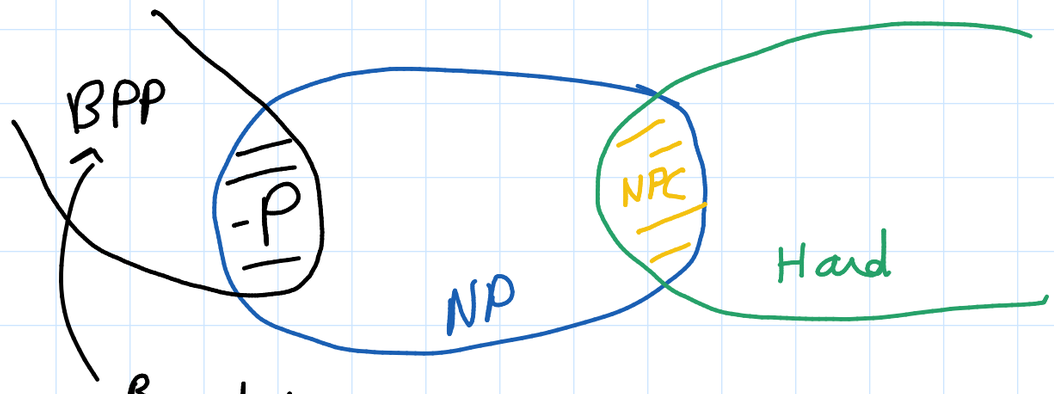
I Machine de Turing

↝ déterministe



$\begin{cases} 0 & w/pr\ 1/2 \\ 1 & w/pr\ 1/2 \end{cases}$ $\Big|$ $a+b=1$ ↓ $\mathbb{R}$

---

0 V^+ orbite basse Spin up

1 V^- orbite haute Spin down

Loi 1 :

un qbit peut être dans un état $|0\rangle$ ou $|1\rangle$ ou une superposition

$|0\rangle$ ↓ amplitude α $\quad$ $|1\rangle$ ↓ amplitude β ↝ $(\alpha,\beta)\in\mathbb{C}$, $|\alpha|^2+|\beta|^2=1$

Ex : $0,8|0\rangle + 0,6|1\rangle$ ↝ $|0,8|^2+|0,6|^2=1$

$0,8|0\rangle - 0,6|1\rangle$ ↝ pareil

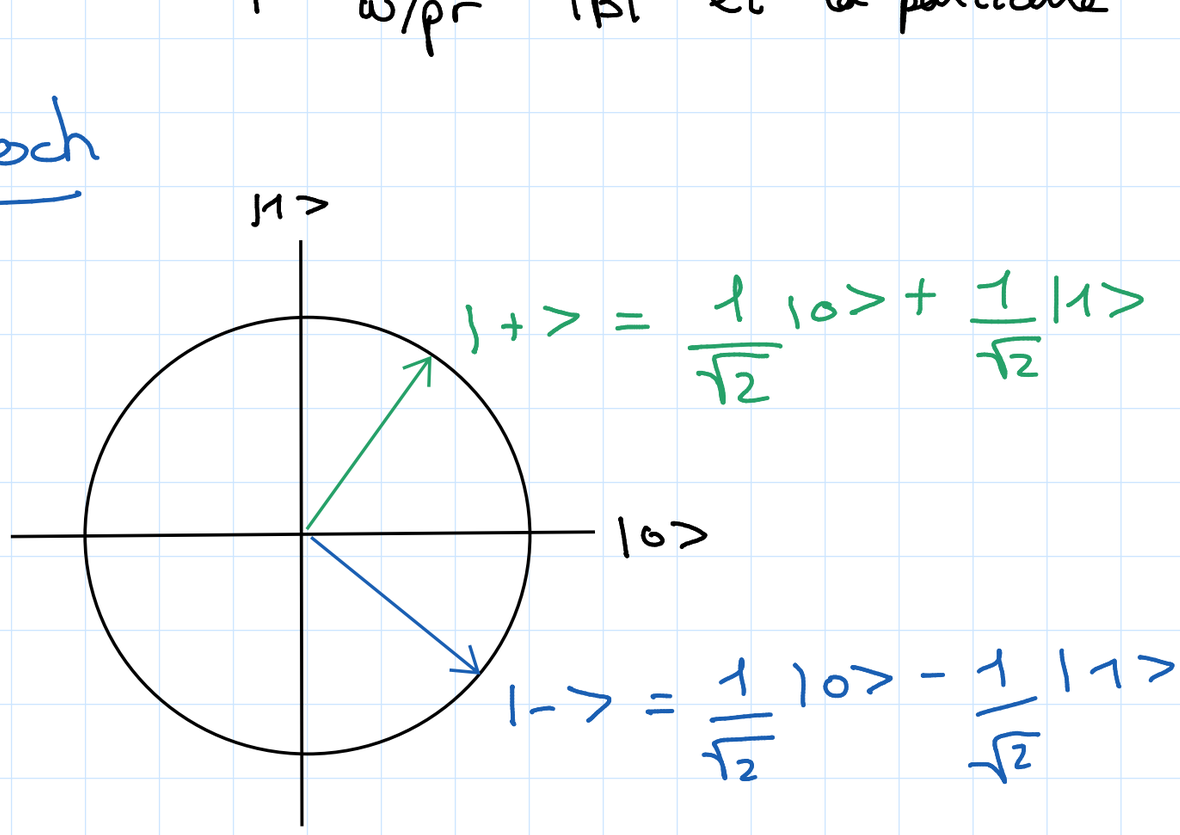
$i|0\rangle - 0|1\rangle$ ↝ pareil

Loi 2 : Mesure

une particule $\alpha|0\rangle + \beta|1\rangle$

sera mesurée
- 0 w/pr $|\alpha|^2$ et la particule prend l'état $|0\rangle$
- 1 w/pr $|\beta|^2$ et la particule prend l'état $|1\rangle$

Bloch

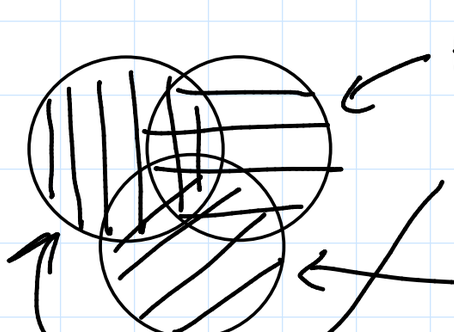


$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$

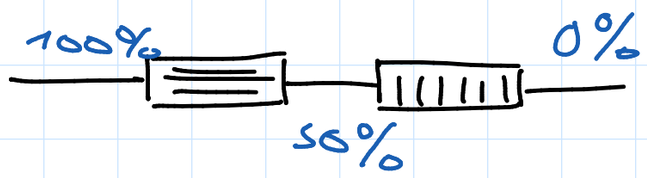
$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

$|+\rangle = \begin{pmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{pmatrix}$

$|-\rangle = \begin{pmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{pmatrix}$



fait passer les photons dans un sens dans l'autre un peu des 2.

