Algorithmes Quantiques et Factorisation des entiers

Factorisation au coeur d'algorithmes de crypto (RSA)

$$p_1 \text{ et } p_2 \; \underset{\text{difficile}}{\overset{\text{facile}}{\rightleftarrows}} \; N = p_1 p_2$$

~~Quantique : essayer toutes les divisions en même temps?~~

<u>Problème</u> : p nombre premier $p > 2$.

$\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, 2, \dots, p-1\} \ni x, y \qquad r \in \mathbb{N}. \qquad \boxed{x^r = y}$

- $x \in \mathbb{Z}_p$ et $r \in \mathbb{N} \rightsquigarrow$ calculer $y = x^r \bmod p$ — Exponentielle modulaire — facile
- $x, y \in \mathbb{Z}_p \rightsquigarrow$ calculer $r \in \mathbb{N}$ tq $x^r = y \bmod p$ — Logarithme discret — difficile

<u>Question du log discret</u> : Étant donné $x \in \mathbb{Z}_p^*$, trouver le plus petit $r \in \mathbb{N}^*$ tel que $x^r = 1 \bmod p$

__Exemple__ : $p = 13$

$x =$	1	2	3	4	5	6	7	8	9	10	11	12
$n =$	1	12	3	6	4	12	12	4	3	6	12	2

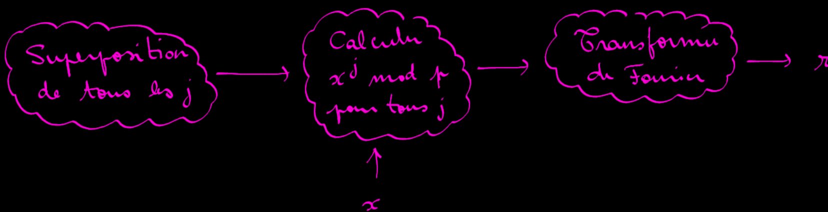
Prenons $x = 9$. On calcule toutes les puissances x^j

$j =$	0	1	2	3	4	5	6	7	8	...
$x^j =$	1	9	3	1	9	3	1	9	3	...

période 3

Idée : calculer tous les x^j pour $j = 0, 1, \dots, p-1$ en une fois à l'aide d'une superposition quantique.

Puis on détecte la période des résultats à l'aide d'une transformée de Fourier.



I) Transformée de Fourier Quantique

Soit $N \in \mathbb{N}$, $N \geq 2$. Suites $\begin{cases} x = (x_j)_{j=0,\dots,N-1} \\ y = (y_k)_{k=0,\dots,N-1} \end{cases}$ $\qquad \omega = e^{2\pi i/N}$

y est la TF de x si $\quad y_k = \dfrac{1}{\sqrt{N}} \displaystyle\sum_{j=0}^{N-1} x_j e^{2\pi i \frac{jk}{N}} = \dfrac{1}{\sqrt{N}} \sum_j x_j \omega^{jk}$

Supposons que $N = 2^n$. On peut écrire en binaire les entiers de 0 à $N-1$ avec n bits.

On considère n qubits, et on encode les entiers.

$$\left(N=4 \qquad \underset{0}{|00\rangle} \qquad \underset{1}{|01\rangle} \qquad \underset{2}{|10\rangle} \qquad \underset{3}{|11\rangle} \right)$$

L'état $|x\rangle = \sum_j x_j |j\rangle$ et $|y\rangle = \sum_k y_k |k\rangle$.

$$|y\rangle = \sum_k \frac{1}{\sqrt{N}} \sum_j x_j \, \omega^{jk} |k\rangle = \sum_j x_j \underbrace{\left(\frac{1}{\sqrt{N}} \sum_k \omega^{jk} |k\rangle\right)} = TF|x\rangle$$

$$TF|x\rangle = \sum_j x_j \underbrace{TF|j\rangle}_{|\tilde{j}\rangle} = \sum x_j |\tilde{j}\rangle$$

$$\boxed{|\tilde{j}\rangle = \frac{1}{\sqrt{N}} \sum_k \omega^{jk} |k\rangle}$$

<u>Exemple</u>

- $n=1$, $N=2$.

