Autour de la cryptographie, dans une classe de Terminale S

Françoise Gaydier(*)

Je suis partie de constats :
– La cryptographie actuelle, dont on n'imagine pas toujours à quel point elle concerne tout un chacun dans sa vie quotidienne, utilise des résultats mathématiques fort vieux, dont on a pu penser à certaines époques qu'ils ne serviraient jamais à rien. Je pense par exemple au théorème d'Euler-Fermat.
– La problématique de la cryptographie, simple à comprendre sans schématisation réductrice, permet non seulement de proposer des exercices touchant à de nombreuses parties du programme de TS, ce que je voudrais illustrer ici, mais également d'aborder des problématiques dans des domaines voisins des maths ou nouveaux en maths : programmation et complexité d'algorithmes.
– De plus, dans la mesure où c'est une problématique ouverte (il demeure des questions sans réponses à ce jour concernant la sûreté des cryptages actuels), il est possible de trouver des articles à donner à lire, qui soient mieux que des textes sur mesure pour lycéens éveillés : de vrais textes d'information ; ainsi par exemple, sur le site de la Société Mathématique de France, l'article de Jean-Louis Nicolas de l'Université Claude-Bernard (Lyon 1), mais aussi de temps en temps dans la presse l'annonce d'un record battu concernant la taille d'un entier décomposé en produit de deux facteurs premiers, annonce parfois assortie d'explications sur ce qui est en jeu autour de ces records : la sûreté des cryptages.

1. La problématique

Il faut, au départ, faire réaliser aux élèves qu'il y a dans la langue courante une ambiguïté sur le sens de « code » : l'enfant de 8 ans qui dit qu'il envoie à ses amis un message codé, sous-entend qu'il s'agit d'un message contenant un secret que seuls les destinataires sont censés pouvoir décoder, autrement dit, il s'agit d'un message crypté.
Dans un contexte de classe Terminale, « coder » ne doit plus sous-entendre une idée de secret : les élèves doivent réaliser que beaucoup de choses sont codées, par des codes souvent transparents ; ainsi 2005 est la représentation en base dix d'un nombre : on a utilisé pour ce faire un code qui permet de représenter n'importe quel nombre entier, mais ce code n'est pas secret, tout le monde sait en principe le déchiffrer.
De même l'utilisation du code morse pour envoyer un SOS exige que le message ainsi codé puisse être déchiffré par n'importe quel capitaine de bateau croisant à proximité…
Tout cela paraît évident, mais il n'est pas si facile que cela de faire exprimer cette

(*) Lycée Claude Monet, Paris 13º.

évidence lorsque l'on pose à un groupe d'élèves de TS, spécialité maths la question : « **codage et cryptage, qu'est-ce que cela évoque pour vous ?** ».

Ces questions de terminologie éclaircies, il s'agit alors de dégager la problématique du cryptage : A et B doivent échanger des messages secrets ; pour cela ils conviennent de les crypter.
De façon classique, A et B se rencontraient ou utilisaient un intermédiaire pour convenir des codes secrets qu'ils allaient utiliser pour crypter et décrypter les messages secrets qu'ils voulaient s'envoyer.
Cette manière de procéder posait plusieurs problèmes :

Premier problème : un tiers caché dans les buissons pouvait intercepter ces codes ; c'est le problème de l'échange des clés (de cryptage et, éventuellement, de décryptage).
Deuxième problème : le plus souvent, lorsque l'on connaissait le code pour crypter, on pouvait connaître le code pour décrypter : le codage par translation ou le codage affine (**annexe 2**) en sont de bons exemples. Non seulement un tiers malveillant pouvait, s'étant procuré la clé de cryptage, envoyer un faux message, mais il pouvait aussi, en s'étant emparé seulement de la clé de cryptage, fabriquer la clé de décryptage et déchiffrer le message secret. Ce problème rendait donc encore plus cruciale la confidentialité de l'échange des clés.
Troisième problème : il fallait faire un code secret suffisamment robuste pour qu'un tiers interceptant le message crypté ne puisse pas, bien que ne connaissant ni le procédé de cryptage ni celui de décryptage, déchiffrer quand même le message (voir l'activité 7 de décryptage par analyse fréquentielle).