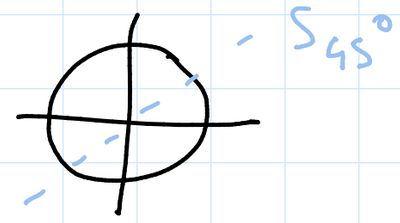


Loi 3 : transformation

① $U^+ U = Id$
② $U^+ = U^{-1}$
③ les colonnes forment une base orthonormé et leur norme est 1.
④ $\forall a,b$ qbits $\langle a|b\rangle = \langle U_a | U_b \rangle$

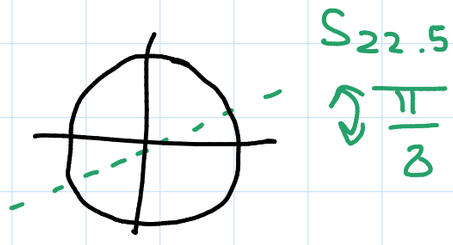
$\varphi = \alpha|0\rangle + \beta|1\rangle$

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix}$$

H → porte de Hadamard → $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$



→ symétrie qui permet d'obtenir $S_{45°}$

Loi 4 : état joint

Soit $|\phi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{pmatrix}$ $|\varphi\rangle = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_e \end{pmatrix}$

Leur état joint est $|\phi\rangle \otimes |\varphi\rangle = \begin{pmatrix} \alpha_1 \beta_1 \\ \alpha_1 \beta_1 \\ \vdots \\ \alpha_d \beta_e \end{pmatrix}$

- $|0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
- $|0\rangle \otimes |+\rangle = |0+\rangle = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix}$
- $|00\rangle \otimes |1\rangle = |0\rangle \otimes |01\rangle \neq |01\rangle \otimes |0\rangle$

Loi 5 : mesure particules

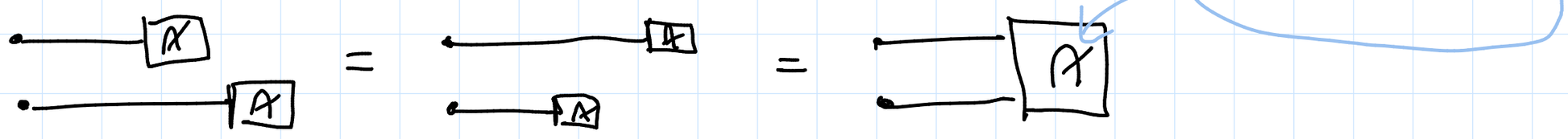
$$|\varphi\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

on 1 sur le qbit 1 renverra $\begin{cases} 0 & w/pr\ |\alpha_{00}|^2 + |\alpha_{01}|^2 \\ 1 & w/pr\ |\alpha_{10}|^2 + |\alpha_{11}|^2 \end{cases}$ et l'état devient $\begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ 0 \\ 0 \end{pmatrix} \cdot \frac{1}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$

→ $\begin{pmatrix} 0 \\ \\ \end{pmatrix}$

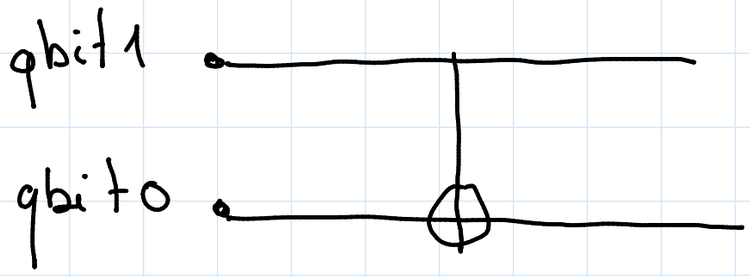
$$\begin{pmatrix} 0 \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

Mesure différée



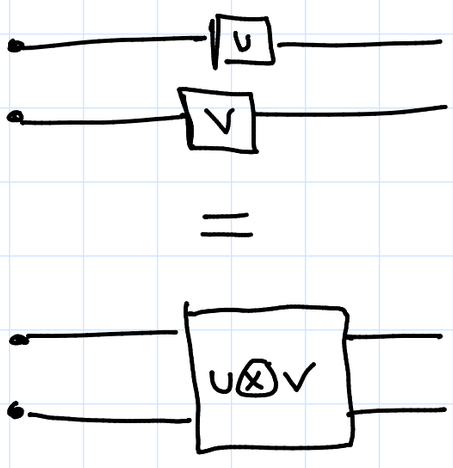
Loi 6 : Unitaires

CNOT (seconde porte avec Hadamard)



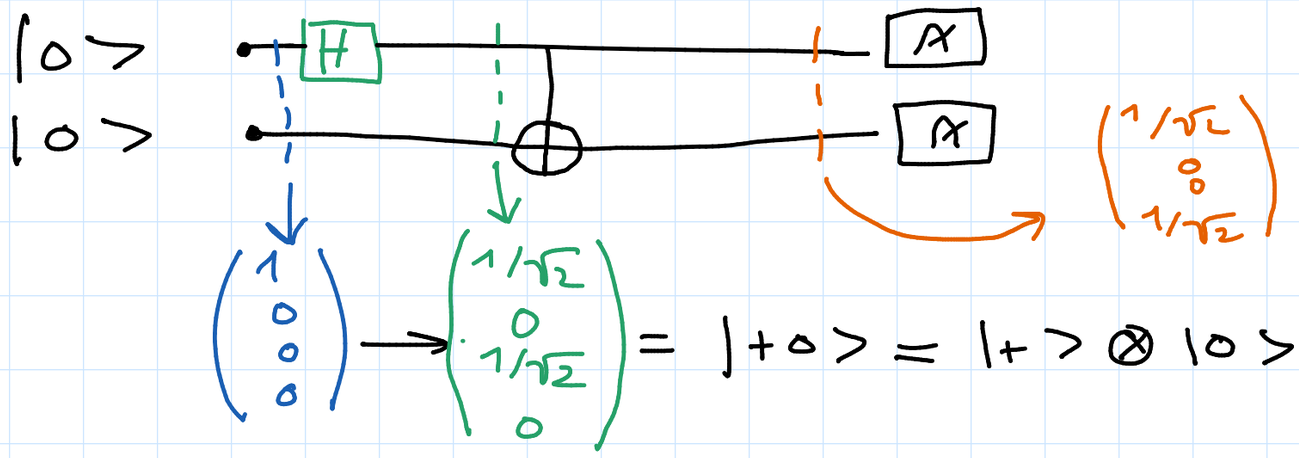
$$|00\rangle \rightarrow |00\rangle$$
$$|01\rangle \rightarrow |01\rangle$$
$$|10\rangle \rightarrow |11\rangle$$
$$|11\rangle \rightarrow |10\rangle$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



$U|\psi\rangle \otimes V|\varphi\rangle$

$(U \otimes V) \cdot |\psi\varphi\rangle$



$$\begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \\ 0 \end{pmatrix} = |+0\rangle = |+\rangle \otimes |0\rangle = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Bell
Paire PR
Qbits intriqués
$|00\rangle + |11\rangle$

$\not\exists \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$ = Ball state

$$\begin{pmatrix} \alpha_0 \beta_0 \\ \alpha_0 \beta_1 \\ \alpha_1 \beta_0 \\ \alpha_1 \beta_1 \end{pmatrix}$$

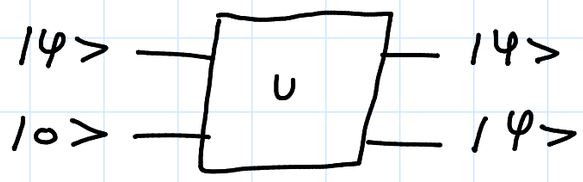
Alice (1er qbits) mesure

$$\begin{cases} 0 & \text{w/pr } \left|\frac{1}{\sqrt{2}}\right|^2 \text{ et l'état devient } \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle \\ 1 & \text{w/pr } \frac{1}{2} \text{ et l'état devient } \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle \end{cases}$$

