Chapitre 5

L'algorithme de Shor

Sommaire

5.1 RSA 200
5.1.1 Les mathématiques derrière RSA 200
5.1.2 Comment Shor fragilise RSA 201
5.2 L'arithmétique derrière Shor 202
5.2.1 Quelques notations 202
5.2.2 Périodicité dans l'élévation à la puissance 202
5.2.3 Exemple trivial mais concret 203
5.2.4 En bref 204
5.3 Algorithme de Simon 204
5.3.1 Approche classique 204
5.3.2 Approche quantique 205
5.3.3 Avantage quantique dans l'algorithme de Simon . . 206
5.4 Implémentation de l'algorithme de Shor 207
5.4.1 Analyse du circuit 207
5.4.2 Récupération de la période 212
5.4.3 Retour sur l'hypothèse simplificatrice 213
5.4.4 Récapitulation de l'algorithme et analyse des coûts 215

Dans ce chapitre, nous allons voir un algorithme qui est sans conteste la star de l'informatique quantique. L'algorithme de Shor[52] permet de factoriser de très grand nombre avec une accélération exponentielle par rapport aux méthodes HPC classiques. Il fragilise donc les bases des algorithmes de

cryptographie actuels sur lesquels reposent une grande partie de la sécurité informatique. Sa simple existence a justifié une grande partie de l'intérêt des scientifiques pour l'informatique quantique. Cette section présente les différentes notions qui sont impliquées dans cet algorithme, à savoir :

— les fondements arithmétiques de l'algorithme ;
— l'implémentation quantique de la transformée de Fourier discrète ;
— la mise en oeuvre de ces composants pour construire l'algorithme de Shor.

5.1 RSA

Le chiffrement RSA[49] (du nom de Rivest-Shamir-Adieman) a été créé en 1977 et il pose les bases des chiffrements utilisés désormais un peu partout dans les communications réseaux.

Dans RSA, on construit des paires de clefs publiques et privées. La connaissance de ses propres clefs privées et des clefs publiques de ses interlocuteurs permet de mettre en place un channel de communication protégée entre eux.

5.1.1 Les mathématiques derrière RSA

RSA repose sur des fondements arithmétiques. Soit $N = p.q$ le produit de deux (grands) nombres premiers p et q.

On définit l'ensemble $\mathbb{Z}/N\mathbb{Z}$, l'ensemble des entiers modulo N, dont le cardinal est N. Si N était premier, cet ensemble serait un groupe multiplicatif (ce n'est pas le cas ici).

On définit l'ensemble $(\mathbb{Z}/N\mathbb{Z})*$ le sous-ensemble de $\mathbb{Z}/N\mathbb{Z}$ des nombres qui sont premiers avec N. Cet ensemble est un groupe multiplicatif. Le cardinal de cet ensemble est **l'indicatrice d'Euler**, on la note $\phi[N)$. Dans le cas où $N = p.q$ avec p et q premiers, on peut montrer que $\phi[N) = (p-1)(q-1)$

L'indicatrice d'Euler est associée au **théorème d'Euler** [1]. Le théorème d'Euler qui affirme que

$$\forall a \in \mathbb{N}, \forall n \in \mathbb{N}, n > 0, \text{ a premier avec n }, a^{\phi(n)} \equiv 1[n]$$

1. Attention, il y a deux théorèmes d'Euler, l'autre parle de mécanique des fluides

Si a et n sont premier entre eux, alors élevé à la puissance $\phi[n)$ est congruent à 1 modulo n.

On choisit deux entiers e et d tels que $e.d = 1$ modulo $\phi[N)$, e premier avec $\phi[N)$. Le nombre d est l'inverse de e dans $(\mathbb{Z}/N\mathbb{Z})*$

On construit les clefs privées et publiques de la façon suivante :

— la clef publique est le produit $N.d$

— la clef privée est le produit $N.e$

Supposons que Alice veuille envoyer l'entier P à Bob. Alice construit le chiffrement C défini par $C = P^e(modN)$.

La magie de RSA, ce résume dans cette formule : $C^d(modN) = P$, on chiffre connaissant e, on déchiffre connaissant d.

Démontrons cette formule, sachant que $e.d = 1[\phi[N)]$ donc $\exists k, e.d = k\phi(N) + 1$, et $\phi[N) = (p-1)(q-1)$. Il est toujours possible de choisir N, p, q de telle sorte qu'ils soient grands par rapport à P. En fait, on découpe le message à envoyer en tronçons qui sont assez petits pour être plus petit que p et q, et donc d'être premier avec N. Par conséquent, d'après le théorème d'Euler, pour tout entier P, $P^{\phi(N)} = 1[N]$

$$C^d[N] = P^{e.d}[N] = P^{k\phi(N)+1)}[N] = P.P^{k\phi(N)}[N] = P.(P^{\phi(N)})^k = P.1^k[N] = P[N]$$

En conclusion, en découpant le message en entiers P suffisamment petits pour être premiers avec N, si Alice et Bon connaissent leurs clefs publiques respectives et leurs propres clefs secrètes, ils peuvent chiffrer et déchiffrer P.

