Printemps des sciences 2007
Mathématiques et cryptographie (Michel Rigo, ULg)
Dossier pédagogique, 4ème – 6ème secondaire

De nos jours, l'envoi de messages secrets requiert la manipulation de nombres ayant plus de cent chiffres décimaux. Nous illustrons une technique cryptographique standard (le RSA) dont la sécurité réside dans le fait qu'il est "rapide", sur le plus banal des ordinateurs personnels, de calculer le produit de deux "grands" nombres (cela se compte au pire en secondes), alors que le temps nécessaire pour effectuer l'opération inverse de factorisation prend, dans l'état actuel des connaissances mathématiques, énormément plus de temps (que l'on pourrait estimer en milliards d'années même pour un super-calculateur !).

Plan de la présentation :

- Cryptographie : définition et applications
- Compter "modulo"
- Codage de texte et chiffrements "élémentaires"
  - Ecriture en base entière
- Cryptographie : clé secrète vs. clé publique
- Les nombres premiers ("briques de construction des nombres")
- Le RSA
- Aller plus loin ?

Ce dossier pédagogique contient plus de détails que la présentation orale prévue pour le Printemps des Sciences.

Support écrit déjà existant :

`http ://www.discmath.ulg.ac.be/`
⟶ Publications
⟶ Vulgarisation scientifique
⟶ "Peut-on avoir confiance en le commerce électronique ?"

Il s'agit d'un autre support provenant d'une formation continuée organisée en 2005–2006 pour les professeurs de l'enseignement secondaire. Cette dernière contient des détails techniques omis dans cette présentation.

Définitions et applications :

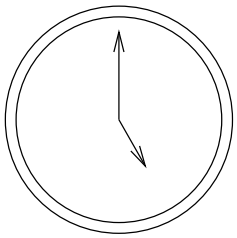
CRYPTOGRAPHIE. n. f. *Art d'écrire en chiffres ou d'une façon secrète quelconque. Ensemble des principes, méthodes et techniques dont l'application assure le chiffrement et le déchiffrement des données, afin d'en préserver la confidentialité et l'authenticité.*

Les applications cryptographiques sont nombreuses :
- Armée, gouvernement
- Banques, transactions bancaires, bancontact, ...
- Internet, paiement en ligne par carte de crédit, ...
- Vote électronique

– GSM (identification, code PIN)
– Télévision payante (à la carte)
– Signatures électroniques, recommandés électroniques, ...
– Mots de passe informatiques, ...

Compter modulo :

On vous a toujours dit que "$1 + 1 = 2$", hé bien, il y a moyen de manière tout à fait rigoureuse d'obtenir un autre résultat. A condition, bien entendu, de savoir exactement de quoi l'on parle ! Nous allons donc vous poser la question suivante : "Savez-vous compter modulo 12 ?", nous allons voir ensemble comment y répondre affirmativement et surtout, nous allons voir ce que cela signifie...



" 17 heures = 5 heures de l'après-midi ! "

Dans cette affirmation, d'une certaine façon, nous identifions (ne serait-ce même qu'inconsciemment) les nombres 5 et 17. Ainsi, lorsqu'il s'agit d'heures, il paraît naturel à tout un chacun lorsqu'il ajoute 5 heures à 22 heures, d'obtenir comme résultat 3 heures (du matin) et donc d'étendre ces conventions de comptage à des opérations comme

$$5 + 22 = 3.$$

Ce calcul, placé dans son contexte, est tout à fait envisageable et conduit à la définition suivante.

Définition : On dit que deux nombres (entiers) x et y sont *congrus modulo* 12 s'ils ont même reste après division par 12. (Ou de manière équivalente, si leur différence est un multiple de 12.)

Exemple : $17 = 1.12 + 5$ et donc 5 et 17 sont congrus modulo 12. Idem, $27 = 2.12 + 3$ et donc 27 et 3 sont aussi congrus modulo 12. Dans le tableau suivant, tous les nombres d'une même colonne sont congrus deux à deux (leur reste, après division par 12, se trouve en tête de colonne).

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	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
⋮											

On s'autorisera à écrire "$5 = 17$", étant sous-entendu que l'on travaille modulo 12 ! Ce que l'on notera aussi $5 = 17 \pmod{12}$ pour éviter toute confusion.

On peut alors sans problème calculer modulo 12, il suffit de ramener le résultat de toute opération arithmétique à son reste après division par 12 ...

$$\begin{array}{rcll} 7+9 & = & 4 \pmod{12} & \text{car} \quad 16 = 1.12+4 \\ 3.11 & = & 9 \pmod{12} & \text{car} \quad 33 = 2.12+9 \\ 5^2 & = & 1 \pmod{12} & \text{car} \quad 25 = 2.12+1 \end{array}$$

Bien sûr, on n'est nullement limité au choix particulier de 12. On aurait d'ailleurs pu s'intéresser aux minutes et pas aux heures et alors compter modulo 60. Ainsi, pour tout entier $m \geq 2$, on dira de deux nombres (entiers) x et y qu'ils sont *congrus modulo* m s'ils ont même reste après division par m. (Ou de manière équivalente, si leur différence est un multiple de m.)