Bob mesure
si Alice a mesuré 0 => 0 w/pr 1
si Alice a mesuré 1 => 1 w/pr 1

Crypto quantique

Théorème de non-clonage : $\exists U$:



BB4D : Quantum Key Distribution (QKD)

0				1			
+		×		+		×	
|0>		|+>		|1>		|->	
+	×	+	×	+	×	+	×
0	1/2 0 1/2 1	1/2	0	1	1/2	1/2	1
+	×	+	×	+	×	+	×
✓	×	×	✓	✓	×	×	✓

Alice pioche un bit une base

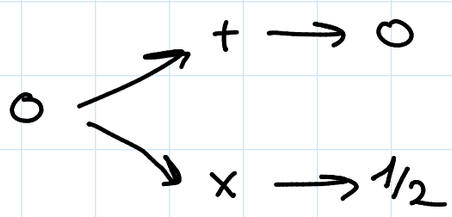
Alice envoie

Bob pioche une base

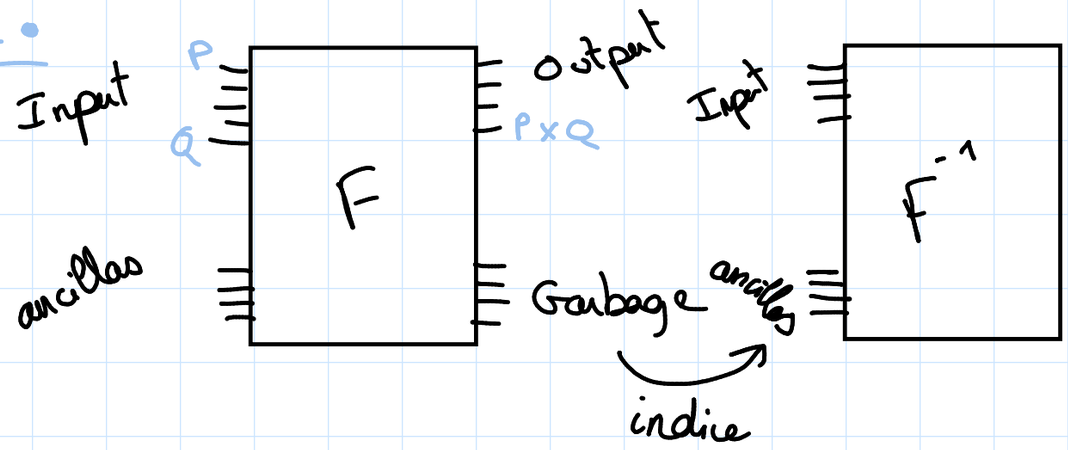
Bob mesure

Bob envoie la base

même base ?

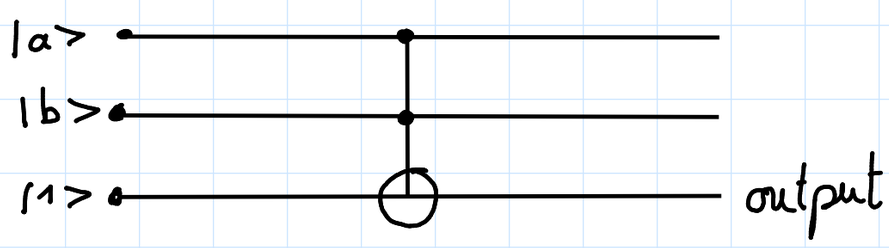


exemple :



crypto post-quantique => chiffrement asymétrique sinon on double juste la taille de la clé en chiffrement symétrique.

CCNOT : Conditional Conditional NOT



y'a rien qu'un ordinateur classique peut faire qu'un ordinateur quantique ne peut pas faire (ici NAND donc turing complet)

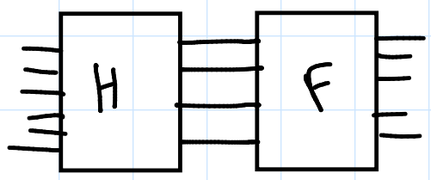
Technique 1 : Rotate compute rotate

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

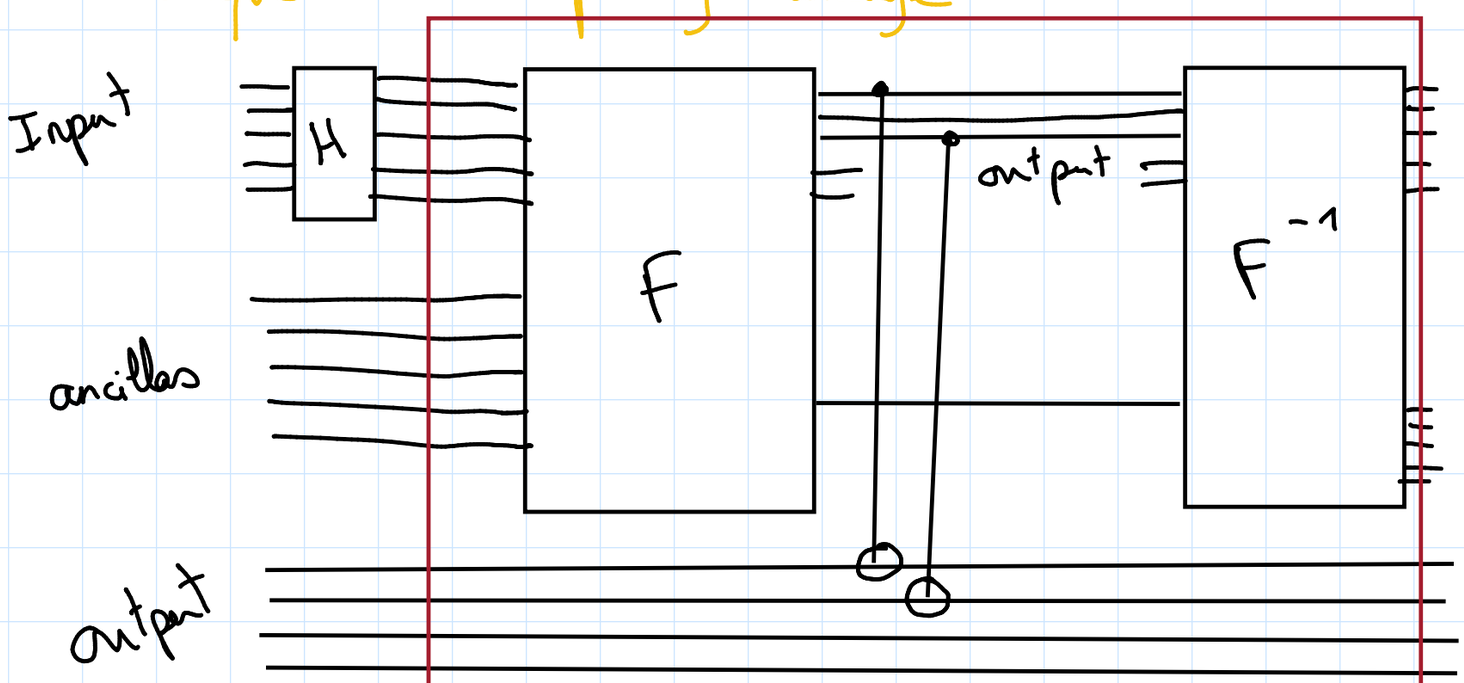
$$H^{\otimes 2}|00\rangle = \frac{1}{\sqrt{2}^2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