$\omega = -1$

$$|0\rangle \xrightarrow{TF} |\tilde{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|1\rangle \xrightarrow{TF} |\tilde{1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Porte de <u>Hadamard</u>

- $n=2 \quad N=4$

$\omega = i$

$$|0\rangle = |00\rangle \longrightarrow |\widetilde{00}\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) = \frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|0\rangle + |1\rangle\right)$$

$$|1\rangle = |01\rangle \longrightarrow |\widetilde{01}\rangle = \frac{1}{2}\left(|00\rangle + i\,|01\rangle - |10\rangle - i\,|11\rangle\right) = \frac{1}{2}\left(|0\rangle - |1\rangle\right)\left(|0\rangle + i\,|1\rangle\right)$$

$$|2\rangle = |10\rangle \longrightarrow |\widetilde{10}\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right) = \frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|0\rangle - |1\rangle\right)$$

$$|3\rangle = |11\rangle \longrightarrow |\widetilde{11}\rangle = \frac{1}{2}\left(|00\rangle - i\,|01\rangle - |10\rangle + i\,|11\rangle\right) = \frac{1}{2}\left(|0\rangle - |1\rangle\right)\left(|0\rangle - i\,|1\rangle\right)$$

Matrice :

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix}$$

$$|j\rangle = |j_1 j_2\rangle \longrightarrow |\tilde{j}\rangle = |\widetilde{j_1 j_2}\rangle$$

$$\boxed{|\tilde{j}\rangle = \frac{1}{2}\left(|0\rangle + e^{2\pi i j/2}|1\rangle\right)\left(|0\rangle + e^{2\pi i j/4}|1\rangle\right)}$$

<u>Exercice</u> : montrer que pour tout $n \geq 1$, si $j = j_1 j_2 \cdots j_n$

$$\boxed{|\tilde{j}\rangle = \frac{1}{\sqrt{N}}\left(|0\rangle + \underbrace{e^{2\pi i j/2}}_{e^{2\pi i\, 0,j_n}}|1\rangle\right)\left(|0\rangle + \underbrace{e^{2\pi i j/4}}_{e^{2\pi i\, 0,j_{n-1}j_n}}|1\rangle\right) \cdots\cdots \left(|0\rangle + \underbrace{e^{2\pi i j/2^n}}_{e^{2\pi i\, 0,j_1 \cdots j_n}}|1\rangle\right)} \quad (*)$$

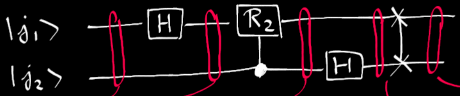
Remarque : $|\tilde{0}\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} |k\rangle$

Circuit quantique : $\underline{n=1}$: $|j_1\rangle - \boxed{H} -$

$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

$\underline{n=2}$:



$$|00\rangle \longrightarrow \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \longrightarrow \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \longrightarrow \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

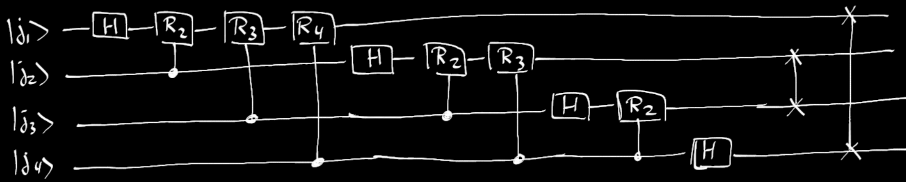
$$|01\rangle \longrightarrow \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|01\rangle + i|11\rangle) \longrightarrow \frac{1}{2}(|00\rangle - |01\rangle + i(\underline{|10\rangle} - |11\rangle)) \longrightarrow \frac{1}{2}(|00\rangle + i|01\rangle - |10\rangle - i|11\rangle)$$

$$|j_1 j_2\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_1}|1\rangle)|j_2\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle)|j_2\rangle \longrightarrow \frac{1}{2}(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle)(|0\rangle + e^{2\pi i 0.j_2}|1\rangle)$$

$$\longrightarrow \frac{1}{2}(|0\rangle + e^{2\pi i 0.j_2}|1\rangle)(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle)$$

Exercice : Montrer que pour tout n, un circuit implémentant $(*)$ est :

$(n=4)$



Complexité : # portes $\simeq 1+2+3+\dots+n \sim \boxed{O(n^2)}$ ← grâce au quantique gain exponentiel.

