L'algorithme de Shor : Quand la physique défie les mathématiques

Factorisation quantique d'entiers

28 octobre 2025

Table des matières

Résumé

L'algorithme de factorisation de Peter Shor, présenté en 1994, constitue l'une des avancées les plus significatives en informatique quantique. Il démontre qu'un ordinateur quantique, s'il était réalisé à grande échelle, pourrait résoudre en temps polynomial des problèmes considérés comme intraitables pour les ordinateurs classiques, notamment la factorisation d'entiers et le logarithme discret. Ces problèmes forment le fondement de la quasi-totalité de la cryptographie à clé publique moderne, en particulier le système RSA.

Cet article propose une analyse détaillée de l'algorithme de Shor. Nous commençons par contextualiser son impact en rappelant les fondements de la cryptographie RSA. Nous procédons ensuite à la réduction mathématique du problème de factorisation à un problème de recherche de période d'une fonction modulaire. C'est cette étape qui ouvre la voie au traitement quantique. Nous explorons l'architecture de l'algorithme, en se concentrant sur le parallélisme quantique, l'exponentiation modulaire et le rôle central de la Transformée de Fourier Quantique (QFT) pour extraire la périodicité via l'interférence. Finalement, nous détaillons la procédure classique de post-traitement (utilisant l'algorithme des fractions continues) pour récupérer les facteurs premiers à partir de la période mesurée, et nous concluons sur les conséquences profondes de cet algorithme pour la sécurité des communications et l'avènement de la cryptographie post-quantique.

1 Introduction : Pourquoi Shor change tout

1.1 Une promesse révolutionnaire

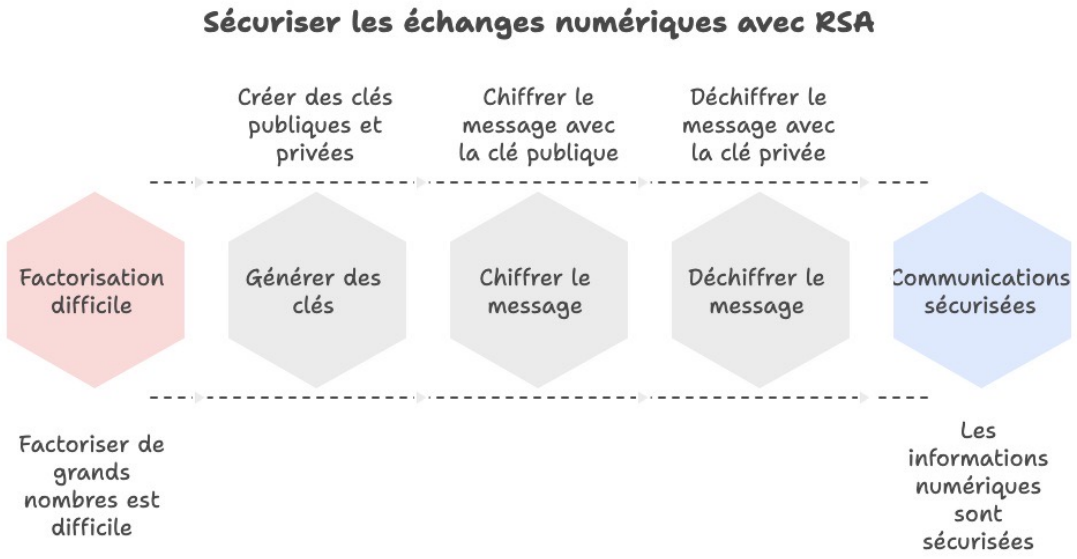
La sécurité des échanges d'informations numériques à l'ère moderne repose en grande partie sur des protocoles de cryptographie asymétrique. Le plus répandu d'entre eux, l'algorithme RSA (Rivest-Shamir-Adleman, 1977), fonde sa sécurité sur une conjecture simple : s'il est trivial de multiplier deux grands nombres premiers p et q pour obtenir leur produit $N = pq$, il est en revanche algorithmiquement très difficile, pour un ordinateur classique, de retrouver p et q à partir de N.