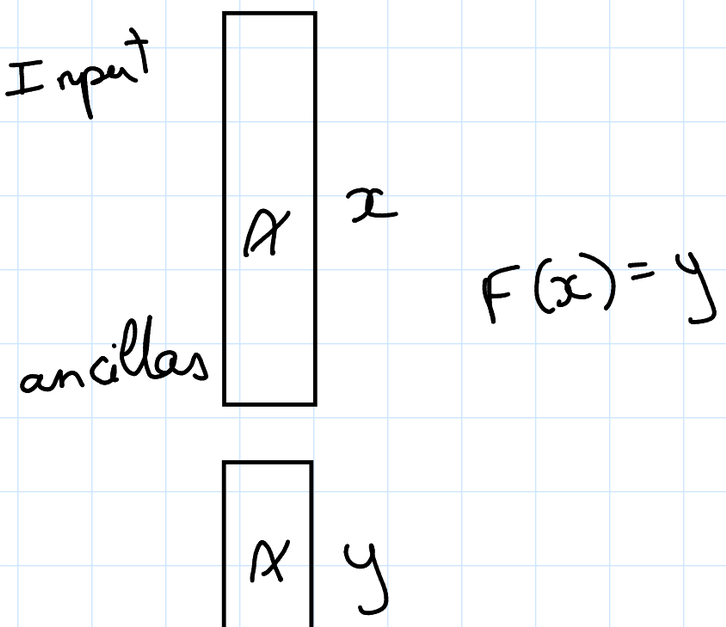
...

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2}^n}\begin{pmatrix}1\\\vdots\\1\end{pmatrix} \updownarrow 2^n$$