Deux personnes donc, A et B, veulent se communiquer une information que de tierces personnes ne doivent pas connaître. Plus précisément, A doit envoyer à B une information qui doit rester secrète.
L'avancée dans la cryptographie moderne se situe dans la « **dé-symétrisation** » des rôles de A et de B : A est l'*émetteur* et B le *récepteur* du message.
Le coup de génie dans le cryptage moderne est de rendre publique, par le récepteur B, la clé de cryptage. Évidemment cela exige que la clé de décryptage ne puisse pas se déduire de la clé de cryptage : elle reste privée, c'est-à-dire connue du seul récepteur B.

Le livre de Simon Singh (J.-C. Lattès) : *Histoire des codes secrets*, détaille les étapes de cette invention du **cryptage à clé publique** qui a abouti à la fin des années 1970 ($\approx$1977).
La solution aujourd'hui le plus souvent retenue, le cryptage R.S.A., est fondée sur le théorème d'Euler-Fermat (**annexe 3**).

Résumons la situation actuelle : on utilise un cryptage à clé publique (par exemple le cryptage RSA) : le récepteur B diffuse sans se cacher la clé avec laquelle les émetteurs doivent crypter les messages confidentiels qu'ils veulent lui envoyer.
Le procédé de cryptage est tel que lorsque l'on connaît seulement la clé de cryptage, on peut crypter (plus ou moins facilement) mais on ne peut pas décrypter (voir

cependant l'activité 8).
Seul B dispose de la clé de décryptage qui est sa clé privée.

En fait, de tels cryptages sont coûteux en temps. Pour réduire les temps de codage/décodage du message, on utilise un procédé en deux étapes : le cryptage à clé publique évoqué ci-dessus est utilisé pour crypter un second code secret avec lequel sera crypté le message, ce second code étant tout à la fois moins gourmand en temps de codage, et cependant assez sophistiqué pour que le décryptage du message ainsi codé soit laborieux, ce qui doit protéger le secret suffisamment longtemps. Le premier cryptage, qui permet la distribution des clés, doit, lui, être inviolable. Ce procédé permet de répondre au premier problème évoqué : celui de la distribution des clés.

2. Diverses activités menées avec les élèves sur le thème de la cryptographie.

1°) **lecture d'un article scientifique** (cité plus haut) en début d'année : j'ai eu peu de retour de la part des élèves, à part la difficulté du paragraphe concernant les courbes elliptiques que j'avais suggéré de survoler. J'entends par « peu de retour » le fait que les élèves ont dit avoir globalement compris le texte et n'ont pas posé de question en dehors du paragraphe sur les courbes elliptiques.

2°) **programmation sur calculatrice d'un test : un nombre est-il premier ? (annexe 1)**.
Le problème de la difficulté de la décomposition d'un nombre en produit de nombres premiers a été abordé lors du chapitre sur les nombres premiers : où sont les nombres premiers ? comment savoir si un nombre est premier ? comment faire avec une machine ? enregistrer tous les nombres premiers n'est pas possible, etc.
Mon objectif était de montrer que le temps de calcul pour tester la primalité d'un nombre augmente vite lorsqu'on augmente la taille du nombre, afin d'introduire l'idée de **complexité d'algorithme**.
Résultats obtenus : quelques élèves ont fait un programme, mais ils ne semblaient pas trouver le temps de calcul long. J'ai fini par réaliser que nous n'avions pas le même outil de calcul : avec ma vieille TI 89, il faut être patient… Avec une TI 83 ou équivalent, ce n'est en fait pas très spectaculaire : une dizaine de minutes pour déclarer premier un nombre de 10 chiffres.
Il me faudra l'an prochain faire la manipulation avec une calculatrice formelle pour que la manipulation soit parlante (voir aussi annexe 4).

3°) **Devoir à la maison pour l'enseignement de spécialité : codage de César, codage affine** : sujet inspiré, avec son aimable autorisation, d'un travail de Martine Bühler (**annexe 2**). Le devoir a été réussi, mise à part la dernière question (voir plus loin).

