Chapitre 3 : Algorithme de Shor

Informatique quantique pour la recherche opérationnelle
Dimitri Watel - ENSIIE

2022

1 Introduction

L'algorithme de Shor permet de résoudre le problème suivant: connaissant un entier n qui est le produit de deux nombres premiers p et q, que valent p et q ?

Résoudre ce problème permettrait de rapidement casser des protocoles de chiffrement à base de décomposition en facteurs premiers, comme le protocole RSA. L'existence de algorithme de Shor est donc problématique à ce sujet. Mais pour le moment, en pratique, les ordinateurs quantiques ne sont pas assez performants pour résoudre ce problème avec de grandes valeurs de n.

2 Un peu d'arithmétique

Commençons par choisir un entier a au hasard entre 1 et $n-1$. Si cet entier divise n alors il s'agit de p ou q. Plus généralement, on a le résultat suivant:

Lemme 2.1. *Si $d = PGCD(n, a) \neq 1$ alors $d = p$ ou $d = q$.*

Proof. Puisque $a < n$ alors $d \neq n$ et puisque $d \neq 1$ alors d est un diviseur non trivial de n, donc p ou q. □

On suppose donc dans la suite que $PGCD(a, n) = 1$. On dit que a est *inversible* dans $\mathbb{Z}/n\mathbb{Z}$, ou qu'il appartient à l'ensemble $\mathbb{Z}/n\mathbb{Z}^*$.

Lemme 2.2. *$a \in \mathbb{Z}/n\mathbb{Z}^*$ si et seulement s'il existe un entier $b \in \mathbb{Z}/n\mathbb{Z}$ tel que $ab \equiv 1 \mod n$.*

Proof. D'après le théorème de Bézout, $PGCD(a, n) = 1$ si et seulement s'il existe u et v tels que $au + nv = 1$, si et seulement si il existe u tel que $au \equiv 1 \mod n$. Notre entier b est $u \mod n$. □

Pour trouver p et q nous allons utiliser l'ordre de a.

Définition 1. L'ordre de a est le plus petit entier r tel que $a^r \equiv 1 \mod n$.

Lemme 2.3. *Cet ordre existe toujours.*

Proof. En effet, il suffit d'avoir au moins un entier p pour lequel $a^p \equiv 1 \mod n$. On peut montrer que c'est vrai pour $\varphi(n) = |\mathbb{Z}/n\mathbb{Z}^*|$.

Considérons $P = \prod_{x \in \mathbb{Z}/n\mathbb{Z}^*} x$.

Donc $\prod_{x \in \mathbb{Z}/n\mathbb{Z}^*} ax = a^{\varphi(n)} P$. Mais ce produit est aussi égal à P. En effet, $x \to ax$ est une bijection dans $\mathbb{Z}/n\mathbb{Z}^*$, donc $\prod_{x \in \mathbb{Z}/n\mathbb{Z}^*} ax$ est le produit $\prod_{x \in \mathbb{Z}/n\mathbb{Z}^*} x$ où les x ont été permutés, donc c'est le même produit.

Ainsi $a^{\varphi(n)} P \equiv P \mod n$.

Puisque chaque x est inversible alors P aussi est inversible, $P^{-1} = \prod_{x \in \mathbb{Z}/n\mathbb{Z}^*} x^{-1}$.

Donc $a^{\varphi(n)} \equiv 1 \mod n$. □

Ce dernier lemme est celui qu'on va utiliser pour trouver p et q.

Lemme 2.4. *Si l'ordre de a est r, si r est pair, et si $a^{r/2} + 1 \not\equiv 0 \mod n$, alors $p = PGCD(a^{r/2} + 1, n)$ et $q = PGCD(a^{r/2} - 1, n)$.*

Proof. Par définition de r, on sait que $a^{r/2} - 1 \not\equiv 0 \mod n$. Aussi, $(a^{r/2} - 1) \cdot (a^{r/2} + 1) \equiv a^r - 1 \equiv 0 \mod n$.

Posons $x = a^{r/2} + 1$ et $y = a^{r/2} - 1$. On a $xy \equiv 0 \mod n$, $x \not\equiv 0 \mod n$ et $y \not\equiv 0 \mod n$.