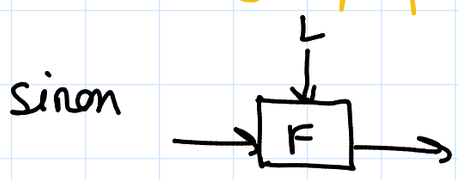
Technique 2 : Uncomputing Garbage





F(x) = y

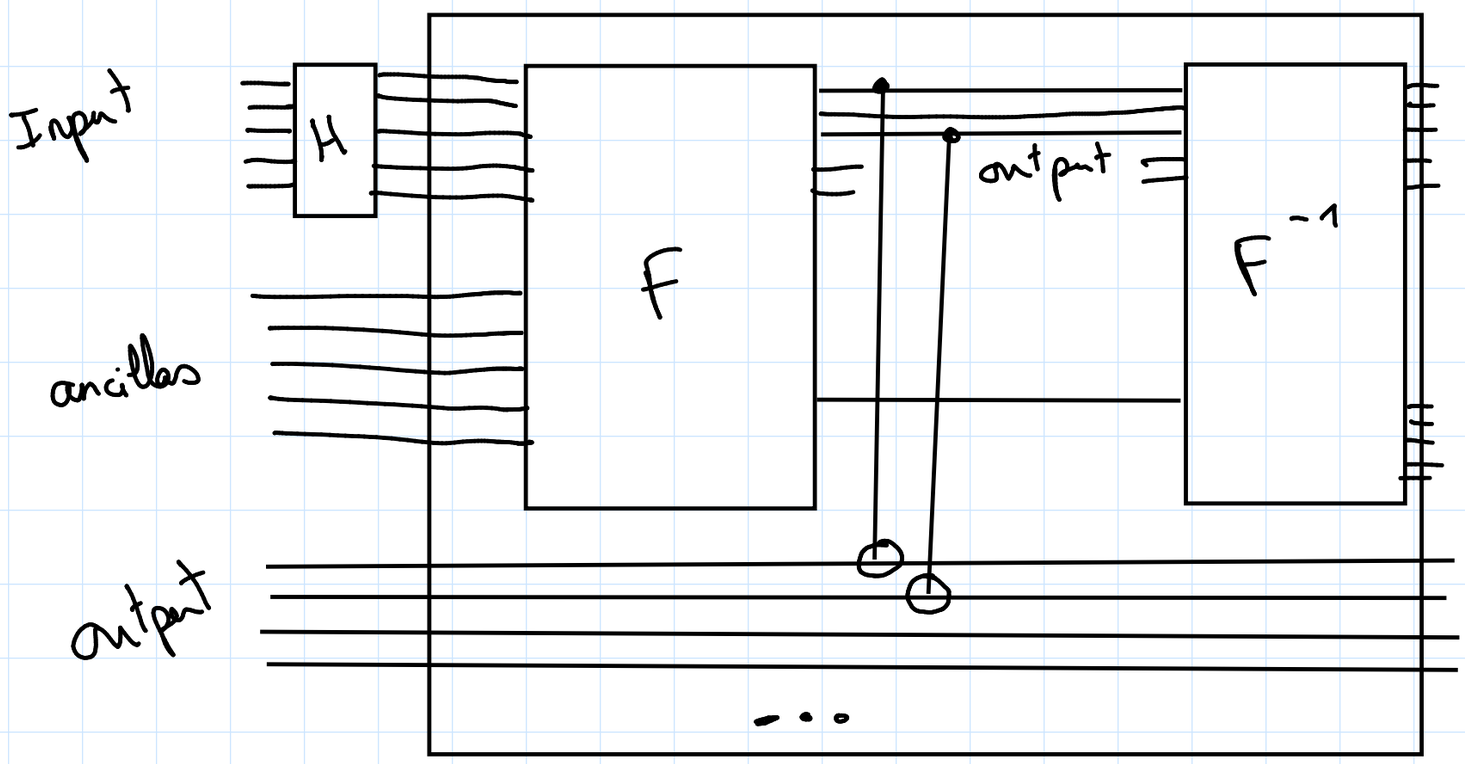
B) Hidden subgroup problem

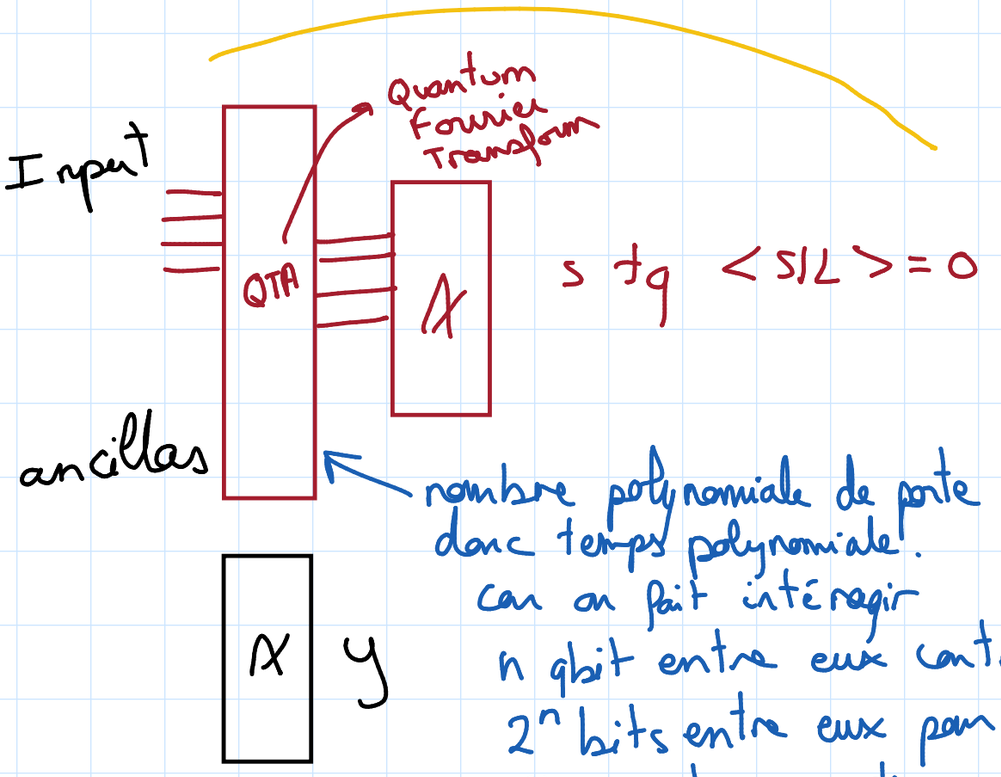


sinon $f(x) = f(y) \Rightarrow x = y \oplus L$

$F(x) = g^x [pq]$ avec QTA.

il essaye d'expliquer un truc



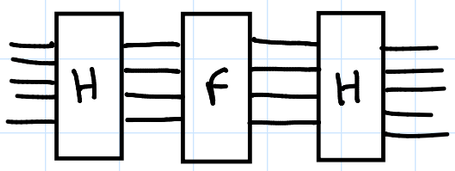


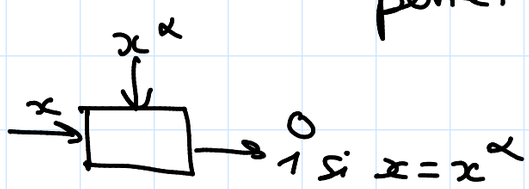
$s \neq q \quad \langle s/L \rangle = 0$

nombre polynomiale de porte donc temps polynomiale. car on fait intéragir n qbit entre eux contre 2^n bits entre eux pour l'informatique quantique

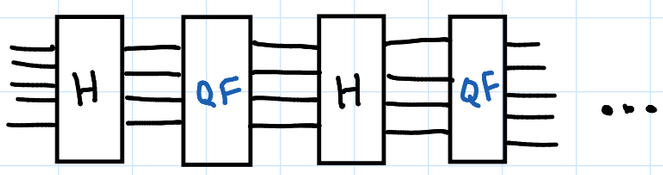
→ le gros schéma c'est **l'algo de SHOR $O(n^3)$**

c) GROVER





permet de faire des condensat (hachage)



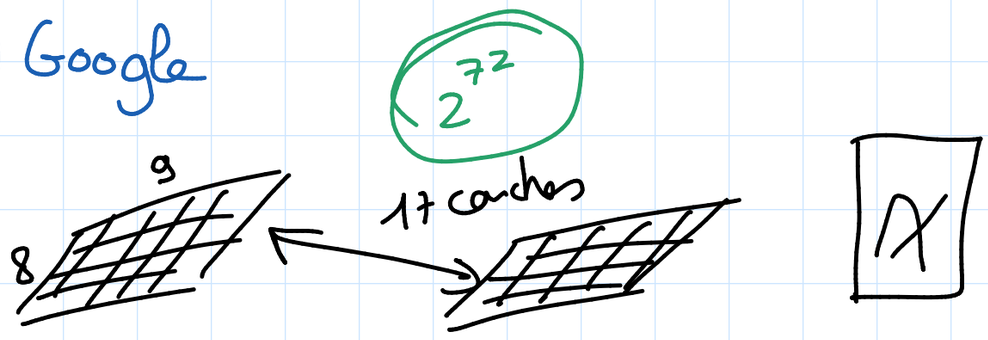
...

RSA 2048 → $\lambda = 112$

2 " → $\lambda = 36$

III Suprématie Quantique

A) Google



B) Fujitsu

C) Microsoft Alice & Bob

D) IBM

New Part : JC Deneuville

→ début de la PLS-SEC

Grover : n entrée pas trié, on recherche en $O(\sqrt{n})$ au lieu de $O(n)$

↳ conséquence : on double la taille de la clé pour avoir le même niveau de sécurité

- sha1 et md5 déprécié. On est sur sha2 et on standardise sha3 (sha512)
  → minimum +33% pour avoir le même niveau de sécurité.

Algo de Shor : meilleur algo $(1.9(\log N)^{1/3}(\log\log N)^{2/3})$ en classique
en quantique c'est de l'ordre du cubique.

↳ conséquence : plus de RSA, DSA, ECDSA, ElGamal ⇝ fin de la crypto asymétrique
→ plus d'échanges de clés.

$\mathbb{F}_p$: corp fini à p éléments
si p premier, alors $\mathbb{Z}_p \simeq \mathbb{F}_p$

$\mathbb{Z}_p$: entier modulo p
$= \mathbb{Z}/p\mathbb{Z}$ (anneau)

$\mathbb{F}_7 = \mathbb{Z}_7$ mais $\mathbb{F}_7 \neq \mathbb{Z}_7$

$\mathbb{F}_q$ existe $\Leftrightarrow q = p^r$ pour p premier et $n \geq 1$

---

p : premier grand
g : générateur $(\mathbb{Z}/p\mathbb{Z})^2$

prenons $\begin{matrix} g = 2 \\ p = 7 \end{matrix}$ (ça marche pas)

$g^1 = 2 = g^4 = 2$
$g^2 = 4 = g^5 = 4$
$g^3 = 1 = g^6 = 1$

n'engendre pas tt le groupe $\mathbb{Z}/7\mathbb{Z}$

↝ $g = 3$ engendre $\mathbb{Z}/7\mathbb{Z}$

inverse de 7 [11] ?

$11 = 7 \times 1 + 4$
$7 = 4 \times 1 + 3$
$4 = 3 \times 1 + 1$

$1 = 4 - 3 \times 1$
$= 4 - (7 - 4 \times 1)$
$= 4 \times 2 - 7$
$= (11 - 7 \times 1) \times 2 - 7$
$= 11 \times 2 - 7 \times 3$
$1 \equiv -7 \times 3 [11]$

donc -3 est l'inverse de 7 donc $-3 + 11 = 8$ est l'inverse de 7

8 est l'inverse de 7 et vice-versa

$g^{\frac{p-1}{2}}[p] = \pm 1$ ↝ +1 ça fonctionne pas
↝ -1 ça fonctionne

$g = 2, p = 7 : 2^3 [7] = +1$
$g = 3, p = 7 : 3^3 [7] = -1$

" si $g^{\frac{p-1}{2}}[p] = -1$ alors g engendre $\mathbb{Z}/p\mathbb{Z}$ et inversement "

ce qu'il faut retenir pour l'exam :

- error correcting codes
- lattices
- multivariate
- hash function
- elliptic curves isogenies

} possibilités en post quantique

Lattice-based

rappels : norme : $\|x\| \geq 0 \ \forall x$, $\|x\| = 0 \Leftrightarrow x = 0 \ \forall x$, $\|x+y\| \leq \|x\| + \|y\| \ \forall x, y$
$\vec{u}$ et $\vec{v}$ linéairement indépendant $\Leftrightarrow \alpha\vec{u} + \beta\vec{v} = \vec{0} \Leftrightarrow \alpha = \beta = 0$

déterminant non nul ⇝ inversible ⇝ revoir les calculs de déterminant et inversions