Quelques exemples[1] :

$$4+3=2 \pmod{5},\ 3.7=3 \pmod{6},\ 1+1=0 \pmod{2}.$$

Une application immédiate :

Pour faire de la cryptographie, il faut d'abord pouvoir encoder (identifier chaque symbole à un entier) un texte avant de le chiffrer (il est plus facile de travailler avec des nombres qu'avec des lettres). Nous décidons dès lors de coder les lettres $\texttt{A}, \ldots, \texttt{Z}$ de l'alphabet par les entiers de 0 à 25,

A	B	C	⋯	X	Y	Z
0	1	2	⋯	23	24	25

Vu la taille de notre alphabet, nous allons alors travailler modulo 26 (on pourrait adapter ces considérations à des alphabets d'autres tailles pour par exemple inclure des lettres minuscules ou des symboles de ponctuation ; le codage `ASCII` contient 256 symboles, l'alphabet cyrillique contient quant à lui 32 lettres).

Voici un premier exemple particulièrement simple (et peu sûr pour des données sensibles). L'histoire attribue ce cryptosystème à Jules César : on y remplace chaque nombre n de la suite par $n+3$ modulo 26. Par exemple, un codage suivi d'un chiffrement donne

$$\texttt{BONJOUR} \to 1, 14, 13, 9, 14, 20, 17 \xrightarrow{+3} 4, 17, 16, 12, 17, 23, 20 \to \texttt{ERQMRXU}$$

et pour le déchiffrement, il suffit de soustraire 3,

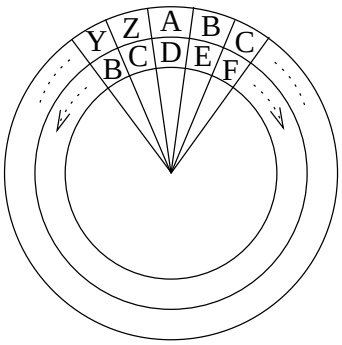
$$\texttt{ERQMRXU} \to 4, 17, 16, 12, 17, 23, 20 \xrightarrow{-3} 1, 14, 13, 9, 14, 20, 17 \to \texttt{BONJOUR}$$

Voici le tableau correspondant au chiffrement de Jules César :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T

[1] Le professeur du secondaire pourrait aussi considérer des problèmes "modulo 7" pour déterminer des jours de la semaine ou encore "modulo 365" si le problème s'étale sur plusieurs années : "*Sachant que le 14 juillet 1789 était un mardi, déterminer quel jour de la semaine était le 14 juillet 1985 ?*" (Il suffit de décaler d'un jour par année car $365 = 1 \pmod 7$ et de deux pour les années bissextiles, les années multiples de 100 mais pas de 400 n'étant pas bissextiles. Ainsi, de 1789 à 1985, on a 196 années dont 47 bissextiles. De plus, $196 + 47 = 5 \pmod 7$, il faut donc "avancer" de 5 jours, la réponse est donc *dimanche*.)

R	S	T	U	V	W	X	Y	Z
17	18	19	20	21	22	23	24	25
20	21	22	23	24	25	0	1	2
U	V	W	X	Y	Z	A	B	C



Deux disques tournant pour des systèmes du type “Jules César”.

On peut aussi facilement ajouter et retrancher une autre quantité que l'entier 3 et obtenir un cryptosystème analogue. On dira que 3 est la clé du cryptosystème. En effet, c'est grâce à cette information qu'une personne peut coder et décoder des messages. Ainsi, au temps de Jules César, seul César et ses généraux avaient convenu en secret du moyen utilisé pour s'envoyer des messages illisibles aux yeux des ennemis. De nos jours, un tel cryptosystème n'est absolument pas sûr. Il est aisé, par simple inspection, de retrouver l'entier utilisé pour réaliser le décalage (par exemple, dans un texte suffisamment long et rédigé en français, il est très probable que la lettre “e” apparaisse avec la plus grande fréquence. De là, on peut facilement supposer la valeur de la clé).

Une idée assez naturelle peut à présent venir à l'esprit du lecteur. Pourquoi ne pas réaliser d'autres opérations que l'ajout d'un entier fixé pour obtenir d'autres cryptosystèmes ? Considérons l'effet de la multiplication par un entier fixé. L'effet d'une multiplication par 5 (toujours modulo 26) est repris dans le tableau ci-dessous :

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

R	S	T	U	V	W	X	Y	Z
17	18	19	20	21	22	23	24	25
7	12	17	22	1	6	11	16	21
H	M	R	W	B	G	L	Q	V

Ainsi, il est facile de chiffrer un texte

$$\texttt{BONJOUR} \to 1, 14, 13, 9, 14, 20, 17 \xrightarrow{\times 5} 5, 18, 13, 19, 18, 22, 7 \to \texttt{FSNTSWH}$$

La question naturelle est à présent de déterminer l'opération permettant de déchiffrer le texte obtenu. Bien évidemment, il suffit de lire la table construite

précédemment de bas en haut. Mais néanmoins, on désirerait une réponse plus générale, de nature purement arithmétique. On peut remarquer que

$$21.5 = 105 = 4.26 + 1 \quad \text{et donc que} \quad 21.5 = 1 \pmod{26}.$$

Dans ce cas, on dira que 21 est l'inverse de 5 modulo 26 (cette dénomination étend[2] la situation rencontrée sur $\mathbb{Q}$ ou $\mathbb{R}$ où l'inverse de 3 est $\frac{1}{3}$ car $3.\frac{1}{3} = 1$). Ainsi, la multiplication par 21 (modulo 26) répond à la question :

$$\texttt{FSNTSWH} \to 5, 18, 13, 19, 18, 22, 7 \xrightarrow{\times 21} 1, 14, 13, 9, 14, 20, 17 \to \texttt{BONJOUR}.$$

On pourrait croire naïvement que tout se passe toujours aussi bien. Nous allons voir qu'il n'en est rien, en considérant à présent la multiplication par 2 (modulo 26) et le tableau correspondant.