5.1.2 Comment Shor fragilise RSA

Pour casser RSA, il faut pouvoir calculer d connaissant e et N, donc il faut calculer l'indicatrice d'Euler qui est égale à $\phi(N) = (p-1)(q-1)$, il faut donc calculer p et q, donc factoriser $N = p.q$.

L'algorithme de Shor permet de faire cette factorisation dans une durée raisonnable, il affaiblit les bases mêmes de l'algorihme RSA.

5.2 L'arithmétique derrière Shor

L'arithmétique derrière l'algorithme de Shor est finalement assez simple. Elle repose sur des bases d'arithmétique modulo N.

5.2.1 Quelques notations

Si $n \in \mathbb{Z}$, on notera $\mathbb{Z}/n\mathbb{Z}$ le groupe quotient défini par la relation d'équivalence "a le même reste lors d'une division entière par n".

Si n est un nombre premier, alors $\mathbb{Z}/n\mathbb{Z}$ est un corps fini, puisque c'est un anneau quotienté par l'un de ses idéaux premiers.

On notera $(\mathbb{Z}/n\mathbb{Z})^*$ la sous-partie de $\mathbb{Z}/n\mathbb{Z}$ qui ne comprend que les éléments qui sont premiers avec n, c'est-à-dire qui n'ont pas de facteurs premiers communs avec n. On peut montrer que $(\mathbb{Z}/n\mathbb{Z})^*$ forme un groupe fini vis-à-vis du produit, en d'autre termes

$$\forall a \in (\mathbb{Z}/n\mathbb{Z})^*, \exists b \in (\mathbb{Z}/n\mathbb{Z})^*, x \times b = 1[n]$$

Il est légitime dès lors de parler d'élévation à la puissance dans $(\mathbb{Z}/n\mathbb{Z})^*$.

5.2.2 Périodicité dans l'élévation à la puissance

Soit N un entier que l'on souhaite factoriser. Soit a un entier plus petit que N.

L'idée de base de l'algorithme de Shor consiste à rechercher les valeurs r telles que

$$a^r \equiv 1[N], \text{ c'est à dire } a^r = 1 \text{ dans } (\mathbb{Z}/n\mathbb{Z})^*$$

Décrivons la fonction f_a suivante :

$$f_a : (\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z}^*), f_a : x \mapsto a^x$$

Si r est tel que $a^r \equiv 1[N]$ alors pour tout p, on a $a^{r+p} = a^r.a^p = a^p$ ce qui revient à dire que $f_a(r+p) = f_a(p)$ donc que f_a admet une période r.

Inversement, si on découvre une valeur r telle que f_a est$r$-périodique, alors on a trouvé une valeur r telle que $a^r \equiv 1[N]$.

Si cette valeur r est impaire, on ne peut rien en faire, mais si r est paire alors on peut trivialement trouver p tel que $r = 2p$. Par ailleurs, le lecteur sait depuis longtemps que $x^2 - 1 = (x-1)(x+1)$, par conséquent

$$\text{Si } a^r \equiv 1[N], r = 2p, a^{2p} - 1 = (a^p - 1)(a^p + 1) = 0 \text{ dans } (\mathbb{Z}/n\mathbb{Z})^*$$

Si l'on sort de $(\mathbb{Z}/n\mathbb{Z})^*$, cela signifie que

$$\exists k \in \mathbb{N}, (a^p - 1)(a^p + 1) = kN$$

Cela signifie que les facteurs premiers de N sont répartis entre $a^p - 1$ et $a^p + 1$.

Calculer le PGCD [2] est trivial, par exemple en utilisant l'algorithme d'Euclide[8]. Il suffit dès que de calculer $PGCD(a^p - 1, N)$ et $PGCD(a^p + 1, N)$ pour trouver les facteurs recherchés.

5.2.3 Exemple trivial mais concret

Supposons que 15 soit un nombre très difficile à factoriser, et appliquons la recette précédente. On a donc $N = 15$.

Prenons un nombre a inférieur 15, par exemple $a = 7$, et évaluons $f(a) = 7^a[15]$.

a	7^a	f(a)
1	7	7
2	49	4
3	343	13
4	2401	1

On s'arrête ici car $7^4 = 1[15]$, et en plus 4 est le double de 2 ! Vous noterez que l'on fai ici des mathématiques très complexes !

Dès lors, on calcule $PGCD(7^2+1, 15) = PGCD(50, 15) = 5$ et $PGCD(7^2 - 1, 15) = PGCD(48, 15) = 3$.

