Un Mooc d'arithmétique et de cryptographie

Arnaud Bodin

Bilan du Mooc de six semaines proposé au printemps 2015 sur la plateforme FUN (France Université Numérique) avec pour thèmes l'arithmétique, la cryptographie et un peu de programmation. Il s'agissait de la réédition d'un cours proposé un an et demi auparavant.

C comme Cours

Le cours est à la fois un cours d'arithmétique assez standard de niveau Licence première année avec en parallèle une étude de méthodes cryptographiques. Le programme mathématique est le suivant : division euclidienne, pgcd, nombres premiers, nombres premiers entre eux, calcul avec les modulos, petit théorème de Fermat. Deux points un peu moins standards : l'exponentiation rapide et une version améliorée du petit théorème de Fermat :

Théorème : soit p et q deux nombres premiers distincts et soit $n = pq$. Si un entier a n'est divisible ni par p ni par q, alors : $a^{(p-1)(q-1)} \equiv 1 \pmod{n}$.

La finalité du cours est de comprendre le chiffrement RSA. Mais auparavant on retrace, par ordre chronologique et de difficulté, les méthodes cryptographiques classiques : le chiffrement par décalage simple (César) ; par décalage de blocs (Vigenère) ; le chiffrement parfait ; on étudie des versions très simplifiées de la machine Enigma et du DES. Tous ces chiffrements sont dits à « clé secrète », ce qui pose le problème majeur de transmettre au préalable cette clé secrète. Une révolution est l'apparition à la fin des années 1970 de systèmes de chiffrement à clé publique, dont le plus connu est RSA. La philosophie est bien différente puisqu'il n'y a pas d'échange de clé secrète, ce qui en fait un système adapté aux communications sur Internet. Le principe de chiffrement/déchiffrement est basé sur le petit théorème de Fermat amélioré énoncé ci-dessus et sur le fait que connaissant deux nombres premiers p et q (très grands) il est facile de calculer $n = pq$, mais connaissant n (très très grand) il est presque impossible de retrouver les deux facteurs premiers p et q.

En plus de la théorie mathématique et cryptographique, nous souhaitions que les étudiants mettent en œuvre les différents codages/décodages rencontrés. C'est pourquoi nous présentions une initiation au langage Python afin de « casser » quelques codes secrets.

O comme Online

Le cours se déroule entièrement en ligne :

(*) email : Arnaud.Bodin@math.univ-lille1.fr

- les vidéos de trois parties de cours (arithmétique, cryptographie, Python) et aussi des vidéos d'exercices corrigés d'arithmétique,
- le cours « papier » de ces mêmes parties.

Les apprenants commencent par regarder les vidéos, ils reviennent vers le support papier pour une seconde lecture.

La différence entre un Mooc et une juxtaposition de ressources en ligne réside dans la temporalité : le cours a un début et une fin. Ici, le cours durait six semaines (après une semaine 0 de mise en route). Chaque semaine il y a donc à étudier une partie du cours et pour vérifier la compréhension, deux types de tests sont proposés chaque semaine :

- 10 questions type QCM
- 3 énigmes de difficultés croissantes qui sont des codes secrets à « craquer » (c'est-à-dire déchiffrer le message, bien qu'il ne nous soit pas adressé).

Ces tests donnent une note et les étudiants ayant obtenu plus de 70 % de réussite obtiennent une attestation de réussite. Il n'y a bien sûr aucune obligation de réussir, ni même de passer les tests mais c'est une énorme motivation pour les étudiants.

Enfin, comme tout le monde travaille en même temps sur la même partie du cours, cela rend le forum de discussions plus vivant : un étudiant propose un début de solution, un autre corrige et complète… C'est le forum qui crée l'appartenance à une classe de cours.

O comme Ouvert

La seule condition pour suivre le cours était de s'inscrire sur la plateforme de Mooc FUN mise en place par le ministère de l'éducation nationale. Le cours et les vidéos sont en accès libre sur Internet et le restent après la fin du cours.