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

R	S	T	U	V	W	X	Y	Z
17	18	19	20	21	22	23	24	25
8	10	12	14	16	18	20	22	24
I	**K**	M	O	Q	S	U	W	Y

Le lecteur perspicace aura très vite remarqué le problème : des lettres distinctes (par exemple F et S) sont chiffrées de la même manière. Cela est inconcevable pour ensuite obtenir le déchiffrement. Comme on le voit ci-dessous, le déchiffrement d'un simple mot n'est plus unique ! Et le destinataire légitime du texte chiffré n'a pas de moyen de décider quel est le texte original. L'exemple ci-dessous montre tous les déchiffrements possibles d'un mot de trois lettres,

$$\texttt{OUI} \to 14, 20, 8 \xrightarrow{\times 2} 2, 14, 16 \to \texttt{COQ} \longrightarrow \begin{cases} \texttt{OUI} \\ \texttt{OUV} \\ \texttt{OHI} \\ \texttt{OHV} \\ \texttt{BUI} \\ \texttt{BUV} \\ \texttt{BHI} \\ \texttt{BHV} \end{cases}.$$

Mathématiquement, l'explication est très simple. Nous avons remarqué que modulo 26, 5 possède un inverse (à savoir 21). Par contre 2 n'en possède pas. Autrement dit, quel que soit l'entier n envisagé, $2.n$ diffère toujours de 1 modulo 26. Ainsi, pour construire un cryptosystème valide basé sur la multiplication par un entier fixé, il faut que ce dernier possède un inverse. On dispose du résultat suivant.

Théorème : Un entier $x > 0$ possède un inverse modulo $m \geq 2$ si et seulement si x et m sont premiers entre eux (c'est-à-dire que le seul diviseur commun à x et m est 1).

[2] Il y a une quinzaine d'années, la notion de "groupe" était encore au programme de la quatrième année.

En combinant les deux techniques illustrées ci-dessus, on peut construire des cryptosystèmes affins dont la fonction de chiffrement est donnée par

$$n \mapsto a.n + b \pmod{26}$$

où a est bien sûr premier avec 26.

Pour terminer cette section, nous allons considérer une variante du chiffrement de Jules César faisant intervenir cette fois un codage par blocs, en place d'un codage lettre à lettre (il n'y a rien de difficile, on "change juste d'unité"). On pourrait coder un texte en le découpant en unités de longueur 2 (encore une fois, le passage à des blocs plus longs n'est pas difficile à mettre en oeuvre).

`CR|YP|TO|GR|AP|HE`

Dans ce cas, il faut alors convenir d'un codage ad hoc pour les blocs, celui repris ci-dessous calque la construction réalisée pour le codage des lettres (blocs de longueur 1) : on énumère les blocs par ordre alphabétique.

AA	AB	AC	$\cdots$	AX	AY	AZ
0	1	2	$\cdots$	23	24	25
BA	BB	BC	$\cdots$	BX	BY	BZ
26	27	28	$\cdots$	49	50	51
$\vdots$						$\vdots$
ZA	ZB	ZC	$\cdots$	ZX	ZY	ZZ
650	651	652	$\cdots$	673	674	675

Il n'est en fait pas difficile d'obtenir le code d'un couple de lettres. Il suffit de prendre le code de la première lettre multiplié par 26 auquel on ajoute le code de la seconde lettre. Par exemple, `EK` a pour code $4.26 + 10 = 114$. Pour un codage sur plus de deux lettres, il faut alors faire intervenir des puissances successives de 26 et on remarque que les choses se passent exactement comme pour le système de numération[3] en base 10 pour lequel interviennent les unités, les dizaines, les centaines, etc... (ici, on remplace juste les puissances de 10 par les puissances du nombre de lettres dans l'alphabet). Voici une illustration pour un bloc de longueur 4, `EKBM` donne

$$4.26^3 + 10.26^2 + 1.26 + 13 = 77103.$$

Quel est l'avantage de cette construction ? Avec des blocs de longueur 1, le nombre de clés disponibles pour un chiffrement de Jules César était de 26. Pour des blocs de longueur 2, on travaille à présent modulo $26^2 = 676$ et on dispose donc de 676 clés possibles. Avec des blocs de taille n, on dispose de 26^n clés distinctes ce qui complique la tâche d'une personne mal intentionnée désirant déchiffrer le message sans connaissance de la clé. Néanmoins, ce genre de cryptosystème est inefficace pour une situation réelle (ces chiffrements ne résistent pas aux attaques classiques même d'un petit ordinateur personnel).

[3]Ce serait une occasion pour le professeur du secondaire de présenter les systèmes de numération en base entière comme les systèmes binaire, octal ou hexadécimal que l'on retrouve sur nombre de calculatrices de poche.