On peut donc conclure sur ce résultat exceptionnel, on sait maintenant que $15 = 3 \times 5$!

2. Plus Grand Commun Diviseur

5.2.4 En bref

La complexité arithmétique s'arrête là. Il suffit de trouver un entier a et une période r permet dans la fonction qui élève a à la puissance x pour factoriser N. On choisira a au hasard, mais des calculs simples montre qu'on peut se limiter aux valeurs inférieures à $\sqrt{N}$.

5.3 Algorithme de Simon

Daniel Simon a décrit l'algorithme qui porte son nom en 1997[53]. Celui-ci permet de déterminer la période d'une fonction booléenne périodique. Cet algorithme exploite une nouvelle idée, à savoir l'effondrement causé par la mesure d'un état.

L'algorithme de Simon revêt une grande importance car il a servi d'inspiration à l'algorithme de Shor.

Il s'agit ici de résoudre le problème suivant : soit $f : \{0,1\}^n \to \{0,1\}^m$, la fonction f est périodique, c'est-à-dire qu'il existe une chaine binaire $s \in \{0,1^n_\}$ telle que $\forall x, f(x \oplus s) = f(x)$. On cherche à déterminer la valeur de s.

Remarque préalable : une fonction périodique booléenne qui a N arguments possible prend exactement N valeurs. En effet, si f est L-périodique, alors $f(x) = f(x \oplus L) = f(x \oplus L \oplus L)$.

Ici on opère avec l'opérateur XOR (c'est le sens du symbole $\oplus$), donc $x \oplus L \oplus L = x$, on revient à son point de départ en appliquant deux fois la période. On ne peut avoir plus de $N/2$ valeurs. Par ailleurs, si j'ai moins de $N/2$ valeurs alors je vais pouvoir trouver un $L' \neq Letunx$ tels que $f(x \oplus L') = f(x)$ et la fonction est à la fois L-périodque et L'-périodique, ce qui n'a pas de sens.

5.3.1 Approche classique

Si l'on a n bits, alors le cardinal de $\{0,1\}^n$ est 2^n. Il faut faire des tirage de x et de calcul de $f(x)$ et s'arrêter lorsque l'on voit pour la seconde fois une valeur $f(x)$ que l'on a déjà vue. Dans le cas le plus défavorable, il faudra effectuer $2n - 1 + 1$ tirages pour être sûr de faire un tel tirage et de deviner ainsi la période.

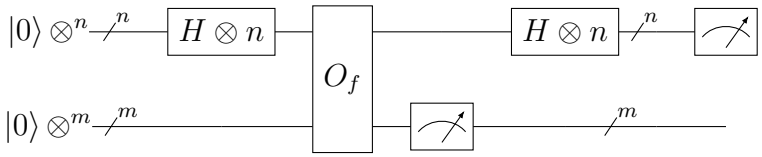


FIGURE 5.1 – Circuit implémentant l'algorithme de Simon

5.3.2 Approche quantique

Le circuit de l'algorithme de Simon est décrit par la figure 5.1. J La bordée initiale de portes H va produire une superposition uniforme sur les n qubits de données. On aura alors l'état suivant :

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \, |0\rangle^{\otimes m}$$

Lorsqu'on applique l'oracle, les n qubits de données sont inchangés, mais les m ancillae portent les valeurs de retour. On a alors l'état suivant :

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \, |f(x)\rangle$$

C'est ici que réside toute l'astuce de l'algorithme de Simon : on va venir faire une mesure des m ancillae. Ce faisant on provoque l'état de la superposition vers un état compatible avec cette mesure. Cela revient à choisir un état $|f(a)\rangle$. Or, puisque la fonction f est s-périodique, seuls deux valeurs possible de x font produire $f(a)$, c'est à dire a et $a + s$. L'état global du système devient alors

$$\frac{1}{\sqrt{2}}(|a\rangle + |a + s\rangle) \, |f(a)\rangle)$$

L'action des n portes H sur les n qubits du haut (cf 4.3.2 va donner l'état final suivant (en utilisant la notation de produit pointé que nous avons alors introduite) :

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} ((-1)^{a \cdot y} + (-1)^{(a \oplus s) \cdot y}) \, |y\rangle$$

que nous pouvons réécrire

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{a \cdot y} (1 + (-1)^{s \cdot y}) \, |y\rangle$$

Prenons le temps d'analyser ce résultat :

— si $s \cdot y = 1$, on a ajoute 1 et -1 et les contributions s'annulent ;

— si $s \cdot y = 0$, on ajoute les contributions qui subsistent.

Il ne subsiste dans l'état que les composantes telles que $s \cdot y = 0$. Quand on vient faire al dernière étape, qui consiste à mesurer les n premiers qubits, on va trouver une valeur qui vérifie la propriété d'avoir un produit pointé nul avec la période s recherchée. On ne connaît pas encore s.