4°) **Travail en tronc commun** :

a) **Dénombrement des codages caractères par caractères** (ou **monographiques**) : il y en a 26 !

Puis, calcul de durées quand on essaye tous les codages (le pire des cas) en supposant qu'on ait besoin de deux secondes par essai.
J'ai mené ce calcul de durées car il m'est apparu, lors de la correction du devoir évoqué en 3°), que les élèves pensaient que la fragilité d'un codage monographique résidait dans le fait que l'on pouvait trouver facilement la clé : « il y en a peu, donc avec un ordinateur … ».
J'ai alors découvert que l'ordre de grandeur de 26 ! ne saute pas aux yeux : ce n'est après tout qu'un entier qui s'écrit avec 3 signes…
Remarque : j'ai constaté également un peu plus tard que prendre x grand sur calculatrice, c'est prendre $x = 50$ *par exemple… Je voulais mettre en défaut la calculatrice pour conjecturer la limite à l'infini de* $\left(1+\frac{1}{x}\right)^x$. *J'ai obtenu timidement* $x = 1\,000$ *ou* $x = 10\,000$. *Ma proposition* $x = 10^{48}$ *les a surpris…*

b) **Exercice de probabilités** :
Sachant qu'un message a été codé caractères par caractères, que le code a été choisi au hasard et que l'on a besoin d'une durée Δt pour essayer un code, quelle est la durée moyenne pour décrypter le message ?

5°) **Devoir en classe de spécialité : le théorème de Fermat-Euler** (**annexe 3**). Je n'ai malheureusement gardé de ce devoir (d'une durée d'une heure) que la note globale obtenue par chaque élève pour le DS de quatre heures dont cet exercice était une composante. En tout cas, lors de l'exposé évoqué plus loin, les élèves se souvenaient bien du résultat démontré.

6°) **Introduction à la notion de bijectivité** sur la base du devoir à la maison (**annexe 2**).
Ce fut ma transition tout à fait inattendue entre l'arithmétique et les similitudes…
J'avais à revenir sur la question 1°) du *Cas général*. Je me suis alors aperçue que certains élèves n'avaient pas vu l'importance pour le décryptage que TOTO et TATA ne soient pas codés de la même manière.
En faisant un schéma pour expliquer le problème du décodage, je me suis rendu compte que j'étais en train de faire les diagrammes que j'avais l'intention de faire l'heure suivante pour commencer mon cours sur les similitudes : « une similitude est une **bijection** qui conserve les rapports de longueurs ».
J'ai fait mes diagrammes et j'ai enchaîné sur la feuille polycopiée que j'avais préparée sur la notion de bijectivité. J'ai pu illustrer mon propos non seulement avec les transformations déjà connues et leurs bijections réciproques, les fonctions « carré » et ln, « racine carrée » et exp, mais aussi les fonctions de codage et décodage que nous venions d'étudier.

7°) **Décryptage par analyse fréquentielle** d'un texte crypté caractères par caractères : ce texte est le texte proposé en première étape d'un concours par Simon Singh dans son livre déjà évoqué (p. 382).
J'ai vidéoprojeté le texte à décrypter préalablement scanné et transformé en fichier Word par un logiciel de reconnaissance de caractères. Les élèves l'ont décrypté en

utilisant des données statistiques sur la fréquence des caractères dans la langue française, et quelques commandes Word (« rechercher » et « remplacer ») avec un petit problème technique à résoudre quand on remplace un caractère par un autre…
Ce fut rapide, plaisant et probant quant à la faiblesse du cryptage monographique. Pour être honnête, les espaces étant en place, l'outil informatique était utile mais non nécessaire…