Une application "arithmétique" : x est-il divisible par 3 ou par 9 ?

Puisque nous savons à présent manipuler les "entiers modulo", il est clair qu'un nombre x est divisible par 9 si et seulement si $x = 0 \pmod 9$. C'est la définition même de la congruence modulo 9. Si le nombre x s'écrit en base 10 comme

$$\mathtt{x}_n\mathtt{x}_{n-1}\cdots\mathtt{x}_1\mathtt{x}_0$$

où les $\mathtt{x}_i$ sont des chiffres usuels compris entre 0 et 9 (ainsi, $\mathtt{x}_0$ est le chiffre des unités, $\mathtt{x}_1$ celui des dizaines, $\mathtt{x}_2$ celui des centaines, etc...), alors cela signifie que

$$x = \mathtt{x}_n 10^n + \mathtt{x}_{n-1}10^{n-1} + \cdots + \mathtt{x}_1 10 + \mathtt{x}_0. \tag{1}$$

Remarquons que $10^j = 9 \times \underbrace{1\cdots1}_{j \text{ fois}} + 1$ et donc $10^j = 1 \pmod 9$, pour tout $j \leq n$.
Ainsi, si l'on reprend l'égalité (1) et qu'on la considère modulo 10, on obtient

$$x = \mathtt{x}_n + \mathtt{x}_{n-1} + \cdots + \mathtt{x}_1 + \mathtt{x}_0 \pmod 9.$$

De là, on en déduit la règle bien connue :

"*un nombre est divisible par* 9 *si (et seulement si) la somme des chiffres constituant* x *est elle-même divisible par* 9".

Ce fait d'apparence "extraordinaire" réside donc simplement dans quelques règles de calcul modulo. Par exemple, si toutes les égalités ci-dessous sont considérées modulo 9, alors

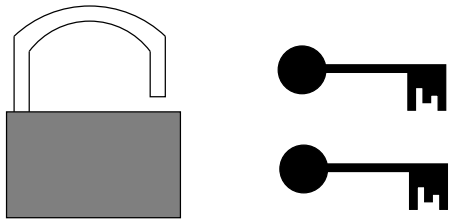
$$12431761938 = 1+2+4+3+1+7+6+1+9+3+8 = 45 = 4+5 = 9 = 0.$$

Ceci signifie que 12431761938 est divisible par 9 (par contre, la règle ne nous apprend rien sur le quotient résultant de la division). On peut voir facilement que cette règle s'adapte aussi à la divisibilité par 3. En effet, on vérifie facilement que $10^j = 1 \pmod 3$, pour tout j. On peut donc reproduire le même raisonnement. Raisonnement qu'il ne sera d'ailleurs pas possible de reproduire pour d'autres diviseurs. Par exemple, si on regarde les puissances de 10 modulo 7, on trouve un suite périodique qui ne donne pas lieu à une règle simple à appliquer ou à énoncer :

n	1	2	3	4	5	6	7	8	$\cdots$
$10^n \pmod 7$	3	2	6	4	5	1	3	2	$\cdots$

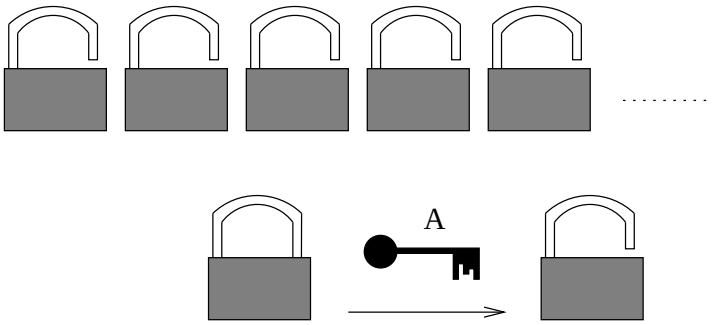
Cryptographie : clé secrète vs. clé publique

Dans l'exemple de Jules César, la personne chiffrant et la personne déchiffrant le message partagent, avant tout échange, la même information : une même clé. De manière imagée, dans un cryptosystème à clé secrète, on dispose d'un seul cadenas (le procédé de chiffrement) permettant de sceller une boîte (le message à envoyer). Pour ce cadenas, on dispose d'un jeu de deux (ou plusieurs, si les échanges doivent impliquer plus de deux personnes) clés identiques utilisées tant pour verrouiller que pour déverrouiller le cadenas.



Ainsi, les deux utilisateurs légitimes peuvent à leur guise ouvrir et fermer le cadenas aussi souvent qu'ils le désirent. Le problème principal de ce type d'échange réside dans la distribution des clés entre les utilisateurs légitimes. En effet, ceux-ci doivent au préalable se rencontrer dans un endroit secret pour fabriquer le jeu de clés. Une personne non autorisée à ouvrir le cadenas doit, pour forcer ce dernier, découvrir par des moyens détournés (vol, copie, essais/erreurs) la forme exacte de la clé.