Formellement, la clé publique (N, e) permet de chiffrer un message M par $C \equiv M^e \pmod{N}$. Le déchiffrement $M \equiv C^d \pmod{N}$ nécessite la clé privée d, qui est l'inverse multiplicatif de e modulo $\varphi(N)$, où $\varphi(N) = (p-1)(q-1)$ est l'indicatrice d'Euler. La connaissance de d est donc conditionnée à la connaissance de $\varphi(N)$, qui est elle-même conditionnée à la factorisation de N.

La meilleure méthode classique connue pour factoriser N est le *Crible Général de Corps de Nombres* (GNFS), dont la complexité est sous-exponentielle, estimée en

$$O\left(\exp\left(c(\log N)^{1/3}(\log\log N)^{2/3}\right)\right).$$

Pour des clés de 2048 bits (environ 617 chiffres décimaux, soit $\approx 10^{616}$), le temps de calcul requis se chiffre en milliers d'années, voire des milliards d'années selon les estimations, garantissant la sécurité de nos transactions bancaires, communications sécurisées et secrets d'État.



1.2 L'ordinateur quantique entre en scène

En 1994, Peter Shor, alors chercheur aux Bell Labs, a démontré qu'un ordinateur quantique hypothétique pourrait exécuter un algorithme capable de factoriser N en temps polynomial, plus précisément en $O((\log N)^3)$ opérations avec un nombre de qubits en $O(\log N)$. Cette transition d'une complexité sous-exponentielle à une complexité polynomiale représente un changement de paradigme fondamental. Un problème jugé intraitable, pierre angulaire de notre sécurité, deviendrait trivial pour une machine exploitant les lois de la mécanique quantique. Ce saut exponentiel remet en question la sécurité de RSA à long terme.

Le qubit : fondement du calcul quantique

Le concept de **qubit** (bit quantique) est central : c'est un vecteur dans un espace de dimension 2 (base $\{|0\rangle, |1\rangle\}$). Un état pur de qubit s'interprète géométriquement sur la *sphère de Bloch*, où chaque point de surface correspond à un état superposé :

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$$

Un registre de n qubits peut simultanément exister dans une superposition cohérente de 2^n états de la base de calcul $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$. C'est le principe de **superposition quantique**.