Supposons que $PGCD(x, n) = 1$, alors il existe u et v tels que $ux + vn = 1$, donc $y = uxy + vny$, donc $y \equiv 0 \mod n$ ce qui est exclu. Donc même si $PGCD(y, n) = 1$. Donc ces deux PGCD sont différents de 1, alors ils divisent n et on trouve alors p et q. □

On a donc l'algorithme (classique) suivant pour trouver p et q.

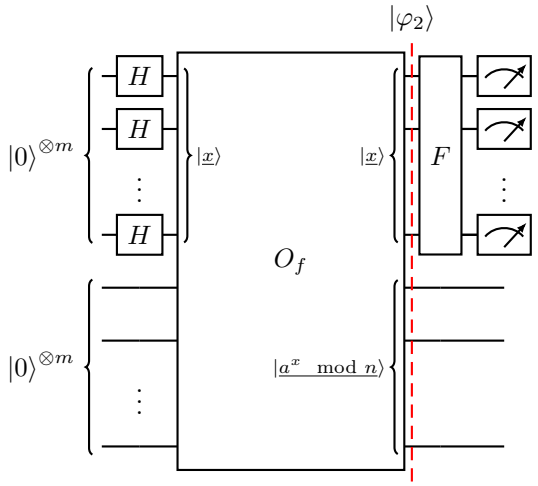
1: **Boucle infinie**
2: $a \leftarrow$ élément de $\mathbb{Z}/n\mathbb{Z}$ au hasard
3: **Si** $(PGCD(a, n) = d \neq 1)$ **Renvoyer** d
4: $r \leftarrow$ ordre de a
5: **Si** r est pair et $a^{r/2} + 1 \not\equiv 0 \mod n$ **Alors**
6: **Renvoyer** $PGCD(a^{r/2} + 1, n)$ et $PGCD(a^{r/2} - 1, n)$

L'algorithme de Shor va s'intéresser particulièrement à trouver r qui est un problème pour lequel on ne connait pas d'algorithme classique polynomial.

3 Un peu de quantique

Pour trouver r, on va utiliser un algorithme à oracle, mais la porte de l'oracle sera légèrement différente des précédentes car la sortie fait plus que 1 qbit.

On pose $2^m \geq n^2$. Le circuit de l'algorithme est le suivant, avec $2m$ qbits en entrées. Dans ce circuit, les m premiers qbits sont ceux devant les portes H. Les m derniers qbits sont les autres.



où

- $\omega = e^{\frac{2i\pi}{2^m}}$
- $F|\underline{x}\rangle = \frac{1}{\sqrt{2^m}} \sum_{y=0}^{2^m-1} \omega^{xy} |\underline{y}\rangle$

On reviendra sur cette porte.
Une fois l'oracle effectué, on obtient

$$|\varphi_2\rangle = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |\underline{x}\rangle \otimes |\underline{a^x \mod n}\rangle$$

Notez qu'il peut exister plusieurs x tels que $a^x \mod n$ donne la même valeur. Soit $a^y \mod n$ cette valeur. Alors le premier m-qbit vaut un x tel que $a^x \equiv a^y \mod n$. Donc $a^{x-y} \equiv 1 \mod n$, ce qui signifie que $x-y$ est un multiple de r.

$$|\varphi_2\rangle = \frac{1}{\sqrt{2^m}} \sum_{y=0}^{r-1} \sum_{\substack{x=0 \\ x-y=kr}}^{2^m-1} |\underline{x}\rangle \otimes |\underline{a^y \mod n}\rangle$$

3.1 Cas simple : r divise 2^m

On suppose ici que r est une puissance de 2, ce qui est très improbable, mais cela simplifie les calculs.

Soit $q = \frac{2^m}{r}$. Puisque r divise 2^m alors q est entier et il y a exactement q multiples de r inférieurs à 2^m.

$$|\varphi_2\rangle = \frac{1}{\sqrt{r}} \sum_{y=0}^{r-1} \frac{1}{\sqrt{q}} \sum_{k=0}^{q-1} |\underline{kr+y}\rangle \otimes |\underline{a^y \mod n}\rangle$$

Soit $|\varphi_3\rangle$ le qbit qu'on obtiendrait si on faisait une mesure du 2e m-qbit après O_f et avant F. Si on mesure a^y alors

$$|\varphi_3\rangle = \frac{1}{\sqrt{q}} \sum_{k=0}^{q-1} |\underline{kr+y}\rangle \otimes |\underline{a^y \mod n}\rangle$$