Algorithme classique de FFT : $O(n\,2^n)$

Notation :

transformation unitaire

↳

Dans la suite :

II) Arithmétique

$N \in \mathbb{N}$, $N \geq 2$. $(\mathbb{Z}_N, +, \times)$ anneau.

Tous les éléments non nuls ne sont pas forcément inversibles :

$N = 10$ $\quad 2 \times 5 = 10 = 0 \mod 10$ mais $2 \neq 0$ et $5 \neq 0$.

Si N est premier, $(\mathbb{Z}_N, +, \times)$ est un corps, et $\mathbb{Z}_N^* = \{1, 2, \ldots, N-1\}$.

éléments inversibles de $\mathbb{Z}_N$.

Pour N quelconque, $\boxed{\mathbb{Z}_N^* = \text{groupe des inversibles de } \mathbb{Z}_N}$

$$\boxed{\varphi(N) = |\mathbb{Z}_N^*|}$$

$$\mathbb{Z}_N^* = \{\text{entiers } 0 \leq k \leq N-1 \mid \gcd(k, N) = 1\}$$

Exemple : $N = 15$.

$(N = p_1 p_2)$

	~~0~~	1	2	3	4
0	~~0~~	~~6~~	~~12~~	~~3~~	~~9~~
1	~~10~~	1	7	13	4
2	~~5~~	11	2	8	14

$\mathbb{Z}_5^*$

$\mathbb{Z}_3^*$

C_2

C_4

$\mathbb{Z}_{15}^*$

$\varphi(15) = 8$

$C_k \simeq \mathbb{Z}/k\mathbb{Z}$

Théorème : Si $N = p_1^{\alpha_1} p_2^{\alpha_2} \cdots$ **impair**, alors $\mathbb{Z}_N^* = C_{p_1^{\alpha_1} - p_1^{\alpha_1 - 1}} \times C_{p_2^{\alpha_2} - p_2^{\alpha_2 - 1}} \times \cdots$

Exemple : $N = 15 = 3 \cdot 5$ $\quad \mathbb{Z}_N^* = C_2 \times C_4$

Définition. L'ordre de $x \in \mathbb{Z}_N^*$ est $|\{1, x, x^2, \ldots\}|$, c'est-à-dire le plus petit $r \in \mathbb{N}$, $r > 0$ tel que $x^r = 1 \mod N$

Exemple : $N = 15$

$$\mathbb{Z}_{15}^* = \left\{ \begin{matrix} 1 & 7 & 13 & 4 \\ 11 & 2 & 8 & 14 \end{matrix} \right\} \xrightarrow{r} \left\{ \begin{matrix} 1 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \end{matrix} \right\}$$

$N = 13 \qquad \mathbb{Z}_{13}^* = \{1, \ldots, 12\}$.

Générateurs de $\mathbb{Z}_{13}^*$ sont $2, 6, 7, 11$.

Problème du logarithme discret : Étant donné N et $x \in \mathbb{Z}_N^*$, trouver l'ordre r de x modulo N.

Soit p premier, $p > 2$. Soit d maximal tel que $2^d \mid p-1$.

$\mathbb{Z}_p^*$ est cyclique d'ordre $p-1$: $\mathbb{Z}_p^* = \{1, g, g^2, \dots, g^{p-2}\}$ pour un certain $g \in \mathbb{Z}_p^*$. Quel est l'ordre r de g^k $(k \in \{0, \dots, p-2\})$?

