Travaux Pratiques
Séance n° 5

I. Calcul d'indices et factorisation

On reprend les notations du dernier TP sur les méthodes de calcul d'indices pour le problème du logarithme discret. On pourra reprendre aussi certaines des fonctions implémentées à cette occasion.

En particulier, on rappelle qu'un entier n est *B-friable* si tous ses facteurs premiers sont plus petits que B.

1. Soient a, b, n trois entiers tels que $a^2 = b^2 \bmod n$ et $a \neq \pm b \bmod n$. Montrer que n n'est pas premier et expliquer comment retrouver un facteur non trivial de n.
   Application : factoriser 20003 sachant que $245^2 = 16 \bmod 20003$.
2. Factoriser 30049 sachant que $177^2 = 1280 = 2^8.5 \bmod 30049$ et que $361^2 = 10125 = 3^4.5^3 \bmod 30049$. Indication : combiner les deux équations pour trouver une congruence de carrés.

Trouver des congruences de carrés est encore aujourd'hui la méthode de factorisation la plus rapide. La version élémentaire de la méthode de calcul d'indices pour factoriser un entier n est la suivante :

— On se fixe une borne de friabilité B et on considère la base de factorisation $\mathcal{F} = \{q_1, \dots, q_{\pi(B)}\} = \{q \in \mathbb{N} \mid q \text{ est premier et } q \leq B\}$.

— On teste pour plusieurs $x \in \mathbb{Z}/n\mathbb{Z}$ si le représentant de x^2 dans $[\![0, n-1]\!]$ est B-friable. Cela permet de produire des relations de la forme

$$x_i^2 = \prod_{j=1}^{\pi(B)} q_j^{a_{ij}} \bmod n.$$

— Une fois que $N = \pi(B) + 1$ relations ont été trouvées, on les combine pour obtenir une congruence de carrés.

Pour cette dernière étape, on considère la matrice $A = \begin{pmatrix} \overline{a_{11}} & \dots & \overline{a_{1\pi(B)}} \\ \vdots & & \vdots \\ \overline{a_{N1}} & \dots & \overline{a_{N\pi(B)}} \end{pmatrix} \in \mathcal{M}_{N,\pi(B)}(\mathbb{Z}/2\mathbb{Z})$
(chaque coefficient est réduit modulo 2).
On note $V = (\epsilon_1 \ \epsilon_2 \dots \epsilon_N) \in \mathcal{M}_{1,N}(\mathbb{Z}/2\mathbb{Z})$ un élément non nul du noyau à gauche de A, c'est-à-dire que $V.A = 0$.

3. En identifiant les ϵ_i avec leurs représentants dans $\{0, 1\}$, montrer que

$$\prod_{\substack{1 \leq i \leq N \\ \epsilon_i = 1}} \prod_{j=1}^{\pi(B)} q_j^{a_{ij}} = \prod_{j=1}^{\pi(B)} q^{\sum_{i=1}^{N} \epsilon_i a_{ij}}$$

   est un carré (dans $\mathbb{N}$), puis en déduire une congruence de carrés modulo n.

Exemple jouet : avec $B = 5$ (et donc $\mathcal{F} = \{2, 3, 5\}$) et $n = 3053$, en démarrant avec $\lceil \sqrt{3053} \rceil = 56$, les relations obtenues sont

$$\begin{array}{rcl} 79^2 & = & 3^3.5 \bmod 3053 \\ 97^2 & = & 2.5^3 \bmod 3053 \\ 125^2 & = & 2^3.3^2.5 \bmod 3053 \\ 127^2 & = & 2^5.3 \bmod 3053 \end{array}$$

correspondant à la matrice des relations $M = \begin{pmatrix} 0 & 3 & 1 \\ 1 & 0 & 3 \\ 3 & 2 & 1 \\ 5 & 3 & 0 \end{pmatrix}$.

On obtient alors modulo 2 la matrice $A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$, et le vecteur $V = (0\ 1\ 1\ 0)$ vérifie clairement $V.A = (0\ 0\ 0)$ (noter qu'il est possible que SageMath renvoie un autre vecteur dans le noyau à gauche de A). En revenant aux relations, on obtient alors la congruence

$$97^2.125^2 = 2^4.3^2.5^4 \bmod 3053.$$

Maintenant $97.125 = 2966 \bmod 3053$ et $2^2.3.5^2 = 300$, donc finalement $\gcd(2966 - 300, 3053) = 43$ est un facteur non trivial de 3053.

4. Écrire un programme qui factorise un entier donné n en utilisant des variantes des programmes écrits dans le dernier TP.
   Afin d'accélerer la recherche de relations, on parcourra les entiers x en démarrant à $x = \lceil \sqrt{N} \rceil$.
   La fonction `A=M.change_ring(Zmod(2))` peut être utilisée pour réduire les coefficients de la matrice M modulo 2 ; il est également recommandé d'utiliser la commande `A.kernel()[1]` ou `A.left_kernel()[1]`.
5. (a) Trouver la factorisation de 20099 en prenant $B = 15$.
   (b) Trouver la factorisation de $10^{13} + 73$ en prenant $B = 1000$.

II. Algorithme rho de Pollard pour la factorisation

L'algorithme rho de Pollard a été conçu (en 1975) pour factoriser un entier donné n. L'idée de base est la même que pour calculer des logarithmes discrets : on itère une fonction $F : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$ qui part d'un élément aléatoire x_0, i.e. qui calcule la suite $(x_k)_{k\in\mathbb{N}}$ telle que $x_{k+1} = F(x_k)$.

Les contraintes sur F sont néanmoins légèrement différentes :
- F doit être facile à calculer ;
- pour tout diviseur d de n, pour tous $x, y \in \mathbb{Z}/n\mathbb{Z}$, si $x \equiv y\ [d]$ alors $F(x) \equiv F(y)\ [d]$;
- F se comporte comme une fonction aléatoire.

Ainsi, F induit une application $\bar{F} : \mathbb{Z}/d\mathbb{Z} \to \mathbb{Z}/d\mathbb{Z}$ pour tout diviseur d de n, et la suite $(\bar{x}_k)_{k\in\mathbb{N}} = (x_k \bmod d)_{k\in\mathbb{N}}$ satisfait $\bar{x}_{k+1} = \bar{F}(\bar{x}_k)$.

1. Soit p le plus petit facteur premier de n. Montrer que la première collision modulo p se produit après environ $O(\sqrt{p})$ itérations, autrement dit que le plus petit entier $k \in \mathbb{N}^*$ tel qu'il existe $i < k$ avec $x_i \equiv x_k\ [p]$ satisfait $k = O(\sqrt{p})$.
   Est-il probable que $x_i = x_k \pmod n$ lors de cette première collision ?
2. Lorsque $x_i \neq x_k$, expliquer comment tester si $x_i \equiv x_k$ modulo un facteur premier (non connu !) de n, et pourquoi cela permet de retrouver un facteur non premier de n.

Lorsque la factorisation de n n'est pas connue, la seule façon naturelle de satisfaire la deuxième contrainte sur F est d'utiliser des applications polynomiales. En pratique, les fonctions de la forme $x \mapsto x^2 + c$ sont bien appropriées ; bien qu'elles ne soient pas aléatoires, elles se comportent de façon suffisamment chaotiques en pratique pour que les estimées précédentes restent vraies.

3. Implémenter et tester cet algorithme de factorisation. Pour détecter les collisions, on considèrera uniquement les paires de la forme (x_i, x_{2i}) — c'est l'algorithme de détection de cycle de Floyd. Donner une estimée de sa complexité en temps et en mémoire.