Ce type de cryptosystème n'est donc en pratique pas toujours possible à mettre en oeuvre puisqu'il nécessite que les deux interlocuteurs désirant converser de manière secrète puissent se rencontrer en toute sécurité (impensable par exemple sur le réseau Internet). Pour pallier à cet inconvénient majeur, on a introduit le concept de *cryptographie à clé publique*. De manière imagée, dans ce type de cryptosystème, un utilisateur A fournit un moyen pour produire facilement des cadenas ouverts. Ces cadenas (ouverts) peuvent être obtenus par quiconque et ils ont la particularité suivante : tout un chacun peut les utiliser pour sceller une boîte, mais une fois le cadenas verrouillé, seul A dispose de la clé permettant d'ouvrir le cadenas.



En effet, lorsque A met à la disposition de tous des cadenas, il ne donne jamais d'information permettant d'ouvrir un cadenas déjà fermé. Ainsi, tout un chacun peut envoyer des messages que seul le destinataire légitime A sera en mesure de lire. Si deux utilisateurs A et B désirent converser, ils devront chacun fournir une méthode de fabrication de cadenas qui leur est propre : A utilisera les cadenas de B et B, ceux de A. Nous verrons dans les prochaines sections comment construire de tels systèmes.

Nombres premiers :

Le cryptosystème RSA dont il sera question à la section suivante repose sur les nombres premiers. Un nombre $p \geq 2$ est *premier* s'il est divisible uniquement par 1 et par lui-même.

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151,
157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233,
239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317,
331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419,
421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503,
509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607,
613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701,
709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811,
821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911,
919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997

Les nombres premiers inférieurs à 1000.

Théorème fondamental de l'arithmétique : Tout nombre entier se décompose de manière unique (à l'ordre des facteurs près) comme produit de nombres premiers.

Par exemple, on a $1960 = 2^3.5.7^2$. Ce théorème montre en quelque sorte que tout nombre est construit à partir de nombres premiers. Cette décomposition en facteurs premiers permet par exemple de retrouver facilement le p.g.c.d. ou le p.p.c.m. de deux nombres.

Revenons au comptage modulo et à un résultat dû à Fermat (aussi attribué à Euler dans sa version généralisée) dont nous aurons usage à la section suivante.

Pierre de Fermat, 1601-1665.

Théorème (Fermat) : Soient p un nombre premier et a un entier qui n'est pas multiple de p. On a

$$a^{p-1} = 1 \pmod{p}.$$

Exemples : $3^4 = 1 + 16.5$ et donc $3^4 = 1 \pmod 5$ ou encore, $132^{642} = 1 \pmod{643}$ (notez que 132^{642} est un nombre de plus de 1350 chiffres en base 10. Il n'est donc pas raisonnable d'effectuer le calcul explicite !).

Leonhard Euler, 1707-1783

Le théorème de Fermat possède la généralisation suivante.
Théorème (Euler) : Soient $n \geq 2$ un entier et a un entier premier avec n, alors

$$\boxed{a^{\varphi(n)} = 1 \pmod{n}.}$$

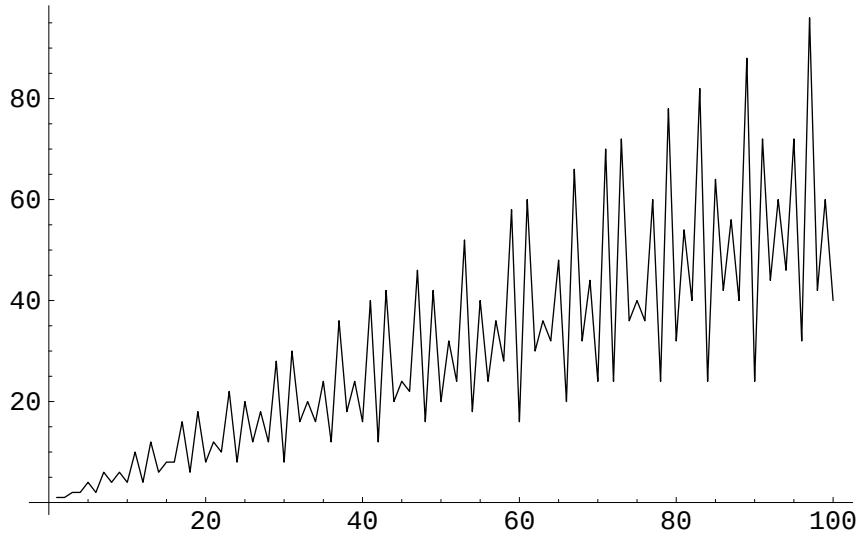
La fonction $\varphi(n)$, appelée *fonction indicatrice d'Euler*[4], compte le nombre de nombres inférieurs à n et premiers avec n (i.e., qui ont 1 comme seul diviseur commun avec n). En voici les premières valeurs, à chaque fois, on a repris les nombres premiers avec l'entier considéré

$$\begin{array}{l|l} \varphi(2) = 1 & 1 \\ \varphi(3) = 2 & 1, 2 \\ \varphi(4) = 2 & 1, 3 \\ \varphi(5) = 4 & 1, 2, 3, 4 \\ \varphi(6) = 2 & 1, 5 \\ \varphi(7) = 6 & 1, 2, 3, 4, 5, 6 \\ \vdots & \end{array}$$

On remarque en particulier que si p est premier, alors $\varphi(p) = p - 1$ et on retrouve le théorème de Fermat comme cas particulier du théorème d'Euler. Voici quelques exemples numériques : $3^2 = 1 \pmod 4$, $5^2 = 1 \pmod 6$, ...

