103 302 304 305	Congruences dans $\mathbb{Z}$, anneau $\mathbb{Z}/n\mathbb{Z}$, applications. Exercices: Congruences et divisibilité ds $\mathbb{Z}$. Exercices: Théorème de Bézout Exercices faisant intervenir les nombres premiers	Système RSA	Terracher T.S.

(Wikipedia) Ronald Rivest, Adi Shamir et Leonard Adleman, dans "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", paru en 1978.

Soient p et q deux nombres premiers distincts ($p\neq 2$ *et* $q\neq 2$). On pose $n=pq$ et on désigne par e un entier t.q.: $1<e<(p-1)(q-1)$ et e premier avec $(p-1)(q-1)$.

1°) Montrer qu'il existe d unique tel que:

$1\le d<(p-1)(q-1)$ et $ed\equiv 1\left[(p-1)(q-1)\right]$.

2°) Prouver que, pour tout $m\in\mathbb{N}$, $m^{ed}\equiv m\left[n\right]$.

I. Outils:

- Congruences,
- Théorème de Bézout,
- Petit théorème de Fermat.

II. Développement:

A. Lemme.

Lemme: Soient a et b des entiers premiers entre eux (a>1 et b>1), alors ∃! $u_0\in\mathbb{N}$ tq : $\begin{cases}1\le u_0\le b-1\\ au_0\equiv 1\left[b\right]\end{cases}$.

Existence: D'après le **théorème de Bézout**, ∃u, v tq: $au+bv=1$. (1)

On effectue la division Euclidienne de u par b, il vient: $u=qb+u_0$, avec $0<u_0<b$, i.e. $1\le u_0\le b-1$.

Dans (1), il vient: $a(qb+u_0)+bv=1\Rightarrow aqb+au_0+bv=1\Rightarrow au_0=1+b(-aq-v)$, d'où $au_0\equiv 1\left[b\right]$.

Unicité: a, b premiers entre eux, a>1, b>1. Supposons qu'il existe:

$u_0\in\mathbb{N}$ tq : $1\le u_0\le b-1$; $au_0\equiv 1\left[b\right]$, et $u_1\in\mathbb{N}$ tq : $1\le u_1\le b-1$; $au_1\equiv 1\left[b\right]$.

Alors, en soustrayant membre à membre: $a(u_0-u_1)\equiv 0\left[b\right]$, i.e. $b\mid a(u_0-u_1)$. or $b\nmid a$, donc $b\mid(u_0-u_1)$.

Soit donc $k\in\mathbb{Z}$ tq : $(u_0-u_1)=kb$.

Les conditions $1\le u_0\le b-1$ et $1\le u_1\le b-1$ impliquent $2-b\le kb\le b-2$, donc a fortiori $-b<kb<b\Rightarrow -1<k<1\Rightarrow k=0$.

Donc $u_1=u_0$, ce qui prouve l'unicité.

B. Question 1.

Soient p et q deux nombres premiers distincts ($p\neq 2$ *et* $q\neq 2$). On pose $n=pq$ et on désigne par e un entier t.q.: $1<e<(p-1)(q-1)$ et e premier avec $(p-1)(q-1)$.

1°) Montrer qu'il existe d unique tel que:

$1\le d<(p-1)(q-1)$ et $ed\equiv 1\left[(p-1)(q-1)\right]$.

On applique le Lemme avec d→u_0, e→a, (p-1)(q-1)→b.

Les hypothèses sont bien vérifiées: $e\wedge(p-1)(q-1)$; e>1; $\begin{cases}p>2\\ q>2\end{cases}\Rightarrow(p-1)(q-1)>1$.

Donc : ∃!d∈ℕ tq : $\begin{cases}1\le d\le(p-1)(q-1)-1 \text{ i.e. } 1\le d<(p-1)(q-1)\\ ed\equiv 1\left[(p-1)(q-1)\right]\end{cases}$.

D'où l'existence et l'unicité de d.

Soit un tel d.

C. Question 2.

2°) Prouver que, pour tout $m \in \mathbb{N}$, $m^{ed} \equiv m[n]$.

Il s'agit de mq. $n \mid m^{ed} - m$. Mais on a n=pq et p et q premiers distincts, donc il suffit de mq. $p \mid m^{ed} - m$ et $q \mid m^{ed} - m$.

D'après la question 1°), ∃k∈ℤ* tq. $ed = 1 + k(p-1)(q-1)$; Soit un tel k; d'après l'égalité précédente, on a en fait k∈ℕ*.

- ➔ Si p|m, alors on a directement $p \mid m^{ed} - m$.
- ➔ Si $p \nmid m$, alors on utilise le **petit théorème de Fermat**:

∀n∈ℕ*, ∀p premier, $p \mid (n^p - n)$.

et si $p \nmid n$, alors $p \mid (n^{p-1} - 1)$, i.e. $n^{p-1} \equiv 1[p]$.

Ici, $m^{p-1} \equiv 1[p]$.