- Si k est pair, $(g^k)^{\frac{p-1}{2}} = (g^{p-1})^{k/2} = 1 \bmod p$ donc $r \mid \frac{p-1}{2}$. Donc $\boxed{2^d \nmid r}$
- Si k est impair, $g^{kr} = 1 \bmod p$ donc $p-1 \mid kr$. Comme k est impair, $2^d \mid p-1 \mid kr$ implique $\boxed{2^d \mid r}$

Donc : $$\mathbb{Z}_p^* = \underbrace{\left\{\begin{array}{l}\text{éléments d'ordre} \\ \text{divisible par } 2^d\end{array}\right\}}_{\# = \frac{p-1}{2}} \sqcup \underbrace{\left\{\begin{array}{l}\text{éléments d'ordre non} \\ \text{divisible par } 2^d\end{array}\right\}}_{\# = \frac{p-1}{2}}$$

III) Logarithme discret quantique

Soit $N \in \mathbb{N}$, $N > 2$, et $x \in \mathbb{Z}_N^*$. On veut trouver l'ordre r de x mod N.

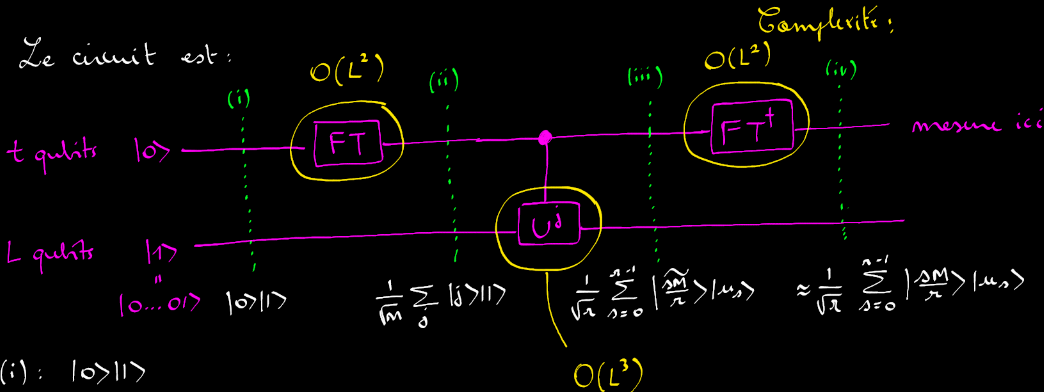
Soit $L = \lceil \log N \rceil$ le nombre de bits pour écrire N.

On définit l'opérateur unitaire $U|y\rangle = \begin{cases} |xy \bmod N\rangle & \text{si } y \leqslant N-1 \\ |y\rangle & \text{si } y \geqslant N. \end{cases}$

pour $y \in [\![0, \dots, 2^L - 1]\!]$

Soit $\varepsilon > 0$, et $t = 2L + 1 + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right) \right\rceil$ et $M = 2^t$.

Le circuit est :



(i) : $|0\rangle|1\rangle$

(ii) : $\frac{1}{\sqrt{M}}\sum_{j=0}^{M-1} |j\rangle|1\rangle$

(iii) : $\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |j\rangle |x^j \bmod N\rangle$

$$|x^j \bmod N\rangle = \sum_{k=0}^{r-1} \delta_{kj} \, |x^k \bmod N\rangle = \sum_{k=0}^{r-1} \frac{1}{r} \sum_{s=0}^{r-1} e^{2\pi i \frac{s}{r}(j-k)} |x^k \bmod N\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i \, sj/r} \underbrace{\left[\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \, sk/r} |x^k \bmod N\rangle \right]}$$

$$= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i \, sj/r} |u_s\rangle$$

$$= \frac{1}{\sqrt{M}} \frac{1}{\sqrt{r}} \sum_{j=0}^{M-1} \sum_{s=0}^{r-1} e^{2\pi i \frac{sjM}{rM}} |j\rangle |u_s\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|\widetilde{\frac{sM}{r}}\right\rangle |u_s\rangle$$

⚠ En général, $\frac{sM}{r} \notin \mathbb{N}$!

$$\left|\widetilde{\frac{sM}{r}}\right\rangle = \frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} e^{2\pi i \frac{sjM}{rM}} |j\rangle$$

En mesurant le 1er registre, on obtient une valeur approchée de $\frac{sM}{r}$, plus précisément, on connaît $2L+1$ bits de $\frac{s}{r}$, avec s aléatoire dans $\{0, \dots, r-1\}$.