Nous verrons bientôt comment ces résultats de nature arithmétique et apparemment peu "appliqués" vont nous permettre de réaliser des chiffrements utilisés quotidiennement dans des millions de transactions.

[4]Lorsqu'on s'intéresse aux racines (complexes) de l'unité, on dit que ω est une racine n-ième *primitive* de l'unité si le plus petit entier $t > 0$ tel que $\omega^t = 1$ est $t = n$. Il n'est alors pas difficile de voir que $e^{2ik\pi/n}$ est une racine primitive si et seulement si k est premier avec n. En conclusion, le nombre de racines n-ièmes de l'unité qui sont primitives vaut $\varphi(n)$.



Les premières valeurs de la fonction φ.

1, 1, 2, 2, 4, 2, 6, 4, 6, 4, 10, 4, 12, 6, 8, 8, 16, 6, 18, 8, 12, 10, 22,
8, 20, 12, 18, 12, 28, 8, 30, 16, 20, 16, 24, 12, 36, 18, 24, 16, 40, 12, 42,
20, 24, 22, 46, 16, 42, 20, 32, 24, 52, 18, 40, 24, 36, 28, 58, 16, 60, 30,
36, 32, 48, 20, 66, 32, 44, 24, 70, 24, 72, 36, 40, 36, 60, 24, 78, 32, 54,
40, 82, 24, 64, 42, 56, 40, 88, 24, 72, 44, 60, 46, 72, 32, 96, 42, 60, 40

Les nombres premiers interviennent dans de nombreuses constructions très importantes en mathématiques (ou en sciences de manière générale) et bien qu'ils soient particulièrement simples à définir, certaines de leurs propriétés restent encore à découvrir. Par exemple, la conjecture[5] des nombres premiers jumeaux stipule qu'il existe une infinité de nombres premiers de la forme p et $p+2$. Ainsi, 3 et 5 ou encore 641 et 643 sont des nombres premiers jumeaux. Mais on ne sait pas avec certitude s'il existe ou non une infinité de nombres de cette forme.

Tester si un nombre est ou non premier, peut paraître enfantin. En effet, il suffit d'essayer tous les diviseurs possibles pour savoir si le nombre envisagé est ou non premier. Cependant, une telle procédure peut prendre énormément de temps. Imaginer vouloir tester si le nombre[6]

> 3107418240490043721350750035888567930037346022842727545720
> 1619488232064405180815045563468296717232867824379162728380
> 3341547107310850191954852900733772482278352574238645401469
> 1736602477652346609

est ou non premier... La racine carrée de ce nombre vaut approximativement 10^{91}. Disposant d'une machine capable de tester 10^9 diviseurs à la seconde, il faudrait environ 10^{74} années pour passer tous ces diviseurs en revue, cela dépasse sans commune mesure l'âge de l'univers ! Ainsi, il paraît clair que la recherche

[5] Une *conjecture* est une propriété non vérifiée (i.e., non prouvée) que l'on pense (à tort ou à raison) être vraie. Une telle supposition repose souvent sur l'expérimentation ou sur l'intuition qu'acquièrent les chercheurs.

[6] *RSA Challenge number* `RSA-640` factorisé le 5 novembre 2005 après un calcul de plusieurs mois sur des centaines d'ordinateurs en réseau.

en mathématique doit être poursuivie pour espérer obtenir des méthodes satisfaisantes pour résoudre ce genre de problème. Car, comme nous le verrons, il s'agit d'une question cruciale pour les systèmes informatiques d'aujourd'hui.

A ce jour, le plus grand nombre premier connu vaut $2^{32582657}-1$ qui s'il était écrit en base 10, s'écrirait avec 9 808 358 chiffres ! De la même manière, les plus grands nombres premiers jumeaux connus sont $100314512544015.2^{171960} \pm 1$ qui s'écrivent avec 51 780 chiffres décimaux.

Le RSA

Nous l'avons suffisamment annoncé, il est grand temps d'introduire le cryptosystème à clé publique RSA, des noms de ses concepteurs : Rivest, Shamir, Adleman. L'idée repose sur le simple constat suivant. Il est très simple de calculer le produit de deux nombres entiers (par calcul écrit, comme à l'école primaire, cela fonctionne très bien même avec des nombres "très grands"). Par contre, comme on a déjà pu s'en convaincre naïvement, il est *a priori* très long et pénible d'effectuer l'opération inverse (à savoir, si l'on dispose d'un nombre qui est le produit de deux autres nombres, retrouver ces derniers).

Ron Rivest, Adi Shamir et Leonard Adleman (1977).

Détaillons-en les ingrédients. Imaginons avoir deux protagonistes, Alice et Bob, désirant converser secrètement. Bob désire envoyer un message chiffré à Alice (et qu'elle seule pourra déchiffrer).