8°) **Exposé de trois élèves sur le cryptage RSA** (**annexe 5**).
J'avais proposé assez tôt dans l'année à quatre bons (voire très bons) élèves de faire un exposé sur le cryptage. Ils en étaient d'accord, puis une élève trop prise par ses activités extra scolaires s'est dédite.
Plus tard j'ai fixé avec eux une date : après les soutenances de TPE et le Bac Blanc. Je les ai laissés travailler en autonomie.
La compréhension de cet exposé par l'ensemble de la classe devait être facilitée par le travail entrepris avec le DS cité plus haut. J'avais, dans le même but, proposé aux élèves de traduire sur leur calculatrice un algorithme permettant la détermination de solutions entières particulières pour l'équation $au + bv = 1$ lorsque a et b sont premiers entre eux (un élève avait essayé tout seul sans aboutir, et je voulais pousser les autres à programmer).
L'exposé a été remarquable :
– Un bon plan.
– Les trois élèves ont parlé, sans cabotinage.
– Le contenu était solide et amené de manière plaisante. Le groupe a donné la preuve du décryptage dans le cas général (cf. **annexe 5**). Il a été jusqu'à évoquer le problème de la génération de grands nombres premiers (ou qui ont une forte probabilité de l'être), et a su expliquer ce qui faisait la solidité de RSA, mais aussi peut-être sa fragilité : on ne sait pas actuellement « casser » n en sa décomposition $n = pq$ en un temps raisonnable, mais on n'a pas prouvé que c'est impossible.
– Les autres élèves ont bien participé et posé de bonnes questions (par exemple : et si p et q ne sont pas premiers ? comment fait-on pour avoir de grands nombres premiers ?).
Le point faible a été la complexité d'algorithme : ils ont essayé de parler de calculs en temps polynomial ou en temps exponentiel, mais sans conviction.
Je me suis moi-même « plantée » en tentant d'improviser sur le sujet. Les élèves qui faisaient l'exposé étant demandeurs, j'ai essayé de préparer un travail sur la question (cf. **annexe 4**), mais, en dehors des convergences de suites, bien qu'ayant plus de dix ans de pratique de l'enseignement de l'algorithmique, je ne trouve pas facilement d'algorithmes abordables dans ce cadre pour évaluer des temps de calcul : les algorithmes de tri ou de résolution de systèmes me paraissent difficiles dans ce contexte.

9°) **Travail sur la complexité d'algorithme avec calculatrice**
en tronc commun par une étude de rapidité de convergences de suite ou d'algorithme :

– calcul de $\sqrt{2}$ par la méthode de Héron : $u_0 = 2$ et, pour tout n,

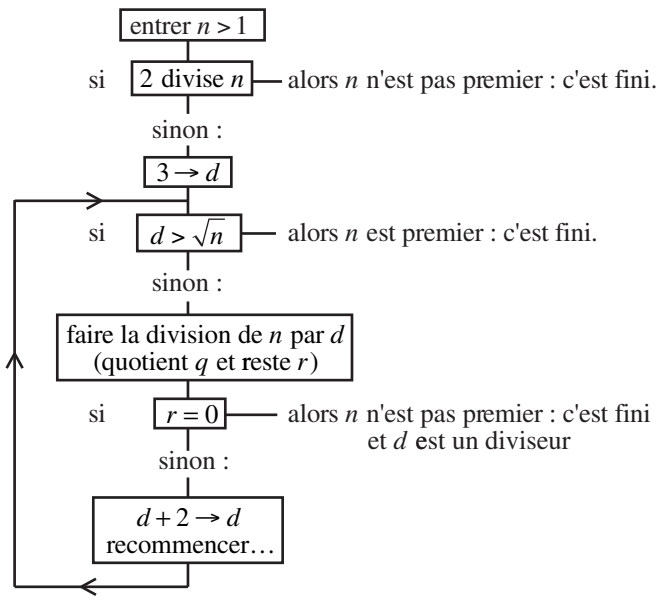
$$u_{n+1} = \frac{1}{2}\left(u_n + \frac{2}{u_n}\right) ;$$

- calcul de π par la méthode de Monte-Carlo en DM : simulation avec la calculatrice de gouttes de pluie tombant aléatoirement sur un disque inscrit dans un carré ;
- calcul de π par la méthode des polygones inscrits et exinscrits en DM.

