1$	$2^{19937}-1$
$2^{21701}-1$	$2^{23209}-1$	$2^{44497}-1$	$2^{86243}-1$
$2^{110503}-1$	$2^{132049}-1$	$2^{216091}-1$	$2^{756839}-1$
$2^{859433}-1$	$2^{1257787}-1$	$2^{1398269}-1$	$2^{2976221}-1$
$2^{3021377}-1$	$2^{6972593}-1$	$2^{13466917}-1$	$2^{20996011}-1$
$2^{24036583}-1$	$2^{25964951}-1$		

Les prochains sont les suivants :

$$\begin{aligned} M_{30402457} &= 2^{30402457}-1,\\ M_{32582657} &= 2^{32582657}-1,\\ M_{37156667} &= 2^{37156667}-1,\\ M_{42643801} &= 2^{42643801}-1,\\ M_{43112609} &= 2^{43112609}-1,\\ M_{57885161} &= 2^{57885161}-1, \end{aligned}$$

ce qui veut dire que tous les M_p intercalés qui n'apparaissent pas sont *composés*.

Le dernier, découvert en janvier 2013 :

$$2^{57885161}-1,$$

est composé de plus de 17 millions de chiffres en base 10, et son écriture complète remplirait une bonne dizaine de livres de 500 pages environ.

À titre de comparaison, il faut moins de 100 chiffres pour effectuer le décompte du nombre de particules (neutrons, protons et électrons) contenues dans tout l'univers !

Après tous ces préliminaires qui se sont étendus en longueur, il est temps, maintenant, d'entrer dans le vif du sujet.

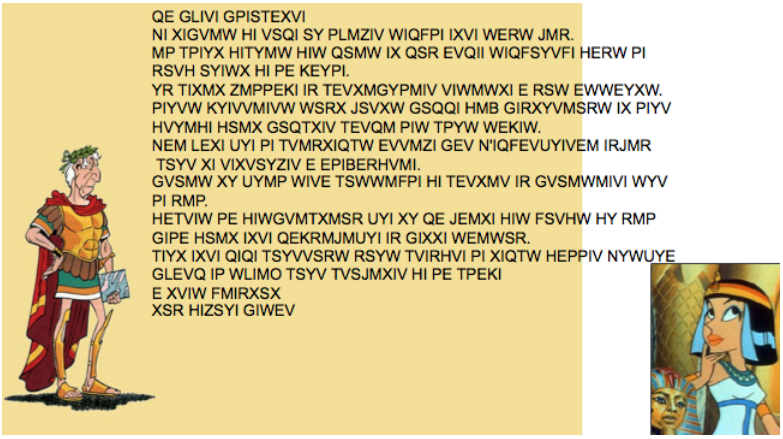
3. Principe de la crytographie RSA

> Our ability to uncover large primes and prove them prime has outstripped our ability to factor, a situation that gives tome comfort to cryptographers. Richard Crandall, Carl Pomerance

Le scénario est le suivant. Bernard [4] souhaite envoyer secrètement un message à sa dulcinée, Alice.

En effet, il est hors de question que des oreilles indiscrètes interceptent ce qui transite dans les canaux de leur communication intime.

L'idée la plus simple, qui remonte à l'Antiquité, consiste à *coder*, ou à *chiffrer*, le message, c'est-à-dire à le rendre illisible par toute autre tierce personne.

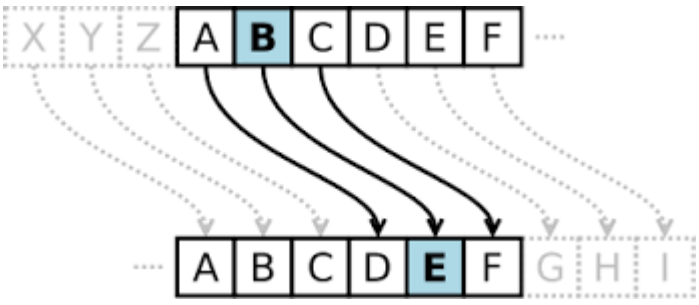


Le *code de Cæsar* — Julius, qui écrivait à Cléopâtre — permute simplement toutes les lettres de l'alphabet :

$$a \longmapsto d, \quad b \longmapsto e, \quad c \longmapsto f, \quad \ldots\ldots, \quad y \longmapsto b, \quad z \longmapsto c.$$

Par exemple, Cæsar signera sa lettre enflammée par :

$$Fdhvdu.$$



Malheureusement, ce cryptosystème est trivialement facile à casser : il y a seulement 26 possibilités à tester, et une fois que la correspondance pour une seule lettre a été trouvée, toutes les autres s'ensuivent car la permutation est circulaire.