Alice choisit deux grands nombres premiers distincts p et q. Elle en calcule (sans problème) le produit

$$n = p.q.$$

Puisque p et q sont premiers, on peut montrer (c'est une des propriétés fondamentales de la fonction φ) que la valeur en n de la fonction indicatrice d'Euler est

$$\varphi(n) = (p-1).(q-1).$$

Alice choisit à présent deux nombres e et d tels que

$$d.e = 1 \mod \varphi(n).$$

Pour ce faire, elle choisit e tel que

$$1 < e < \varphi(n) \quad \text{et} \quad \text{pgcd}(e, \varphi(n)) = 1.$$

En effet, nous avons vu que pour qu'un nombre possède un inverse modulo $\varphi(n)$, autrement dit pour trouver un d tel que $d.e = 1 \pmod{\varphi(n)}$, il doit être premier avec $\varphi(n)$ (souvenez-vous de l'exemple de la multiplication par 2 ou par 5 modulo 26). Alice publie n et e et conserve secrets les autres éléments $d, p, q, \varphi(n)$. On appelle e (resp. d) l'*exposant de chiffrement* (resp. *de déchiffrement*). On dira de plus que le couple $k = (e, n)$ est la *clé* du système. Si les éléments à chiffrer sont des entiers x inférieurs à n, le chiffrement est calculé par la fonction suivante

$$\boxed{e_k(x) := x^e \mod n.}$$

(On peut calculer assez rapidement et sans grande difficulté une telle exponentiation. On parle alors d'exponentiation modulaire.)

Avec les notations précédentes, si $k = (e, n)$ et si $y = e_k(x)$, alors grâce au théorème d'Euler, on peut montrer que

$$\boxed{y^d \mod n = x.}$$

En effet, si x est premier avec n, alors

$$y^d = x^{ed} = x^{1+k\varphi(n)} = x.x^{k\varphi(n)} = x.(x^{\varphi(n)})^k = x.1 \pmod{n}.$$

Si x n'est pas premier avec n, la preuve est un peu plus technique mais est tout de même basée sur le même principe.

Cette proposition nous montre que pour déchiffrer un message, il suffit d'élever le texte chiffré à la puissance d et on posera donc

$$d_k(y) := y^d \mod n.$$

Connaissant d, le déchiffrement est donc semblable au chiffrement.

En résumé, on a les données suivantes :

- Alice publie : $k = (e, n)$,
- Alice conserve secrets : d, p, q, $\varphi(n)$,
- fonction de chiffrement (utilisée par Bob ou tout autre personne envoyant un message à Alice) : $e_k(x) = x^e \mod n$,
- fonction de déchiffrement (utilisée par Alice) : $d_k(y) = y^d \mod n$.

Toute personne voulant envoyer un message à Alice dispose du couple (e, n) publié par exemple dans un annuaire ou sur la page web d'Alice. Par contre, un message chiffré ne peut, en principe, être déchiffré que par Alice, car elle seule dispose de d. Pour retrouver d à partir de (e, n), il faut être en mesure de factoriser n. A ce jour, la factorisation de "très grands" nombres entiers est impraticable en un temps raisonnable. Ainsi, la sécurité du RSA réside dans le fait qu'il est facile de faire le produit n de deux grands nombres premiers p et q mais qu'il est par contre *supposé difficile*[7] de factoriser n. On peut en outre montrer que la connaissance de d ou de $\varphi(n)$ revient à la factorisation de n.

[7] On ne dispose pas de preuve sur ce caractère difficile. On pourrait imaginer que des avancées significatives dans la recherche en mathématique permettraient d'accélérer grandement les méthodes de factorisation et réduire ainsi à néant la sécurité du RSA.

Exemple. Considérons un exemple rudimentaire (et très peu réaliste). Soient les entiers :

$$p = 11,\ q = 23,\ n = p.q = 253,\ e = 3.$$

Puisque $220 = 3.73 + 1$, on en déduit que l'exposant de déchiffrement d est tel que

$$3^{-1} = -73 \mod 220 = 147 \mod 220.$$

Alice publie $k = (3, 253)$. Si Bob veut envoyer le texte clair $x = 165$ à Alice, il calcule

$$e_k(x) = 165^3 \mod 253 = 110.$$

Si Alice reçoit le message $y = 110$, il lui suffit de calculer

$$d_k(y) = 110^{147} = 110^{128}.110^{16}.110^2.110^1 \mod 253 = 165.$$

Pour assurer la sécurité du RSA, il est de coûtume de prendre des nombres premiers p et q de l'ordre de 2^{512}. Un nombre de cette taille écrit en base 10 possède environ

$$\lfloor \log_{10} 2^{512} \rfloor = \left\lfloor 512 \underbrace{\log_{10} 2}_{\sim 0,3} \right\rfloor = 154$$

chiffres décimaux. De plus, on effectue aussi un codage par blocs et non pas lettre à lettre. Pour ne pas alourdir[8] cette présentation, nous ne donnerons pas de détail sur la taille des blocs à envisager. (Cela n'apporterait que des détails techniques sans idée nouvelle).

Enfin, il est à noter que le RSA est aussi utilisé pour la signature et l'authentification de messages électroniques.

Pour aller plus loin :

Bien que nous n'ayons qu'effleuré le problème du chiffrement, le lecteur peut aisément être convaincu que la cryptographie suscite (et suscitera encore longtemps) de nombreuses questions en mathématique. Par exemple, la sécurité du RSA est conditionnée par les avancées des chercheurs travaillant entre autres sur les algorithmes de factorisation, ces derniers reposant souvent sur des concepts mathématiques avancés (tels que le crible quadratique). De plus, pour ne pas mettre tous ses oeufs dans le même panier, il est nécessaire d'explorer d'autres pistes pour construire des cryptosystèmes à clé publique dont la sécurité ne reposerait pas sur la factorisation de grands entiers. De tels systèmes existent et sont par exemple construits sur des courbes elliptiques (et le problème du logarithme discret). L'étude de ces courbes et des structures associées forme à elle seule un domaine de recherche à part entière (une recherche bibliographique rapide donne plus de 1700 articles écrits sur le sujet depuis 2000). De plus, les cryptosystèmes à clé secrète sont eux aussi en perpétuelle évolution, ces derniers étant souvent bien plus rapides et moins gourmands en ressources que les systèmes à clé publique.