Comment trouver r ? On peut grâce à l'algorithme des fractions continues, car $\left|\frac{s}{r} - \underset{\substack{\uparrow \\ \text{valeur} \\ \text{approchée}}}{\varphi}\right| \leqslant \frac{1}{2^{2L+1}} \leqslant \frac{1}{2r^2}$.

C'est un algorithme de complexité $O(L^3)$.

IV Factorisation

Soient p_1 et p_2 deux nombres premiers impairs, et $N = p_1 p_2$.

Lemme 1 : Soit $x \in \mathbb{Z}_N^*$ aléatoire, et r son ordre modulo N.
Alors $\mathbb{P}[\, r \text{ impair } \underline{\text{OU}} \ \{ r \text{ pair } \underline{\text{et}} \ x^{r/2} = -1 \bmod N \}] \leqslant \frac{1}{2}$

Preuve. $\mathbb{Z}_N^* = \mathbb{Z}_{p_1}^* \times \mathbb{Z}_{p_2}^*$ et choisir x aléatoirement revient à choisir aléatoirement x_1 et x_2, restes modulo p_1 et p_2.

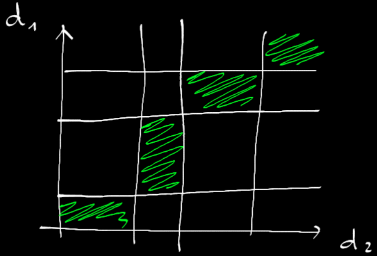
Soit r_i l'ordre de x_i modulo p_i, et d_i maximal tel que $2^{d_i} | r_i$.

Soit r l'ordre de x modulo N, et d maximal tel que $2^d | r$.

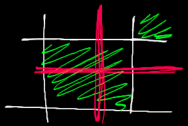
- Si r impair alors r_1 et r_2 impairs et donc $\boxed{d_1 = d_2} = 0$.
- Si r est pair et $x^{r/2} = -1 \bmod N$ alors $x^{r/2} = -1 \bmod p_i$

Donc $r_i \nmid \frac{r}{2}$, mais $r_i | r$ donc $d_i = d$ donc $\boxed{d_1 = d_2}$

Or, " aucun d_i ne couvre plus de la moitié des cas dans $\mathbb{Z}_p^*$ ".



$$\frac{\text{Aire}(\text{▨})}{(p_1-1)(p_2-1)} \leqslant \frac{1}{2}.$$



Lemme 2. Soit $x \in \{2, \dots, N-2\}$ tel que $x^2 = 1 \mod N$.
Alors $\{\gcd(x+1, N), \gcd(x-1, N)\} \cap \{p_1, p_2\} \neq \emptyset$.

Preuve : Si $x^2 = 1 \mod N$ alors $N \mid x^2 - 1 = (x+1)(x-1)$.

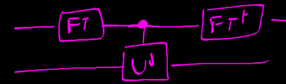
Donc $p_1 \mid (x+1)(x-1)$ donc $p_1 \mid x+1$ ou $p_1 \mid x-1$, CQFD.

Algorithme de Shor (1994).

① Choisir aléatoirement $x \in \{2, \dots, N-2\}$.

Si $\gcd(x, N) > 1$ alors c'est fini. Sinon → ②.

② On sait que $x \in \mathbb{Z}_N^*$. $\boxed{\text{Calculer son ordre } r.}$



$\mathbb{P} \geq 1/2$

③ $\boxed{\text{Si } r \text{ est pair et } x^{r/2} \not\equiv -1 \bmod N}$, calculer $\gcd(x^{r/2} \pm 1, N)$.

(Lemme 2)

Sinon, recommencer → ①.

Complexité : $O((\log N)^3)$ polynomial en le nombre de bits de N.

Meilleurs algorithmes classiques : $O\left(e^{\#(\log N)^{1/3}(\log \log N)^{2/3}}\right)$

L'algorithme quantique est dans "BQP"

Remarque : on a fait de la théorie !