M comme Massif

« Massif » est exagéré, disons qu'il n'y a pas de limite au nombre d'inscriptions. Le public est très varié. Comme lors de la première édition il n'est pas du tout composé d'étudiants de première année, mais plutôt d'un public plus âgé (jusqu'à 78 ans !) ayant déjà un diplôme du supérieur et dont une bonne part ont déjà un bon niveau en mathématiques ou en informatique.

Voici les chiffres pour la seconde édition de ce cours (et entre parenthèses pour la première édition) :

- Nombres d'inscrits : 2700 (1400)
- Nombre de participants à la première semaine : 610 (300)
- Nombre de certificats délivrés : 280 (130)

Bilan

Les avis issus du questionnaire proposé en fin de cursus sont extrêmement positifs. Certains ont apprécié les maths, d'autres plus la cryptographie, ceux qui ont découvert Python ont été ravis. Cependant certains ont été découragés par l'apprentissage d'un langage de programmation afin de résoudre les énigmes. Au

début la machine permet de s'épargner des recherches un peu longuettes, mais devient indispensable pour la factorisation des grands entiers. Et il n'est pas facile de se faire aider pour la mise en route et l'écriture de sa première boucle sur le forum de discussions. Autre bémol, pour tout réussir, les étudiants passaient beaucoup plus de temps que les 3 à 4 heures prévues par semaine. Les énigmes sont très appréciées pour leur côté ludique et très frustrantes pour ceux qui ne les trouvent pas, d'autant plus que les réponses ne sont données qu'à la toute fin du cours. Le public étant varié, certains ont trouvé les énigmes trop dures, d'autres trop faciles !

Côté enseignant le bilan est aussi positif : nous sommes ravis de transmettre notre passion. Les apprenants sont ici très motivés, très agréables et viennent pour le plaisir. Ce Mooc a aussi permis de créer une nouvelle option hybride math/info proposée aux étudiants de l'université Lille 1, mais cette fois qui se déroule dans des salles de classe tout à fait classiques !

Ce cours a été proposé par Pierre Allegraud, Arnaud Bodin, François Recher et Éric Wegrzynowski de l'université de Lille 1.

Un article de présentation de la première édition de ce mooc est paru dans la revue en ligne Sesamath : http://revue.sesamath.net/spip.php?article577

Les videos, les cours écrits, et leurs fichiers-source sont disponibles respectivement aux adresses :
https://www.youtube.com/watch?v=ABqrKsaN8hg&list=SP024XGD7WCIEii2U_HKeprCTJA4xb-uJ6
http://exo7.emath.fr/cours/ch_crypto.pdf
http://exo7.emath.fr/cours/sauv-cours-exo7.tar.gz
https://www.france-universite-numerique-mooc.fr/

Annexe 1 : exemples d'énigmes

Une énigme facile

Le texte suivant a été chiffré par un chiffre mono-alphabétique (c'est-à-dire que chaque lettre de l'alphabet est remplacée par une autre lettre).

SCPGHZ SCE K GXWC G K TCXQC VX WKGZFTHU KG FGPMGDZC NC MGQUHQGH RVHE UX NC EGHE OXC UX P GUUCZSE

Heureusement vous avez réussi à intercepter une partie du principe de chiffrement : 'A' est chiffré en 'G' ; 'B' en 'W' ; 'C' en 'F' ; 'D' en 'S'.

Quel est le nom de l'auteur de ce texte ?

Une énigme plus difficile

Dans cette énigme, il s'agit de trouver un nombre. Vous vous aiderez d'un ordinateur.

On dit qu'un entier $p \geq 2$ vérifie le n-test de Fermat si $n^{p-1} \equiv 1 \bmod p$, où n est un nombre entier fixé. Le petit théorème de Fermat affirme que si p est un nombre premier alors p vérifie le n-test de Fermat, quelque soit n non divisible par p.

Trouver le plus petit entier $p \geq 2$ qui vérifie à la fois le 3-test de Fermat et le 5-test de Fermat mais tel que p ne soit pas un nombre premier.

Annexe 2 : légende des images

Image 1 : accès à une vidéo de cours depuis la plateforme FUN

6.3 Déchiffrement du message

Alice reçoit le message z chiffré par Bruno, elle le décrypte à l'aide de sa clé privée d, par l'opération :

$$m \equiv z^d \pmod{n}$$

qui utilise également l'algorithme d'exponentiation rapide.
Nous allons prouver dans le lemme 4, que par cette opération Alice retrouve bien le message original m de Bruno.