Plus astucieusement, on pourrait utiliser l'une des :

$$26! \approx 4 \cdot 10^{26}$$

permutations de l'alphabet à 26 lettres. Dans un tel cryptosystème qui demeure encore assez primitif, casser le code, cela consiste à reconstituer la permutation. Mais si l'on sait dans quel langage le message est codé, une simple anayse de fréquence des lettres qui apparaissent permet assez rapidement de reconstituer la totalité du message.

4. Dans la langue anglo-saxonne, Bernard est appelé Bob.

Exeunt, donc les codes trop antiques de Julius Cæsar et Cléôpatre, place à la science high-tech d'Alice et de Bernard !

La seule méthode de codage dont on démontre qu'elle est entièrement sécurisée, au sens de l'informatique théorique, est la suivante.

Soit $\nu \geqslant 1$ le nombre de lettres du message à transmettre, par exemple, le message « ECUREUIL» réduit pour simplifier à un seul mot. On choisit une autre séquence auxiliaire de ν lettres produites au hasard, par exemple la séquence « DFTXMQIB ». On effectue la correspondence la plus simple entre les 26 lettres de l'alphabet et les nombres de 0 à 25 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**

On place le message à transmettre ainsi que son compagnon aléatoire l'un au-dessus de l'autre :

4	**2**	**20**	**17**	**4**	**20**	**8**	**11**
E	C	U	R	E	U	I	L
D	F	T	X	M	Q	I	B
3	**5**	**19**	**23**	**12**	**16**	**8**	**1**

On effectue l'addition modulo 26 de leurs chiffres respectifs, et on retranscrit le résultat en lettres :

7	**7**	**14**	**15**	**11**	**16**	**18**	**12**
H	H	O	P	L	Q	S	M

ce qui est le code du message d'origine.

C'est le seul procédé de codage qui soit démontrablement *incassable*.

Mais il présente quelques inconvénients. Premièrement, les clés doivent avoir la même longueur que les messages. Deuxièmement, il est préférable de ne pas ré-utiliser plusieurs fois des parties d'une même séquence aléatoire de lettres.

Le système de cryptographie le plus répandu et qui a supplanté tous ses concurrents est le système dit RSA — *voir* la Section 5 *infra* pour des informations historiques —, et il est basé sur l'utilisation de grands nombres premiers, le plus complexe des mystères mathématiques.

L'idée profonde est que la personne, Alice, qui souhaite recevoir un message secret de Bernard (de la CIA), prépare à l'avance chez elle (à la Maison Blanche) certaines données qui lui permettront à elle seule de comprendre les messages que Bernard aura envoyés. *Alice se prépare, en secret, à devenir la seule personne au monde qui pourra comprendre ce que voudra lui dire Bernard !*

Si on admet temporairement que le message de Bernard peut être découpé en blocs de caractères d'imprimerie ayant des longueurs approximativement égales, chaque morceau correspondant d'une certaine manière à un nombre, le problème reviendra alors à envoyer secrètement une suite de nombres entiers, écrits en base 10, de tailles approximativement égales. Dans la Section 4, nous justifierons un tel *codage préliminaire*, qui transforme des blocs de caractères d'imprimerie en une suite de nombres entiers.

Algorithme: Cryptographie RSA

► Un bloc du message de Bernard est représenté par un nombre $m \in \mathbb{N}$ de taille raisonnable.

► Alice, qui est cryptographe professionnelle, choisit dans sa collection personnelle deux très grands nombres premiers, soient :

$$p \qquad \text{et} \qquad q,$$

comportant tous deux environ 300 chiffres en base 10. Lorsque soit p, soit q est trop petit, le protocole a des risques d'être attaqué.

