Université de Montréal

Les codes correcteurs quantiques et leurs applications cryptographiques

par

Christian Paquin

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

décembre 2000

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

Les codes correcteurs quantiques et leurs applications cryptographiques

présenté par:

Christian Paquin

a été évalué par un jury composé des personnes suivantes:

Michel Boyer

(président-rapporteur)

Gilles Brassard

(directeur de recherche)

Geňa Hahn

(membre du jury)

Mémoire accepté le:

20 Avril 2001

Sommaire

L'informatique et la mécanique quantique sont deux des plus importantes théories du 20^{e} siècle. Elles se combinent élégamment pour former la théorie de l'information quantique. Depuis que Shor [Sho97] a découvert qu'un ordinateur quantique peut factoriser les grands nombres en temps polynomial (un problème considéré difficile classiquement), beaucoup d'attention a été portée sur ce domaine.

Depuis, plusieurs résultats surprenants montrent que l'ordinateur quantique peut faire des choses qu'on ne pourrait jamais faire classiquement, autant du point de vue algorithmique (comme l'algorithme de Grover [Gro97]) que celui du traitement d'information (comme par exemple, la téléportation [BBC^{+}93] ou la pseudo-télépathie [BCT99]). Cependant, une question importante se pose : est-ce que l'ordinateur quantique pourra être un jour construit ? En effet, en plus des défis technologiques à relever pour atteindre un tel but, un autre obstacle mine l'avenir de l'ordinateur quantique. L'information quantique est par nature très fragile. Le simple fait d'y accéder ou de la garder en mémoire trop longtemps risque de la détruire. De plus, chaque opération effectuée sur un registre quantique risque de ruiner le calcul en y introduisant des erreurs. Les plus pessimistes disaient qu'on ne pourrait jamais construire un ordinateur quantique à cause de ces erreurs incontrôlables [Lan95, Unr94, CLSZ95, HR96]. L'opinion de plusieurs personnes a changé lorsque Shor [Sho95] a présenté le premier code correcteur quantique en 1995 (i.e. un procédé algorithmique qui augmente la résistance aux erreurs des systèmes quantiques). Durant cette même année, le domaine des codes correcteurs quantiques est né et a littéralement explosé, car plusieurs chercheurs ont porté leurs efforts pour développer cette nouvelle théorie. Plusieurs papiers importants

ont vu le jour [Ste96b, CS96, CL95, BDSW97, Sho96], donnant bon espoir que l'ordinateur quantique, malgré la malédiction des erreurs, pourrait un jour être construit grâce à ces codes correcteurs.

Le fait que l'algorithme de factorisation de Shor mette en péril les systèmes cryptographiques à clés publiques les plus populaires (comme RSA et ElGamal) a attiré une attention particulière sur la cryptographie quantique. Bennett et Brassard ont fondé cette branche quantique de la cryptographie en présentant, en 1984, leur protocole BB84 de distribution quantique de clés. Le protocole permet à deux personnes d'échanger une clé secrète de façon parfaitement sécuritaire. On a découvert dernièrement des liens importants entre les domaines de la cryptographie quantique et de la correction d'erreurs. Ce mémoire s'intéresse aux applications cryptographiques des codes correcteurs d'erreurs quantiques. Nous présenterons dans cet ouvrage deux protocoles cryptographiques qui utilisent les codes correcteurs quantiques.

Le mémoire est divisé de deux parties. La première, intitulée la correction d'erreurs classiques et quantiques, contient trois chapitres d'introduction. Le chapitre 1 présente brièvement la théorie des codes correcteurs classiques. Une introduction à la théorie de l'information quantique est présentée au chapitre 2. On combine ces deux premiers chapitres pour former le chapitre 3 qui décrit la théorie des codes correcteurs quantiques. Cette première partie se veut une référence pour chacun des sujets traités. L'auteur a pris soin de décrire chaque concept de façon complète et claire en donnant plusieurs exemples afin qu'un informaticien non familier avec ces notions puisse comprendre la deuxième partie. Le résultat est un ouvrage volumineux qui contient plus d'information que ce qui est nécessaire pour comprendre la deuxième partie. La notation du chapitre 2 peut sembler étrange à un mathématicien ou à un informaticien, mais c'est la notation utilisée dans le domaine et l'auteur s'inspire spécialement de [Bra].

La deuxième partie décrit les applications cryptographiques des codes correcteurs quantiques. Le chapitre 4 présente le protocole de distribution quantique de clés BB84 ainsi que la preuve de sécurité de Shor et Preskill [SP00] qui utilise les codes correcteurs. On y décrit aussi la notion de répéteurs quantiques qui peuvent être utilisés pour

augmenter les distances atteintes pour la distribution quantique de clés. Le chapitre 5 présente la notion de partage de secrets quantiques. Nous y verrons le protocole de Cleve, Gottesman et Lo [CGL99], qui généralise le partage de secrets classiques de Shamir. De plus, l'auteur y présente une version simplifiée du protocole CGL qui en facilite la présentation et l'implantation.

Table des matières

Table des figures

Remerciements

Plusieurs personnes ont contribué au succès de ce travail. J'aimerais remercier

- Gilles Brassard, mon directeur de recherche, pour m'avoir accueilli au laboratoire d'informatique théorique et quantique au début des mes études, pour son enseignement dynamique, pour ses encouragements continuels et pour son aide financière.
- Anton Stiglic et Frédéric Légaré, des amis très chers qui m'ont beaucoup aidé dans mes années d'études et dans la rédaction de ce mémoire.
- Les membres du LITQ avec qui j'ai eu plusieurs entretients très intéressants.
- le CRSNG qui m'a soutenu dans mes études graduées.
- Pierre McKenzie, pour son enseignement rigoureux et très stimulant. Son cours de complexité a été le plus dur et le plus stimulant de mes études.
- Claude Crépeau pour ses cours de cryptographie et de théorie des codes desquels j'ai tiré ma spécialité dans le monde quantique.
- John Preskill, pour ses publications qui m'ont permis de comprendre le domaine de la correction d'erreur à l'époque où la littérature du domaine était très difficile à suivre.

Je remercie tout particulièrement mes parents, François et Lucille, pour leur soutien moral et financier tout au long de mes études.

Finalement, j'aimerais dédier cet ouvrage à la personne la plus chère dans ma vie, Sophie, sans qui je n'aurais pu compléter ce mémoire.

Première partie

La correction d'erreurs classiques et quantiques

Chapitre 1

Les codes correcteurs d'erreurs

Supposons qu'Alice désire transmettre une chaîne de bits $m = m_1 \dots m_k$ à Bob. Cependant, elle ne dispose que d'un canal bruyant (ou imparfait) à sens unique, i.e. une voie de communication telle que si elle envoie un bit b par le canal, alors avec une certaine probabilité $p < 1/2$, Bob recevra la négation $\bar{b}$ du bit (le bit restant intact avec probabilité $1-p$). De plus, Bob ne peut communiquer avec Alice d'aucune autre façon.

Comment peuvent-ils s'assurer que le message a bien été transmis ? Un moyen simple serait d'envoyer chaque bit du message plusieurs fois. Si Alice transmet chaque bit trois fois, Bob déduirait le bit en effectuant un vote majoritaire. Par exemple, si elle envoie 000 et que Bob reçoit 010 (une erreur ayant affecté le deuxième bit), il conclurait qu'Alice a transmis le bit 0. Cette procédure ne fonctionnera pas si deux ou trois erreurs affectent la transmission. Cette situation se produit avec probabilité

$$\binom{3}{2} p^2(1-p) + \binom{3}{3} p^3 = 3p^2 - 2p^3 \tag{1.1}$$

qui est inférieure à p si $p < 1/2$. Alice et Bob gagnent donc à encoder le message de cette façon. Ils peuvent augmenter leur confiance si Alice transmet $2n+1$ copie de b ($n = 1$ correspond au cas précédent). Ainsi, la probabilité que des erreurs affectent plus de $n+1$ bits tend vers 0 quand n tend vers l'infini.

Par contre, le prix à payer pour cette meilleure fiabilité est l'augmentation de la

longueur du message. Le taux de transmission, i.e. le rapport entre le nombre de bits encodés et le nombre de bits envoyés, devient $\frac{1}{2n+1}$, ce qui n'est pas très efficace.

La théorie des codes correcteurs étudie les façons de transmettre des messages sur des canaux imparfaits de sorte qu'on soit capable de les reconstituer avec grande probabilité. Le but est de trouver des procédures d'encodage $\mathcal{C}$ et de décodage $\mathcal{D}$ efficaces telles que si on envoie $\mathcal{C}(m)$ à travers le canal $\mathcal{X}$ et qu'on décode ensuite avec $\mathcal{D}$, on retrouve m avec bonne probabilité sans trop réduire le taux de transmission. Autrement dit, si $c = \mathcal{C}(m)$, $x = \mathcal{X}(c)$ et $m' = \mathcal{D}(x)$, alors on veut que la probabilité que $m = m'$ et que le ratio $\frac{|m|}{|c|}$ soient tous deux grands (i.e. le plus près de 1 possible).

Ce chapitre fait un survol de la théorie des codes. Les preuves des théorèmes qui y sont présentés seront omises pour éviter d'allonger inutilement le mémoire. On peut les retrouver dans des livres d'algèbre linéaire (comme [Gri90]) ou de théorie des codes (comme [Rom92, MS77]).

1.1 Les canaux de transmission

Un canal est la représentation mathématique de ce qui peut arriver à un message lorsqu'il est envoyé sur une voie de transmission. La description d'un canal comprend trois choses : 1) un alphabet d'entrée qui contient les symboles que l'on peut transmettre à travers le canal, 2) un alphabet de sortie qui contient les symboles d'entrée et possiblement d'autres symboles que l'on peut recevoir à l'autre bout du canal, et 3) une fonction probabiliste qui décrit comment les symboles d'entrée sont transformés par le canal. La plupart du temps, l'alphabet d'entrée est un corps fini[1]. Dans cet ouvrage, $\mathbb{F}_q$ représente un corps avec q éléments et $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, p étant premier, est le corps avec addition et multiplication modulo p.

Définition 1.1 Un *canal discret inconscient* est un triplet (P, A, S) où $A = \{a_1, \ldots, a_q\}$ est l'alphabet d'entrée, $S = \{s_1, \ldots, s_t\} \supseteq A$ est l'alphabet de sortie et $P : A \times S \to [0, 1] \subset \mathbb{R}$ donne la probabilité que le symbole $s \in S$ sorte du canal si

[1] Si vous n'êtes pas familier avec les corps finis, voir l'annexe A.