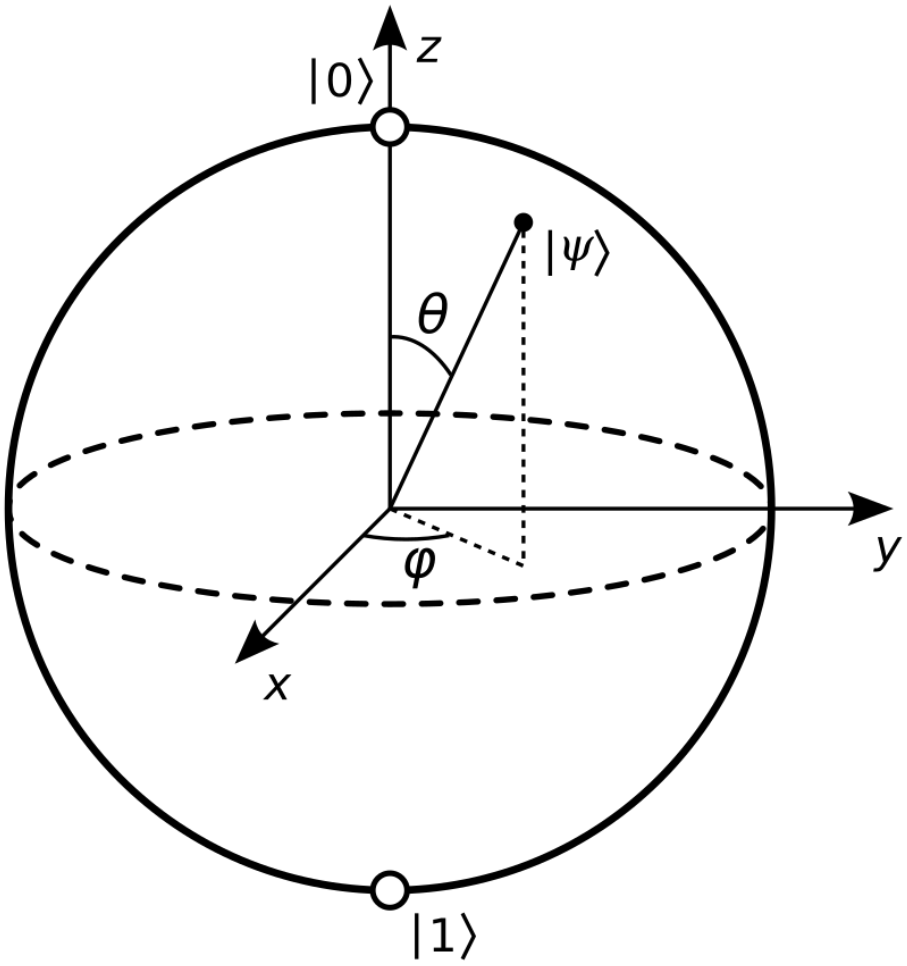


FIGURE 1 – Représentation d'un qubit sur la sphère de Bloch

La question centrale qui émerge est la suivante : *comment une machine, en exploitant des principes physiques tels que la superposition et l'intrication, peut-elle résoudre un problème de théorie des nombres purement abstrait, là où l'arsenal algorithmique classique échoue ?*

Pour comprendre, il faut d'abord disséquer le problème mathématique et le reformuler. L'astuce de Shor n'est pas d'attaquer la factorisation de front, mais de la réduire à un problème structurel différent : la recherche d'une période cachée. C'est dans la résolution de ce second problème que la physique quantique offre un avantage exponentiel.

2 Le problème mathématique : factoriser un nombre

2.1 Le défi de la factorisation

La difficulté de la factorisation est non-intuitive car elle est facile pour de petits nombres. Par exemple, $15 = 3 \times 5$ est immédiat. 143 demande déjà plus de réflexion (test de divisibilité par 2, 3, 5, 7, 11... on trouve $143 = 11 \times 13$). Pour un nombre de 300 chiffres, le nombre d'opérations nécessaires pour le GNFS est hors de portée de toute la puissance de calcul mondiale combinée. Aucun algorithme classique polynomial (en $\log N$) n'est connu.

2.2 Motivation cryptographique

La sécurité de RSA repose sur cette difficulté. Si factoriser devenait facile, il suffirait de rompre une clé privée RSA en factorisant son modulus public N. Shor a montré qu'un ordinateur quantique effectuant son algorithme en temps polynomial briserait RSA en réduisant drastiquement la complexité de factorisation.

2.3 Changer de point de vue : la périodicité cachée

L'algorithme de Shor ne cherche pas les facteurs p et q directement. Il exploite une réduction du problème de la factorisation à celui de la recherche de l'ordre d'un élément dans un groupe multiplicatif.

Théorème 2.1 (Réduction à la recherche de période)**.** *Soit N un entier composite à factoriser.*

1. *Choisir un entier aléatoire $a < N$ tel que $\gcd(a, N) = 1$. (Si $\gcd(a, N) > 1$, alors ce PGCD est un facteur non trivial de N, et nous avons terminé).*
2. *Considérer la fonction $f : \mathbb{Z} \to \mathbb{Z}_N$ définie par*
$$f(x) = a^x \pmod N.$$
3. *Cette fonction est périodique. Soit $r = ord_N(a)$ le plus petit entier positif tel que $f(x+r) = f(x)$, c'est-à-dire $a^r \equiv 1 \pmod N$. Cet r est l'ordre de a modulo N.*
4. *Si r est pair et $a^{r/2} \not\equiv -1 \pmod N$, alors N partage des facteurs non triviaux avec $a^{r/2} - 1$ et $a^{r/2} + 1$.*