► Alice multiplie ces deux nombres, ce qui est très facile, et elle obtient le nombre :

$$n := p\,q,$$

qui comporte environ 600 chiffres.

► Alice rend public ce nombre n, *mais elle conserve secrètement chez elle les deux précieux facteurs premiers p et q*. Autrement dit, Bernard, qui fait partie du public d'Alice, peut prendre connaissance du nombre n, par exemple sur internet, et le monde entier, d'ailleurs, peut aussi prendre connaissance de n. *L'intérêt fantastique du protocole RSA, c'est que Bernard n'aura aucunement besoin de connaître les secrets d'Alice pour pouvoir lui écrire sans que personne n'y comprenne rien*. Autrement dit, aucun secret ne transitera dans les canaux de communication, tel est le génie du RSA !

Le secret reste intérieur !

► Sur le plan pratique, la sécurité du protocole RSA tient au fait qu'il est démontrablement extrêmement difficile de factoriser, même avec un réseau d'ordinateurs superpuissants, des nombres entiers à environ 600 chiffres. Donc tout le monde a beau connaître n, personne n'est assez fort pour retrouver les deux facteurs premiers p et q, *à moins d'y passer des milliards de milliards d'années*.

Comment choisir ses clés ?

Soit p et q deux nombres premiers,
on le produit n = p*q

Soit e, un grand nombre, choisi au hasard,
premier avec (p-1)*(q-1)

Soit d, tel que e*d ≡ 1 [(p-1)*(q-1)]

► Ensuite, Alice considère le groupe multiplicatif des éléments inversibles :

$$\big(\mathbb{Z}/(p-1)(q-1)\mathbb{Z}\big)^{\times},$$

et en appliquant des recherches aléatoires automatiques répétées, elle finit par trouver un élément inversible :

$$d \in \big(\mathbb{Z}/(p-1)(q-1)\mathbb{Z}\big)^{\times},$$

dont la taille soit assez grande pour que le protocole ne puisse pas être cassé.

► Alice détermine alors l'unique inverse e de d dans ce groupe, qui satisfait donc :

$$d\,e \equiv 1 \bmod (p-1)(q-1).$$

► Enfin, Alice rend aussi public cet inverse e, et alors, ce qu'on appelle la *clé publique* d'Alice, visible par tout le monde dans le monde entier, est le couple :

$$(n,\, e),$$

que Bernard se hâte de noter sur ses ordinateurs.

► Tout est maintenant prêt pour que Bernard se décide enfin à coder puis à envoyer son message, qui est un nombre $m \in \mathbb{N}$, car ... Alice est prête !

► C'est très simple : en utilisant l'Algorithme (très efficace !) d'exponentiation modulaire, Bernard prend son entier m et lui fait subir :

$$m \longmapsto m^e \bmod n,$$

ce qu'il peut effectivement faire, puisqu'il connaît n et e. *De la sorte, le message initial signifiant m est remplacé par un message $m^e \bmod n$ dans lequel un très grand désordre numérique a été introduit.*

► Bernard envoie alors à Alice ce message illisible $m^e \bmod n$, que le KGB n'essaie même plus d'intercepter sur internet.

► Mais comment Alice va-t-elle réussir à déchiffer cela ?

► C'est très simple, Alice fait subir à m^e une nouvelle exponentiation modulaire :

$$\begin{aligned} m^e \bmod n &\longmapsto \left(m^e\right)^d \bmod n \\ &= m \bmod n, \end{aligned}$$

et la Proposition 3.1 ci-dessous stipule que Alice retrouvera bien m.

Système RSA

clé privée (n, d)
clé publique (n, e)

M est le message que l'on veut chiffrer.

Chiffrement : $C \equiv M^e \ [n]$

Déchiffrement : $M \equiv C^d \ [n]$

► Afin d'assurer qu'il n'y ait aucune ambiguïté dans la reconstitution de m à travers le module n, il suffit que Bernard s'arrange à l'avance pour découper son message en blocs codés par des entiers m qui soient tous $\leqslant n-1$.

