Recherche de générateurs

Soit $p \geq 3$ un nombre premier. Le groupe $\mathbf{G} = \mathbb{Z}_p^*$ est cyclique. Le but de cet exercice est de trouver un générateur de ce groupe, i.e., un élément g tel que $\mathbb{Z}_p^* = \{g^k, k \in \mathbb{Z}\}$. Pour $g \in \mathbf{G}$, l'ordre de g est le plus petit $k > 0$ tel que $g^k = 1$, noté $\mathrm{ord}(g)$.

Question 1. *Montrer que pour tout $g \in \mathbf{G}$, $\mathrm{ord}(g) \,|p-1$.*

Solution. *Le groupe multiplicatif $\mathbf{G} = \mathbb{Z}_p^*$ est un groupe cyclique d'ordre $p-1$ (car il contient tous les éléments non nuls modulo p et il est cyclique par hypothèse).*

Pour tout élément $g \in \mathbf{G}$, l'ordre de g, noté $\mathrm{ord}(g)$, est le plus petit entier $k > 0$ tel que $g^k = 1$. Par définition de l'ordre d'un élément dans un groupe, on sait que $g^{\mathrm{ord}(g)} = 1$.

Puisque g appartient à un groupe de taille $p-1$, l'ordre de g doit diviser l'ordre du groupe. Par conséquent, $\mathrm{ord}(g)$ divise $p-1$ selon le théorème de Lagrange.

Question 2. *Donner un élément de $\mathbf{G}$ qui n'est pas un générateur de $\mathbf{G}$. Combien d'éléments de $\mathbf{G}$ sont générateurs ?*

Solution. *Un générateur de $\mathbf{G}$ est un élément $g \in \mathbf{G}$ tel que l'ordre de g est exactement $p-1$, c'est-à-dire que g génère tous les éléments de $\mathbb{Z}_p^*$.*

Soit d un diviseur propre de $p-1$ (c'est-à-dire $d \neq 1$ et $d \neq p-1$). Un élément $h \in \mathbf{G}$ d'ordre d ne peut pas être un générateur de $\mathbf{G}$ parce que les puissances de h ne couvrent qu'un sous-groupe de $\mathbf{G}$, de taille d.

Le nombre d'éléments générateurs de $\mathbf{G}$ est donné par le nombre de k tels que $1 \leq k < p$ et $\mathrm{pgcd}(k, p-1) = 1$ (car ce sont les éléments d'ordre $p-1$). Ce nombre est donc $\varphi(p-1)$.

On se restreint au cas où $p - 1 = 2q$ pour q premier.

Question 3. *Donner un algorithme efficace qui trouve un générateur de $\mathbf{G}$. Comment trouver un tel premier p ?*

Solution. *Pour trouver un générateur de $\mathbf{G}$, on peut utiliser l'algorithme suivant :*

1. Choisir un élément $g \in \mathbf{G}$ aléatoirement. 2. Calculer $g^{(p-1)/2} \mod p$. Si $g^{(p-1)/2} = 1$, alors g n'est pas un générateur, car son ordre divise $(p-1)/2 = q$. 3. Si $g^{(p-1)/2} \neq 1$ et $g^{(p-1)/q} \neq 1$, alors g est un générateur de $\mathbf{G}$ car son ordre est $p-1$.

Ce test est efficace parce qu'il ne nécessite que quelques exponentiations mod p.

Pour trouver un tel premier p, on peut partir de q et vérifier si $p = 2q + 1$ est également premier.

Algorithme de Shanks

Soit $\mathbf{G}$ un groupe cyclique généré par g, d'ordre premier q connu, et $h \in \mathbf{G}$.

Question 4. *Soit $m = \lceil \sqrt{q} \rceil$. Montrer qu'il existe $i, j \leq m$ tels que : $g^i = h(g^{-m})^j$.*

Solution. *Soit x le DL en base g de h, on fait sa division euclidienne par m, il existe $i, j < m$ tels que $x = i + jm$.*

Question 5. *En déduire un algorithme de résolution du DLP, et sa complexité.*

Solution. *Construire une table de hachage pour les g^i (baby step). Calculer les $(g^{-m})^i$ (giant step) et chercher dans la table. Complexité en $\mathcal{O}(\sqrt{q})$.*

Algorithm 1 Algorithme Rho de Pollard.

Input : deux éléments $h, g \in \mathbf{G}$
Output : $x \in [0; q-1]$ tel que $h = g^x$ ou $\perp$

$i \leftarrow 1$, $x \leftarrow 0$, $\alpha \leftarrow h$
$y \leftarrow F(\alpha), \beta \leftarrow H(\alpha)$
while $\alpha \neq \beta$ **do**
    $x \leftarrow x + F(\alpha) \mod q, \alpha \leftarrow H(\alpha)$
    $y \leftarrow y + F(\beta) \mod q, \beta \leftarrow H(\beta)$
    $y \leftarrow y + F(\beta) \mod q, \beta \leftarrow H(\beta)$
    $i \leftarrow i + 1$
end while
if $i < q$ **then**
    Return $(x - y)i^{-1} \mod q$
else
    Return $\perp$
end if

Algorithme Rho de Pollard

Soit $\mathbf{G}$ comme dans la question précédente, $F : \mathbf{G} \to \mathbb{Z}_q$ une fonction non-nulle, et $H : \mathbf{G} \to \mathbf{G}$ définie par $H(\alpha) = \alpha \cdot h \cdot g^{F(\alpha)}$. Nous étudions l'algorithme suivant.