Exemple 1. $e = 40$, $d = 13$, $n = 85$ donc

$$z^d \equiv (40)^{13} \pmod{85}.$$

Calculons à la main $40^{13} \pmod{85}$ on note que $13 = 8 + 4 + 1$, donc $40^{13} = 40^8 \times 40^4 \times 40$.

$$\begin{aligned} 40^2 &\equiv 1600 \equiv 70 \pmod{85} \\ 40^4 &\equiv (40^2)^2 \equiv 70^2 \equiv 4900 \equiv 55 \pmod{85} \\ 40^8 &\equiv (40^4)^2 \equiv 55^2 \equiv 3025 \equiv 50 \pmod{85} \end{aligned}$$

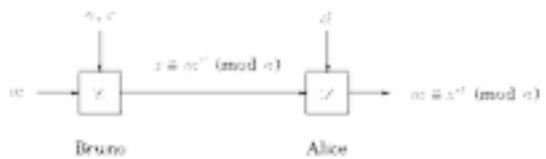
Donc

$$z^d \equiv 40^{13} \equiv 40^8 \times 40^4 \times 40 \equiv 50 \times 55 \times 40 \equiv 10 \pmod{85}$$

qui est bien le message m de Bruno.

Exemple 2. $e = 10\,378$, $d = 8743$, $n = 10\,403$. On calcule par ordinateur $z^d \equiv (10\,378)^{8743} \pmod{10\,403}$ qui vaut exactement le message original de Bruno $m = 1234$.

6.4 Schéma



Clés d'Alice :
- publique : n, e
- privée : d

6.5 Lemme de déchiffrement

Le principe de déchiffrement repose sur le petit théorème de Fermat amélioré.

Lemme 4. Soit d l'inverse de e modulo $\varphi(n)$.

$$\text{Si } c \equiv m^e \pmod{n} \text{ alors } m \equiv x^d \pmod{n}.$$

Ce lemme prouve bien que le message original m de Bruno, chiffré par clé publique d'Alice (e, n) en le message z, peut-être retrouvé par Alice à l'aide de sa clé secrète d.

Démonstration. – Que d soit l'inverse de e modulo $\varphi(n)$ signifie $d \cdot e \equiv 1 \pmod{\varphi(n)}$. Autrement dit, il existe $k \in \mathbb{Z}$ tel que $d \cdot e = 1 + k\varphi(n)$.

Image 2 : le cours « papier »

QCM 6 (10 points)

Ce quiz comporte 10 questions, chacune notée sur 1 point. Lorsqu'il y a des cases à cocher, il faut choisir toutes les bonnes réponses. Les questions concernent le programme mathématique de cette semaine et la cryptographie. Munissez vous d'un papier et d'un crayon ! Il n'y a pas de limite de temps et vous pouvez passer ce qcm deux fois !

1. Quelles sont les assertions vraies ?

✔
- ☑ Si a est impair et divise b+c et b-c alors a divise b et a divise c.
- ☐ Si a est multiple de b, et si c est multiple de d, alors a+c est multiple de b+d.
- ☐ Si 21 divise bc alors 21 divise b ou 21 divise c.
- ☑ Si 4 ne divise pas bc alors b ou c est impair.

2. Soit n un entier supérieur ou égal à 1. Quelles sont les assertions vraies ?

✔
- ☐ pgcd(3n,3n+3) vaut 3n.
- ☑ n(n+1) est pair.
- ☑ pgcd(n,n+1) vaut 1.
- ☐ pgcd(n,2n) est pair.

3. Quelles sont les assertions vraies ?

✔
- ☑ Il existe une infinité de nombres premiers impairs.
- ☑ Le pgcd de deux nombres premiers distincts est 1.
- ☐ Tout diviseur d'une entier est un nombre premier.
- ☑ Tout nombre strictement supérieur à 1 et qui n'est pas premier a au moins trois diviseurs distincts.

Image 3 : un qcm