Proposition 3.1. *Étant donné deux nombres entiers premiers $p, q \in \mathscr{P}$, et étant donné un élément inversible :*

$$d \in \left(\mathbb{Z}/(p-1)(q-1)\mathbb{Z}\right)^\times$$

d'inverse e :

$$1 \equiv d\,e \bmod (p-1)(q-1),$$

pour tout entier $m \in \mathbb{Z}$, on a toujours :

$$\left(m^e\right)^d \bmod p\,q \equiv m \bmod p\,q.$$

Démonstration. Par hypothèse, il existe un entier $k \in \mathbb{Z}$ tel que :

$$e\,d = 1 + k\,(p-1)(q-1).$$

Assertion 3.2. *On a :*

$$m^{e\,d} \equiv m \bmod p.$$

Preuve. Deux cas sont à considérer. Premièrement, lorsque p divise m, on gagne facilement :

$$\begin{aligned} m^{e\,d} &\equiv 0 \bmod p \\ &\equiv m \bmod p. \end{aligned}$$

Deuxièmement, lorsque p ne divise pas m, d'où $\mathsf{pgcd}(p, m) = 1$, le petit Théorème de Fermat donne :

$$m^{p-1} \equiv 1 \bmod p,$$

d'où :

$$\begin{aligned} m^{e\,d} &= m \cdot m^{k\,(p-1)\,(q-1)} \\ &\equiv m \cdot 1 \bmod p \end{aligned}$$

et là encore, on a gagné. □

Par symétrie, on obtient aussi :

$$m^{e\,d} \equiv m \bmod q.$$

Autrement dit, $m^{e\,d} - m$ est multiple de p et est multiple de q, et comme p et q sont premiers, c'est que $m^{e\,d} - m$ doit être (exercice mental) multiple de leur produit $p\,q$, ce qui conclut. □

Le système RSA est dit *à clé publique*, ce qui signifie que :

▷ ni l'algorithme de calcul ni la clé de codage ne sont cachés ;

▷ la connaissance de la clé publique d'Alice permet à tous ses interlocuteurs potentiels : Bernard, Gustave, Gédéon, Prosper, Léopold, de crypter les messages qu'ils lui destinent, mais seule Alice peut décrypter les messages qu'elle reçoit, grâce à sa clé de décodage privée, qu'elle cache soigneusement.

Les clés publiques peuvent être publiées dans un annuaire ou sont obtenues, à la demande, en contactant préalablement celui à qui l'on veut faire parvenir un message.

Ce type de système possédant une clé de décodage différente de la clé de codage (le système est dissymétrique) présente un avantage sur les systèmes classiques (dits symétriques, car une seule et même clé sert à la fois au codage et au décodage) : avant un échange, les deux interlocuteurs n'ont pas besoin de se rencontrer pour convenir d'une clé secrète, qui devra rester connue d'eux seuls, ni n'ont besoin de faire circuler — sur un réseau informatique ou autre — une clé secrète, transmission qui bien sûr engendre un risque. Seule la clé publique, insuffisante pour le décryptage, est connue préalablement aux échanges cryptés entre les deux interlocuteurs.

Pour terminer cette section, posons-nous la :

Question 3.3. *Combien de temps faudrait-il pour factoriser un entier n à 600 chiffres en base 10 en appliquant l'algorithme naïf d'Eratosthène ?*

Il s'agit de supprimer tous les multiples de nombres premiers $p \leqslant \sqrt{10^{600}} = 10^{300}$. Le nombre de nombre premiers inférieurs à 10^{300} est approximativement égal à :

$$\frac{10^{300}}{\log 10^{300}} \approx 1,5 \cdot 10^{297}.$$

Un processeur ayant une forte puissance de 10 Ghz (optimiste) fait environ 10^{10} opérations à la seconde.

Supposons sans sourciller qu'il y ait 10^{11} tels processeurs de part le monde (ce qui ferait 10 processeurs par être humain !).

On obtiendrait une puissance de calcul de 10^{21} opérations à la seconde.

Puisqu'il y a $60 \cdot 60 \cdot 24 \cdot 365 \approx 3 \cdot 10^7$ secondes dans une année, cela ferait au mieux $3 \cdot 10^{28}$ opérations par an.