En conclusion : ce thème de la cryptographie m'a permis de donner la possibilité à de bons élèves d'aller plus loin en restant dans le cadre du programme de l'année en cours et sans déflorer celui de l'année suivante.
Je pense (j'espère) que ce travail sur la cryptographie a permis aussi à l'ensemble du groupe de voir des applications des mathématiques dans un champ où on ne les imagine pas. Il m'a permis également, en évoquant (sommairement certes) les problèmes de complexité d'algorithme et la conjecture P ≠ NP de leur suggérer que les mathématiques sont bien vivantes et pas seulement prestataires de service.
Je n'ai aucun moyen pour mesurer l'impact de ce travail : je pense qu'il fait partie de ce travail de fourmis que nous devons mener quotidiennement pour éveiller des curiosités, relancer des motivations et parfois susciter des enthousiasmes en mathématiques.

Annexe 1
TS4 – Spécialité math –

Un algorithme simple est représenté par l'organigramme ci-dessous : il permet de tester si un nombre entier naturel n est premier ou non.
Traduisez cet algorithme dans le langage de votre calculatrice.



Annexe 2
Devoir à la maison (2004/2005)

Systèmes de codages monographiques

Dans ce type de codage, chaque lettre de l'alphabet est transformée par codage en une autre lettre de l'alphabet.
Question : combien y a-t-il de permutations des lettres de l'alphabet ?

Dans la suite, chaque lettre de l'alphabet est associée à un nombre entier compris entre 0 et 25 (à l'aide de son rang dans l'alphabet). Un système de codage monographique est donc défini par une application f de $\{0, 1, 2, 3, \ldots, 25\}$ dans lui-même.

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

EXERCICE 1 : codage par translation (système de César)

Soit x le nombre associé à une lettre de l'alphabet (c'est-à-dire son rang dans l'alphabet), et $f(x)$ le nombre associé à la lettre qui crypte ce caractère. On considère le codage défini par :

$$f(x) \equiv x + 3\,[26] \quad \text{avec } 0 \leq f(x) \leq 25.$$

1°) Codez le mot « CHOIX ».
2°) Décodez le mot « PHVVDJH ».

EXERCICE 2 : codage par transformation affine

Exemple : la fonction de codage est définie par $f(x) \equiv 7x + 15\,[26]$.
Coder est très simple, mais comment décoder « GTGRVRCSKTVPCDMR » ?

a) Montrez qu'il existe a' entier relatif tel que $7a' \equiv 1\,[26]$.

b) Démontrez que l'on peut trouver b' entier relatif tel que pour tous x et y de $\{0, 1, 2, 3, \ldots, 25\}$, $y \equiv f(x) \Rightarrow x = a'y + b'\,[26]$.

c) Prenez b' de plus petite valeur absolue possible, puis décodez le message.

Cas général : la fonction de codage est définie par :

$$f(x) \equiv ax + b\,[26] \quad \text{avec } 0 \leq f(x) \leq 25.$$

Le codage n'est utilisable que si deux lettres différentes sont codées différemment.

1°) a) On suppose que a est premier avec 26. Démontrez qu'alors :

$$f(x_1) = f(x_2) \Rightarrow x_1 = x_2.$$

b) On suppose que PGCD(a,26) = d avec $d > 1$. On a alors $26 = d \times k$ avec $0 < k < 26$.

Démontrez que dans ce cas, $f(k) = f(0)$.

2°) On suppose dans la suite que a est premier avec 26. Détermination de la fonction de décodage :

a) Démontrez qu'il existe a' entier relatif tel que $a \times a' \equiv 1\,[26]$.

b) Déterminez une fonction de décodage.

Question : pourquoi a-t-on abandonné ces systèmes de codages ?

Annexe 3
DS3 de spécialité (2004/2005) – 1 heure

NB : *le petit théorème de Fermat avait été démontré la semaine précédente. Je n'aurais pas demandé cette « ROC » sinon. À tort ? Par ailleurs, la deuxième question est inutile, je ne l'ai réalisé qu'après…*

Dans tout le problème, p et q sont deux nombres premiers distincts.

On rappelle que si p est un nombre premier, alors p est premier avec tout entier qu'il ne divise pas, et que $a^k - b^k = (a-b)(a^{k-1} + a^{k-2}b + a^{k-3}b^2 + \ldots + b^{k-1})$.