D'où, comme $ed = 1 + k(p-1)(q-1)$, avec k∈ℕ*, il vient: $m^{ed} = m^{1+k(p-1)(q-1)} = m \times \underbrace{\left(m^{(p-1)}\right)^{k(q-1)}}_{\equiv 1[p]}$, donc $m^{ed} \equiv m[p]$.

En échangeant les rôles de p et q, on montre de la même manière que $m^{ed} \equiv m[q]$.

Comme p et q sont premiers et distincts, on a finalement: $m^{ed} \equiv m[n]$, ce qui achève la démonstration.

D. Fonctionnement – Application.

A (Alice) veut transmettre des informations à B (Bob)

Explication	Exemple "naïf"
B **choisit**: • **p** et **q** premiers distincts • **e** vérifiant $1 < e < (p-1)(q-1)$ et $e \wedge (p-1)(q-1)$ B **calcule** l'unique entier **d** tel que: $1 \le d < (p-1)(q-1)$ et $ed \equiv 1[(p-1)(q-1)]$ B **diffuse** largement les entiers **n** = pq et **e**. Les entiers p, q et d restent secrets.	p = 41 et q = 53; alors n = pq = 2 173; (p-1)(q-1) = 2 080. e = 1 427 (premier avec 2 080) ← Algorithme d'Euclide* *L'Algo. d'Eucl. donne au passage d = 1 083 Quand on calcule les coefs de Bézout: e.**u**+(p-1)(q-1).**v**=1, d est le reste de la division Euclidienne de **u** par (p-1)(q-1).
Pour envoyer un message à B, A **convertit** ce message en une suite de nombres **m < n**. A **chiffre** chaque nombre m en calculant le nombre **c** ∈ $[\![1; n]\!]$ tq. $m^e \equiv c[n]$. A **transmet** chaque nombre **c** à B. B reçoit le message, et le décrypte en utilisant le fait que $c^d \equiv m^{ed} \equiv m[n]$.	Message M = 356 453 213. Découpage en tranches de 3 chiffres <u>en partant de la droite</u>: $m_1 = 213$; $m_2 = 453$; $m_3 = 356$. (tous <2 173). Avec e = 1 427, il vient: $\begin{cases} m_1^{\,e} \equiv 1273[n] \\ m_2^{\,e} \equiv 907[n] \\ m_3^{\,e} \equiv 1297[n] \end{cases}$ d'où: $\begin{cases} c_1 = 1273 \\ c_2 = 0907* \\ c_3 = 1297 \end{cases}$ On normalise l'écriture des c_i en nombres de 4 chiffres (ce sera un "zéro de position"). On liste ces nombres <u>en partant de la gauche</u>. Le message transmis sera donc: 127309071297. Bob découpe ce nombre en tranches de 4 chiffres, retrouvant c_1, c_2 et c_3. Puis en calculant c_1^d, c_2^d et c_3^d, il retrouve les nombres m_1, m_2 et m_3.

III. Commentaires:

Comment être sûr que le message vient d'Alice? *(Terracher)*

Alice dispose également d'une fonction trappe f_A.

f_A est publique, mais f_A^{-1} n'est connue que d'Alice.

Alice envoie à Bob: (son message; une double signature). La double signature est de la forme: $\left(S; f_A^{-1}(S)\right)$.

Alors Bob peut s'assurer que le message vient bien d'Alice en calculant $f_A\left(f_A^{-1}(S)\right)$ et en vérifiant qu'il trouve bien S.

(Terracher)

Cryptologie: *Etude des techniques mathématiques liées à la sécurité de l'information.*
Cryptographie: *Etude des moyens de parer aux attaques humaines sur les communications sécurisées.*
Cryptanalyse: *tentative de déchiffrement d'un message chiffré.*

Un système à clef publique:
Le couple (n,e) connu de tout un chacun permet donc à tout public de transmettre un message à Bob.
La sécurité du système:
Si l'on sait factoriser n sous la forme n=pq, connaissant e (qui est public), on trouve d sans problème et la lecture du courrier électronique de Bob se fait comme dans un livre ouvert.
En 1999, un nombre de 155 chiffres (qui servait de clef à un système RSA) a été décomposé en un produit de deux facteurs premiers de 78 chiffres. Le temps cumulé de calcul a été d'environ 8 000 millions d'instructions par seaconde pdt 1 an. Depuis, la société RSA Data Security recommande d'utiliser des nbres de plus grande taille (309 voire 617 chiffres – calcul de taille en bits).
De même, on montre que, connaissant n, e et d, on trouve rapidement p et q.
La sécurité du système tient pour l'essentiel dans:

- *La construction de "grands" nombres premiers*
- *La difficulté de décomposer un grand nombre et produit de deux nombres premiers.*

Mais personne ne sait si une attaque du RSA par un biais autre que le factorisation est impossible...
Les fonctions à sens unique
La fonction one-way de Bob f_B *est la fonction de* $X=\{1,2,...,n\}$ *dans* X *définie par* $f(m)=m^e\,[n]$ *est connue de tout le monde. En revanche, la fonction réciproque* $f_B^{-1}: m \mapsto m^d\,[n]$ *n'est connue que de Bob, et est pratiquement impossible à calculer, sauf à disposer d'une information supplémentaire (trapdoor information), d'où le nom donné à une telle fonction: "fonction trappe".*