$a \in A$ y a été transmis, avec la contrainte que

$$\forall_{1 \leq i \leq q}, \sum_{j=1}^{t} P(a_i, s_j) = 1.$$

◇

Si on transmet une chaîne de bits à travers un canal, on suppose que chaque bit sera affecté de façon indépendante. Cette hypothèse n'est pas toujours réaliste, mais elle simplifie de beaucoup l'étude des codes correcteurs. Si un symbole est modifié par le canal, on dit qu'il y a eu une erreur de *canal* ou de *transmission.*

Le canal présenté dans l'introduction est très important et reviendra souvent au cours de ce mémoire. Il est formellement défini comme suit.

Définition 1.2 Le canal *binaire symétrique* BS_p est un canal $(P, \{0,1\}, \{0,1\})$ où

$$P(x,y) = \begin{cases} 1-p & \text{si } x = y \\ p & \text{si } x \neq y \end{cases}$$

En d'autres mots, le canal applique, avec une probabilité p, une négation au bit d'entrée. ◇

Un autre canal que nous utiliserons est le canal d'effacement, dans lequel les symboles ne sont pas modifiés, mais effacés.

Définition 1.3 Le *canal d'effacement* Σ_p est un canal $(P, A, A \cup \{\epsilon\})$ où

$$P(x,y) = \begin{cases} 1-p & \text{si } x = y \\ p & \text{si } y = \epsilon \end{cases}$$

Le canal, en recevant le symbole x, l'efface avec probabilité p ou le transmet intact avec probabilité complémentaire $1-p$. Dans ce cas particulier, une erreur de canal est appelée *erreur d'effacement.* ◇

Nous considérons les canaux comme des fonctions probabilistes qui, selon le contexte, s'appliquent sur un symbole ou sur une chaîne de symboles. Par exemple,

si $b \in \mathbb{Z}_2$ et $m = m_1 \dots m_n \in \mathbb{Z}_2^n$ (l'ensemble des chaînes[2] de n bits), alors

$$BS_p(b) = \begin{cases} \bar{b} & \text{avec probabilité } p \\ b & \text{avec probabilité } 1-p \end{cases}$$

et

$$BS_p(m) = BS_p(m_1) \dots BS_p(m_n) = m'_1 \dots m'_n = m'.$$

Avant de passer à la section suivante, définissons une notion utile qui nous permettra de décrire l'action du canal sur l'ensemble de la chaîne transmise. Notons que $\mathbb{F}_q^n$ est l'ensemble des chaînes de n éléments de $\mathbb{F}_q$.

Définition 1.4 Soit $\mathcal{X}$ un canal avec $\mathbb{F}_q$ comme alphabet d'entrée et de sortie. Si $m \in \mathbb{F}_q^n$ et que $x = \mathcal{X}(m)$, alors le *vecteur d'erreurs* est la chaîne $e \in \mathbb{F}_q^n$ telle que $x = m + e$ (l'addition se fait dans le corps $\mathbb{F}_q$, position par position). ◇

1.2 Codes correcteurs

Un code est un ensemble de mots représentant des messages. Les mots du code sont en général plus longs que les messages qu'ils représentent pour ajouter une certaine redondance à ceux-ci. Les symboles composant les mots du code sont tirés d'un alphabet précis. Un code est formellement défini comme suit.

Définition 1.5 Soit A un ensemble fini $\{a_1, \dots, a_q\}$. Un *code* q-aire est un sous-ensemble $C \subseteq A^n$ non-vide. Les éléments de C sont les *mots de code*. Si $|C| = M$, alors on dit que M est la *taille* du code, n est sa *longueur* et on dit que C est un code (n, M). Le *taux de transmission* d'un code q-aire est $R = (\log_q M)/n$ ($\log_q M$ étant la longueur nécessaire pour exprimer les M mots de code de façon minimale). ◇

Dans cet ouvrage, l'alphabet A sera toujours un corps fini. Avec un code (n, M), on peut encoder M messages différents. Pour expliquer comment la procédure de décodage

[2] On abuse la notation en considérant les chaînes de n bits comme des vecteurs de $\mathbb{Z}_2^n$.

fonctionnera, nous avons besoin de définir une notion importante, celle de la distance de Hamming.

Définition 1.6 Soit $x, y \in A^n$. La *distance de Hamming* entre x et y, notée $\Delta(x, y)$, est définie par

$$\Delta(x, y) = \Big|\{ i : x_i \neq y_i,\ 1 \leq i \leq n \}\Big|$$

En d'autres mots, $\Delta(x, y)$ est le nombre de positions où x et y diffèrent. ◇

Exemple 1.7 $\Delta(10011, 11010) = 2$, $\Delta(ab, ac) = 1$ et $\Delta(123, 123) = 0$. ◇

Il est facile de vérifier que (A^n, Δ) forme un espace métrique, i.e. pour $x, y, z \in A^n$, Δ vérifie les propriétés suivantes :

1) Δ est définie positive : $\Delta(x, y) \geq 0$ (et $= 0 \iff x = y$)
2) Δ est symétrique : $\Delta(x, y) = \Delta(y, x)$
3) Inégalité du triangle : $\Delta(x, y) \leq \Delta(x, z) + \Delta(z, y)$

Définition 1.8 La *distance minimale* d d'un code C, est la distance de Hamming minimale entre toutes les paires de mots de code, i.e.

$$d = \min_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} \Delta(c_1, c_2).$$

On dit d'un code avec longueur n, taille M et distance minimale d qu'il est un code (n, M, d). ◇

On peut maintenant voir comment un code peut aider à corriger des erreurs de transmission. Si on envoie $c \in C$ sur le canal $\mathcal{X}$ et que l'on reçoive la chaîne x, notre but est de déterminer quel c a été originalement transmis. Notre choix sera le $c \in C$ le plus probable.

Les canaux que nous étudierons ont la propriété que le mot le plus probable est celui qui est le plus "proche" de x, i.e. celui qui minimise la distance de Hamming entre lui et x.

Théorème 1.9 Pour les canaux BS_p et Σ_p, si on reçoit la chaîne x, le mot de code c le plus probable est celui qui minimise $\Delta(x, c)$. On dit alors que le décodage se fait à *distance minimale.* ◇

Notons que pour les codes avec distance minimale paire, la procédure de correction à distance minimale peut être ambiguë. Par exemple, si $C = \{0011, 1100\}$ avec distance minimale de 4, alors la chaîne $x = 1010$ est à la même distance de 0011 que de 1100. Cette égalité est reportée comme une erreur, i.e. on ne peut pas corriger x à un mot de C.

Définition 1.10 Un code C *détecte* t *erreurs* lorsque entre 1 et t erreurs affectent un mot de code, le résultat n'est pas un mot de code. Un code C *corrige* t *erreurs* si le décodage à distance minimale corrige jusqu'à t erreurs. ◇

Pour les codes où le décodage se fait à distance minimale, il y a un lien important entre la distance minimale du code et le nombre d'erreurs qu'il peut détecter et corriger.

Théorème 1.11 Soit d la distance minimale du code C, alors on a les relations suivantes :

1) C détecte exactement $d - 1$ erreurs,
2) C corrige exactement $\lfloor (d-1)/2 \rfloor$ erreurs.

◇

Exemple 1.12 Soit le code de répétition $R = \{000, 111\} \subset \mathbb{Z}_2^n$ (le même que dans l'introduction). Le code R a les paramètres $(3, 2, 3)$. Comme expliqué dans l'introduction, le décodage se fait par vote majoritaire, ce qui est un exemple de décodage à distance minimale. Le code R détecte 2 erreurs et peut en corriger 1. Ceci implique que si une ou deux erreurs de canal se produisent, on pourra dire que le x reçu a été modifié. Cependant, s'il y a eu deux erreurs, on ne pourra pas corriger, car on va décoder au mauvais mot de code. ◇

Avant de passer à la prochaine section, donnons quelques dernières définitions sur les codes.

Définition 1.13 Soit C un code (n, M, d). À chaque mot c de C, on associe une *sphère* qui contient toutes les chaînes à distance de Hamming non supérieure à $(d-1)/2$ de c, i.e.

$$\Phi_c = \{\, x \in A^n \,:\, \Delta(x, c) \leq (d-1)/2 \,\}.$$

On dit que le code C est *parfait* si

1) $\forall_{x \in A^n} \exists_{c \in C}$ tel que $x \in \Phi_c$, et

2) $\forall_{x \in A^n} \nexists_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}}$ tels que $x \in \Phi_{c_1}$ et $x \in \Phi_{c_2}$.

Autrement dit, un code est parfait si toutes les sphères de distance de Hamming $(d-1)/2$ entourant les mots de codes sont disjointes et couvrent complètement A^n. Notons que si d est pair, alors le code ne peut pas être parfait. ◇

Exemple 1.14 Le code de répétition R est parfait, car

$$\Phi_{000} = \{000, 001, 010, 100\},$$
$$\Phi_{111} = \{111, 110, 101, 011\},$$
$$\Phi_{000} \cap \Phi_{111} = \emptyset \quad \text{et} \quad \Phi_{000} \cup \Phi_{111} = \mathbb{Z}_2^3.$$

◇

Exemple 1.15 Soit $\Pi = \{\, a_1 \ldots a_8 \,:\, a_i \in \mathbb{Z}_2 \text{ et } \sum_{i=1}^{8} a_i \pmod 2 = 0 \,\}$. Le code de parité Π est un code $(8, 2^7, 2)$ qui peut détecter une erreur mais qui ne peut pas en corriger. Le code Π n'est pas parfait, car sa distance minimale est paire. Ce code est souvent utilisé dans les mémoires d'ordinateurs. On fixe le dernier bit d'un octet pour que la somme modulo 2 des bits soit 0. Ainsi, en lisant l'octet, on peut détecter si une erreur a affecté un des bits. En effet, si un bit a été changé, la parité sera 1. On ne peut cependant pas savoir quel bit est erroné. ◇

1.3 Codes linéaires

La classe de codes la plus importante est celle des codes linéaires. (Tous les codes correcteurs quantiques que nous définirons seront linéaires.) Les codes linéaires corres-

pondront à des espaces vectoriels, ce qui leur donnera une structure importante. Pour ce qui suit, on suppose que l'alphabet du code est le corps $\mathbb{F}_q$.

Définition 1.16 Un code $C \subseteq \mathbb{F}_q^n$ est *linéaire* si C est un sous-espace vectoriel de $\mathbb{F}_q^n$. Soit $k < n$ la dimension de C, alors on dit que C est un code $[n, k]$, et si d est sa distance minimale, on dit alors que c'est un code $[n, k, d]$. ◇

On utilise les crochets pour exprimer qu'un code est linéaire. Notons que tout code linéaire $[n, k, d]$ est un code (n, q^k, d). On remarque que la chaîne $0^n = \overbrace{0 \ldots 0}^{n}$ fait partie de tout code linéaire C. De plus, pour $c_1, c_2 \in C$, l'addition des deux mots de code est aussi un mot de code, i.e. $c_1 + c_2 \in C$. De plus, la taille et le taux de transmission sont $M = q^k$ et $R = k/n$, respectivement.