Justification de (4). Si r est la période, nous avons $a^r \equiv 1 \pmod N$, soit $a^r - 1 \equiv 0 \pmod N$. Si r est pair, nous pouvons écrire $(a^{r/2})^2 - 1 \equiv 0 \pmod N$, ce qui se factorise en :

$$(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \pmod N$$

Cela signifie que N divise le produit $(a^{r/2} - 1)(a^{r/2} + 1)$. Si :

— $a^{r/2} \not\equiv 1 \pmod N$ (ce qui est garanti car r est la plus petite période), et

— $a^{r/2} \not\equiv -1 \pmod N$ (ce que nous posons comme condition),

alors N ne divise aucun des deux termes du produit individuellement. Par conséquent, N doit partager un facteur premier avec chaque terme. Les facteurs p et q de N sont alors retrouvés (avec une forte probabilité) en calculant :

$$p = \gcd(a^{r/2} - 1, N) \quad \text{et} \quad q = \gcd(a^{r/2} + 1, N)$$

On pourra vérifier facilement que ces facteurs se multiplient pour donner N. En pratique, on répète le choix d'un a aléatoire si r est impair ou donne des facteurs triviaux.

L'idée clé : le problème de factorisation, un problème de recherche, est transformé en un problème de recherche de période. C'est ce dernier que l'ordinateur quantique va résoudre. Cette étape de recherche de période est la clé : Shor fait essentiellement une *quantum phase estimation* pour extraire r via la transformée de Fourier quantique (QFT).

3 L'intuition quantique : la force du parallélisme

3.1 Le calcul classique : une boucle après l'autre

Comment trouver r classiquement ? Il n'y a pas de méthode efficace connue. L'approche naïve consiste à calculer $f(1), f(2), f(3), \ldots$ jusqu'à trouver r tel que $f(r) = 1$. Dans le pire des cas, r peut être de l'ordre de N, rendant cette recherche aussi longue que la factorisation elle-même.

Par exemple, avec $a = 7$ et $N = 15$, la fonction $f(x) = 7^x \pmod{15}$ donne la suite de valeurs $(1, 7, 4, 13, 1, 7, 4, 13, \ldots)$ qui se répètent avec une période $r = 4$.

3.2 Le calcul quantique : toutes les valeurs à la fois

L'ordinateur quantique utilise le principe de **superposition**. L'algorithme de Shor utilise deux registres quantiques :

1. Un **registre d'entrée** (ou registre de contrôle) de n qubits (où $2^n \geq N^2$ pour une précision suffisante), initialisé à $|0\rangle^{\otimes n}$.
2. Un **registre de sortie** (ou registre de cible) de m qubits (où $m \approx \log_2 N$), initialisé à $|0\rangle^{\otimes m}$ (ou parfois $|1\rangle$).

L'état initial du système est $|\Psi_0\rangle = |0\rangle_n |0\rangle_m$.

Étape 1 : Superposition

On applique une porte d'Hadamard H à chaque qubit du premier registre. La transformation $H^{\otimes n}$ crée une superposition uniforme de tous les états d'entrée possibles :

$$|\Psi_1\rangle = (H^{\otimes n}|0\rangle_n)|0\rangle_m = \left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\right) |0\rangle_m$$

Étape 2 : Oracle (Exponentiation modulaire)

C'est le cœur du calcul. On construit un circuit quantique U_f qui implémente la fonction $f(x) = a^x \pmod{N}$. Cet opérateur agit sur les deux registres :

$$U_f |x\rangle_n |y\rangle_m = |x\rangle_n |y \oplus f(x)\rangle_m$$

où $\oplus$ est typiquement une addition modulaire.