À ce stade, on doit effectuer plusieurs k tirages pour construire un système d'équations linéaires :

$$\begin{cases} y_1 \cdot s = 0 \\ y_2 \cdot s = 0 \\ \vdots \\ y_k \cdot s = 0 \end{cases}$$

On peut démontrer qu'avec $n - 1$ tirages, la probabilité d'avoir un système d'équations indépendantes, qui permette de trouver la valeur de s, est de 25%. En réalisant au plus $4n$ tirages, on est sûr d'avoir un système qui permette de connaître s.

5.3.3 Avantage quantique dans l'algorithme de Simon

L'approche classique suppose d'invoquer 2^{n-1} fois la fonction f quand l'approche quantique requiert au plus $4n$ appels. On a d'un côté un temps exponentiel et de l'autre un temps linéaire. Le gain est clair, surtout si on imagine qu'il est difficile, ou lent, ou coûteux, d'utiliser la fonction f. On observe, sur l'algorithme de Simon un réel avantage quantique.

Une autre considération globale est à comprendre également : les algorithmes quantiques ne permettent pas de faire 2^n actions à la fois, puisque la mesure ne permet de ne voir qu'un seul état. En revanche, un algorithme quantique peut être vu, du point de vue physique, comme une expérience d'interférences dans la physique quantique. Les résultats souhaités subissent des interférences constructives quand les autres sont détruits par des interférences destructives. Les algorithmes quantiques, via la superposition quantique, agissent sur l'ensemble des valeurs possibles, ce qui les rend efficaces pour détecter des propriétés globales. Dans le cas de l'algorithme de Simon, cette propriété globale est la période d'une fonction booléenne.

5.4 Implémentation de l'algorithme de Shor

L'algorithme de Shor est dans l'esprit très voisin de l'algorithme de Simon, on utilisera en particulier une étape de mesure destinée à faire s'effondrer un état superposé que l'on traitera ensuite.

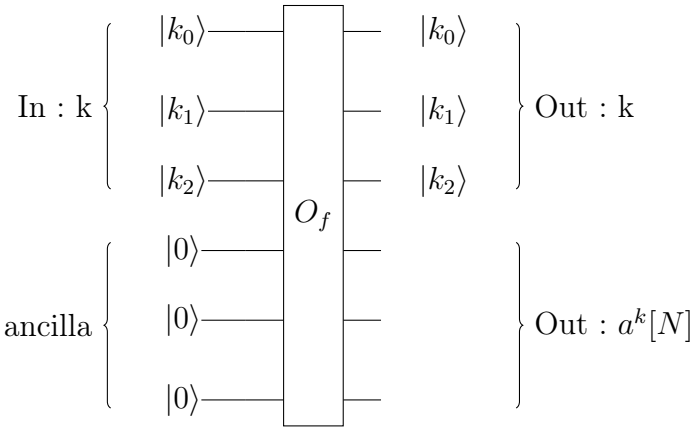
Dans toute cette partie, pour alléger les schémas, je ne représente des encodages que sur 3 qubits.

On souhaite factoriser l'entier N, on dispose de n qubits tels que $2^n \geq N$, soit a un entier inférieur à N. Soit f la fonction

$$f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z}, k \mapsto a^k \text{ (modulo N)}$$

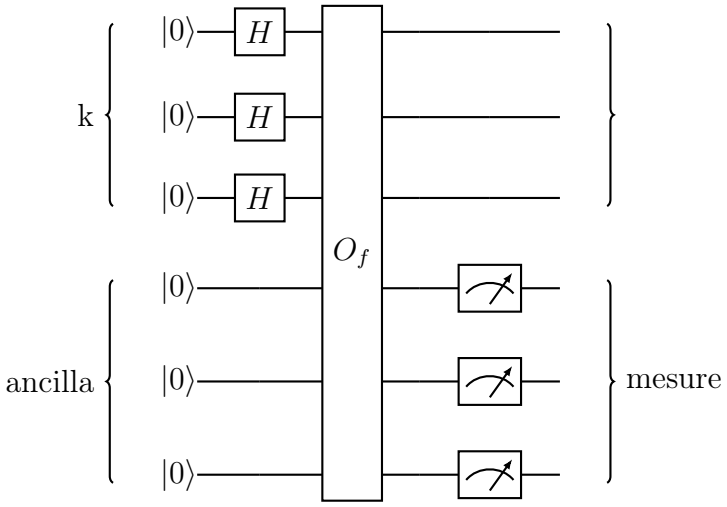
On dispose de l'oracle suivant, qui implémente la fonction f. Les qubits d'entrées représentent la valeur k sous forme binaire, ses bits sont k_0, k_1, ..., k_p.