Une fois ce qbit passé dans la porte F, on obtient

$$|\varphi_4\rangle = (F \otimes Id)|\varphi_3\rangle$$
$$|\varphi_4\rangle = F\left(\frac{1}{\sqrt{q}} \sum_{k=0}^{q-1} |\underline{kr+y}\rangle\right) \otimes |\underline{a^y \mod n}\rangle$$
$$|\varphi_4\rangle = \frac{1}{\sqrt{q2^m}} \sum_{z=0}^{2^m-1} \left(\sum_{k=0}^{q-1} \omega^{krz}\right) \omega^{yz} |\underline{z}\rangle \otimes |\underline{a^y \mod n}\rangle$$

Lemme 3.1.

$$\sum_{k=0}^{q-1} \omega^{krz} = \begin{cases} q \text{ si } z \equiv 0 \mod q \\ 0 \text{ sinon} \end{cases}$$

Proof. $\sum_{k=0}^{q-1} \omega^{krz} = \sum_{k=0}^{q-1} e^{\frac{2i\pi}{2^m} krz} = \sum_{k=0}^{q-1} e^{\frac{2i\pi}{q} kz}$

Si $z \equiv 0 \mod q$, alors z est multiple de q, donc $\frac{2i\pi}{q} kz$ est multiple de $2i\pi$ alors l'exponentielle vaut 1. Il s'agit donc d'une somme de q fois la valeur 1.

Si $z \not\equiv 0 \mod q$, alors la somme vaut $\frac{(e^{\frac{2i\pi}{q}k})^q - 1}{e^{\frac{2i\pi}{q}k} - 1}$ dont le numérateur vaut 0. □

En utilisant ce lemme, on obtient le résultat suivant:

$$|\varphi_4\rangle = \frac{1}{\sqrt{r}} \sum_{\substack{z=0 \\ z \equiv 0 \mod q}}^{2^m-1} \omega^{yz} |\underline{z}\rangle \otimes |\underline{a^y \mod n}\rangle$$
$$= \frac{1}{\sqrt{r}} \sum_{k=0}^{2^m/q-1} \omega^{ykq} |\underline{kq}\rangle \otimes |\underline{a^y \mod n}\rangle$$

Ainsi, en mesurant le premier qbit, on obtient un multiple de q. On applique alors l'algorithme suivant

1: **Si** $(k = 0)$ Tout recommencer
2: Calculer $\frac{kq}{2^m} = \frac{k}{r}$
3: Simplifier la fraction.
4: **Si** Le dénominateur vérifie $a^r \equiv 1 \mod n$ **Alors**
5: Le renvoyer
6: **Sinon**
7: Tout recommencer

Le deuxième **Si** arrive si le PGCD de k et r vaut 1. Le dénominateur est alors nécessairement r (sinon la fraction serait simplifiée).

3.2 Comment construire F ?

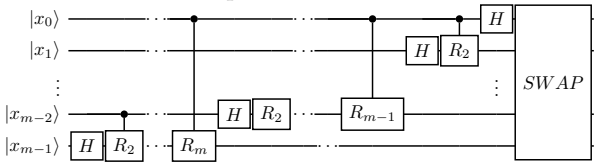
Le théorème suivant est un résultat permettant rapidement de démontrer que le circuit ci-après est correct.

Théorème 3.1. *Posons* $\underline{x} = x_{m-1} \dots x_2 x_0$

$$F|\underline{x}\rangle = \frac{1}{\sqrt{2^m}} \bigotimes_{k=1}^{m} \left(|0\rangle + \exp(2i\pi \cdot \sum_{j=0}^{k-1} \frac{x_j}{2^{k-j}}) |1\rangle \right)$$

Proof. On peut facilement montrer que $F|\underline{x}\rangle = \frac{1}{\sqrt{2^m}} \bigotimes_{k=1}^{m} \left(|0\rangle + \exp(2i\pi \cdot \frac{x}{2^k}) |1\rangle\right)$ en développant ce produit et en remarquant qu'il est égal à $\sum_{y=0}^{2^m-1} \omega^{xy} |\underline{y}\rangle$. On remplace ensuite x par $\sum_{j=0}^{m-1} x_j 2^j$ pour obtenir le résultat souhaité. □

Le circuit suivant implante cette formule de F.



où R_k est la porte $\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2i\pi}{2^k}} \end{pmatrix}$.