Définition 1.17 Le *poids* $w(x)$ pour $x \in \mathbb{F}_q^n$ est le nombre de positions où $x_i \neq 0$. Le poids w_C d'un code linéaire C est le poids minimal de tous les $c \in C - \{0^n\}$. ◇

Notons que pour $x, y \in \mathbb{F}_q^n$, $\Delta(x, y) = w(x - y)$.

Proposition 1.18 La distance minimale d d'un code C est égale au poids minimal w_C du code. ◇

Puisque C est un sous-espace vectoriel, il suffit de donner une base pour le décrire. La plupart du temps, on la représente sous forme de matrice où chaque ligne est un vecteur de base. Nous choisissons comme convention qu'une chaîne $x_1 \ldots x_n$ est équivalente, selon le contexte, au vecteur ligne $(x_1, \ldots, x_n)$.

Définition 1.19 Soit C un code $[n, k]$. Une *matrice génératrice* G de C est une matrice $k \times n$ telle que les lignes de G forment une base pour C. Ainsi,

$$C = \{ mG : m \in \mathbb{F}_q^k \}.$$

◇

Avant de continuer, présentons quelques notations que nous utiliserons avec les matrices. Le symbole $M_{i \times j}$ représente une matrice de i lignes et de j colonnes, M_{ij}

désigne l'élément de la matrice M à la i^{e} ligne et à la j^{e} colonne et M^{T} représente la matrice transposée de M. La matrice identité $j \times j$ est notée I_j. Soit deux matrices $A_{i\times j}$ et $B_{i\times k}$, la matrice $M_{i\times(j+k)} = (A_{i\times j}|B_{i\times k})$ est formée en concaténant B à la droite de A. Soit deux matrices $A_{i\times k}$ et $B_{j\times k}$, la matrice $M_{(i+j)\times k} = \left(\frac{A_{i\times k}}{B_{j\times k}}\right)$ est formée en concaténant B sous la matrice A. Pour $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, on note le produit scalaire entre x et y par

$$x \cdot y = x_1 y_1 + \ldots + x_n y_n$$

où les additions et les multiplications se font dans $\mathbb{F}_q$.

Définition 1.20 Une matrice génératrice $G_{k\times n}$ est en *forme standard* si la sous-matrice $G_{k\times k}$ est la matrice identité $k \times k$, i.e. si $G = (I_k|M_{k\times(n-k)})$. Pour toute matrice génératrice G, il existe une matrice génératrice en forme standard G' qui génère le même code. ◇

La matrice génératrice G d'un code C $[n, k]$ permet d'encoder les chaînes de $\mathbb{F}_q^k$ par des mots de code de C. Si G est en forme standard, l'encodage se fait simplement en ajoutant $n - k$ bits (qui dépendent des $n - k$ dernières colonnes de G) au k bits de la chaîne initiale. L'interprétation d'un mot de code $c \in C$ consiste à faire l'opération inverse, i.e. trouver le $m \in \mathbb{F}_q^k$ tel que $mG = c$. Encore une fois, si G est en forme standard, l'interprétation se fait en ne gardant que les k premiers bits de c.

Tout code linéaire C est lié à un code qui lui est orthogonal, qu'on appelle le code dual de C. La définition de ce code suit.

Définition 1.21 Soit C un code $[n, k]$. Le *code dual* de C, noté $C^\perp$, est défini par

$$C^\perp = \{\, x \in \mathbb{F}_q^n \;:\; x \cdot c = 0,\, \forall_{c\in C} \,\}$$

◇

Théorème 1.22 Soit $C \subseteq \mathbb{F}_q^n$ un code $[n, k]$ avec matrice génératrice G, alors on a que

1) $C^\perp = \{\, x \in \mathbb{F}_q^n \;:\; xG^{\mathsf{T}} = 0^k \,\}$,

2) $C^{\perp}$ est un code $[n, n-k]$,

3) $C^{\perp\perp} = C$.

◇

Définition 1.23 Un code $[n,k]$ C est *semi-dual* si $C^{\perp} \subset C$ (i.e. $C^{\perp} \subseteq C$ mais $C \nsubseteq C^{\perp}$). Le code est *auto-dual* si $C^{\perp} = C$. Dans ce cas, la longueur n doit être paire et on a nécessairement que $k = n/2$. ◇

La matrice génératrice nous donne un moyen d'encoder les codes linéaires. Nous verrons maintenant comment les décoder (i.e. corriger les erreurs afin de retrouver le message original). Soit $G = (I_k|M)$ la matrice génératrice d'un code $[n,k]$ C. Considérons la matrice

$$P = (-M^{\mathsf{T}}|I_{n-k}).$$

On vérifie facilement que les lignes de P sont orthogonales aux lignes de G. En effet, si X_{i*} est la i^{e} ligne de la matrice X, alors

$$G_{i*}(P_{i*})^{\mathsf{T}} = (I_{i*}|M_{i*})\,(-M_{i*}^{\mathsf{T}}|I_{i*})^{\mathsf{T}} = -M_{ii} + M_{ii} = 0.$$

De plus, puisque $\text{rang}(P) = n-k = \dim(C^{\perp})$, alors P est une matrice génératrice pour $C^{\perp}$, le dual de C.

Définition 1.24 Une matrice P de dimension $(n-k) \times n$ est une *matrice de parité*[3] pour le code C si on a que $c \in C \iff cP^{\mathsf{T}} = 0^{n-k}$. ◇

Définition 1.25 Soit C un code $[n,k]$, P une matrice de parité pour C et $x \in \mathbb{F}_q^n$. Le *syndrome* de x, noté $\text{Syn}(x)$, est le vecteur xP^{T}. Un *coset* est un ensemble de vecteurs de même syndrome, i.e.

$$\text{Coset}(x) = \{\, y \in \mathbb{F}_q^n \,:\, \text{Syn}(x) = \text{Syn}(y) \,\}$$

◇

[3]Dans la littérature, H est souvent utilisé pour représenter les matrices de parité. Mais nous réservons le H pour un autre usage dans cet ouvrage.

On remarque que, pour tout code linéaire C, $\text{Coset}(0^n) = C$ et que $\text{Coset}(x) = \{ y \in \mathbb{F}_q^n \ : \ \exists_{c \in C} \ x = y + c \}$. De plus, notons que si c est le mot transmis, e est le vecteur d'erreurs appliquées par le canal et x est la chaîne reçue (i.e. $x = c + e$), alors le syndrome de x est

$$\text{Syn}(x) = \text{Syn}(c + e) = (c + e)P^{\mathsf{T}} = cP^{\mathsf{T}} + eP^{\mathsf{T}} = eP^{\mathsf{T}}$$

(car $cP^{\mathsf{T}} = 0^{n-k}$). Ainsi, le calcul du syndrome ne nous apprend rien sur le mot de code envoyé, il nous renseigne seulement sur l'erreur survenue (ceci aura une importance primordiale dans les codes correcteurs quantiques). Une fois le syndrome calculé, on peut corriger l'erreur (si le syndrome n'est pas 0^{n-k}) en cherchant le mot de code le plus près de x. On peut montrer que le mot de code à distance minimale de x est $x - y$ où y est le vecteur de poids minimal[4] dans le coset de x (i.e. pour $u \in \text{Coset}(x)$, $w(u) \leq w(y)$). La figure 1.1 décrit comment encoder les messages et détecter/corriger les erreurs avec un code linéaire.

Nous présentons maintenant un théorème qui nous sera utile au chapitre 4.

Théorème 1.26 Soit C_1 et C_2 deux codes linéraires (avec paramètres $[n, k_1]$ et $[n, k_2]$ respectivement) tels que $C_2 \subset C_1$. Il existe une matrice de parité P_2 du code C_2 qui peut être écrite comme

$$P_2 = \left(\frac{P_1}{K} \right)$$

où P_1 est une matrice de parité du code C_1 et K est une matrice $(k_1 - k_2) \times n$. ◇

Nous énonçons maintenant un lemme qui sera utile dans la section 3.4.

Lemme 1.27 Soit C un code $[n, k]$ et son dual $C^{\perp}$. Alors,

$$\sum_{v \in C} (-1)^{v \cdot u} = \begin{cases} 2^k & \text{si } u \in C^{\perp} \\ 0 & \text{si } u \notin C^{\perp} \end{cases} . \tag{1.2}$$

◇

[4] Le problème de trouver le vecteur de poids minimal d'un coset est très difficile (c'est un problème NP-complet). L'efficacité de la procédure de décodage est un facteur important dans le choix d'un code linéaire.

Soit un canal $\mathcal{X}$ à décodage à distance minimale. Soit $m \in \mathbb{F}_q^k$ le message à transmettre. Soit C un code $[n, k, d]$ avec matrice génératrice G et matrice de parité P.

Encodage : Alice calcule $c = mG$.

Transmission : Alice envoie c sur le canal, Bob reçoit $x = c + e = \mathcal{X}(c)$.

Décodage : Bob calcule $\mathrm{Syn}(x) = xP^{\mathsf{T}}$.

Détection : Si $\mathrm{Syn}(x) \neq 0^{n-k}$, alors Bob sait qu'une erreur est survenue (fonctionne toujours si $w(e) \leq d - 1$).

Correction : Bob cherche le vecteur d'erreurs y de poids minimal dans le $\mathrm{Coset}(x)$. Il corrige alors les erreurs en calculant $c' = x - y$ (fonctionne toujours si $w(e) \leq \lfloor (d-1)/2 \rfloor$).

Interprétation : Bob déduit le message d'Alice en cherchant l'unique $m' \in \mathbb{F}_q^k$ tel que $m'G = c'$ (si G est en forme standard, $m' = c'_1 \dots c'_k$). Si $m' \neq m$ (i.e. si $y \neq e$), il y a eu erreur de décodage.

FIG. 1.1: Procédure d'encodage et de décodage pour les codes linéaires

Preuve Tout d'abord, si $u \in C^{\perp}$, alors par définition $v \cdot u = 0$ pour tout $v \in C$. Donc, trivialement, $\sum_{v \in C}(-1)^{v \cdot u} = 2^k$.