On applique l'opération U_a qui multiplie par a modulo N : $U_a|y\rangle = |a\,y \bmod N\rangle$ de façon contrôlée en fonction des bits de x (cela équivaut à $U_a^{2^j}$ selon la décomposition binaire de x). Appliqué à notre état superposé, U_f calcule $f(x)$ pour toutes les 2^n valeurs de x simultanément :

$$|\Psi_2\rangle = U_f|\Psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n |a^x \pmod{N}\rangle_m$$

Cet état $|\Psi_2\rangle$ est un état massivement intriqué contenant, en superposition, l'ensemble des paires $(x, f(x))$. C'est le **parallélisme quantique**.

Analogie musicale

L'image est celle d'un orchestre jouant toutes les notes (tous les x) en même temps. Le son résultant (l'état intriqué) contient toutes les informations, mais de manière superposée. Il nous faut maintenant un moyen d'extraire la fréquence dominante, *id est* la période r.

4 Trouver la période avec la Transformée de Fourier Quantique (QFT)

4.1 Une analogie musicale

En analyse de signal classique, la Transformée de Fourier Discrète (TFD) est un outil qui décompose un signal complexe (temporel) en ses fréquences fondamentales (spectre fréquentiel). Si un son contient une note pure, la TFD produira un pic à la fréquence de cette note.

La **Transformée de Fourier Quantique (QFT)** est l'analogue unitaire de la TFD. Elle n'agit pas sur un signal, mais sur les amplitudes d'un état quantique. Elle opère une transformation de base, passant de la base de calcul (les $|x\rangle$) à la base de Fourier (les $|k\rangle$, ou "base des fréquences") :

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i x k/2^n} |k\rangle$$

La QFT peut se réaliser en $O(n^2)$ portes quantiques (Hadamard + rotations contrôlées) pour n qubits, bien plus efficace que la DFT classique ($O(n2^n)$ opérations). L'intuition est que la QFT détecte efficacement la période r encodée dans la superposition grâce à l'interférence constructive des états correspondants.

4.2 Le cœur de Shor : extraction de la période

Reprenons à l'état $|\Psi_2\rangle$. Pour extraire la période, nous procédons en deux temps.

Étape 3 : Mesure du second registre

Nous effectuons une mesure sur le second registre (celui contenant $f(x)$). Selon les postulats de la mécanique quantique, le système s'effondre. Nous mesurons une valeur aléatoire, disons $y_0 = a^{x_0} \pmod N$. Le premier registre s'effondre corrélativement en une superposition contenant uniquement les valeurs x qui auraient produit ce résultat y_0.

En raison de la périodicité de f, ces x sont $x_0, x_0+r, x_0+2r, \ldots, x_0+Kr$ (où K est le nombre de répétitions dans l'intervalle $[0, 2^n - 1]$). L'état du premier registre devient (en ignorant la normalisation et le second registre) :

$$|\Psi_3\rangle \propto \sum_{j=0}^{K-1} |x_0 + jr\rangle$$

Nous avons isolé un état périodique pur, dont la période est r. Le registre de contrôle contient maintenant un état où la périodicité de $f(x)$ est codée dans les phases.

Étape 4 : Application de la QFT

Nous appliquons maintenant la QFT sur le premier registre, qui est dans l'état $|\Psi_3\rangle$:

$$|\Psi_4\rangle = \mathrm{QFT}|\Psi_3\rangle = \sum_{k=0}^{2^n-1} C_k |k\rangle$$

L'amplitude C_k associée à chaque état de fréquence $|k\rangle$ est la superposition des phases $e^{2\pi i(x_0+jr)k/2^n}$:

$$C_k \propto \sum_{j=0}^{K-1} e^{2\pi i(x_0+jr)k/2^n} = e^{2\pi i x_0 k/2^n} \sum_{j=0}^{K-1} \left(e^{2\pi i r k/2^n}\right)^j$$

C'est une somme géométrique. Cette somme s'annule presque partout (interférence destructive), sauf lorsque le terme de phase $e^{2\pi irk/2^n}$ est proche de 1. L'interférence est constructive si et seulement si l'exposant est proche d'un entier, c'est-à-dire :

$$\frac{rk}{2^n} \approx \text{entier} \quad \Longrightarrow \quad k \approx \frac{j \cdot 2^n}{r} \quad \text{pour } j \in \mathbb{Z}$$

Appliquer la QFT fait converger l'amplitude de probabilité vers des états correspondant à des multiples entiers de $2^n/r$. L'étape de QFT fait interfèrer ces états de façon à accentuer ceux compatibles avec la périodicité r (les « pics de Fourier »).