$u = (1,0,1,1) = 1 + X^2 + X^3$
$v = (0,1,1,0) = X + X^2$

$\Rightarrow u * v = (1+X^2+X^3)(X+X^2) = X + X^3 + X^4 + X^2 + X^4 + X^5$
$= X^2 + X^3$

$u * v = (0,0,1,1)$ produit de convolution

$\mathbb{F}_2^4 \simeq \mathbb{F}_2[X]/(X^4-1)$

$X^4 - 1 = 0 \Rightarrow \begin{cases} X^4 = 1 \\ X^5 = X \\ X^6 = X^2 \\ X^7 = X^3 \end{cases}$

$$(1\,0\,1\,1)\cdot\begin{pmatrix} 0&1&1&0 \\ 0&0&1&1 \\ 1&0&0&1 \\ 1&1&0&0 \end{pmatrix} = (0\,0\,1\,1)$$

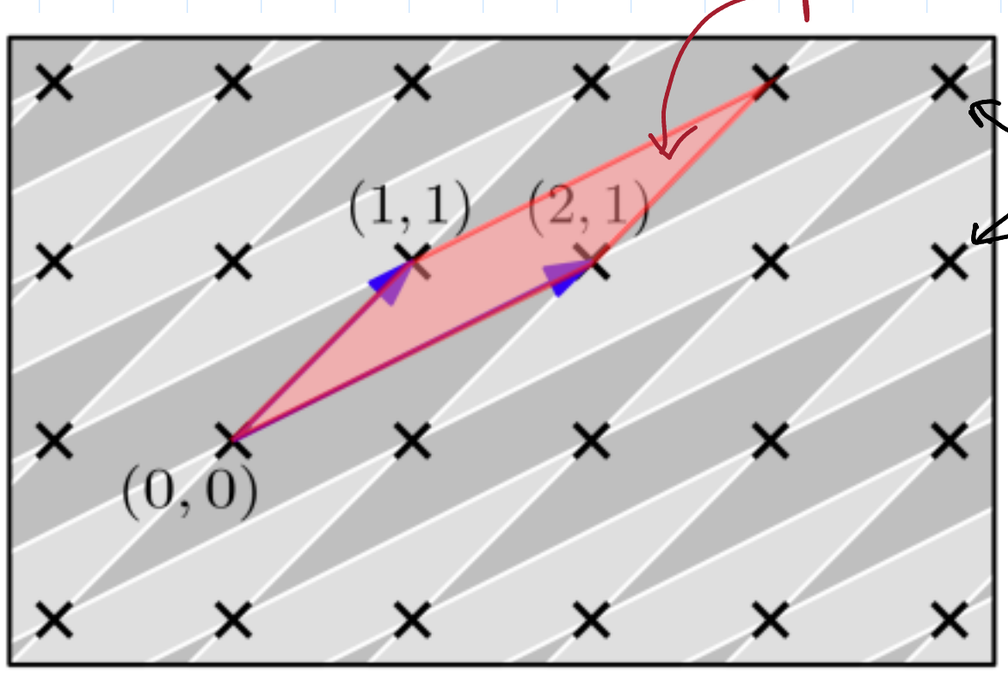
End_revisions ;

Lattice : réseau euclidien

un reseau de dimension n est un sous groupe discret de $\mathbb{R}^n$.

$(b_1, \dots, b_n) \in \mathbb{R}^m$, $A(b_1, \dots, b_n)$

$$A = \left\{ \sum_{i=1}^{n} x_i b_i \; ; x_i \in \mathbb{Z} \right\} \subset \mathbb{R}$$



parallélépipède fondamentale

solutions entières générés par la base

représentation matricielles : $B = (b_1 | \dots | b_n) \in \mathbb{Z}^{n \times n}$, le réseau généré par B est :
$\Lambda(B) = \{B \cdot x \; ; x \in \mathbb{Z}^n\}$

$B = (b_1 | \dots | b_n) \in \mathbb{Z}_q^{m \times n}$, q premier
$A_q(B) = \{B \cdot x \; [q], \; x \in \mathbb{Z}^n\}$
$A_q^{\perp}(B) = \{y \in \mathbb{Z}^m, \; y^t B = 0 \; [q]\}$

- une matrice est unimodulaire si son déterminant est ± 1
- $B' = BU$ pour des matrices unimodulaire $U \Rightarrow \Lambda(B') = \Lambda(B)$
- $\forall n \geq 2$, les réseaux à dimension n ont une infinité de bases

minimums successifs : un réseau $\mathcal{L}$, la distance minimale $\lambda_1(\mathcal{L})$ est la distance minimale entre 2 points du réseaux :

$$\lambda_1(\mathcal{L}) = \inf\{\|x - y\|, \; x \neq y \in \mathcal{L}\}$$

Hards problems : SVP, small integers solutions, closest vector problem, learning with errors,

Lattice problems : "on va utiliser la bonne base en clé privé, et la mauvaise en clé publique"
↳ avec une "bonne" base ⇝ facile
↳ hard problem avec la mauvaise base
→ on est en dimension ≥ 512 (le problem serait trivial en dimension 2)

$\mathbb{Z}_4 = \{0,1,2,3\} \neq \mathbb{F}_4 = \{0,1,\alpha,\alpha+1\}$, $\mathbb{F}_4[x] = \mathbb{F}_2[x]/(x^2+x+1)$

$1^{-1} = 1$

$\alpha^{-1} = \alpha$ ou $\alpha+1$?

α est racine de $\alpha^2+\alpha+1=0$
$\alpha^2 = \alpha+1$

$\alpha(\alpha+1) = \alpha^2+\alpha = \alpha^2+1+\alpha$
$\Rightarrow \alpha^{-1} = \alpha+1$

X^2+1 / X^2+X / X^2+X+1 } pas irréductible

heuristique optimale : heuristique de Gauss :

How orthogonal can a basis be?

$$\delta(\mathcal{L}) = \frac{\prod_{i=1}^{n} \|\mathbf{b}_i\|}{\det(\mathcal{L}) = \sqrt{\det(\mathbf{B}^\top \mathbf{B})}} \geq 1$$

Notice that equality holds $\iff$ basis is orthogonal.

How short can a vector be?

$$\lambda_1(\mathcal{L}) \approx \frac{\Gamma(n/2+1)^{1/n}}{\sqrt{\pi}} \cdot \det(\mathcal{L})^{1/n} \approx \sqrt{\frac{n}{2\pi e}} \cdot \det(\mathcal{L})^{1/n}$$

Gaussian heuristic predicts the length of the shortest vector in a random lattice.