Nous définissons inductivement la séquence (γ_i) par $\gamma_1 = h$ et $\gamma_{i+1} = H(\gamma_i)$ pour $i \geq 1$.

Question 6. *Montrer que dans la boucle While de l'algorithme, on a $\alpha = \gamma_i = g^x h^i$ et $\beta = \gamma_{2i} = g^y h^{2i}$.*

Solution. *Par induction. L'important ici est de remarquer qu'on passe d'un élément aléatoire à un autre tout en connaissant leurs DLs.*

Question 7. *Montrer que si on sort de la boucle avec $i < q$, alors l'algorithme renvoie le logarithme discret de h en base g.*

Solution. *La condition de sortie est $\alpha = \beta \implies g^x hi = g^y h^{2i}$. Par conséquent $g^{(x-y)i^{-1}} = h$.*

Question 8. *Soit j le plus petit entier tel que $\gamma_j = \gamma_k$ pour $k < j$. Montrer que $j \leq q + 1$ et qu'en sortant de la boucle on a $i < j$.*

Solution. *Il faut s'intéresser à la fonction H. Elle est à valeurs dans un ensemble de taille q. L'ensemble :*

$$\{h, H(h), H^2(h), \ldots\}$$

est donc de taille q au plus. Et par conséquent il existe un entier j tel que $\gamma_j = \gamma_k$ pour un plus petit k, et ce j est au plus $q + 1$.

Quand on sort de la boucle on obtient le plus petit i tel que $\gamma_i = \gamma_{2i}$. (Faire des dessins !). Écrivons la division euclidienne de i par j : $i = jq + r$. Alors : $\gamma_i = \gamma_{jq+r} = \gamma_{kq+r}$. De même $\gamma_{2i} = \gamma_{2kq+2r}$ (ce n'est pas trivial à écrire, il faut isoler des groupes de j itérations et les remplacer par des groupes de k itérations à chaque fois par hypothèse). Or $k < j$ donc si $q > 0$, on a $kq + r < jq + r$, ce qui contredirait le fait que i est le plus petit satisfaisant l'hypothèse.

Question 9. *Montrer que si F est une fonction aléatoire, alors le temps d'exécution moyen de l'algorithme est en $\mathcal{O}(q^{1/2})$ multiplications dans $\mathbf{G}$. Comment l'implémenter dans la pratique ?*

Solution. *Paradoxe des anniversaires. En supposant que la séquence des γ_i est aléatoire, la probabilité qu'il existe une collision après $\mathcal{O}(q^{1/2})$ itérations est constante.*

Par conséquent l'algorithme va terminer après $\mathcal{O}(q^{1/2})$ itérations.

Pour l'implémenter, on peut utiliser une fonction de hachage cryptographique, mais une fonction de hachage quelconque suffit.

Remarquer que l'algorithme Rho a un très gros avantage sur le BSGS en terme de complexité mémoire.

Collisions des Fonctions de Hachage

L'algorithme rho est un exemple particulier de l'algorithme de recherche de cycles de Floyd, que nous allons maintenant utiliser pour trouver une collision dans une fonction de hachage $H : \{0,1\}^n \to \{0,1\}^n$ (notez que pour une fonction de hachage quelconque, il doit d'abord réduire arbitrairement la taille de son entrée à n bits).

Soit X_0 une valeur de départ ; on définit la séquence $X_{i+1} := H(X_i)$. Comme elle prend des valeurs dans un ensemble fini, elle est périodique après un certain point.

Soit c la longueur de la *pré-période* $X_0, \ldots, X_{c-1}$ et ℓ la longueur du *cycle*, tels que : • $X_0, \ldots, X_{c+\ell-1}$ sont tous distincts ; • pour tout $i \geq c$, $X_{i+\ell} = X_i$. On admet que pour une fonction aléatoire, $c = \mathcal{O}(2^{n/2})$ et $\ell = \mathcal{O}(2^{n/2})$.

On exécute l'algorithme Algorithm 2 donné ci-dessous.

Algorithm 2 Algorithme de recherche de cycle de Floyd.

```
x ← H(X_0)
y ← H(H(X_0))
while x ≠ y do
    x ← H(x)
    y ← H(H(y))
end while
```

Question 10. *Justifier que l'algorithme renvoie un élément x appartenant au cycle.*

Solution. *On a : $H^{2i}(X_0) = H^i(X_0) = x$ donc $H^i(x) = x$ donc par définition x appartient au cycle et i est un multiple de ℓ.*

Question 11. *Donner un algorithme pour trouver la longueur ℓ du cycle.*

Solution. *Il suffit de repartir de x, de calculer $H^i(x)$ jusqu'à avoir $H^i(x) = x$. Le premier tel i est la longueur du cycle ℓ.*

Si $c > 0$ et $\ell > 1$, on a :

$$H(X_{c-1}) = X_c = X_{c+\ell} = H(X_{c+\ell-1})$$

et par définition $X_{c-1} \neq X_{c+\ell-1}$. La paire $(X_{c-1}, X_{c+\ell-1})$ est donc une collision de H. Pour la trouver, on calcule deux nouvelles séquences partant de X_0 et X_ℓ. Le premier indice i tel que $X_i = X_{i+\ell}$ est égal à c.

Question 12. *En déduire qu'on peut calculer une collision de la fonction H en temps $\mathcal{O}(2^{n/2})$ et mémoire $\mathcal{O}(1)$.*