3.3 Cas difficile : r ne divise pas 2^m

On pose $2^m = qr + r'$ avec $r' > 0$. On a donc $\omega^r = e^{\frac{2i\pi r}{2^m}} \neq e^{\frac{2i\pi}{q}}$ et donc $\sum_{k=0}^{q-1} \omega^{krz} \neq \begin{cases} q \text{ si } z = 0 \mod q \\ 0 \text{ sinon} \end{cases}$.

Mais cette égalité est vraie si z est un multiple de $\frac{2^m}{r}$. Puisque cette fraction n'est plus entière, alors les entiers les plus proches de cette fraction devraient donner une somme proche de q. Ainsi, au lieu de mesurer un multiple de q comme dans le cas simple, on va mesurer un entier proche d'un multiple de $\frac{2^m}{r}$.

Comment retrouver r à partir de cet entier qui n'est pas un multiple de $\frac{2^m}{r}$ mais seulement proche de celui-ci? Soit z l'entier mesuré alors on va développer $\frac{z}{2^m}$ en fraction continue.

Définition 2. Soit une suite $(u_n)_{n\in\mathbb{N}}$ d'entiers, alors la fraction continue associée à cette suite est la limite de la fraction:

$$x = \lim_{n\to+\infty} f_n = u_0 + \frac{1}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}}$$

La n^{ieme} fraction f_n est un rationnel qui approche x.
On va ensuite utilier le résultat suivant.

Théorème 3.2. *Si* $\left|x - \frac{p}{q}\right| < \frac{1}{2q^2}$, *alors* $\frac{p}{q}$ *est une des fractions du développement en fractions continues de* x.

Si z est bien l'entier le plus proche d'un multiple de $\frac{2^m}{r}$, alors $\left|z - \frac{k2^m}{r}\right| \leq \frac{1}{2}$. Puisque $2^m \geq n^2$ alors $\left|\frac{z}{2^m} - \frac{k}{r}\right| < \frac{1}{2r^2}$. on peut donc appliquer le théorème précédent avec $x = \frac{z}{2^m}$ et on obtient la fraction $\frac{k}{r}$, on peut ensuite appliquer le même raisonnement que pour le cas simple.

Le développement en fraction continue se fait rapidement. On obtient la fraction $\frac{k}{r}$ en $O(\log(r)^3)$.

3.4 Probabilité de succès

Il existe de nombreux endroits où l'algorithme échoue et où on doit tout recommencer. Quelle est la probabilité de ne pas recommencer ?

- Connaissant un multiple $\frac{2^m k}{r}$ de $\frac{2^m}{r}$, la probabilité de mesurer un entier z le plus proche de $\frac{2^m k}{r}$ est de $\frac{1}{3r}$ (Source : papier original de Shor).
- Parmis tous les multiples $\frac{2^m k}{r}$, on en veut un où $PGCD(k, r) = 1$, donc $k \in \mathbb{Z}/r\mathbb{Z}^*$. Il y a donc $\varphi(r) = |\mathbb{Z}/r\mathbb{Z}^*|$ valeurs possibles mesurées pour k qui sont satisfaisantes.
- Donc pour mesurer un entier z plus proche multiple de $\frac{2^m}{r}$ dont le développement en fraction continue donne $\frac{k}{r}$ où $k \in \mathbb{Z}/r\mathbb{Z}^*$ est $\frac{\varphi(r)}{3r}$
- Or, $\frac{\varphi(r)}{r} = \Theta\left(\frac{1}{\log\log(r)}\right)$. (Source : wikipedia).

  On a environ une chance sur $\log\log(r)$ de tomber sur r.
- On recommence si r est impair ou $a^{r/2} + 1 \equiv 0 \mod n$. Ce cas arrive avec une probabilité inférieure à $\frac{1}{2}$ (Source : livre *Quantum Un peu de mathématiques pour l'informatique quantique* cité dans les sources).
- L'algorithme de Shor recommence donc en moyenne $\Theta(\log\log(r))$ fois avant de trouver r.

On recommence en moyenne $O(\log\log(r))$ fois

- le circuit avec $m = O(\log(n))$ qbits
- des portes H et O_f de tailles $O(m)$
- la porte F de taille $O(m^2)$
- la recherche de r avec les fractions continues en $O(\log(r)^3)$

avec $m = O(\log(n))$ et $r = O(n)$.
Donc l'algorithme est polynomial.