Pour $w \in \{0,1\}^k$ non-nul ($w \neq 0^k$), il est facile de vérifier que

$$\sum_{v \in \{0,1\}^k} (-1)^{v \cdot w} = 0. \tag{1.3}$$

Soit G la matrice génératrice de C. Tout mot de code $v \in C$ peut être exprimé comme $v = mG$ pour un $m \in \{0,1\}^k$. Considérons un $u \notin C^{\perp}$. Puisque G est la matrice de parité de $C^{\perp}$, alors $uG^{\mathsf{T}} \neq 0^k$. On obtient donc que

$$\begin{aligned}
\sum_{v \in C}(-1)^{v \cdot u} &= \sum_{v \in C}(-1)^{vu^{\mathsf{T}}} \\
&= \sum_{m \in \{0,1\}^k} (-1)^{mGu^{\mathsf{T}}} \\
&= \sum_{m \in \{0,1\}^k} (-1)^{m(uG^{\mathsf{T}})^{\mathsf{T}}} \\
&= \sum_{m \in \{0,1\}^k} (-1)^{m \cdot (uG^{\mathsf{T}})} = 0 \quad \text{par l'équation 1.3.}
\end{aligned}$$

□

Nous définirons maintenant une classe importante de codes : les codes cycliques. La définition et la remarque qui suivent contiennent des notions avancées d'algèbre que le lecteur ne possède peut-être pas. Elles ne sont présentées que pour donner une idée au lecteur avancé de quoi ont l'air les codes BCH (présenté à la section 1.5). On peut sauter à la section suivante sans influencer la compréhension du reste du mémoire.

Définition 1.28 Un code linéaire C est *cyclique* si pour $c = (c_1, c_2, \ldots, c_n) \in C$ alors $c' = (c_n, c_1, \ldots, c_{n-1}) \in C$. Alternativement, on peut définir les codes cycliques de façon algébrique. Soit $A_n = \mathbb{F}_q[x]/(x^n - 1)$ l'anneau des polynômes à coefficients dans $\mathbb{F}_q$ modulo $x^n - 1$. On associe à chaque chaîne $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ le polynôme $c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1}x^{n-1} \in A_n$. On dira alors qu'un code cyclique C est formé de polynômes de A_n plutôt que d'éléments de $\mathbb{F}_q^n$.

Voici donc une autre définition pour les codes cycliques. Un code linéaire C est *cyclique* si C est un idéal[5] de A_n. ◇

[5]Un idéal I d'un anneau A est un sous-espace linéaire de A tel que si pour tout $i \in I$ et $a \in A$, on

Remarque 1.29 Soit C un code cyclique de A_n. Soit q un nombre relativement premier à n et soit m le plus petit entier tel que $q^m \equiv 1 \pmod{n}$. Soit α une n^{e} racine primitive de l'unité de $\mathbb{F}_{q^m}$ (si γ est un élément primitif de $\mathbb{F}_{q^m}$, alors $\alpha = \gamma^{(q^m-1)/n}$ sera une n^{e} racine primitive).

1) Il existe un *polynôme générateur* $g(x)$ tel que $C = \{\, g(x)r(x) \,:\, r(x) \in A_n \,\}$.

2) Il existe un *polynôme de parité* $p(x)$ tel que $C = \{\, c(x) \,:\, c(x)p(x) \equiv 0 \pmod{x^n - 1} \,\}$. Notamment, $p(x) = (x^n - 1)/g(x)$ est un polynôme de parité.

3) Le code dual $C^\perp$ est aussi un code cyclique et est généré par $g(x)^\perp = x^{\deg(p)}p(1/x)$.

4) Il existe $K \subseteq \{0, 1, \ldots, n-1\}$ tel que $g(x) = \prod_{i \in K}(x - \alpha^i)$ et $p(x) = \prod_{i \notin K}(x - \alpha^i)$.

5) Le coset cyclotomique de $s \pmod{n}$ est

$$C_s = \{s, sq, sq^2, \ldots, sq^{m_s}\}$$

où m_s est le plus petit entier tel que $sq^{m_s} \equiv s \pmod{n}$. Le polynôme minimal de α^s est

$$M^{(s)}(x) = \prod_{i \in C_s} (x - \alpha^i).$$

◇

1.4 Exemple : le code de Hamming

Nous verrons dans cette section un code simple ayant une grande importance. En effet, il est à la base d'un code quantique que nous verrons dans la section 3.3.

Exemple 1.30 Soit $H \subset \mathbb{Z}_2^7$ le code linéaire binaire généré par

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{1.4}$$

a que $ai \in I$.

i.e.

$$\begin{aligned} H &= \{mG : m \in \mathbb{Z}_2^4\} \\ &= \{0000000, 0001111, 0010110, 0011001, 0100101, 0101010, 0110011, 0111100, \\ &\quad 1000011, 1001100, 1010101, 1011010, 1100110, 1101001, 1110000, 1111111\} \end{aligned}$$

H est le code [7,4,3] binaire de Hamming[6]. Puisque G est en forme standard, l'encodage d'une chaîne $m = m_1m_2m_3m_4$ se fait très simplement :

$$mG = c = m_1m_2m_3m_4 \,|\, m_2 + m_3 + m_4 \,|\, m_1 + m_3 + m_4 \,|\, m_1 + m_2 + m_4$$

Une matrice de parité de H est

$$P = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{1.5}$$

On remarque que la i^{e} colonne de P est la représentation binaire du nombre i.

La distance minimale du code de Hamming est 3. En effet, si $u, v \in H$, alors $\Delta(u, v) \geq 3$ et pour $u = 0001111$, $v = 0101010 \in H$, $\Delta(u, v) = 3$. Le code de Hamming permet donc de détecter 2 erreurs et d'en corriger une. Soit $e^{(i)}$ le vecteur de longueur n avec un 1 en i^{e} position et des 0 partout ailleurs ($e^{(i)}$ représente un vecteur d'erreurs où une seule erreur est survenue en position i). Si $x = BS_p(c) = c + e^{(i)}$, alors $\text{Syn}(x) = e^{(i)}P^{\mathsf{T}} = (P_{*i})^{\mathsf{T}}$, où P_{*i} est la i^{e} colonne de P. Si on interprète ce vecteur ligne comme un nombre binaire, alors le syndrome nous indique la position i de l'erreur. On peut donc facilement la corriger en appliquant une négation sur le bit i (on voit que la procédure de décodage du code de Hamming est très efficace).

Par exemple, supposons que $m = 0101$. Alors $c = mG = 0101010$. Si $x = 0101\underline{1}10$ (une erreur ayant affecté le 5^{e} bit), alors $\text{Syn}(x) = xP^{\mathsf{T}} = 101$. Puisque $101_2 = 5$, on inverse le 5^{e} bit pour retrouver le c original.

Par contre, si plus d'une erreur affectent un mot de code, la procédure de décodage ne fonctionnera pas. Par exemple, si $c = 0111100$ et que $x = 0\underline{0}111\underline{0}1$, alors $\text{Syn}(x) =$

[6]Les codes de Hamming sont en fait une famille plus large que le code présenté ici. Voir l'exemple 1.35.

101. Si on applique une négation sur le 5^{e} bit, on ajoute une erreur supplémentaire et on obtient $c' = 0\underline{0}11\underline{0}0\underline{1} \neq c$.

Est-ce que le code de Hamming nous aide à protéger les messages des effets du canal BS_p ? Si on transmet les quatre bits du message directement, le message sera erroné si au moins une erreur affecte le message, ce qui survient avec probabilité

$$p_{\text{direct}} = 1 - \text{prob(aucune erreur)} = 1 - (1-p)^4.$$

Par contre, si on encode le message avec le code de Hamming, une erreur de décodage survient si plus d'une erreur affectent la transmission, ce qui se produit avec probabilité

$$p_{\text{Hamming}} = 1 - \text{prob(aucune ou une seule erreur)} = 1 - \Big((1-p)^7 + 7p(1-p)^6\Big).$$

On peut vérifier que $p_{\text{Hamming}} < p_{\text{direct}}$ pour $0 < p < 1/2$. Par exemple, si $p = 0,1$, alors $p_{\text{direct}} \approx 0,344$ et $p_{\text{Hamming}} \approx 0,15$.

H nous protège moins bien que le code de répétition R. En effet, en encodant indépendamment chaque bit du message par un bloc de 3 bits, on obtient une erreur de décodage si un des blocs contient au moins deux erreurs. Comme calculée dans l'équation 1.1, la probabilité d'erreur de décodage d'un bloc est $3p^2 - 2p^3$, donc la probabilité d'erreur de décodage du message est

$$p_{\text{rép}} = 1 - \text{prob(aucun bloc est mal décodé)} = 1 - \Big(1 - (3p^2 - 2p^3)\Big)^4.$$

On a que $p_{\text{rép}} < p_{\text{Hamming}}$ pour $0 < p < 1/2$ (par exemple, pour $p = 0,1$, $p_{\text{rép}} \approx 0,107$). Cependant, le taux de transmission de H, qui est de $4/7$, est meilleur que le taux de $1/3$ de R. ◇

Exemple 1.31 Le dual $H^{\perp}$ du code de Hamming H est un code $[7,3,4]$ généré par la matrice de parité P de H. Le code $H^{\perp}$ contient tous (et seulement) les mots de code de H de poids pair :

$$\begin{aligned} H^{\perp} &= \{\, u \in H \,:\, w(u) \text{ est pair} \,\} \\ &= \{0000000, 1010101, 0110011, 1100110, 0001111, 1011010, 0111100, 1101001\} \end{aligned}$$

Donc, $H^{\perp} \subset H$. Ce fait sera utile dans la section 3.4. La matrice de parité du code $H^{\perp}$ est G, la matrice génératrice du code H. Cependant, selon le théorème 1.26, on peut la transformer sous la forme

$$G' = \left(\frac{P}{K}\right) = \left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \tag{1.6}$$

(G est la forme échelonnée réduite de G'). De plus, pour $c \in H$, on a que $cG'^{\mathsf{T}} = 000\kappa$ avec $\kappa = 0$ si $c \in H^{\perp}$ et $\kappa = 1$ sinon. ◇

1.5 Bornes et codes efficaces

Les codes que nous avons vus jusqu'à maintenant ne sont pas très efficaces. Nous verrons dans cette section quelques constructions qui donnent des codes qui permettent de corriger plusieurs erreurs en gardant de bons taux de transfert. On cherche des codes $[n, k, d]$ avec de grands k et de grands d. Cependant, on voit que la dimension et la distance minimale d'un code linéaire sont deux valeurs conflictuelles, i.e. soit que le code a peu de mots de code et peut corriger beaucoup d'erreurs ou soit qu'il possède plusieurs mots de code et peut corriger moins d'erreurs. Il existe des bornes qui décrivent les paramètres optimaux pour un code.

Théorème 1.32 (Borne de Hamming) Soit C un code linéaire $[n, k, 2t+1]$. Alors

$$\sum_{i=0}^{t} \binom{n}{i} \leq 2^{n-k} \tag{1.7}$$

◇

Théorème 1.33 (Borne de Singleton) Soit C un code linéaire $[n, k, d]$. Alors

$$n - k \geq d - 1 \tag{1.8}$$

◇

Théorème 1.34 (Borne de Gilbert-Varshamov) Il existe un code C $[n, k, d]$ tel que

$$\sum_{i=0}^{d-2} \binom{n-1}{i} \leq 2^{n-k} \tag{1.9}$$

◇

Certaines constructions permettent de créer des codes efficaces. Sans donner trop de détails[7], voici quelques exemples de familles importantes de codes linéaires.