L'action de l'oracle transforme $|k\rangle \otimes |0\rangle$ en $|k\rangle \otimes |a^k\rangle$



5.4.1 Analyse du circuit

On commence à construire le circuit de manière analogue à celui qui implémente l'algorithme de Simon (cf 5.3)



L'état initial est $|\phi_0\rangle = |0\cdots0\rangle \otimes |0\cdots0\rangle$, l'application de n portes de Hadamard va le changer en $|\phi_1\rangle$

$$|\phi_1\rangle = (H^{\otimes n} \otimes I)\,|\phi_0\rangle = (H^{\otimes n} \otimes I) \times (|0\rangle \otimes |0\rangle) = (\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |0\rangle$$

Si on applique l'oracle qui implémente f, on obtient alors $|\phi_2\rangle$

$$|\phi_2\rangle = O_f((\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |0\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |a^k\rangle$$

Il est simple et rapide de vérifier que a est premier avec n, il suffit pour cela d'utiliser par exemple l'algorithme d'Euclide[8]. À cette étape, si a n'est pas premier avec n, on a eu un coup de chance prodigieux en choisissant a puisqu'il possède des facteurs premiers communs avec n ! Si n est le produit de deux grands nombres premiers, on vient de trouver l'un d'entre eux.

On peut faire en sorte que f admette une période r. En effet, si on choisit a dans $(\mathbb{Z}/n\mathbb{Z})*$, comme cet ensemble est fini, on va trouver deux entiers tels que $a^p \equiv a^{p+q}$, donc $a^p \equiv 1$, et on ne peut pas itérer plus de fois que le cardinal de $(\mathbb{Z}/n\mathbb{Z})*$, qui est inférieur à n. Donc la période cherchée est forcément inférieure à l'entier que l'on peut factoriser.

$$k = pr + q, 0 \leq q < r$$

La partie $|k\rangle \otimes |a^k\rangle$ de la somme peut s'écrire ainsi, attendu que $a^r \equiv 1[N]$

$$|k\rangle \otimes |a^k\rangle = |pr+q\rangle \otimes |a^{pr+q}\rangle = |pr+q\rangle \otimes |a^q\rangle$$

On peut dès lors réécrire $|\phi_3\rangle$

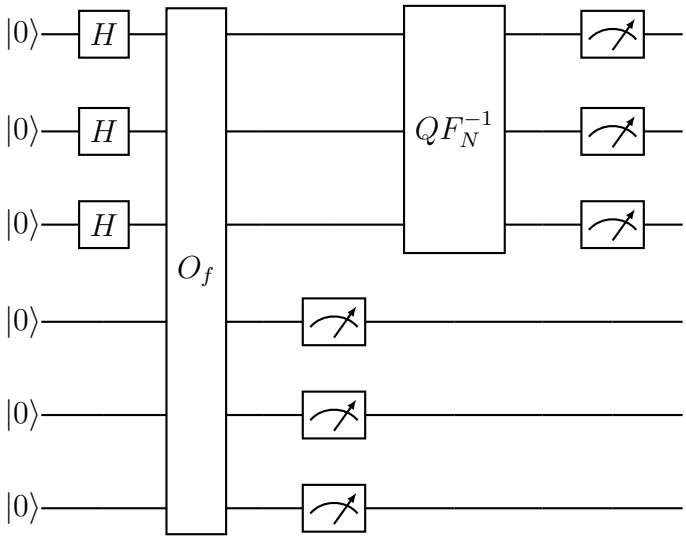
$$\begin{aligned}
|\phi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |a^k\rangle &= \frac{1}{\sqrt{2^n}} \sum_{q=0}^{r-1} \sum_{p=0}^{(2^n/r)-1} |pr+q\rangle \otimes |a^{pr+q}\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{q=0}^{r-1} \sum_{p=0}^{(2^n/r)-1} |pr+q\rangle \otimes |a^q\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{q=0}^{r-1} (\sum_{p=0}^{(2^n/r)-1} |pr+q\rangle) \otimes |a^q\rangle
\end{aligned}$$

On effectue alors une mesure des qubits ancillaires. Ils vont s'effondrer sur un état qui correspond à un certain q_0 dont on sait simplement qu'il est inférieur à r. L'état $|\phi_3\rangle$ s'effondre sur

$$|\phi_4\rangle = \frac{\sqrt{r}}{\sqrt{2^n}} (\sum_{p=0}^{(2^n/r)-1} |pr+q_0\rangle) \otimes |a^{q_0}\rangle$$

La mesure des premiers qubits permet de connaître l'une des valeurs $|pr+q_0\rangle$. Toutefois, on connaît juste a_0^q, donc on ne peut pas décider avec certitude de la valeur de q_0, on ne connaît pas r (c'est la valeur que l'on cherche), on ne peut pas déterminer p. À ce stade, on ne dispose de rien d'utile.