→ norme quadratique : $\sqrt{(x_B-x_A)^2+(y_B-y_A)^2}$

norme euclidienne : $d(A,B) = \|B-A\|$

réduction de réseaux euclidien : Gram-Schmidt ne fonctionne pas car va sortir des base réelle, hors on est dans les entiers
⇝ on va prendre l'entier le ⊕ proche

LBC : (on va générer une base de norme petite (erreurs) ($e \leftarrow \{-1,0,1\}^n$)
on va générer un secret ($s_k = s \in \{-1,0,1\}^n$)
$p_k = (A,b)$ où $b = As+e$
→ à partir de n, m, q, α] Keygen

$r \leftarrow \{0,1\}^n$ ⇝ output $u = r^T A$ et $v = r^T b + \frac{q \cdot m}{2}$] Encrypt

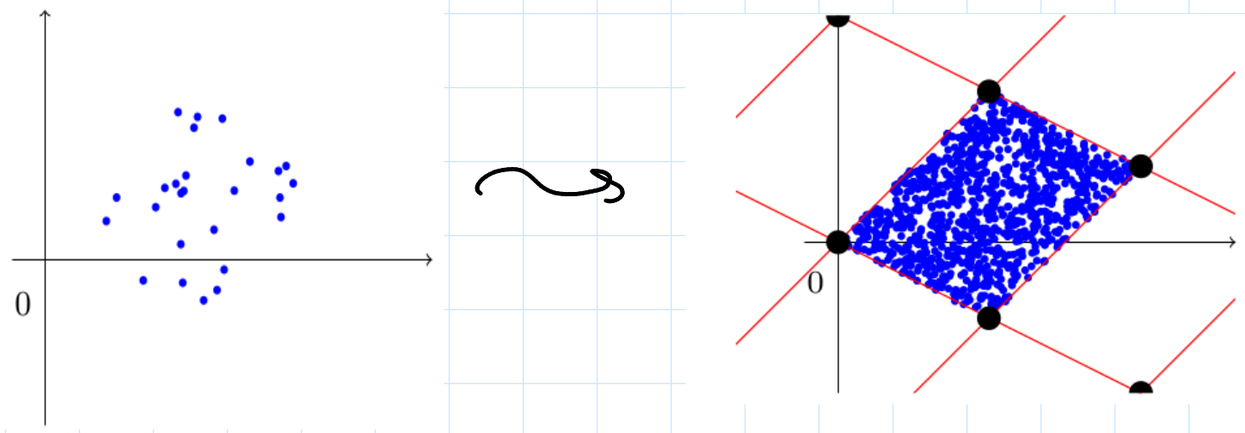
compute $l = v - u^T s$. Si l proche de 0 : 0 sinon 1] Decrypt

→ $l = r^T b + \frac{q}{2} m - r^T A s$
$= r^T(As+e) + \frac{q}{2} m - r^T A s$
$= r^T A s + r^T e + \frac{q}{2} m - r^T A s$

$l = v - u^T s$

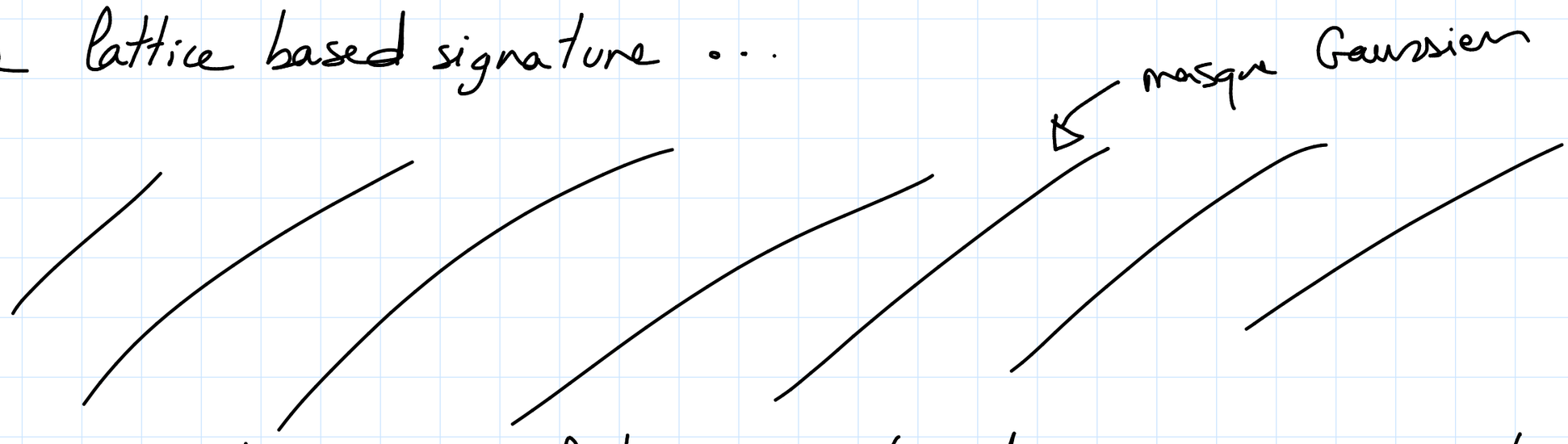
NTRUsign ? pa compri

→ askip on reviendra dessus semaine pro → tsé gwa ça me va



si on l'utilise trop, on leak notre parallélépipède

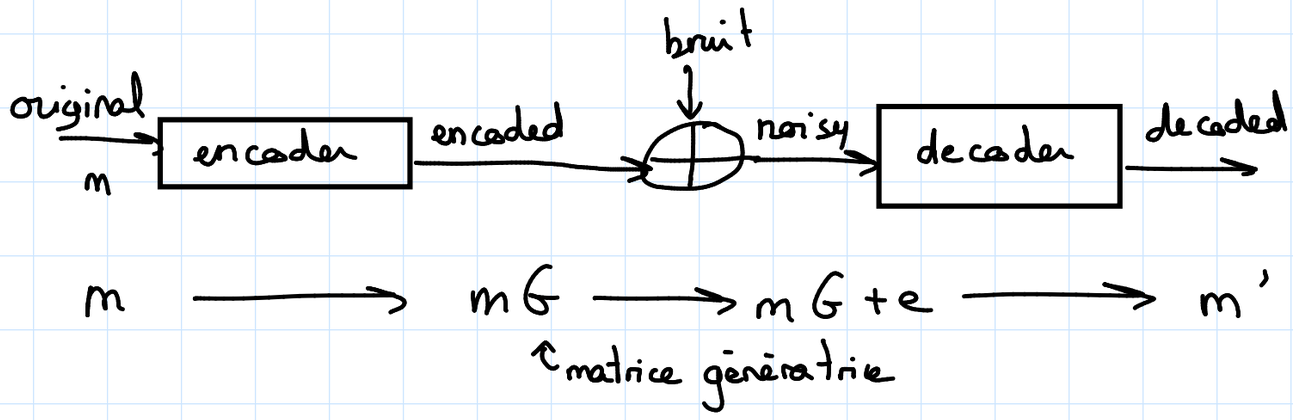
secure lattice based signature ...

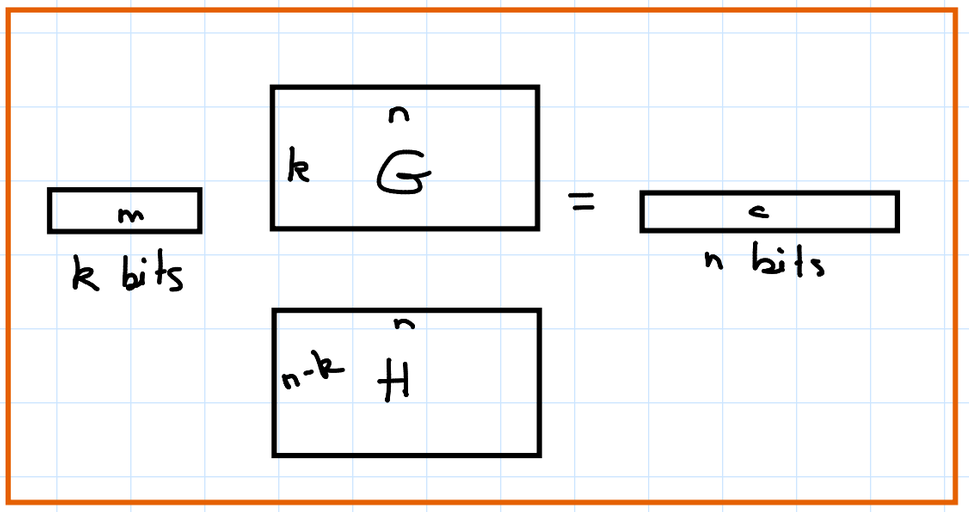


si pas assez randomisé on refait la génération de z, sinon on peut publier la clé z.