Partie A. Préliminaires

1°) Soit X un entier naturel non nul ; démontrez que si X est divisible par p et par q, alors X est divisible par $p \times q$.

2°) Soient b et k deux entiers naturels non nuls ; on suppose que $b - 1$ est divisible par p et que $b^k - 1$ est divisible par q. Démontrez que $b^k - 1$ est divisible par $p \times q$.

3°) Énoncez et démontrez le petit théorème de Fermat (on supposera connu le résultat : si p est un nombre premier, alors p est premier avec $(p - 1)!$).

4°) Soit a un entier strictement positif, premier avec p et avec q. Démontrez que $a^{(p-1)(q-1)} \equiv 1\,[p \times q]$.

Partie B. Indicatrice d'Euler

Les questions 1°), 2°) et 3°) sont indépendantes de la partie A.

Soit n un entier naturel non nul. On note $\varphi(n)$ le nombre d'entiers strictement positifs, inférieurs à n et premiers avec n.

Par exemple :

$\varphi(2) = 1$ car les entiers strictement positifs inférieurs à 2 sont 1 et 2 et seul 1 est premier avec 2,

$\varphi(4) = 2$ car les entiers strictement positifs et inférieurs à 4 sont 1, 2, 3 et 4 et seuls 1 et 3 sont premiers avec 4.

1°) Faîtes un tableau des valeurs de $\varphi(n)$ pour $1 \leq n \leq 10$.

2°) Démontrez que si p est premier, alors $\varphi(p) = p - 1$.

3°) On suppose que $n = p \times q$ (où p et q sont encore deux nombres premiers distincts).

Soit x un entier tel que $1 \leq x < n$. Dire que x n'est pas premier avec n signifie que x admet p ou q comme diviseur.
x ne peut admettre p et q pour diviseurs car d'après A 1 °), il admettrait $n = p \times q$ pour diviseur, ce qui est impossible puisque $x < n$.
Donc soit x admet p pour diviseur, soit x admet q pour diviseur.
Autrement dit, les nombres inférieurs strictement à n et non premiers avec n sont soit multiples de p, soit multiples de q.

a) Combien y a-t-il de multiples de p strictement positifs et strictement inférieurs à n ?
Combien y a-t-il de multiples de q strictement positifs et strictement inférieurs à n ?

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

4°) Démontrez que si $n = p \times q$ où p et q sont des nombres premiers distincts, alors, pour tout a premier avec n,

$$a^{\varphi(n)} \equiv 1\,[n].$$

En fait, ce résultat est vrai pour tout entier naturel n non nul, et est connu sous le nom de « Théorème d'Euler-Fermat ».

Annexe 4 : Complexité d'algorithme (comparaison de temps de calcul)

1°) Test de primalité : analyse du temps de calcul.

Soit x un entier dont on veut tester la primalité. Le programme décrit en **annexe 2** consiste essentiellement en une boucle :

While $d \leq \sqrt{x}$
……
……
$d + 2 \to d$
EndWhile

Admettons que le corps de la boucle nécessite un temps de calcul constant égal à k.

Le temps de calcul total sera de l'ordre de $k\dfrac{\sqrt{x}}{2}$ c'est-à-dire de $\mathrm{K}\sqrt{x}$.

Supposons x écrit en base 10 ; appelons n la longueur de x (c'est-à-dire le nombre de ses chiffres).

À l'unité près, $n \approx \log x$, donc $x \approx 10^n$.

Donc le temps de calcul total est de l'ordre de $\mathrm{K} \times 10^{n/2}$: c'est un temps de calcul exponentiel en la longueur des données.

Illustrons :

1°) Si l'on ajoute un chiffre : on passe de $K\times10^{n/2}$ à $K\times10^{(n+1)/2}=K\times10^{n/2}\times\sqrt{10}$.
Lorsque l'on ajoute un chiffre on multiplie par plus de trois le temps de calcul.