Exemple 1.35 Les codes de *Hamming* généralisés $H_q(r)$ forment une famille de codes linéaires parfaits très importante. Pour $r > 0$, $H_q(r)$ est un code $[n, k, d]$ q-aire où

$$n = \frac{q^r - 1}{q - 1},\ k = n - r,\ \text{et}\ d = 3.$$

Les codes de Hamming les plus répandus sont les codes binaires $[2^r - 1, 2^r - r - 1, 3]$. Leur construction est simple. La matrice de parité du code $H_2(r)$ est :

$$P[H_2(r)] = \left(\beta_r^{\mathsf{T}}(1) \,\middle|\, \ldots \,\middle|\, \beta_r^{\mathsf{T}}(r) \right)$$

où $\beta_r(i)$ est la représentation binaire de i sur r bits. Notons que le code H présenté à la section 1.4 est en fait un code de Hamming $H_2(3)$. Les codes de Hamming furent découverts par Marcel Golay (en 1949) et indépendamment par Richard Hamming (en 1950). ◇

Exemple 1.36 Quatre codes forment la famille *Golay*. Il y a deux codes binaires : G_{24}, un code $[24, 12, 8]$, et G_{23}, un code cyclique parfait $[23, 12, 7]$ (obtenu en perforant G_{24}, i.e. en enlevant n'importe quelle coordonnée du code G_{24}). Le code G_{24} est auto-dual et G_{23} est semi-dual. Le code G_{24} a une importance théorique et pratique importante. En effet, il a été utilisé par la sonde Voyager pour transmettre des images de Jupiter et de Saturne jusqu'à la Terre. Les codes ternaires G_{11} (code parfait de paramètres $[11, 6, 5]$) et G_{12} (de paramètres $[12, 6, 6]$) complètent la famille des codes de Golay. Les codes de Golay ont été introduits par Marcel Golay en 1948. ◇

[7] Pour un traitement plus détaillé, voir [Rom92].

Exemple 1.37 Les codes de *Reed-Muller* sont des codes semi-duaux binaires efficaces (i.e. facile à décoder). Pour r et m tels que $0 \leq r \leq m$, le code de Reed-Muller $R(r,m)$ a les paramètres suivants :

$$n = 2^m,\ k = 1 + \binom{m}{1} + \ldots + \binom{m}{r},\ \text{et}\ d = 2^{m-r}.$$

Les codes de Reed-Muller sont construits à l'aide de polynômes binaires (i.e. à coefficients dans $\mathbb{Z}_2$). Le code $R(1,5)$ a été utilisé pour transmettre des images de Mars captées par la sonde Mariner 9. ◇

Exemple 1.38 La famille des codes BCH (découverts par Bose et Ray-Chaudhuri en 1960 et indépendamment par Hocquenghem en 1959) est une généralisation des codes de Hamming. Les codes BCH sont des codes cycliques avec de bonnes propriétés et sont à la base de plusieurs autres constructions de codes. Pour ce qui suit, on utilise la notation et les notions introduites dans la remarque 1.29. On définit un code BCH de longueur n, pour $b \geq 0$, sur $\mathbb{F}_q$ avec un polynôme générateur

$$g(x) = \text{ppcm}\Big\{M^{(b)}(x), M^{(b+1)}(x), \ldots, M^{(b+\delta-2)}(x)\Big\},$$

On dit alors que δ est la distance de design du code. La distance minimale du code est au moins δ et la dimension est $k = n - \deg(g) \geq n - m(\delta - 1)$. ◇

Exemple 1.39 Les codes de *Reed-Solomon* sont des codes BCH de longueur $n = q - 1$. Un polynôme générateur du code de distance de design δ est

$$g(x) = \prod_{i=0}^{\delta-2} M^{(b+i)}(x) = \prod_{i=0}^{\delta-2} (x - \alpha^{b+i}),$$

où α et $M^{(s)}(x)$ ont été définis dans la remarque 1.29. La dimension du code est $k = n - \delta + 1$ et sa distance minimale est $d = n - k + 1$. Les codes de Reed-Solomon ont été utilisés dans plusieurs programmes de la NASA, notamment dans les missions Galileo, Magellan et Ulysses. ◇

Chapitre 2

Introduction à la théorie de l'information quantique

Jusqu'à très récemment, l'informatique se contentait d'étudier le traitement d'information "classique", i.e. l'information stockée et traitée sur des systèmes régis par la mécanique newtonienne. Pourtant, on sait que la nature suit les lois de la mécanique quantique. La théorie de l'information quantique est un domaine récent dans lequel on se demande qu'est-ce qu'on peut faire comme traitement d'information si les systèmes étudiés sont quantiques. La réponse est surprenante : malgré le fait que l'information quantique ne peut être ni lue ni copiée, elle peut cependant être parallèlement dans plusieurs états à la fois, être téléportée, être corrélée avec d'autres systèmes éloignés, peut servir à faire de la cryptographie parfaitement sécuritaire, et semble permettre la télépathie. D'un point de vue plus algorithmique, un ordinateur quantique permettrait de résoudre certains problèmes plus rapidement que sa contrepartie classique. Par exemple, l'ordinateur quantique peut factoriser les grands entiers en temps polynomial [Sho97] et accélérer la recherche dans une base de données non-triée par un facteur quadratique [Gro97].

Ce chapitre présente les éléments de base de la théorie de l'information quantique. Pour un traitement plus général, consultez [NC00, Bra, Gru99, Per93].

2.1 Les états quantiques

Un système quantique est un système physique qui est régi par les lois de la mécanique quantique. Définissons maintenant la structure algébrique qui nous permettra de décrire les états quantiques.

Définition 2.1 Un *espace de Hilbert* est un espace vectoriel complexe avec produit scalaire. ◇

Pour le reste de ce chapitre, on suppose que l'on travaille sur l'espace de Hilbert $\mathbb{C}^{2^n}$, que l'on note $\mathcal{H}_2^n$ ou plus simplement $\mathcal{H}^n$ (ou $\mathcal{H}$ si $n = 1$) ; qui est un espace vectoriel de dimension 2^n. On verra dans la section 2.6 comment la théorie se généralise pour $\mathcal{H}_q^n = \mathbb{C}^{q^n}$.

On représente l'état d'un système quantique par un vecteur unitaire dans un espace de Hilbert. On dit alors que l'état est *pur*. Nous utilisons la notation de *Dirac* dans laquelle un vecteur $\vec{v} \in \mathcal{H}$ est noté par $|v\rangle$.

En informatique classique, l'unité d'information de base est le bit, qui peut prendre deux valeurs : 0 ou 1. Son analogue quantique sera un vecteur dans $\mathcal{H}$. Soit $\{|0\rangle, |1\rangle\}$ une base orthonormée de $\mathcal{H}$. Alors, tout vecteur $|\psi\rangle \in \mathcal{H}$ peut être écrit comme une combinaison linéaire $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, où les nombres complexes α et β sont les *amplitudes* de l'état. On peut maintenant définir l'unité de base en informatique quantique. Notons d'abord que pour $\alpha \in \mathbb{C}$, le conjugué de α est noté $\alpha^\star$ et sa norme est $|\alpha| = \sqrt{\alpha\alpha^\star}$.

Définition 2.2 Un *bit quantique*, ou *qubit*, est un rayon dans $\mathcal{H}$. On le représente par un vecteur normalisé[1] $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ dans $\mathcal{H}$, où $\alpha, \beta \in \mathbb{C}$ et $|\alpha|^2 + |\beta|^2 = 1$. ◇

Les vecteurs de base représentent les bits classiques 0 et 1. Le qubit est une superposition de ces deux états, i.e. il est 0 *et* 1 en même temps. Il est important de comprendre la différence entre une superposition et une interpolation (comme dans un ordinateur analogique). Dans cette dernière, on interpréterait le qubit comme étant une

[1] Nous utiliserons à quelques rares occasions des états quantiques non-normalisés. On notera ces derniers par $|\widetilde{\psi}\rangle$, pour ne pas les confondre avec les états normalisés.

valeur entre 0 et 1, ce qui est différent d'être dans l'état 0 avec une certaine proportion et d'être dans l'état 1 avec une proportion complémentaire qui dépend des amplitudes.

Plusieurs supports physiques permettent de représenter un qubit. Par exemple, le spin d'un électron peut être dans deux états classiques : haut $= |0\rangle$ et bas $= |1\rangle$. Un autre exemple plus facile à visualiser utilise la polarisation des photons. On choisit une base orthonormée arbitraire pour l'angle de polarisation, par exemple $|0\rangle = 0°$ et $|1\rangle = 90°$. Ainsi, la polarisation θ ($0° \leq \theta < 180°$) d'un photon arbitraire peut être représentée par une superposition des états de base.

On utilise souvent la représentation matricielle pour représenter les éléments de $\mathcal{H}$. Pour les vecteurs de base standard $|0\rangle$ et $|1\rangle$, on prend comme convention que

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

alors, pour un qubit arbitraire $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, on a que

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

En général, si on veut représenter l'état d'un ensemble de qubits, on doit combiner leur espaces de Hilbert avec un *produit tensoriel* (illustré plus bas). Si on combine un état $|\psi\rangle \in \mathcal{H}^m$ avec $|\phi\rangle \in \mathcal{H}^n$, alors l'état résultant sera $|\psi\rangle \otimes |\phi\rangle \in \mathcal{H}^{m+n} = \mathcal{H}^m \otimes \mathcal{H}^n$, où $\otimes$ est le produit tensoriel entre les états ou les espaces. Pour alléger la notation, on laisse souvent tomber le symbole $\otimes$ entre les états. Les notations suivantes sont équivalentes :

$$|\psi\rangle \otimes |\phi\rangle \equiv |\psi\rangle|\phi\rangle \equiv |\psi\phi\rangle.$$

De plus, si on veut combiner plusieurs états $|\psi_i\rangle$ (pour $1 \leq i \leq n$) par produit tensoriel, alors on peut utiliser la notation concise suivante :

$$\bigotimes_{i=1}^{n} |\psi_i\rangle \equiv |\psi_1\rangle \otimes \ldots \otimes |\psi_n\rangle$$

Si les $|\psi_i\rangle$ sont tous égaux, on peut aussi écrire $|\psi_i\rangle^{\otimes n}$. En combinant plusieurs qubits, on forme un registre quantique, défini comme suit.