Ces quelques exemples démontrent que la recherche actuelle en mathématique peut déboucher sur des applications concrètes utilisées quotidiennement

[8] Les détails concernant la taille des blocs ou encore la génération de grands nombres premier se trouvent par exemple dans [3].

et ce, même de manière *a priori* inattendue. Si l'on s'intéresse à la génération de grands nombres premiers (question naturelle pour la mise en oeuvre du RSA), Agrawal, Saxena et Kayal ont montré en 2001 que "`Primes` $\in$ P" (ce qui signifie grosso-modo que tester si un nombre est premier peut être réalisé en un temps raisonnable par rapport à la taille de l'entier). La théorie des nombres, longtemps considérée comme n'ayant aucune application (et dès lors, appelée par certains[9] la discipline reine des mathématiques), recèle en fait de résultats débouchant sur des techniques cryptographiques. L'étude difficile de la distribution des nombres premiers n'est pas en reste : fonction zeta de Riemann[10], conjecture des nombres premiers jumeaux ou encore un résultat récent de Terence Tao, médaille Fields 2006, étudiant les progressions arithmétiques présentes dans la suite des nombres premiers ... Tous ces chercheurs apportent de par leurs résultats des briques nouvelles, utilisées par d'autres, pour construire, améliorer ou encore attaquer des cryptosystèmes.

Les quelques lignes de cette section ont un double but. Elles sont en quelque sorte un plaidoyer pour la recherche tant en mathématiques appliquées qu'en mathématiques pures (on ne saurait prédire quels résultats auront ou non des applications). D'autre part, elles démontrent que les mathématiques tant élémentaires qu'avancées ont des applications dans la vie de tous les jours et ce, même si l'on ne s'en rend pas toujours compte (y pensez-vous lorsque vous entrez le code PIN de votre téléphone portable ?). De cette manière, nous apportons une réponse (partielle) à la question si souvent entendue : "*Mais les maths, ça sert à quoi ?*". En effet, les techniques et les outils employés en cryptographie sont nombreux et variés : analyse, analyse complexe, analyse numérique, algèbre matricielle, algèbre générale, statistique, probabilités, ...

La cryptographie n'est en fait qu'un prétexte pour illustrer une application des mathématiques : saviez-vous par exemple que le système de classement des pages internet effectué par Google repose en fait sur un très beau résultat d'algèbre linéaire, le théorème de Perron-Frobenius, datant du début du XX-ième siècle. Ainsi, les ordinateurs de Google calculent sans cesse des produits de matrices immenses pour mettre à jour les classements des sites internet. De tels exemples peuvent être multipliés à foison, mais ça c'est une autre histoire.

Références

[1] S. Flannery, D. Flannery, In Code : A Mathematical Journey, Algonquin Books, 2001.

[2] M. Rigo, Structures discrètes, notes de cours, licences et troisièmes bacheliers en sciences mathématiques, `http ://www.discmath.ulg.ac.be/`

[3] M. Rigo, Peut-on avoir confiance en le commerce électronique ?, texte de vulgarisation scientifique, `http ://www.discmath.ulg.ac.be/`

[4] A. Salomaa, Public key cryptography, Eatcs Monographs on Theoretical Computer Science, Vol 23, Springer, 1996.

[5] D. Stinson, Cryptographie : Théorie et pratique, Vuibert, 2003.

[9] "*La mathématique est la reine des sciences et la théorie des nombres est la reine des mathématiques.*", Carl Friedrich Gauss 1777-1855.

[10] L'hypothèse de Riemann stipulant que les zéros non triviaux de la fonction ζ ont tous une partie réelle égale à 1/2 est l'un des sept problèmes du Millénaire doté d'un prix d'un million de dollars (Clay Mathematics Institute) et il s'agissait aussi du huitième problème de Hilbert.

Les références [2, 3] sont des textes accessibles en ligne permettant d'aller plus loin que le survol présenté ici. On préférera certainement [3] d'approche plus simple que [2] destiné à un public spécialiste. La référence [5] est un texte "classique" abordant les thèmes majeurs de la cryptographie (des prérequis mathématiques solides s'imposent). Bien qu'en anglais, la référence [4] se lira certainement plus facilement (on y aborde également des points ardus mais le style de l'auteur, membre de l'Académie des Sciences de Finlande, est remarquable). Enfin, on notera le livre [1] qui explique l'histoire d'une lycéenne irlandaise ayant développé son propre cryptosystème. Elle y présente les concepts mathématiques qu'elle a du apprendre pour mener à bien son entreprise (par ailleurs, il faut savoir que son père est professeur de mathématiques).

Michel Rigo
Département de Mathématiques
Université de Liège
Grande Traverse 12 (B37)
4000 Liège
M.Rigo@ulg.ac.be