Étape 5 : Mesure du premier registre

En mesurant le premier registre après QFT, nous n'obtiendrons pas une valeur aléatoire. La distribution de probabilité $|\Psi_4\rangle$ est fortement piquée (pics) aux valeurs k qui sont des multiples entiers de la "fréquence fondamentale" $2^n/r$. On obtient en général un nombre y tel que $y/2^n$ est proche d'un multiple de $1/r$.

5 La partie classique : retrouver les facteurs

L'ordinateur quantique a terminé son travail. Il nous fournit une mesure k. Le reste du travail est classique.

5.1 Recalcul classique et fractions continues

Nous avons mesuré une valeur k telle que $\frac{k}{2^n} \approx \frac{j}{r}$ pour un j entier inconnu. Nous connaissons k (la mesure) et 2^n (la taille de notre registre). Nous cherchons r.

Le problème est de trouver la fraction irréductible $\frac{j}{r}$ qui approxime le mieux le nombre réel $\frac{k}{2^n}$. Ce problème est résolu efficacement par l'**algorithme des fractions continues**. Cet algorithme prend en entrée $\frac{k}{2^n}$ et renvoie la "meilleure" approximation rationnelle dont le dénominateur est inférieur à N. Ce dénominateur est notre candidat pour la période r.

Par une procédure de fractions continues classique, on déduit la valeur exacte de r (ou un de ses diviseurs). Il est possible que j et r partagent un facteur commun, auquel cas l'algorithme des fractions continues nous donnera un diviseur r' de r. Cela est géré en relançant l'algorithme si nécessaire.

5.2 Extraction des facteurs

Une fois la période r (ou un multiple) trouvée, nous vérifions les conditions :

1. r est-il pair ?
2. $a^{r/2} \not\equiv -1 \pmod{N}$?

Si les conditions sont remplies (ce qui arrive avec une probabilité $\geq 1 - 1/2^m$ où m est le nombre de facteurs premiers distincts de N), nous calculons les facteurs :

$$p = \gcd(a^{r/2} - 1, N) \quad \text{et} \quad q = \gcd(a^{r/2} + 1, N)$$

Le calcul du PGCD se fait avec l'algorithme d'Euclide en temps polynomial, très rapide (polynomial en $\log N$), par exemple avec l'algorithme d'Euclide étendu. Si p ou q donne un facteur non trivial ($\neq 1, N$), on aura décomposé $N = p \times q$. Si l'une des conditions échoue, on recommence l'ensemble du processus (quantique + classique) avec une nouvelle valeur aléatoire de a. Le nombre d'essais attendu est faible.

On vérifie alors que $p \times q = N$ (ce qui est garanti mathématiquement). Cette étape de post-traitement est purement classique et très efficace. L'ensemble quantum+classique réalise ainsi la factorisation de N.

5.3 Exemple simplifié : Factorisation de $N = 15$

Prenons $N = 15$. Factorisons-le en détail.

1. **Choix de a :** Choisissons $a = 7$ (on vérifie $\gcd(7, 15) = 1$, premier avec 15).
2. **Recherche de période :** L'ordinateur quantique cherche la période de $f(x) = 7^x \pmod{15}$. Les calculs (classiques pour cet exemple) donnent :

$$\begin{aligned} 7^0 &\equiv 1 \\ 7^1 &\equiv 7 \\ 7^2 &\equiv 49 \equiv 4 \\ 7^3 &\equiv 28 \equiv 13 \\ 7^4 &\equiv 91 \equiv 1 \end{aligned}$$

