Algorithme de Shor

Baptiste Daumen

30 juin 2022

1 Introduction

La factorisation en nombre premier est un problème compliqué. Pour l'instant aucun alogrithme classique ne peut résoudre ce problème en temps polynomial. De nos jours, de nombreux algorithmes de cryptographie se basent sur la complexité de résoudre ce problème. L'algorithme le plus connu est le chiffrement RSA [1].
L'ordinateur quantique ouvre de nouvelles perspectives. La rapidité de calcul et les nouveaux types d'algorithme offrent la possibilité de trouver une façon de résoudre ce problème en un temps raisonnable, de façon déterministe ou probabiliste. La possibilité de résoudre ce problème remettrait en cause de nombreux systèmes de protection, par exemple les systèmes bancaires.
En 1994, un mathématicien américain, Peter Shor, invente un algorithme permettant de factoriser un entier naturel N en nombre premier. La complexité en temps est en $\mathcal{O}(\log(N)^3)$ et en espace en $\mathcal{O}(\log(N))$. La complexité en temps est exxponentiellement fois meilleur que les algorithmes classiques découverts.
Je vais dans ce papier présenté l'algorithme de Shor afin d'en découvrir le contenu théorique et essayer de l'implémenter sur un (vrai) algorithme quantique.

2 Algorithme de Shor

L'algorithme de Shor se décompose en plusieurs étapes. Il repose sur des résultats de la théorie des groupes. Il se sépare en deux parties. Une partie classique, qui peut être implémenté sur un ordinateur classique et une autre quantique qui nécessite un ordinateur quantique.
On note N l'entier naturel qu'on cherche à factoriser.
Premièrement, on peut s'assurer que N n'est pas premier en temps polynomial qur un ordinateur quantique. Soit en utilisant des tests de primalité probabilistes comme le test de Miller Rabin. Soit en utilisant un algortihme déterministe, AKS qui est bien polynomiale mais en pratique pas très utilisable.
Si l'entier N n'est pas premier, on peut alors cherche à le factoriser en produit de nombre premiers.
Je commence par présenter la partie "classique".

2.1 Partie Classique

On choisit un nombre pseudo-aléatoire (tiré aléatoirement sur un ordinateur classique) $a < N$.
A l'aide de l'algorithme d'Euclide (complexité en temps polynomial sur la taille de l'entrée), on détermine $PGCD(a, N)$. Si ce dernier est différent de 1 alors on a trouvé un facteur de N et on peut réappliquer l'algorithme à $\frac{N}{PGCD(a,N)}$ afin de détemriner la factorisation complète. Sinon, le $PGCD$ vaut 1 et a et N sont premiers entre eux.
La résolution du problème de factorisation en nombre premier peut être ramené à un problème de recherche de période d'une fonction, qui donne l'ordre de certains nombres dans l'anneau $(\mathbb{Z}/N\mathbb{Z})^*$.
On pose

$$\begin{array}{rccc} f & : & \mathbb{Z} & \to & \mathbb{Z}/N\mathbb{Z} \\ & & x & \mapsto & a^x \ (mod\ N) \end{array}$$

On sait que a possède un ordre dans $(\mathbb{Z}/N\mathbb{Z})^*$ donc il existe r tel que
$f(x+r) = f(x)$. La partie quantique (qui sera détaillée par la suite va nous permettre de déterminer ce r.
On présente la fin de l'algorithme, on suppose donc qu'on a déterminé r.
On suppose que r est pair. On peut écrire $a^r - 1 = 0$ donc $(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = 0$. On sait que $(a^{\frac{r}{2}} - 1) \neq 0$ car r par définition est le plus entier tel que $a^r = 1$. N n'étant pas premier $(\mathbb{Z}/N\mathbb{Z})^*$ n'est pas forcément intègre et donc on peut trouver des éléments tels que $(a^{\frac{r}{2}} + 1) \neq 0$. On suppose cette hypothèse vraie. Comme $(a^{\frac{r}{2}} + 1) \neq 0$ et $(a^{\frac{r}{2}} - 1) \neq 0$ donc $PGCD(a^{\frac{r}{2}} + 1, N)$ et $PGCD(a^{\frac{r}{2}} + 1, N)$ se calculent en temps polynomial et donne des facteurs non triviaux de N. On reproduit l'algorithme sur les deux facteurs trouvés, on peut ainsi déterminer la décomposition voulue.
Deux questions se posent à ce moment là. La première est : que fait-on si r est impair ou si r est pair mais que $a^{\frac{r}{2}} + 1 = 0$?
Dans ces cas là on réitère l'algorithme en tirant un nouveau $a < N$.
On se rend donc compte que l'algorithme abouti uniquement s'il existe des entiers strictement inférieurs à N tels que r soit pair et $a^{\frac{r}{2}} + 1 \neq 0$. Un résultat d'arithmétique et de théorie des groupes affirment qu'ils existent au moins, plus d'un entier sur 2 entre 0 et $N-1$ qui vérifient ces conditions [2].
Ainsi on a décrit la partie classique de l'algorithme. Je vais présenter maintenant la partie quantique de l'algorithme.