Coding Theory

↳ bruits sur signal → faut ajouter de la redondance sur le message
↳ m.G (G génératrice pour faire de la redondance)





- Pour décoder, il faut résoudre l'erreur => $m' = m$
- Pour HQC, encoder publique, decoder privée

Linear code : un code linéaire de dimension k, taille n dans $\mathbb{F}_q$ est un sous espace de $\mathbb{F}_q^n$ de dimension k.
↳ matrice génératrice $G \in \mathbb{F}_q^{k \times n}$
↳ matrice de parité $H \in \mathbb{F}_q^{(n-k) \times n}$
↳ $c = xG$ ...) voir slide 70
↳

une manière de faire de la correction de code est la repetition.

Distance de Hamming → revoir les slides...

CBC (code-based cryptography)

pour une dimension k, longueur n et distance minimale $d=3$ entre chaque mots du code
↳ capacité de détection : $d-1 = 2$ erreurs détectables
↳ capacité de correction : $\left\lfloor \frac{d-1}{2} \right\rfloor = 1$ erreur corrigeable
↳ rendement $\frac{k}{n} = \frac{1}{3}$

exemple bidon où on répète 3 fois chaque sigle :

Message à envoyer		1			0			1	
Encodage	1	1	1	0	0	0	1	1	1
Message reçu	0	1	1	0	1	0	1	1	0
Message décodé		1			0			1	

Décodage de syndrôme (CBC)

Soit $s \in \mathbb{F}_2^{n-k}$ et $H \in \mathbb{F}_2^{(n-k)n}$. Trouver $x \in \mathbb{F}_2^n$ tq $Hx^T = s$

↳ ce n'est pas un problème difficile (algèbre linéaire)

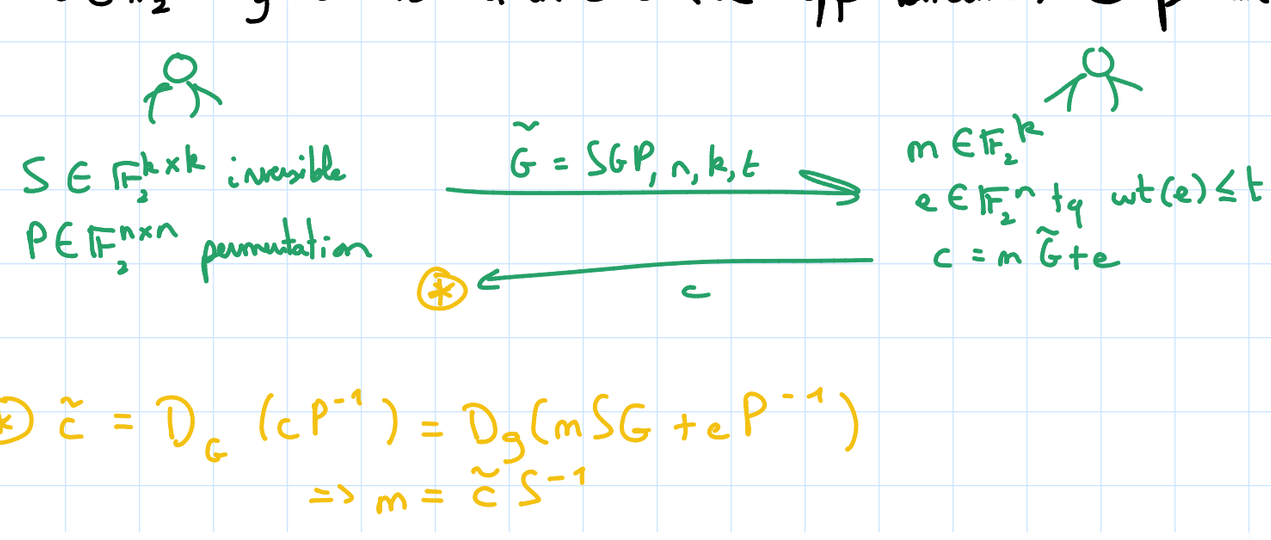
↳ on va donc modifier le problème :

Soit $s \in \mathbb{F}_2^{n-k}$ et $H \in \mathbb{F}_2^{(n-k)n}$. Trouver $x \in \mathbb{F}_2^n$ tq $Hx^T = s$ **et x de poid relativement faible**

↳ le problème devient NP-difficile (donc intéressant cryptographiquement)

McEliece [McE78]. $G \in \mathbb{F}_2^{k \times n}$

Soit $G \in \mathbb{F}_2^{k \times n}$ génératrice d'un code (de Goppa binaire) C pouvant corriger jusqu'à t erreurs avec l'algo D_g



(*) $\tilde{c} = D_G(cP^{-1}) = D_g(mSG + eP^{-1})$

$\Rightarrow m = \tilde{c}S^{-1}$

↳ exemple sur les slides avec code de Hamming

Soit $\mathcal{C}$ le code (de Hamming) admettant pour matrice de parité $\mathbf{H}$:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Soit $\mathbf{s} = (1,0,0,0,1,1,1)$ le mot reçu. Quel était le message envoyé ? Décodons

car le syndrôme se trouve dans la 5e colonnes de H.

$$\mathbf{v} = (1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1) \qquad \mathbf{e} = (0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0)$$

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \mathbf{s}$$

on corrige l'erreur

$$\mathbf{m} = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1)$$

Quasi Cyclic Moderate Density Parity-Check Codes

KeyGen

Sample $\mathbf{h}_0, \mathbf{h}_1 \leftarrow \mathbb{F}_2^r$ of small weight w, $\mathbf{h}_0$ inversible. Compute $\mathbf{h} = \mathbf{h}_1\mathbf{h}_0^{-1}$.

$$\mathbf{H}_{\text{secret}} = \left(\begin{array}{c|c} \mathbf{h}_0 & \mathbf{h}_1 \\ \circlearrowleft & \circlearrowleft \end{array} \right)$$

$$\mathbf{H}_{\text{pub}} = \left(\begin{array}{c|c} (1,0,\ldots,0) & \mathbf{h} \\ \circlearrowleft & \circlearrowleft \end{array} \right)$$

Encryption

As for McEliece, $\mathbf{e}$ of weight t,

$$\mathbf{c} = \mathbf{mG} + \mathbf{e}.$$

Decryption

Use an iterative decoder (e.g. the BitFlipping algorithm) to recover message $\mathbf{m}$.

Suggested parameters: r = 9857, n = 2r, w = 142, t = 134 for 128 bits. Resulting sizes?

- → efficace
- → modulo 2 vs modulo $2^{\sim 1024}$ pour RSA
- → parallélisable
- → taille de clé conséquente (quasi-cyclique ?)
- → hypothèse d'indistagibilité de la famille de code utilisé
  - ↳ trop technique

Hash-Based Cryptography

HQC