

# RACINES CARRÉES DANS LES CORPS FINIS

## 1. Racines carrées : le cas facile

Soit  $q$  une puissance d'un nombre premier  $p$ . Si  $q$  est impair la moitié des éléments de  $\mathbb{F}_q^*$  sont des carrés, tandis que si  $q$  est pair tous éléments de  $\mathbb{F}_q^*$  sont des carrés.

Ce texte est consacré à la recherche d'algorithmes permettant de trouver des racines carrées dans  $\mathbb{F}_q^*$ .

Commençons par remarquer que dans un groupe  $G$  d'ordre  $m$  impair noté multiplicativement, pour tout  $a$  dans  $G$ , l'équation  $x^2 = a$  a une unique solution dans  $G$ , à savoir  $x = a^{(m+1)/2}$ .

Il en suit que dans  $\mathbb{F}_{2^n}^*$ , on a  $\sqrt{a} = a^{2^{n-1}}$ , tandis que si  $q \equiv 3 \pmod{4}$ , dans le groupe d'ordre impair  $(\mathbb{F}_q^*)^2$ , on a  $\sqrt{a} = a^{(q+1)/4}$ . Dans ces deux cas on dispose ainsi d'un algorithme de calcul de  $\sqrt{a}$  en  $O((\log q)^3)$  opérations.

## 2. Racines carrées : algorithme de Tonelli

Dans le cas général où  $q$  est impair, on écrit  $q - 1 = 2^s t$  avec  $t$  impair. Comme  $\mathbb{F}_q^*$  est cyclique, il existe un unique sous-groupe  $H$  d'ordre  $t$  et une chaîne de sous-groupes

$$H = G_0 \subset G_1 \subset \cdots \subset G_s = \mathbb{F}_q^*$$

avec  $G_i/G_{i-1}$  d'ordre 2. Si  $g$  n'est pas un carré dans  $\mathbb{F}_q^*$ , son image dans  $\mathbb{F}_q^*/H$  est un générateur de  $\mathbb{F}_q^*/H$ , et on peut écrire  $a = g^e h$  avec  $h$  dans  $H$ . L'idée est rechercher séparément une racine carrée de  $g^e$  et de  $h$ . Comme  $H$  est d'ordre impair, il est facile d'après la section précédente de calculer une racine de  $h$ . Pour trouver  $e$ , on procède de la façon suivante. SI on connaît  $e$  modulo  $2^{i-1}$  il y a seulement 2 choix possibles pour  $e$  modulo  $2^i$ , et seul le choix correct vérifie  $ag^{-e} \in G_{s-i}$ . Une fois trouvé  $e$ , qui est nécessairement pair, une racine carrée de  $g^e$  est  $g^{e/2}$ . Il reste encore à trouver un  $g$  qui n'est pas un carré. Pour cela on remarque qu'un  $g$  aléatoire dans  $\mathbb{F}_q^*$  n'est pas un carré avec probabilité  $1/2$ .

On obtient ainsi un algorithme randomisé qui échoue avec probabilité  $1/2$ . S'il n'échoue pas, il calcule une racine carrée de  $a$  en  $O(\nu_2(q - 1)(\log q)^3)$  opérations. Ici  $\nu_2(q - 1)$  désigne l'exposant de la plus grande puissance de 2 divisant  $q - 1$ .

### 3. Racines carrées : algorithme de Cipolla

Voici un autre algorithme de calcul de  $\sqrt{a}$  pour  $q$  est impair. C'est un algorithme probabiliste qui fonctionne en  $O((\log q)^3)$  opérations.

L'idée est la suivante. Soit  $a$  dans  $\mathbb{F}_q^*$  dont on recherche la racine carrée. On suppose connue une extension quadratique  $\mathbb{F}_{q^2}$  de  $\mathbb{F}_q$  et un élément  $x$  de  $\mathbb{F}_{q^2}$  de norme  $x^{q+1}$  égale à  $a$ . Alors  $x^{(q+1)/2}$  est une racine carrée de  $a$ . Pour cela on choisit  $t$  dans  $\mathbb{F}_q$  au hasard. Si  $t^2 - 4a$  n'est pas un carré, on a échoué. Si c'est un carré, l'algorithme calcule la classe du polynôme  $X^2 - tX + a$  modulo  $X^{(q+1)/2}$ .

Notons  $\chi(x) = x^{(q-1)/2}$  pour  $x$  non nul dans  $\mathbb{F}_q$ . Pour comprendre pourquoi cela marche, on fait les observations suivantes :

1) Si  $t$  est choisi au hasard dans  $\mathbb{F}$ ,  $\chi(t^2 - 4a) = -1$  avec probabilité  $(q-1)2q$ . (Ceci résulte du fait que parmi les polynômes  $X^2 - tX + a$ ,  $(q+1)/2$  sont scindés sur  $\mathbb{F}_q$  et  $(q-1)/2$  ne le sont pas.)