Définition 2.3 Un *registre quantique* de n qubits est un vecteur normalisé dans $\mathcal{H}^n = \underbrace{\mathcal{H} \otimes \ldots \otimes \mathcal{H}}_{n}$, un espace de dimension 2^n. On peut le représenter par une combinaison linéaire des vecteurs de base $\{\, |x\rangle \,:\, x \in \{0,1\}^n \,\}$

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

où $\sum_x |\alpha_x|^2 = 1$. ◇

La base standard du nouvel espace $\mathcal{H}^n$ est la composition par produit tensoriel de la base standard $\{|0\rangle, |1\rangle\}$. Ainsi,

$$|x_1 \ldots x_n\rangle = |x_1\rangle \otimes \ldots \otimes |x_n\rangle \text{ où } x_i \in \{0,1\} \text{ pour } 1 \leq i \leq n.$$

Lorsqu'on considère les états comme des matrices, le produit tensoriel est équivalent au produit de Kronecker des matrices. Soit A et B deux matrices de dimension $a \times b$ et $c \times d$ respectivement. La produit de Kronecker de A et B est défini par

$$A \otimes B = \begin{pmatrix} A_{1,1}B & A_{1,2}B & \cdots & A_{1,b}B \\ A_{2,1}B & A_{2,2}B & \cdots & A_{2,b}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{a,1}B & A_{a,2}B & \cdots & A_{a,b}B \end{pmatrix}$$

où $A_{i,j}B$ est la matrice B, sans parenthèse, multipliée par le scalaire $A_{i,j}$. La matrice $A \otimes B$ est de dimension $ac \times bd$.

Par exemple, supposons que l'on veuille former un registre avec les qubits $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ et $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$. Le produit tensoriel donne

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= \big(\alpha|0\rangle + \beta|1\rangle\big) \otimes \big(\gamma|0\rangle + \delta|1\rangle\big) \\ &= \alpha\gamma\big(|0\rangle \otimes |0\rangle\big) + \alpha\delta\big(|0\rangle \otimes |1\rangle\big) + \beta\gamma\big(|1\rangle \otimes |0\rangle\big) + \beta\delta\big(|1\rangle \otimes |1\rangle\big) \\ &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle \in \mathcal{H}^2 \end{aligned} \tag{2.1}$$

où

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \; |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \; |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ et } |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix};$$

$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ étant la base standard de $\mathcal{H}^2$. L'équation 2.1 équivaut bien au produit de Kronecker

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix}.$$

Pour un vecteur $|\psi\rangle \in \mathcal{H}^n$, son vecteur conjugué transposé $|\psi\rangle^\dagger$ est noté $\langle\psi|$. Le produit scalaire entre deux états $|\psi\rangle$ et $|\phi\rangle$ est défini par $\langle\phi|\psi\rangle$ (i.e. la multiplication matricielle entre $\langle\phi|$ et $|\psi\rangle$. Puisque les vecteurs de bases sont orthogonaux, on a que, pour $x, y \in \{0,1\}^n$,

$$\langle x|y\rangle = \begin{cases} 1 & \text{si } x = y \\ 0 & \text{si } x \neq y \end{cases} \tag{2.2}$$

Le produit externe entre $|\psi\rangle$ et $|\phi\rangle$ est la matrice $|\phi\rangle\langle\psi|$. Par exemple, si $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ et $|\phi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$ (où $i = \sqrt{-1}$), alors

$$\langle\phi|\psi\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}}^\star & \frac{i}{\sqrt{2}}^\star \end{pmatrix} \begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix} = \frac{\sqrt{3} - i}{2\sqrt{2}}$$

et

$$|\phi\rangle\langle\psi| = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{3}}{2}^\star & 1^\star \\ \overline{2} & \overline{2} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{3}}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{i\sqrt{3}}{2\sqrt{2}} & \frac{i}{2\sqrt{2}} \end{pmatrix}.$$

2.2 Les opérations sur les états

Essentiellement, on peut faire deux choses avec un état $|\psi\rangle$: le transformer ou le mesurer. Nous verrons dans les deux prochaines sections comment fonctionnent ces opérations.

2.2.1 Les transformations unitaires

Une transformation est un opérateur linéaire qui envoie les vecteurs de $\mathcal{H}^n$ vers les vecteurs de $\mathcal{H}^n$. Il suffit de décrire son action sur les vecteurs de base pour savoir comment il opère sur un état en superposition. Par exemple, si on applique la transformation U suivante

$$\begin{aligned} |0\rangle &\xrightarrow{\mathsf{U}} \alpha|0\rangle + \beta|1\rangle \\ |1\rangle &\xrightarrow{\mathsf{U}} \gamma|0\rangle + \delta|1\rangle \end{aligned}$$

sur un qubit $|\psi\rangle = \zeta|0\rangle + \xi|1\rangle$, alors par linéarité, on obtient

$$\begin{aligned} \mathsf{U}|\psi\rangle &= \zeta\mathsf{U}|0\rangle + \xi\mathsf{U}|1\rangle \\ &= \zeta\big(\alpha|0\rangle + \beta|1\rangle\big) + \xi\big(\gamma|0\rangle + \delta|1\rangle\big) \\ &= \big(\zeta\alpha + \xi\gamma\big)|0\rangle + \big(\zeta\beta + \xi\delta\big)|1\rangle. \end{aligned}$$

Chaque transformation a une représentation matricielle. Par exemple, la transformation U définie plus haut est représentée par la matrice

$$\mathsf{U} = \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix}$$

et on vérifie bien que

$$\mathsf{U}|\psi\rangle = \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} \begin{pmatrix} \zeta \\ \xi \end{pmatrix} = \begin{pmatrix} \zeta\alpha + \xi\gamma \\ \zeta\beta + \xi\delta \end{pmatrix}.$$

Les lois de la mécanique quantique nous imposent une restriction sur le type de transformations que l'on peut faire subir aux états. Les opérateurs doivent être *unitaires.*

Définition 2.4 Soit U la représentation matricielle d'une transformation. On dit que U est *unitaire* $\iff \mathsf{U}^{-1} = \mathsf{U}^\dagger$ (où $\mathsf{U}^\dagger$ est la matrice conjuguée transposée de U). De façon équivalente, U est unitaire si, pour tout $|\psi\rangle, |\phi\rangle \in \mathcal{H}^n$ et si $|\psi'\rangle = \mathsf{U}|\psi\rangle$, $|\phi'\rangle = \mathsf{U}|\phi\rangle$; alors on a que $\langle\phi|\psi\rangle = \langle\phi'|\psi'\rangle$. ◇

L'unitarité des transformations est nécessaire afin d'assurer que les qubits garderont leur normalisation après la transformation. On peut appliquer une transformation unitaire sur un registre de plusieurs qubits. La définition donnée est valide pour les états de plus d'un qubit. Si on a deux transformations U et V agissant sur $\mathcal{H}^m$ et $\mathcal{H}^n$ respectivement, on peut les décrire comme étant une transformation sur un registre de $m+n$ qubits en effectuant leur produit tensoriel. En effet, si $|\psi\rangle \in \mathcal{H}^m, |\phi\rangle \in \mathcal{H}^n$, alors

$$(\mathsf{U} \otimes \mathsf{V})\big(|\psi\rangle \otimes |\phi\rangle\big) = \mathsf{U}|\psi\rangle \otimes \mathsf{V}|\phi\rangle.$$

Par exemple, si $|\psi\rangle \otimes |\phi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$ est l'état de l'équation 2.1 et si

$$\mathsf{U} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \text{ et } \mathsf{V} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

alors

$$\begin{aligned}
(\mathsf{U} \otimes \mathsf{V})\big(|\psi\rangle \otimes |\phi\rangle\big) &= \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix} \\
&= \begin{pmatrix} (\alpha+\beta)\gamma/\sqrt{2} \\ -(\alpha+\beta)\delta/\sqrt{2} \\ (\alpha-\beta)\gamma/\sqrt{2} \\ -(\alpha-\beta)\delta/\sqrt{2} \end{pmatrix} \\
&= \begin{pmatrix} (\alpha+\beta)/\sqrt{2} \\ (\alpha-\beta)/\sqrt{2} \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ -\delta \end{pmatrix} \\
&= \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \gamma \\ \delta \end{pmatrix} \\
&= \mathsf{U}|\psi\rangle \otimes \mathsf{V}|\phi\rangle
\end{aligned}$$

Si U est une transformation unitaire, alors $\mathsf{U}^{\otimes n}$ représente le produit tensoriel de U avec elle-même n fois, i.e. $\mathsf{U}^{\otimes n} = \underbrace{\mathsf{U} \otimes \ldots \otimes \mathsf{U}}_{n}$. Il ne faut pas confondre cette notation avec U^n qui représente la composition matricielle de U, i.e. $\mathsf{U}^n = \underbrace{\mathsf{U}\mathsf{U}\ldots\mathsf{U}}_{n}$ (en particulier,

$\mathsf{U}^0 = \mathrm{I}$, la matrice identité). Comme pour les états, on peut représenter le produit tensoriel de n transformation $\mathsf{U}_1, \ldots, \mathsf{U}_n$ par

$$\bigotimes_{i=1}^{n} \mathsf{U}_i \equiv \mathsf{U}_1 \otimes \ldots \otimes \mathsf{U}_n$$

Définition 2.5 Une *ancille* ou *système ancillaire* est un registre quantique qu'on ajoute à un autre pour faire un calcul quantique. ◇

Souvent, lors d'un calcul quantique, on ajoute des qubits dans un état connu (par exemple $|0\ldots0\rangle$) à un registre avant d'effectuer une transformation unitaire. L'ancille sert en quelque sorte d'espace de calcul supplémentaire.

2.2.2 Les mesures

La mesure est l'autre opération que l'on peut effectuer sur les qubits. La mesure d'un qubit est, en principe, un événement probabiliste.

Définition 2.6 Soit un état $|\psi\rangle \in \mathcal{H}^n$ et soit $\{|v_i\rangle\}$ une base orthogonale pour $\mathcal{H}^n$. On peut donc exprimer l'état comme étant une superposition des vecteurs de base, i.e.

$$|\psi\rangle = \sum_i \alpha_i |v_i\rangle$$

où $\sum_i |\alpha_i|^2 = 1$. Alors la *mesure complète* de $|\psi\rangle$ dans la base $\{|v_i\rangle\}$ à l'effet suivant :

1) Un des état de base $|v_i\rangle$ est choisi avec probabilité $|\alpha_i|^2$.

2) Le vecteur $|\psi\rangle$ *devient*[2] l'état $|v_i\rangle$, i.e. $|\psi\rangle = |v_i\rangle$.

3) La mesure nous retourne quel i a été choisi.