En supposant (fort abusivement !) qu'il n'y ait qu'une seule opération à effectuer par nombre premier dans le crible d'Eratosthène, il faudrait donc un nombre d'années égal à environ :

$$\frac{1.5 \cdot 10^{297}}{3 \cdot 10^{28}} \approx 5 \cdot 10^{268},$$

ce qui est insensé !

4. **Codages préliminaires**

Le texte de Bernard est certainement trop long pour être codé d'un seul tenant comme un unique nombre $m \in \mathbb{Z}$.

> Je regrette les temps de la grande Cybèle
> Qu'on disait parcourir, gigantesquement belle,
> Sur un grand char d'airain, les splendides cités ;
> Son double sein versait dans les immensités
> Le pur ruissellement de la vie infinie,
> L'Homme suçait, heureux, sa mamelle bénie,
> Comme un petit enfant, jouant sur ses genoux.
> Parce qu'il était fort, l'Homme était chaste et doux. Rimbaud

Il faut donc le découper en blocs de caractères d'imprimerie ayant tous une longueur raisonnable approximativement équilibrée, par exemple ici, vers par vers :

> [Je regrette les temps de la grande Cybèle]
>
> [Qu'on disait parcourir, gigantesquement belle,]
>
> [Sur un grand char d'airain, les splendides cités ;]
>
> [Son double sein versait dans les immensités]
>
> [Le pur ruissellement de la vie infinie,]
>
> [L'Homme suçait, heureux, sa mamelle bénie,]
>
> [Comme un petit enfant, jouant sur ses genoux.]
>
> [Parce qu'il était fort, l'Homme était chaste et doux.] Rimbaud

Pour simplifier, nous supposerons que les caractères d'imprimerie que Bernard utilise se réduisent aux lettres standard de l'alphabet :

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**

avec aussi un espace blanc :

␣
26

ainsi que les lettres majuscules :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
27	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**

et nous supposerons que Bernard pré-code ces caractères de 0 à 52. Dans la vraie vie de la poésie, on doit coder avec une centaine de caractères. Dans la vraie vie des mathématiques, plusieurs centaines sont nécessaires.

Soit alors une portion quelconque du texte de Bernard qui comporte un nombre $\nu \geqslant 1$ de caractères :

$$\ell_0\, \ell_1\, \ldots\, \ell_{\nu-1},$$

appartenant tous à cet alphabet simplifié de lettres, espace, majuscules. Le précodage associe donc à cette portion une suite de ν nombres :

$$m_0\, m_1\, \ldots\, m_{\nu-1}$$

appartenant tous à $\{0, 1, \ldots, 52\}$. Mais comme les ordinateurs sont en général construits pour afficher en base 10 les calculs qu'ils font en base 2, il faut pouvoir transcrire cela en un certain nombre :

$$m \in \mathbb{N},$$

écrit en base 10.

L'idée la plus naturelle consiste alors à faire représenter la suite des m_λ par le nombre écrit en base 53 :

$$m := m_0 + m_1 \cdot 53^1 + \cdots + m_{\nu-1} \cdot 53^{\nu-1},$$

et l'ordinateur calculera alors automatiquement cette somme comme un certain nombre écrit en base 10. La correspondance est bi-univoque (exercice mental), et le nombre m est toujours majoré par :

$$\begin{aligned} m &\leqslant 52 + 52 \cdot 53 + \cdots + 52 \cdot 53^{\nu-1} \\ &= 52 \left(1 + 53 + \cdots + 53^{\nu-1}\right) \\ &= 52\, \frac{53^\nu - 1}{53 - 1} \\ &= 53^\nu - 1. \end{aligned}$$

Rappelons enfin que pour qu'Alice retrouve l'entier m sans ambiguïté modulo n, il est préférable que Bernard découpe son texte en blocs de caractères auxquels correspondront des nombres :

$$m \leqslant n - 1,$$

ce qui revient à demander que :

$$53^\nu \leqslant n,$$

à savoir à assurer que la longueur ν des blocs de caractères d'imprimerie soit majorée par :

$$\nu \leqslant \frac{\log n}{\log 53},$$

ce que Bernard peut fort aisément faire, puisque Alice a rendu publique sa clé n ! En résumé :