3. **Période trouvée :** La période est $r = 4$.
4. **Extraction quantique :** L'algorithme quantique (phase estimation) mesurerait un multiple de $1/r = 1/4$ en fraction binaire. L'ordinateur quantique, via la QFT, mesure un k tel que $k/2^n \approx j/4$. L'algorithme des fractions continues extrait $r = 4$.
5. **Vérification classique :**
   — $r = 4$ est pair.
   — $a^{r/2} = 7^{4/2} = 7^2 = 49 \equiv 4 \pmod{15}$
   — $4 \not\equiv -1 \pmod{15}$. Les conditions sont valides.
6. **Calcul des facteurs :**

$$\begin{aligned} \gcd(a^{r/2} - 1, N) &= \gcd(4 - 1, 15) = \gcd(3, 15) = 3 \\ \gcd(a^{r/2} + 1, N) &= \gcd(4 + 1, 15) = \gcd(5, 15) = 5 \end{aligned}$$

7. **Résultat :** Les facteurs de 15 sont 3 et 5. Ainsi on obtient $15 = 3 \times 5$.

Cet exemple a bien été démontré sur un petit prototype quantique (IBM l'a réalisé en 2001 avec 7 qubits en RMN). Des expériences ultérieures ont factorisé 21 ou 35 sur 5-7 qubits.

5.4 Circuit quantique simplifié

Schématiquement, l'algorithme s'implémente ainsi :

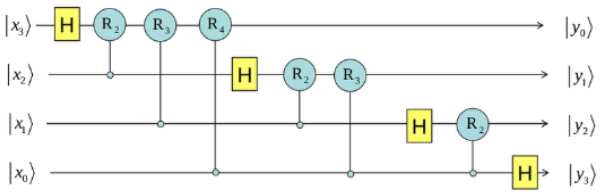


FIGURE 2 – Circuit simplifié

où H sont des portes de Hadamard et $U_a, U_a^2, \ldots$ les opérations de multiplication contrôlée par $a^{2^j} \bmod N$, suivies de la QFT inverse QFT^{-1}.

6 Défis techniques actuels

6.1 Bruit et correction d'erreurs

Les ordinateurs quantiques réels sont bruyants : chaque qubit est sujet à la décohérence, aux erreurs de porte, etc. Pour exécuter Shor à grande échelle, il faut un code de correction d'erreur quantique (par exemple code surface ou codes LDPC). Les qubits logiques (utilisés par l'algorithme) doivent être codés sur des milliers ou millions de qubits physiques pour résister au bruit.

Google et IBM estiment que pour factoriser un RSA-2048 avec fiabilité, il faudrait de l'ordre de dizaines de millions de qubits physiques, organisés en réseau 2D avec correction d'erreurs. Par exemple, en 2019 un travail a calculé qu'il faudrait $\sim$20 millions de qubits bruyants pour casser RSA-2048 en quelques heures. Récemment (2025) Google a montré qu'avec des améliorations algorithmiques et de correction, on pourrait descendre à $\sim$1 million de qubits (environ une semaine de calcul). À titre de comparaison, les machines actuelles (IBM, Google...) ont de l'ordre de centaine de qubits bruyants, soit plusieurs ordres de grandeur en dessous.

6.2 Nombre de qubits requis

La taille requise d'un ordinateur quantique pour Shor est très importante : en théorie $O(\log N)$ qubits logiques pour factoriser N, mais en pratique avec l'erreur on compte souvent $O(n)$ qubits logiques pour un nombre de n bits. Par exemple, des optimisations ont obtenu $\sim 2n + 3$ qubits logiques pour factoriser un n-bit. Avec le recouvrement en codes de correction, on retombe sur des millions de qubits physiques pour $n \sim 2048$.

6.3 Limitations matérielles

Décohérence et fidélité

Les qubits actuels perdent leur cohérence après quelques dizaines à centaines de microsecondes. L'exécution de circuits de grande profondeur (nécessaires pour Shor) est donc très fragile.