2) Si  $\chi(t^2 - 4a) = -1$ ,  $\mathbb{F}[X]/(X^2 - tX + a)$  est une extension de degré 2 de  $\mathbb{F}$  dans laquelle la norme de la classe de  $X$  est égale à  $a$ .

L'algorithme que l'on vient de décrire échoue avec probabilité  $1/2 + 1/2q$ . Sinon, il produit une racine de  $a$  en temps  $O((\log q)^3)$ . Notons que l'évaluation du caractère quadratique  $\chi$  s'effectue  $O((\log q)^2)$  opérations, ce qui est négligeable par rapport au reste de l'algorithme qui repose sur des calculs de puissances.

Exemples : On pourra par exemple calculer une racine carrée de 3615 modulo  $2^{16} + 1$  ou de 552512556430486016984082237 modulo  $2^{89} - 1$ .

### 4. Un algorithme de factorisation (algorithme de Dixon)

Soit  $N$  un nombre entier. Si  $N$  n'est pas premier le calcul de racines carrées modulo  $N$  est essentiellement équivalent à celui de la factorisation de  $N$ . En effet si dans  $\mathbb{Z}/N\mathbb{Z}$  on sait calculer une racine carrée  $y$  de  $a = x^2$ , alors  $(x - y)(x + y) = 0$  dans  $\mathbb{Z}/N\mathbb{Z}$  et donc si  $x \neq \pm y$ , on a une factorisation de  $N$ .

Soit  $N$  un entier impair que l'on se propose de factoriser. On fixe un paramètre  $B$  dans  $\mathbb{R}_{>0}$  et on suppose connue la liste des nombres premiers  $p_1 < p_2 < \dots < p_h$  inférieurs à  $B$  tels que  $n$  soit un carré modulo  $p_i$ . Rappelons qu'il est facile de déterminer si  $n$  est un carré modulo  $p_i$  en utilisant la loi de réciprocité quadratique.

En pratique on choisit  $B$  de l'ordre de  $\exp(1/2(\log n \log \log n)^{1/2})$ . Un entier  $b$  est un  $B$ -entier si le reste de la division de  $b^2$  par  $N$  est produit

de tels nombres premiers. On note alors  $a_i(b)$  la valuation  $p_i$ -adique de ce reste et  $\varepsilon(b)$  le vecteur de composantes  $\alpha_i(b) \pmod 2$  dans  $\mathbb{F}_2^h$ . L'algorithme fonctionne de la façon suivante.

On considère les  $h + 1$  premiers entiers  $b_1 < \dots < b_{h+1}$  de la forme  $[\sqrt{n} + 1], [\sqrt{n} + 2], \dots, [\sqrt{n} + k], \dots$ , qui sont des  $B$ -entiers. Par un algorithme d'algèbre linéaire on produit une sous-famille  $c_1, \dots, c_\ell$  de  $b_1 < \dots < b_{h+1}$  telle que  $\sum_i \varepsilon(c_i) = 0$  dans  $\mathbb{F}_2^h$ . On pose alors  $s = \prod_i c_i$  et  $t = \prod_j p_j^{\gamma_j}$  avec  $\gamma_j = 1/2 \sum_i a_j(c_i)$  et on a  $s^2 \equiv t^2 \pmod N$ . Si  $s \neq \pm t$  on a une factorisation. Sinon on recommence en variant le choix de  $B$  ou de la suite des  $b_i$ .

### 5. Une autre application : le cryptosystème de Rabin

L'observation sur le lien entre calcul de racines carrées modulo  $N$  et factorisation de  $N$  est également à la base du cryptosystème suivant.

Soit  $N$  un entier de la forme  $N = pq$  avec  $p$  et  $q$  deux "grands nombres premiers". Pour envoyer un message  $x$  à Alice, Bob utilise la clé publique  $N$  d'Alice et envoie le reste  $y$  de la division de  $x^2$  par  $N$  à Alice. Celle-ci calcule les racines carrées de  $y$  modulo  $p$  et  $q$  respectivement, par une des méthodes précédentes (noter que rien n'interdit de choisir  $p$  et  $q \equiv 3 \pmod 4$ ), puis à l'aide du théorème chinois elle trouve une racine carrée de  $y$  modulo  $N$  (en fait elle en trouve 4, mais il y a des astuces pour régler ce problème).

---