2.2 Partie quantique

Pour la partie quantique je vais présenter la théorie qui permet de déterminer la phase.
Soit n le plus entier tel que $2^n > N$. On prépare deux états $|00...0\rangle$ de n qubits. On calcule
$(H^{\otimes n} \bigotimes \mathbb{1}^{\otimes n})(|0\rangle \bigotimes |0\rangle) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle |0\rangle$
Soit une porte U_f qui agit en transformant $|k\rangle \bigotimes |0\rangle$ en $|k\rangle \bigotimes |a^k \ (mod\ N)\rangle$. On applique à l'état obtenu précédemment la porte U_f et on obtient :
$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle |a^k\rangle = \frac{1}{\sqrt{N}} \sum_{\beta=0}^{r-1} (\sum_{\alpha=0}^{\frac{2^n}{r}-1} |\alpha r + \beta\rangle |a^{\alpha r+\beta}\rangle)$
On effectue pour la dernière égalité, la division euclidienne de k par r. On suppose par la suite que r divise 2^n. C'est une hypothèse réductrice, le cas sans l'hypothèse peut aussi être traité [3].
r étant l'ordre de a, on obtient $\frac{1}{\sqrt{N}} \sum_{\beta=0}^{r-1} \sum_{\alpha=0}^{\frac{2^n}{r}-1} |\alpha r + \beta\rangle \bigotimes |a^\beta\rangle$.
On fait la mesure du second état, on fixe donc $|a^{\beta_0}\rangle$.
Le premier état est donc $\sum_{\alpha=0}^{\frac{2^n}{r}-1} |\alpha r + \beta_0\rangle$. On applique la transformée de Fourier inverse [4]. L'état devient :
$\sum_{\alpha=0}^{\frac{2^n}{r}-1} (\sum_{j=0}^{2^n-1} e^{\frac{-2i\pi(\alpha r+\beta_0)j}{2^n}} |j\rangle) = \sum_{j=0}^{2^n-1} (\sum_{\alpha=0}^{\frac{2^n}{r}-1} e^{\frac{-2i\pi\alpha j}{\frac{2^n}{r}}}) e^{\frac{-2i\pi\beta_0 j}{2^n}} |j\rangle) = \sum_{j|\frac{2^n}{r}} e^{\frac{-2i\pi\beta_0 j}{2^n}} |j\rangle = \sum_{l=0}^{r-1} e^{\frac{-2i\pi\beta_0 l}{r}} |\frac{2^n l}{r}\rangle$
On peut mesurer l'état, on obtient $|\frac{2^n l_0}{r}\rangle$ avec l_0 compris entre 0 et $r-1$.
$m = \frac{2^n l_0}{r} \Leftrightarrow \frac{m}{2^n} = \frac{l_0}{r}$. Si $PGCD(l_0, r) = 1$, on peut obtnir r qui est le dénominateur de la fraction irréductible de $\frac{m}{2^n}$. Pour être sûr d'obtenir le bon r, on peut répéter ce protocole est avoir choisir le plus grand r obtenu.

2.3 Implémentation

La partie un magique pour l'implémentation semble est au niveau de la porte U_f, c'est une pporte implémentable comme nous l'avons vu le deuxième jour du PAF avec Peter Brown.
Par manque de temps, je n'ai pas pu essayer l'algorithme de Shor pour $N = 15$ sur un ordinateur quantique ou même une simulation.

Références

[1] Wikipédia *Chiffrement RSA*
https://fr.wikipedia.org/wiki/Chiffrement_RSA

[2] Compléments d'arithmétiques
https://exo7math.github.io/quantum-exo7/complement/complement.pdf

[3] Algorithme de Shor
https://exo7math.github.io/quantum-exo7/shor/shor.pdf

[4] Transformée de Fourier Quantique *Transformée de Fourier Quantique - Baptiste Daumen*

[5] Pour une meilleure compréhension de l'algorithme, je conseille la playlist de vidéos sur l'algorithme de Shor de la chaîne Youtube : Quantum.
https://www.youtube.com/watch?v=qPgqcSA8dzclist=PLMP-9Tz3oGTbetMh794L-SesaxSjhfLzG