Topologie et connectivité

La plupart des architectures (supraconducteurs IBM, ions piégés, etc.) ont des contraintes de connectivité locale qui compliquent l'implantation des opérations multi-qubit requises.

Préservation de l'état quantique

Étant donné le grand nombre d'opérations à réaliser, la gestion du "stack" quantique (ordre des portes, évitement des erreurs croisées) est un défi d'ingénierie majeur.

6.4 Implémentations expérimentales

Quelques démonstrations de Shor ont été faites sur de très petits nombres :

— IBM (2001) a factorisé 15 avec 7 qubits (NMR). Des expériences ultérieures ont factorisé 21 ou 35 sur 5-7 qubits.

— Rigetti, Google et autres laboratoires ont exploré des variantes (parfois hybrides quantique-classique) pour de plus grands nombres en théorie, mais sans dépasser une factorisation complète (au-delà de 21).

7 Conséquences : quand la physique menace la cryptographie

7.1 RSA, ECC, et le problème du logarithme discret

L'algorithme de Shor ne s'applique pas seulement à la factorisation. Une structure algorithmique similaire résout également le **problème du logarithme discret (DLP)** en temps polynomial. Le DLP est le fondement de la sécurité d'autres protocoles majeurs, notamment l'échange de clés Diffie-Hellman et la cryptographie sur courbes elliptiques (ECC), cette dernière étant largement utilisée dans les appareils mobiles et les crypto-monnaies.

La construction d'un ordinateur quantique à grande échelle, tolérant aux erreurs (utilisant des milliers de qubits logiques, soit potentiellement des millions de qubits physiques), rendrait donc obsolète l'intégralité de l'infrastructure de sécurité à clé publique sur laquelle repose l'Internet moderne.

7.2 Le post-quantique

Cette menace, bien qu'encore lointaine sur le plan technologique, est prise au sérieux. La communauté cryptographique mondiale travaille activement au développement et à la standardisation de la cryptographie post-quantique (PQC).

Il s'agit d'une nouvelle famille d'algorithmes cryptographiques classiques (non quantiques), conçus pour être sécurisés contre des attaquants disposant à la fois d'ordinateurs classiques et quantiques. Ces nouveaux systèmes reposent sur des problèmes mathématiques supposés difficiles, même pour un ordinateur quantique, tels que :

- **La résolution de problèmes sur les réseaux euclidiens** (lattice) : par exemple LWE (*Learning With Errors*). CRYSTALS-Kyber pour l'échange de clés, CRYSTALS-Dilithium et FALCON pour la signature. Ce sont les "algorithmes principaux" choisis par NIST.
- **La cryptographie basée sur les codes correcteurs d'erreurs** : Classic McEliece (bien que NIST ne l'ait pas encore standardisé, on le considère très sûr) ou HQC (sélectionné en 2025 comme alternative).
- **La cryptographie multivariée** : systèmes basés sur la résolution de systèmes d'équations polynomiales multivariées.
- **La cryptographie basée sur les isogénies de courbes elliptiques** : bien que certains candidats aient été cassés récemment.
- **Basés sur le hachage** : SPHINCS+ (signature stateless) a été retenu par NIST comme sauvegarde pour la signature.

8 Conclusion : la beauté de l'union entre nombres et ondes

L'algorithme de Shor est bien plus qu'une simple menace pour la cryptographie. Il s'agit d'une démonstration profonde de la puissance du modèle de calcul quantique (la classe de complexité BQP). Il prouve qu'un phénomène physique — l'interférence constructive des amplitudes de probabilité, régie par les équations de la mécanique quantique — peut être ingénieusement orchestré pour élucider une propriété mathématique abstraite, la structure périodique d'une fonction en théorie des nombres.

L'algorithme de Shor est une percée théorique majeure : il promet une accélération exponentielle pour un problème jugé intractable classiquement. Cet algorithme nous enseigne que la complexité d'un problème n'est pas absolue, mais relative au modèle de calcul utilisé, et que les lois physiques de l'univers dictent les limites ultimes du calcul.