Proposition 4.1. *Pour pré-coder son message dans le langage de l'arithmétique des nombres entiers, ayant pris connaissance de la clé publique n d'Alice, Bernard découpe son texte en blocs de caractères d'imprimerie :*

$$\big[\ell_0 \dots \ell_{\nu-1}\big] \quad \big[\ell'_0 \dots \ell'_{\nu'-1}\big] \quad \big[\ell''_0 \dots \ell''_{\nu''-1}\big] \quad \big[\ell'''_0 \dots \ell'''_{\nu'''-1}\big] \quad \big[\ell''''_0 \dots \ell''''_{\nu''''-1}\big] \quad \dots\dots$$

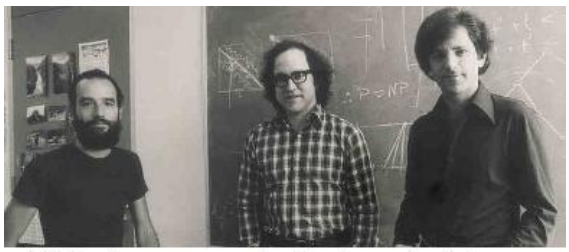
de longueurs $\nu, \nu', \nu'', \dots$ toutes $\leqslant \frac{\log n}{\log 53}$, et il associe à chacun de ces blocs des nombres entiers :

$$\big[m\big] \quad \big[m'\big] \quad \big[m''\big] \quad \big[m'''\big] \quad \big[m''''\big] \dots\dots$$

satisfaisant tous $m, m', m'', \dots \leqslant n-1$, qui seront chacun soumis au processus d'exponentiation RSA. □

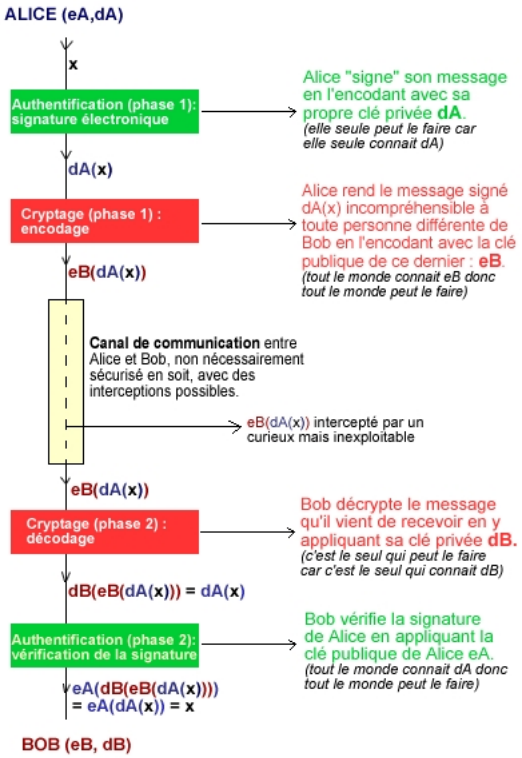
Dans cette approche, on découpe le message en blocs et on utilise RSA sur chaque bloc. Dans la vraie vie, il est plus usuel d'utiliser RSA pour chiffrer une clé et d'utiliser ensuite un chiffrement à clé privée pour faire le chiffrement réel du message. En effet, utiliser RSA sur plusieurs blocs n'est pas sûr : on peut par exemple savoir si le même bloc est transmis plusieurs fois.

5. **RSA : historique et réflexions diverses**



Le système de cryptage RSA a été inventé en 1977 par Rivest, Shamir et Adleman, dont les initiales forment l'acronyme RSA. Ces trois auteurs avaient décidé de travailler ensemble pour établir qu'un nouveau système de codage révolutionnaire, dénommé « *système à clé publique* » que Diffie et Hellman venaient d'inventer, était une impossibilité logique, autrement dit, que tout système de cryptage de cette nature devait présenter certaines failles. Ils ne réussirent pas dans leur projet, mais, au contraire, ils découvrirent un nouveau système à clé publique qui supplanta rapidement celui de Diffie et Hellman.