C'est ici que l'on fait intervenir une QFT inverse. On complète le schéma du circuit ainsi :



On rappelle (paragraphe précédent) que

$$QF_N \times |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jk} |j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2i\pi \frac{jk}{N}} |j\rangle$$

Donc

$$QF_N^{-1} \times |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{-jk} |j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-2i\pi \frac{jk}{N}} |j\rangle$$

Avant la QFT inverse, on a l'état $|\phi_4\rangle$

$$|\phi_4\rangle = \frac{\sqrt{r}}{\sqrt{2^n}} (\sum_{p=0}^{(2^n/r)-1} |pr + q_0\rangle) \otimes |a^{q_0}\rangle$$

La QFT inverse va produire un état $|\phi_5\rangle$

$$|\phi_5\rangle = (QF_N^{-1} \otimes I) \times |\phi_4\rangle = (QF_N^{-1} \otimes I) \times (\frac{\sqrt{r}}{\sqrt{2^n}}(\sum_{p=0}^{(2^n/r)-1} |pr + q_0\rangle) \otimes |a^{q_0}\rangle$$

$$= \frac{\sqrt{r}}{\sqrt{2^n}}(\sum_{p=0}^{(2^n/r)-1} QF_N^{-1} \times |pr + q_0\rangle) \otimes |a^{q_0}\rangle$$

$$= \frac{\sqrt{r}}{\sqrt{2^n}}(\sum_{p=0}^{(2^n/r)-1} \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} e^{-2i\pi \frac{j(pr+q_0)}{N}} |j\rangle) \otimes |a^{q_0}\rangle$$

$$= \frac{\sqrt{r}}{2^n}(\sum_{j=0}^{N-1} \sum_{p=0}^{(2^n/r)-1} e^{-2i\pi \frac{j(pr+q_0)}{N}} |j\rangle) \otimes |a^{q_0}\rangle$$

$$= \frac{\sqrt{r}}{2^n}(\sum_{j=0}^{N-1} (\sum_{p=0}^{(2^n/r)-1} e^{-2i\pi p \frac{jr}{2^n}}) e^{-2i\pi \frac{q_0 j}{2^n}} |j\rangle) \otimes |a^{q_0}\rangle$$

La même situation que dans l'algorithme de Bernstein-Vazirini(4.4 se produit ici. La somme a l'air complexe, mais en fait quantité de ses termes sont en fait nuls. Les physiciens vont parler d'interférences destructives.

Rappelez-vous du calcul effectué en section 1.5.3 pour montrer que la matrice de Vandermonde-Fourier est unitaire à une normalisation près et intéressons-nous au terme $e^{-2ip\pi \frac{jr}{2^n}}$) et à la somme de ces termes :

— si $\frac{jr}{2^n}$ est un entier, on élève $e^{-2i\pi} = 1$ à la puissance et on fait en fait une somme de 1 ;

— si $\frac{jr}{2^n}$ n'est pas un entier, on a une série géométrique de raison $e^{-2i\pi \frac{jr}{2^n}}$)

Or, si $k \neq 1$, on rappelle que $1+k+k^2+k^3+\cdots+k^{N-1} = \sum_{p=0}^{N-1} k^p = \frac{1-k^N}{1-k}$

On a donc deux cas. Dans le premier $\frac{jr}{2^n}$ est un entier et

$$\sum_{p=0}^{(2^n/r)-1} e^{-2i\pi p \frac{jr}{2^n}} = \sum_{p=0}^{(2^n/r)-1} 1 = \frac{2^n}{r}$$

Dans le second cas $\frac{jr}{2^n}$ n'est pas un entier et

$$\sum_{p=0}^{(2^n/r)-1} e^{-2i\pi p \frac{jr}{2^n}} = \frac{1 - e^{-2i\pi \frac{jr}{2^n} \frac{2^n}{r}}}{1 - e^{-2i\pi \frac{jr}{2^n}}} = \frac{1 - e^{-2i\pi j}}{1 - e^{-2i\pi \frac{jr}{2^n}}} = \frac{1-1}{1 - e^{-2i\pi \frac{jr}{2^n}}} = 0$$

Par conséquent, dans la décomposition de $|\phi_5\rangle$, seuls sont à considérer les termes tels que $\frac{jr}{2^n}$ est un nombre entier. Cela simplifie énormément l'écriture

$$\begin{aligned}
|\phi_5\rangle &= \frac{\sqrt{r}}{2^n}(\sum_{j=0}^{N-1}(\sum_{p=0}^{(2^n/r)-1} e^{-2i\pi p\frac{jr}{2^n}})e^{-2i\pi\frac{q_0 j}{2^n}}|j\rangle)\otimes|a^{q_0}\rangle \\
&= \frac{\sqrt{r}}{2^n}(\sum_{j=0,\frac{jr}{2^n}\in\mathbb{N}}\frac{2^n}{r}e^{-2i\pi\frac{q_0 j}{2^n}}|j\rangle)\otimes|a^{q_0}\rangle \\
&= \frac{1}{\sqrt{r}}(\sum_{j=0,\frac{jr}{2^n}\in\mathbb{N}} e^{-2i\pi\frac{q_0 j}{2^n}}|j\rangle)\otimes|a^{q_0}\rangle
\end{aligned}$$