2°) Si l'on double la longueur : on passe de $K\times10^{n/2}$ à $K\times10^{n}$; on a multiplié le temps de calcul par $10^{n/2}$.
Exemple : si on passe de 10 chiffres à 20 chiffres, on multiplie le temps de calcul par 10^5.
Si l'on a besoin de 10 minutes pour un nombre de 10 chiffres, on aura besoin de 10^6 minutes pour un nombre de 20 chiffres soit 10^6 / 381 600 ≈ 2,6 ans.
Si l'on passe de 100 chiffres à 200 chiffres, on multiplie le temps de calcul par 10^{100}.

2°) Recherche du plus grand élément d'un tableau T de longueur *n* :

Des nombres sont rangés dans n cases d'un tableau. Chaque case est repérée par son rang. Le contenu de la case de rang i se trouve dans l'ordinateur à l'adresse T(*i*).

Max ← T(1) ; *i* ← 1
Répéter (*n* − 1) fois
    i ← *i* + 1
    si T(*i*) > Max alors Max ← T(*i*)
FinRépéter
Afficher Max

Si le corps de la boucle nécessite un temps de calcul égal à *k*, cet algorithme s'exécute en ≈ *kn*.
Si on multiplie la longueur du tableau par 2, on multiplie la durée de calcul par 2.

3°) Tri d'un tableau T de longueur *n* :

Un algorithme simple est en n^2. Si on multiplie la longueur du tableau par 2, on multiplie la durée de calcul par 4.

Annexe 5 : Le cryptage RSA (d'après l'exposé du 4 avril 2005).

A, l'envoyeur, doit envoyer à B, le récepteur, un message secret. Il le code en un nombre entier *m* tel que $m < n$ (par exemple 00 pour A, 01 pour B, … 25 pour Z ce qui donne $m = 010002$ pour « bac »).

C'est ce nombre que A veut envoyer à B : B sera capable de décoder *m*.

Domaine de B	Domaine public	Domaine de A
B crée deux grands nombres premiers p et q B calcule $n = p \times q$ B calcule $(p-1) \times (q-1)$. B crée d et e tels que : $d \times e \equiv 1\ [(p-1) \times (q-1)]$ d est la *clé publique* e est la *clé secrète*		A veut envoyer un message m
B envoie $(n\ ;\ d)$ à A ⟶	⟶ $n\ ;\ d$ ⟶	⟶ A reçoit $(n\ ;\ d)$ A code m en c par : $c \equiv m^d\ [n]$ et $0 \leq c < n$
B reçoit c ⟵ B décrypte le message c en calculant m' : $m' \equiv c^e\ [n]$ tel que $0 \leq m' < n$	⟵ c ⟵	⟵ A envoie c à B
« c'est magique » : $m' = m$.		

Un exemple (choisi par la classe : les trois élèves demandent des nombres p, q et m pas trop grands pour pouvoir calculer avec leurs calculatrices TI 83).
$p = 17$; $q = 13$
alors $n = 221$ et $(p-1) \times (q-1) = 192$.
Cécile choisit d premier avec 192 : $d = 5$.
Cécile calcule avec sa calculatrice e tel que $d \times e \equiv 1\ [(p-1) \times (q-1)]$ et trouve $e = 77$.

Le message à crypter est $m = 65$.
Il faut donc calculer c, le reste de 65^5 dans la division par 221.

Corentin qui doit faire le calcul avec sa calculatrice explique qu'il y a un problème : le calcul des puissances va amener de trop grands nombres pour la calculatrice et présente la solution :
L'exponentiation rapide : on décompose l'exposant en base 2 :
$5 = 4 + 1 = 2^2 + 2^0$ donc :
$65^5 = 65^4 \times 65$; on calcule avec la calculatrice :
$65^2 \equiv 26\ [221]$
$65^4 \equiv 26^2 \equiv 13\ [221]$
$65^5 \equiv 13 \times 65 \equiv 182\ [221]$
donc $c = 182$.

Décryptage : il faut calculer c^{77} :
$77 = 64 + 8 + 4 + 1 = 2^6 + 2^3 + 2^2 + 2^0$
$182^2 \equiv 195$

$182^4 \equiv 13$
$182^{16} \equiv 52$
$182^{32} \equiv 52$
$186^{64} \equiv 52$
d'où $182^{77} \equiv 65$ [221]
On a bien retrouvé m…

Preuve du cas général
Elle utilise le théorème d'Euler-Fermat démontré en DS (**annexe 4**) : si p et q sont deux nombres premiers distincts et si $n = p \times q$ alors, pour tout a premier avec n, $a^{(p-1)(q-1)} \equiv 1\,[n]$.