◇

La mesure la plus simple consiste à mesurer un qubit dans la base standard $\{|0\rangle, |1\rangle\}$, i.e. étant donné un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, la mesure retourne 0 avec probabilité $|\alpha|^2$ et l'état devient $|\psi\rangle = |0\rangle$, ou bien la mesure retourne 1 avec probabilité complémentaire $|\beta|^2$ et l'état devient $|1\rangle$. Pour que les probabilités somment à

[2] On dit que la superposition $|\psi\rangle$ *se réduit* à $|v_i\rangle$.

1, le qubit doit être de norme unitaire (c'est pourquoi on a cette restriction dans la définition du qubit).

Il est important de remarquer que la mesure modifie le qubit. Si on mesure un qubit arbitraire, on n'obtient à la fin qu'un vecteur de la base de mesure et l'état original est perdu. Notons aussi que si le qubit est un des vecteurs de cette base, alors la mesure ne modifie pas le qubit. En effet, si $|\psi\rangle = 1|0\rangle + 0|1\rangle = |0\rangle$, alors la mesure retourne 0 avec probabilité 1 et "transforme" le qubit à l'état $|0\rangle$, le laissant donc intact.

La mesure telle que décrite plus haut est valable pour les états de dimensions arbitraire. Par exemple, si $|\psi\rangle$ est l'état de l'équation 2.1, alors la mesure complète de $|\psi\rangle$ dans la base $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ donne 00 avec probabilité $|\alpha\gamma|^2$, 01 avec probabilité $|\alpha\delta|^2$, et ainsi de suite.

Une autre base importante dans laquelle nous pouvons mesurer est la base diagonale $\{|\nearrow\rangle, |\nwarrow\rangle\}$ où $|\nearrow\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ et $|\nwarrow\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ (nous verrons la significations de ces symboles au chapitre 4). Un qubit arbitraire $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ s'exprime, dans la base diagonale, par

$$|\nearrow\rangle\langle\nearrow|\psi\rangle + |\nwarrow\rangle\langle\nwarrow|\psi\rangle = \frac{\alpha+\beta}{\sqrt{2}}|\nearrow\rangle + \frac{\alpha-\beta}{\sqrt{2}}|\nwarrow\rangle. \tag{2.3}$$

Donc, si on mesure $|\psi\rangle$ dans la base diagonale, on obtient $|\nearrow\rangle$ avec probabilité $\left|\frac{\alpha+\beta}{\sqrt{2}}\right|^2$ et $|\nwarrow\rangle$ avec probabilité $\left|\frac{\alpha-\beta}{\sqrt{2}}\right|^2$.

Ces deux mesures seront les seules que nous utiliserons dans ce travail. En général, si on n'a aucune information sur un qubit $|\psi\rangle$, on ne peut pas le mesurer précisément.

Théorème 2.7 Soit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ un qubit arbitraire. Il n'existe pas de procédure (une combinaison de transformations et de mesures) qui peut déterminer les amplitudes α et β du qubit. ◇

Par contre, si on sait que l'état d'un registre a été choisi parmi un ensemble d'états orthogonaux, on peut apprendre lequel à coup sûr.

Proposition 2.8 Soit $|\psi\rangle \in \{|\psi_1\rangle, \ldots, |\psi_k\rangle\} \subset \mathcal{H}^n$ où $\langle\psi_i|\psi_j\rangle = 0$ pour $i \neq j$. Alors, il existe une mesure $\mathcal{M}$ qui distingue les $|\psi_i\rangle$ avec probabilité 1, i.e. la mesure retourne j si $|\psi\rangle = |\psi_j\rangle$. ◇

Si on a un registre entre les mains, on peut mesurer que certains qubits sans toucher aux autres. On dit alors qu'on fait une *mesure partielle* du registre. Si le registre est dans l'état $|\psi\rangle = |\psi_1\rangle \otimes \ldots \otimes |\psi_k\rangle$ (nous verrons dans la section 2.3 que certains états ne peuvent pas s'exprimer ainsi), alors la mesure de chaque qubit se fait indépendamment. Par exemple pour $k > 4$, si on mesure le premier et le troisième qubit et qu'on obtient 0 et 1 respectivement, alors l'état du registre devient

$$|\psi'\rangle = |0\rangle \otimes |\psi_2\rangle \otimes |1\rangle \otimes |\psi_4\rangle \otimes \ldots \otimes |\psi_k\rangle.$$

Par contre, si le registre est intriqué (voir section 2.3), la mesure de quelques qubits affecte l'état du reste du registre. Nous donnerons un exemple dans la section 2.3.

2.2.3 Non-clonage

Avant de terminer la section, énonçons un théorème important qui a plusieurs conséquences. En gros, le théorème dit qu'on ne peut pas copier (ou *cloner*) les états quantiques [WZ82].

Théorème 2.9 Il n'existe pas d'opération quantique qui, pour un état $|\psi\rangle$ arbitraire, transforme $|\psi\rangle$ en $|\psi\rangle \otimes |\psi\rangle$. ◇

Preuve Nous n'illustrerons que le cas unitaire (voir [WZ82] pour la preuve complète). Supposons donc qu'il existe une transformation unitaire C qui, appliquée sur $|\psi\rangle$, produit $|\psi\rangle \otimes |\psi\rangle$. Notons que si on applique C aux états de base $|0\rangle$ et $|1\rangle$, on obtient

$$\begin{aligned} |0\rangle &\xrightarrow{\mathsf{C}} |0\rangle|0\rangle \\ |1\rangle &\xrightarrow{\mathsf{C}} |1\rangle|1\rangle \end{aligned}$$

Si on applique C à la superposition $\alpha|0\rangle + \beta|1\rangle$, on obtient

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle \qquad (2.4)$$

Cependant, par linéarité, on a aussi que

$$\begin{aligned} \mathsf{C}(\alpha|0\rangle + \beta|1\rangle) &= \alpha\mathsf{C}|0\rangle + \beta\mathsf{C}|1\rangle \\ &= \alpha|00\rangle + \beta|11\rangle \end{aligned} \tag{2.5}$$

Puisque (2.4) $\neq$ (2.5), on obtient une contradiction. □

Voici un théorème relié très utile en cryptographie quantique. Il a été prouvé par [BBM92] et nous l'utiliserons dans la deuxième partie du mémoire.

Théorème 2.10 Soit $|\psi\rangle$ un état quantique arbitraire. Il n'existe pas d'algorithme quantique qui peut obtenir de l'information sur $|\psi\rangle$ sans le modifier. Sans perte de généralité, ceci est équivalent à dire qu'aucune transformation unitaire U, appliquée à $|\psi\rangle$ et à un état ancillaire $|a\rangle$, ne peut laisser $|\psi\rangle$ intact et transformer l'ancille pour qu'elle dépende de $|\psi\rangle$. ◇

Preuve On va montrer le théorème pour un cas particulier. Supposons que $|\phi\rangle$ et $|\psi\rangle$ sont deux états non-orthogonaux. Supposons qu'il existe une transformation unitaire U et un état ancillaire $|a\rangle$ tels que

$$\mathsf{U}\big(|\psi\rangle \otimes |a\rangle\big) = |\psi\rangle \otimes |a_\psi\rangle \text{ et } \mathsf{U}\big(|\phi\rangle \otimes |a\rangle\big) = |\phi\rangle \otimes |a_\phi\rangle$$

où $|a_\psi\rangle \neq |a_\phi\rangle$. Il est facile de vérifier que pour des vecteurs $|a\rangle, |b\rangle, |c\rangle, |d\rangle$,

$$\langle a \otimes c | b \otimes d\rangle = \langle a|b\rangle\langle c|d\rangle. \tag{2.6}$$

Par définition, l'application d'une transformation unitaire ne change pas le produit scalaire entre deux états. Donc, on a que

$$\begin{aligned} \langle\phi|\psi\rangle &= \langle\phi|\psi\rangle\langle a|a\rangle && (\text{car } \langle a|a\rangle = 1) \\ &= \langle\phi \otimes a|\psi \otimes a\rangle && (\text{par 2.6}) \\ &= \langle\phi \otimes a_\phi|\psi \otimes a_\psi\rangle && (\text{en appliquant } \mathsf{U}) \\ &= \langle\phi|\psi\rangle\langle a_\phi|a_\psi\rangle && (\text{par 2.6}) \end{aligned}$$

Puisque $|\phi\rangle$ et $|\psi\rangle$ sont non-orthogonaux, alors $\langle\phi|\psi\rangle \neq 0$. Donc on conclut que $\langle a_\phi|a_\psi\rangle = 1$ ce qui implique que les ancilles sont indistingables, i.e. $|a_\psi\rangle = |a_\phi\rangle$. □

2.3 L'intrication

Il arrive parfois qu'un état ne peut pas être représenté comme un produit tensoriel de qubits. Par exemple, considérons l'état

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle$$

sur deux qubits. $|\Psi^-\rangle$ est un état légitime de $\mathcal{H}^2$ (car $|1/\sqrt{2}|^2 + |1/\sqrt{2}|^2 = 1$). Est-ce qu'on peut trouver $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ et $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$ tels que $|\Psi^-\rangle = |\psi\rangle \otimes |\phi\rangle$? On aurait donc les conditions suivantes, imposées par l'équation 2.1 :

$$\alpha\gamma = 0 \tag{2.7}$$

$$\alpha\delta = \frac{1}{\sqrt{2}} \tag{2.8}$$

$$\beta\gamma = -\frac{1}{\sqrt{2}} \tag{2.9}$$

$$\beta\delta = 0 \tag{2.10}$$

Si on multiplie (2.7) et (2.10), on obtient que $\alpha\beta\gamma\delta = 0$. Si on multiplie (2.8) et (2.9), on obtient que $\alpha\beta\gamma\delta = -1/2$. Cette contradiction implique donc qu'on ne peut pas *factoriser* (i.e. séparer en produit tensoriel) l'état $|\Psi^-\rangle$. On dit alors que l'état est *intriqué.*

Définition 2.11 Soit $|\Gamma\rangle \in \mathcal{H}^d$. L'état $|\Gamma\rangle$ est *intriqué* s'il n'existe pas $|\psi\rangle \in \mathcal{H}^m$ et $|\phi\rangle \in \mathcal{H}^n$ tels que $|\Gamma\rangle = |\psi\rangle \otimes |\phi\rangle$ et $d = m + n$. Un état qui n'est pas intriqué est *séparable.* ◇

L'état $|\Psi^-\rangle$ a une importance capitale en informatique quantique. C'est l'un des quatre états de Bell, qui sont définis comme suit :

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle, \tag{2.11}$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle, \tag{2.12}$$

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle, \tag{2.13}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle. \tag{2.14}$$