On va faire ici une hypothèse simplificatrice sur laquelle nous reviendrons ensuite car elle n'est pas toujours réalisée : on va émettre l'hypothèse que la période r que l'on cherche divise 2^n. Dans ce cas, de 0 à $2^n - 1$, il y a r fois où $\frac{jr}{2^n}$ est entier, on change l'index précédent pour écrire l'état de la manière suivante

$$\begin{aligned}
|\phi_5\rangle &= \frac{1}{\sqrt{r}}(\sum_{j=0,\frac{jr}{2^n}\in\mathbb{N}} e^{-2i\pi\frac{q_0 j}{2^n}}|j\rangle)\otimes|a^{q_0}\rangle \\
&= \frac{1}{\sqrt{r}}(\sum_{l=0}^{r-1} e^{-2i\pi q_0\frac{l}{r}}|\frac{2^n l}{r}\rangle)\otimes|a^{q_0}\rangle
\end{aligned}$$

La mesure qui suit la QFT inverse va faire s'effondrer la superposition d'état vers l'un des états de bases qui la compose, on va voir un certain état $|\phi_6\rangle = |\frac{2^n l_0}{r}\rangle\otimes|a^{q_0}\rangle$

Le circuit nous permet donc de connaître une valeur entière $\frac{2^n l_0}{r}$, mais pas encore la valeur de la période r que nous cherchons.

5.4.2 Récupération de la période

La suite est avant tout une affaire d'arithmétique. l'étape précédente permet de connaître $\frac{2^n l}{r}$, comment déterminer r ?

Ce n'est pas aussi simple qu'il y parait : supposons, après avoir divisé par 2^n que l'on obtienne 0.5, doit-on en conclure qu'il s'agit de la fraction $1/2$ (donc $l = 1, r = 2$), de $2/4$ (donc $l = 2, r = 4$) ou $4/8$ (donc $l = 4, r = 8$) ?

Notons qu'il est toujours possible, d'écrire l/r comme une fraction. En effet, on a mesuré un entier $m = \frac{2^n l}{r}$, on sait donc que $l/r = m/2^n$. On sait donc écrire l/r de manière irréductible. Peut on pour autant conclure ?

Ce n'est pas aussi simple qu'il y parait : supposons, après avoir divisé par 2^n que l'on obtienne 0.5, doit-on en conclure qu'il s'agit de la fraction $1/2$ (donc $l = 1, r = 2$), de $2/4$ (donc $l = 2, r = 4$) ou $4/8$ (donc $l = 4, r = 8$) ?

À de rares cas près, il sera indispensable de lancer le circuit plusieurs fois et d'obtenir une série de couples (l_i, r_j). Si l'on découvre un couple, disons $(l_1, r1)$ et (l_2, r_2) tels que $r_1 = r_2$ et si l_1 et l_2 sont premiers entre eux alors on peut conclure sur la valeur de r.

Par exemple, supposons qu'on ait échantillonné $3/8$, $1/2$, $7/8$. Les couples (3,8) et (7,8) nous permettent de conclure que r = 8.

Comme pour l'algorithme de Simon, il faut faire plusieurs runs du circuit pour conclure. On peut démontrer qu'il y a une probabilité supérieure à 5% d'avoir deux l_i premiers entre eux, on est donc certain de trouver deux valeurs qui permettent de conclure avec 20 tirage

5.4.3 Retour sur l'hypothèse simplificatrice

Le diable est dans les détails. Dans le calcul précédent, on a formulé l'hypothèse que r divise 2^n. Il est évident que cela ne sera pas toujours le cas. Que se passe-t-il dans ce cas ?

Si l'on reprend l'équation 5.4.1, les choses seront sensiblement les mêmes. Si r divise 2^n, alors chaque cas $|\frac{2^n l_0}{r}\rangle$ sera $2^n/r$ fois. Dans le cas où r ne divise pas 2^n, elles seront vues soit $\lfloor 2^n/r \rfloor$ fois, soit ou $\lceil 2^n/r \rceil$ fois [3]. Les occurrences des valeurs possibles suivent une distribution qui ressemble à la figure 5.2 qui correspond au cas N=512 (9 qubits) et la période cherchée égale à 6.

Considérons le cas N=512, r=6 pour illustrer la suite. La mesure sera s=427 pour une factorisation de l'entier M=21.