Le récepteur calcule $c^e = \left(m^d\right)^e = m^{de}$.

Or $de \equiv 1$ $[(p-1)(q-1)]$ donc il existe un entier k tel que $de - 1 = k\,(p-1)(q-1)$ soit $de = k\,(p-1)(q-1) + 1$.

D'où $m^{de} = m^{k(p-1)(q-1)+1} = m \times m^{k(p-1)(q-1)}$.

Or $m^{k(p-1)(q-1)} = \left(m^{(p-1)(q-1)}\right)^k$ et, si m est premier avec n, $m^{(p-1)(q-1)} \equiv 1\,[n]$ donc $m^{k(p-1)(q-1)} \equiv 1\,[n]$ donc $m^{de} \equiv m\,[n]$.

Le choix de p et q.

p et q doivent être très grands.

Dans la pratique, on se contente de choisir des nombres p et q « premiers avec une grande probabilité », en utilisant un test de primalité qui repose sur le petit théorème de Fermat :

« si p est premier et ne divise pas a, alors $a^{p-1} \equiv 1\,[p]$ ».

Donc si a^{p-1} n'est pas congru à 1 $[p]$ alors p n'est pas premier, mais si $a^{p-1} \equiv 1\,[p]$ alors p a de bonnes chances d'être premier (on sait même évaluer la probabilité pour qu'il le soit).

Comment casser un cryptage RSA ?

Il s'agit de résoudre $c \equiv m^d\,[n]$ où c, d et n sont donnés et m l'inconnue.

Évidemment, si on arrive à calculer e, c'est fini.
e se calcule facilement si l'on connaît p et q, c'est-à-dire si l'on arrive à décomposer n en produit de facteurs premiers, ce qui est hors de portée (actuellement) si p et q sont très grands, car cela nécessite un temps de calcul trop long : on n'a pas trouvé d'algorithme rapide pour le faire, mais on ne sait pas prouver qu'il n'y en a pas.

Il semble que les tentatives pour trouver e sans casser n soient aussi en général hors de portée si e est assez grand, mais on ne sait pas le prouver.

La solidité du cryptage RSA est validée par l'expérience d'une part, et par des conjectures mathématiques dans une branche des mathématiques qui étudie la *complexité d'algorithmes* d'autre part.

Bibliographie

– Du chiffrement de César à la mathématique de la carte bancaire. Repères 2002 nº 46, p. 59-90

– Machine à résoudre les congruences. Mnémosyne (IREM de Paris VII) nº 17, 2002.

– Histoire des codes secrets, de l'Égypte des Pharaons à l'ordinateur quantique. Simon Singh (Le livre de Poche 2001 et J.-C. Lattès 1999).

– Quelques activités arithmétiques liées aux codes correcteurs et à la cryptographie. Bulletin APMEP 2001 nº 432, p. 81-94.

– Difficulté d'évaluer la difficulté en arithmétique. Bulletin APMEP 2001 nº 434, p. 328-334.

– Les Mathématiques ne se sont pas faites en un jour. Brochure IREM de Nantes, 2001.

– Initiation à la cryptographie. Brochure IREM de Bordeaux, 2000.

– Cryptographie et arithmétique. L'Ouvert (IREM/APMEP de Strasbourg) nºs 100 et 101, 2000.

– Arithmétique et cryptographie. Repères 1999 nº 37, p. 41-62.

– Apprenons l'arithmétique élémentaire pour comprendre la cryptographie moderne. Brochure IREM de Limoges, 1998.

– Cryptographie classique et cryptographie publique à clé révélée. Bulletin APMEP 1996 nº 406, p. 568-581.

– Cryptographie publique. Bulletin APMEP 1982 nº 336, p. 783-792.

– Mathématiques en liaison avec des problèmes concrets (Niveau TS et Première année d'université). Vol. 1. Ouvrage collectif. IREM de Marseille, fin 2005.