Rappelons que la dissymétrie fondamentale qui sous-tend le protocole, est qu'il est immédiatement facile de multiplier deux nombres premiers ayant un grand nombre de chiffres, alors qu'il est extrêmement difficile de déterminer les facteurs premiers d'un grand nombre donné. Si l'on s'y prend naïvement, il faut diviser ce nombre par beaucoup de nombres plus petits et déterminer le reste : si celui-ci est nul, on a déterminé ainsi un diviseur. Les mathématiciens ont mis au point des moyens plus rapides que les divisions successives, mais le nombre d'opérations à effectuer reste considérable pour des nombres de taille importante. Ainsi il est actuellement impossible de « factoriser » des nombres de plus de 300 chiffres, même avec les ordinateurs les plus rapides et les algorithmes les plus performants.



Insistons sur le fait que le jeu numérique élémentaire du protocole RSA est aujourd'hui un système universel servant dans une multitude d'applications. Au cours des années, il a devancé tous ses concurrents.

Dear Tim,....please find our revenues and profit statement for the last business year attached. This is confidential information.....Best regards..	QE+ygO5sbQrESVnc71TcgU+U0tEOqhTPNqBr1V/ez7RXS0stkgNGafvEYc3Vw1JDkv4PVJ+Lk1HFhSmZgQ2hcjtFF1ZvkoFu+y3fAUd4LN/q6TrR8YSnL81Fidsi16CrN7nMAgB36mBVL2gL4hYYGhC+z06K+6PJ1WEZXtMONYqZj3PE1whz8UIZCUsCpnEE

Le RSA est programmé aussi dans les systèmes d'exploitation de Microsoft, d'Apple, de Sun. Il est intégré dans les cartes Ethernet et dans certaines cartes à puce bancaires. Un très grand nombre d'institutions gouvernementales, universitaires ou militaires, l'utilisent de façon interne. Le réseau Internet en fait un usage systématique pour assurer la confidentialité du courrier électronique et authentifier les utilisateurs. Bref :

le protocole RSA est partout !

Sur le plan pratique, les experts recommandaient au début des années 2000 d'utiliser des nombres n de 768 bits (232 chiffres décimaux) pour mettre en œuvre le RSA dans le cas de données pas trop importantes, mais ils conseillaient 1 024 bits (309 chiffres décimaux) pour un usage commercial et 2 048 bits (617 chiffres décimaux) pour avoir une garantie se prolongeant sur une longue période de temps. Les clés de 512 bits (155 chiffres décimaux),

encore utilisées, ne devraient plus l'être, car on a réussi, en août 1999, à factoriser un nombre n (produit de deux grands nombres premiers) de 512 bits.

La confiance dans la sécurité du RSA n'est pas due à la démonstration théorique que ce système est sûr, car une telle démonstration n'existe pas. La confiance affichée provient de l'échec, répété depuis plus de 30 ans, de toutes les tentatives entreprises pour casser ce système, tentatives qui n'ont conduit qu'à la formulation de quelques recommandations pour le choix des paramètres p, q, e, d.

Sur le plan théorique, la situation est décevante et le restera certainement longtemps. Celui qui sait factoriser $n = p\,q$ retrouve ensuite facilement d. Inversement, les mathématiques montrent que celui qui connaît n, e et d peut trouver rapidement p et q. La robustesse du RSA apparaît donc liée à la difficulté de la factorisation. Malheureusement, il n'est pas exclu que quelqu'un, sans réussir à obtenir d ni p ni q, puisse décrypter un message : autrement dit, il n'a pas été prouvé que la difficulté du RSA est équivalente à celle de la factorisation.

Deux attaques théoriques du RSA sont donc envisageables. Celle passant par la factorisation : quiconque sait factoriser les nombres de la taille de $n = p\,q$ sait lire comme à livre ouvert tous les messages cryptés par le RSA. Celle contournant la factorisation, dont personne n'a su établir qu'elle était impossible et pour laquelle, au contraire, on a proposé récemment des arguments indiquant qu'elle devait être sérieusement crainte. Boneh et Venkatesan ont en effet établi en 1998 que casser le RSA, lorsqu'il est utilisé avec des exposants e trop petits, n'est pas équivalent à factoriser n.