Soit $s = \frac{2^n l_0}{r}$, la mesure effectuée, il est possible de trouver un entier d tel que

$$|s.r - d.N| \leq \frac{r}{2}$$

3. Les notations $\lfloor x \rfloor$ et $\lceil x \rceil$ désignent les fonctions plancher et plafond, c'est-à-dire les entiers inférieur et supérieur les plus proches de x

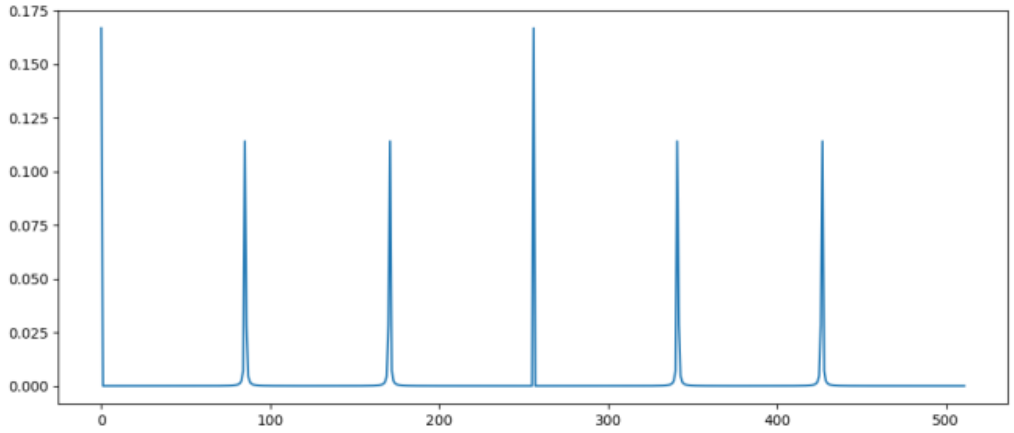


FIGURE 5.2 – Recherche de période dans l'algorithme de Shor

et donc

$$|\frac{s}{N} - \frac{d}{R}| \leq \frac{1}{2N} \leq \frac{1}{M^2}$$

Il suffit de trouver deux nombres rationnels assez proches, pour ce faire, nous allons utiliser les fractions continues décrites à la section 1.6.

Dans notre exemple, on a mesure $s = \frac{247}{512}$, on peut écrire

$$\frac{427}{512} = 0 + \frac{1}{\frac{512}{427}} = 0 + \frac{1}{1 + \frac{85}{427}}$$

Mais

$$\frac{85}{427} = \frac{1}{\frac{427}{85}} = \frac{1}{5 + \frac{2}{85}}$$

Et ainsi de suite... On en déduit la décomposition en fractions continues $427/512 = [0\,;\, 1,\, 5,\, 42,\, 2]$. On calcule les différents n-réduites et on obtient 0, 1, 5/6, 211/253, 427/512.

On sait que la période est inférieur à M, donc à 21 dans notre cas, on retient donc l'approximation 5/6, la dernière à avoir un dénominateur acceptable comme valeur de période (les autres dénominateurs sont plus grands que M=21).

On procède ensuite de la même manière que dans le cas précédent : on réalise plusieurs tirages jusqu'à avoir deux fractions qui ont le même dénominateur mais des numérateurs premiers entre eux.

De là, on peut connaître la valeur de la période et terminer l'algorithme.

5.4.4 Récapitulation de l'algorithme et analyse des coûts

Récapitulons les phases de l'algorithme :

1. choisir un entier a inférieur ou égale à la racine du nombre à factoriser, vérifier qu'il est bien premier avec ce dernier grâce à l'algorithme d'Euclide. S'il n'est pas premier, on est très chanceux et on a un résultat (et on s'arrête ici) [4]
2. utiliser la QFT pour trouver une période à la fonction entière $f(k) = a^k$, implémentée sous la forme d'un oracle dans le circuit quantique (avec utilisation d'Euclide et des fractions continues pour trouver la valeur de la période)
3. si cette période est impaire, on ne peut rien en faire, on abandonne et on choisit un autre a
4. si cette période est paire, alors sa moitié, notée p permet de trouver deux facteurs a^p+1 et a^p-1 dont le PGCD avec le nombre à factoriser (calculé avec l'algorithme d'Euclide) permet de trouver les facteurs recherchés.

En utilisant des ressources HPC classiques, la factorisation d'un produit de deux grands nombres n'est pas un problème polynomial, son coût croît exponentiellement avec la taille du problème.

Avec l'informatique quantique, si on note P l'entier à factoriser et $p = log(P)$, et si l'on note n le nombre de qubits impliqués, on dénombre les coûts suivants :

— choir un nombre a premier avec P via l'algorithme d'Euclide requiert $O(p^3)$ opérations ;
— la recherche de la période de la fonction $f(x) = a^x$ requiert
  — de l'ordre de $O(n)O(p^2)$ pour évaluer l'oracle qui implémente la fonction,

4. objectivement, si cela vous arrive, courrez acheter un billet de Loto...