(Wikipedia) Dans la pratique, deux problèmes majeurs apparaissent :

➔ *choisir un nombre premier de grande taille*

Une méthode simple pour choisir un nombre premier de grande taille est de créer une suite aléatoire de bits, puis de le tester avec le test de primalité. Un problème apparaît pour cette deuxième opération : la méthode naïve serait d'utiliser le crible d'Ératosthène, mais elle est trop lente. En pratique, on utilise un test de primalité probabiliste (test de primalité de Fermat par exemple). Ce test n'assure pas que le nombre est premier, mais il y a une forte probabilité pour qu'il le soit. On peut également utiliser un test de primalité déterministe en temps polynomial qui assure que le nombre est premier (comme le test de primalité AKS [(1)]). Bien que moins rapide, il assure la primalité du nombre.

➔ *calculer* $M=c^d \mod n$

Le calcul de $M=c^d \mod n$ *peut être assez long. Calculer d'abord* c^d, *puis calculer le modulo avec* n *est coûteux en temps et en calculs. Dans la pratique, on utilise l'exponentiation modulaire.*
On peut conserver une forme différente de la clé privée pour permettre un déchiffrement plus rapide à l'aide du théorème des restes chinois.

IV. Notes (et notes de notes).

[(1)] ***Test de primalité AKS*** *(Wikipedia). Le test de primalité AKS (aussi connu comme le test de primalité Agrawal-Kayal-Saxena et le test cyclotomique AKS) est un algorithme déterministe de preuve de primalité découvert et publié le 6 août 2002 par trois scientifiques indiens nommés Manindra Agrawal, Neeraj Kayal et Nitin Saxena dans un article scientifique intitulé « PRIMES is in P ».*
L'algorithme détermine si un nombre est premier ou composé (au sens de la factorisation). La complexité temporelle originale est en O((log n)[12]).
L'algorithme repose sur l'identité AKS, généralisation du petit théorème de Fermat : si n *est un nombre entier et* a *un nombre premier avec n alors:* n premier $\Leftrightarrow (X+a)^n \equiv X^n + a\,[n]$.

L'algorithme AKS n'est pas le premier test de primalité général s'exécutant en un temps polynomial. Il possède cependant une différence clé par rapport à tous les algorithmes généraux de preuve de primalité précédents : il ne repose pas sur une hypothèse non démontrée (telle que l'hypothèse de Riemann[(2)]) pour être vrai et pour avoir un temps polynomial démontrable pour toutes ses entrées. De plus c'est un algorithme déterministe : il permet de déterminer de façon certaine si un nombre est premier (tout comme le crible d'Ératosthène) par opposition aux tests probabilistes, qui permettent seulement de déterminer si un nombre est un nombre premier probable et qui comportent de fait une certaine probabilité d'erreur sur la réponse donnée lorsque celle-ci est affirmative.
Quelques mois après la découverte, de nombreuses variantes sont apparues : Lenstra 2002, Pomerance 2002, Berrizbeitia 2003, Cheng 2003, Bernstein 2003a/b, Lenstra et Pomerance 2003. Elles ont amélioré la vitesse d'exécution de l'algorithme AKS à différentes ampleurs. Ces multiples variantes de l'algorithme sont référencées sous la notion de « classe AKS », introduite par Crandall et Papadopoulos en 2003.

(2) ***Hypothèse de Riemann*** *(Wikipedia). En mathématiques, l'hypothèse de Riemann est une conjecture formulée en 1859 par le mathématicien allemand Bernhard Riemann. Elle dit que les zéros non triviaux de la fonction zêta de Riemann[(3)] ont tous pour partie réelle 1/2. Sa démonstration améliorerait la connaissance de la répartition des nombres premiers.*
Cette conjecture constitue l'un des problèmes non résolus les plus importants des mathématiques du début du XXIe siècle : elle est l'un des fameux problèmes de Hilbert proposés en 1900, et l'un des sept problèmes du prix du millénaire et des dix-huit problèmes de Smale. Comme pour les six autres problèmes du millénaire, l'énoncé exact de la conjecture à démontrer est accompagné d'une description détaillée[1], fournissant de nombreuses informations sur l'historique du problème, son importance, et l'état des travaux à son sujet[2]; beaucoup des remarques informelles de cette page en proviennent.

(3) ***Fonction Zêta de Riemann*** *(Wikipedia). La fonction* ζ *de Riemann est une fonction analytique complexe méromorphe* et définie, pour R(s)>1 (partie réelle de s), par la série de Dirichlet :* $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$.

La série ne converge pas en $s = 1$ *: on a* $\sum_{k=1}^{k=m} \frac{1}{k} \geq \int_{1}^{m+1} \frac{du}{u} = \ln(m+1)$, *qui tend vers l'infini avec m.*

**fonction holomorphe dans tout le plan complexe, sauf éventuellement sur un ensemble de points isolés dont chacun est un pôle pour la fonction*