Remarque 2.12 Les états de Bell forment une base orthonormée pour $\mathcal{H}^2$. En effet, tout $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \in \mathcal{H}^2$, peut être exprimé comme

$$|\Phi^+\rangle\langle\Phi^+|\psi\rangle + |\Phi^-\rangle\langle\Phi^-|\psi\rangle + |\Psi^+\rangle\langle\Psi^+|\psi\rangle + |\Psi^-\rangle\langle\Psi^-|\psi\rangle$$
$$= \frac{(\alpha+\delta)}{\sqrt{2}}|\Phi^+\rangle + \frac{(\alpha-\delta)}{\sqrt{2}}|\Phi^-\rangle + \frac{(\beta+\gamma)}{\sqrt{2}}|\Psi^+\rangle + \frac{(\beta-\gamma)}{\sqrt{2}}|\Psi^-\rangle \quad (2.15)$$

◇

La mesure partielle d'un état intriqué est plus complexe que si l'état était un produit tensoriel (comme on l'a vu dans la section 2.2.2). Donnons l'exemple d'une mesure d'un état de deux qubits. Si on veux mesurer le premier qubit de l'état $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, on réécrit l'état comme

$$|\psi\rangle = \sqrt{p_0}|0\rangle|\psi_0\rangle + \sqrt{p_1}|1\rangle|\psi_1\rangle,$$

où $p_0 = |\alpha|^2 + |\beta|^2$, $p_1 = |\gamma|^2 + |\delta|^2$, et

$$\begin{aligned} |\psi_0\rangle &= \frac{\alpha}{\sqrt{p_0}}|0\rangle + \frac{\beta}{\sqrt{p_0}}|1\rangle, \\ |\psi_1\rangle &= \frac{\gamma}{\sqrt{p_1}}|0\rangle + \frac{\delta}{\sqrt{p_1}}|1\rangle. \end{aligned}$$

Alors, la mesure du premier qubit donne 0 avec probabilité p_0 et l'état devient $|0\rangle\otimes|\psi_0\rangle$, ou elle donne 1 avec probabilité p_1 et l'état devient $|1\rangle \otimes |\psi_1\rangle$. Par exemple, si on veut mesurer le premier qubit de l'état $|\psi\rangle = \frac{\sqrt{3}}{4}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle - \frac{1}{4}|11\rangle$, on obtient 0 avec probabilité 7/16 ou on obtient 1 avec probabilité 9/16. De plus,

$$\begin{aligned} |\psi_0\rangle &= \frac{\sqrt{3}}{\sqrt{7}}|0\rangle + \frac{2}{\sqrt{7}}|1\rangle, \text{ et} \\ |\psi_1\rangle &= \frac{2\sqrt{2}}{3}|0\rangle - \frac{1}{3}|1\rangle. \end{aligned}$$

Cette méthode se généralise directement aux états de $\mathcal{H}^n$.

L'intrication est probablement l'élément qui donne le plus de puissance à l'ordinateur quantique. Considérons l'expérience suivante. Supposons qu'on ait deux particules (qubits) qui sont conjointement dans l'état $|\Phi^+\rangle$. Si on sépare physiquement les particules (l'état ne change pas) et qu'on mesure la première, alors elle retournera $b \in \{0, 1\}$

avec probabilité 1/2. La mesure est purement probabiliste. Cependant, si on mesure l'autre particule, la réponse sera toujours identique à b, malgré la séparation spatiale. Pourtant, si on avait mesuré la deuxième particule avant la première, le résultat aurait été aléatoire. L'intrication est en fait une corrélation qui lie les deux particules à travers l'espace. C'est ce lien étrange qui permet entre autres la téléportation des qubits [BBC$^+$93] et la pseudo-télépathie [BCT99]. On obtient le même résultat si on a plusieurs paires $|\Phi^+\rangle$:

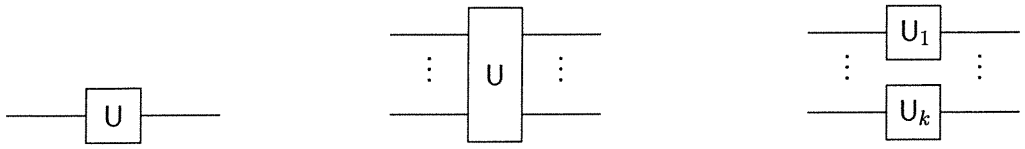
Proposition 2.13 Si n paires $|\Phi^+\rangle$ sont séparées entre Alice et Bob (i.e. pour chaque paire, Alice a la première particule (ou premier qubit) et Bob a la deuxième), alors la mesure complète de chaque registre dans la base standard donnera la même chaîne de bits et cette chaîne sera aléatoire. ◇

L'intrication est en grande partie responsable des erreurs qui affectent les qubits lors de leur transmission à travers un canal quantique. Cependant, nous verrons au prochain chapitre comment l'intrication peut servir à protéger les qubits des dommages qu'elle cause !

2.4 Portes et circuits quantiques

On décrit souvent un calcul quantique à l'aide d'un circuit. Nous prenons la peine de décrire les circuits, car ils aident à visualiser les opérations effectuées sur les registres quantiques. De plus, en donnant un circuit associé à une transformation complexe, on peut évaluer la complexité de cette transformation. Un circuit quantique ressemble en plusieurs points à un circuit classique. Voici une description informelle d'un circuit.

Un circuit a un nombre n d'entrées (ou de fils). Chaque fil contient un qubit, l'espace de Hilbert combiné est donc $\mathcal{H}^n$. Les opérations sur les qubits sont effectuées par des *portes quantiques*. On représente la transformation U agissant sur $|\psi\rangle \in \mathcal{H}$ par le diagramme de la figure 2.1(a) : l'état $|\psi\rangle$ *entre* à gauche et l'état $\mathsf{U}|\psi\rangle$ *sort* à droite. Si une transformation U agit sur plusieurs qubits qui vivent sur des fils voisins, on la



(a) Porte sur un qubit (b) Porte sur plusieurs qubits (c) Portes en produit tensoriel

FIG. 2.1: Diagrammes pour les portes quantiques

représente comme dans la figure 2.1(b). De plus, si U peut être exprimée comme un produit tensoriel $U_1 \otimes \ldots \otimes U_k$, on peut la représenter séparément comme dans la figure 2.1(c).

Voyons maintenant quelques portes quantiques qui reviendront souvent au cours de ce travail. Pour chacune, nous donnons son effet sur les états de base, sa matrice associée et le diagramme décrivant la porte dans le circuit. Notons d'abord par I_k (ou I s'il n'y a pas d'ambiguïté) la matrice ou l'opérateur identité sur k qubits.

Une des portes les plus importantes est la porte de Walsh-Hadamard, notée H, définie par

$$\begin{array}{l} |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{array} \qquad H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \qquad \text{—[H]—} \tag{2.16}$$

Il est facile de vérifier que $HH = I$ (i.e. H est auto-inverse) et que pour $|x\rangle \in \{0,1\}^n$,

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle, \tag{2.17}$$

où $x \cdot y$ est le produit scalaire entre deux chaînes de bits. La porte de Walsh-Hadamard permet de transformer les états de la base standard à ceux de la base diagonale. En effet,

$$\begin{array}{l} |0\rangle \xleftrightarrow{H} |\swarrow\rangle \\ |1\rangle \xleftrightarrow{H} |\searrow\rangle \end{array} \tag{2.18}$$

La porte suivante, notée N, définit la négation logique sur les qubits.

$$\begin{array}{ll} |0\rangle & \xrightarrow{\mathsf{N}} \ |1\rangle \\ |1\rangle & \xrightarrow{\mathsf{N}} \ |0\rangle \end{array} \qquad \mathsf{N} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{(2.19)}$$

La porte de négation échange les amplitudes de $|0\rangle$ et de $|1\rangle$. En effet, $\mathsf{N}\big(\alpha|0\rangle + \beta|1\rangle\big) = \beta|0\rangle + \alpha|1\rangle$.

La porte de changement de phase P permet de changer la phase relative entre $|0\rangle$ et $|1\rangle$. Elle est définie comme suit.

$$\begin{array}{ll} |0\rangle & \xrightarrow{\mathsf{P}} \ |0\rangle \\ |1\rangle & \xrightarrow{\mathsf{P}} \ -|1\rangle \end{array} \qquad \mathsf{P} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{(2.20)}$$

Si on l'applique sur un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, on obtient $\mathsf{P}|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$.

La porte la plus importante sur deux qubits est sans doute le ou-exclusif[3] (souvent appelée la négation contrôlée). Le ou-exclusif est une porte non séparable ; on ne peut donc pas la remplacer par deux portes qui agissent indépendamment sur deux qubits. On note la porte par O et on la définit par :

$$\begin{array}{ll} |00\rangle & \xrightarrow{\mathsf{O}} \ |00\rangle \\ |01\rangle & \xrightarrow{\mathsf{O}} \ |01\rangle \\ |10\rangle & \xrightarrow{\mathsf{O}} \ |11\rangle \\ |11\rangle & \xrightarrow{\mathsf{O}} \ |10\rangle \end{array} \qquad \mathsf{O} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \text{(2.21)}$$

Nous ne décrirons pas de portes plus complexes. En effet, le ou-exclusif et les portes sur un qubit sont suffisants pour tout calcul quantique, comme l'énonce le théorème suivant.

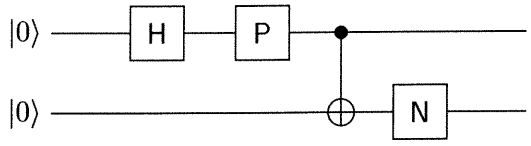
Théorème 2.14 Toute transformation unitaire sur $\mathcal{H}^n$ peut être implantée par un circuit composé de portes unaires (sur un qubit) agissant sur $\mathcal{H}$ et de ou-exclusifs. $\diamond$

De plus, tout ce qu'on peut calculer classiquement, on peut le faire avec un circuit quantique.

[3]Son nom vient du fait qu'elle envoit l'état $|ab\rangle$ sur $|a(a \oplus b)\rangle$ pour $a, b \in \{0, 1\}$.

Théorème 2.15 Tout circuit classique peut être transformé en un circuit quantique de taille comparable. ◇

Comme exercice, analysons l'effet du circuit suivant sur le registre $|00\rangle$.



L'espace du premier fil est $\mathcal{H}_A = \mathcal{H}$ et celui du deuxième est $\mathcal{H}_B = \mathcal{H}$. Pour différencier les transformations faites sur le premier ou le deuxième qubit, on peut les indexer par A ou B. Par exemple, si U est appliquée sur le premier qubit, on peut la noter par U_A, ce qui est équivalent à appliquer $(U \otimes I)$ sur le registre.

L'état du registre évolue comme suit.

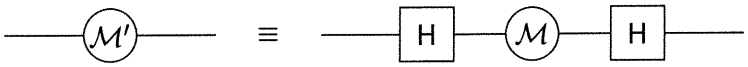
$$\begin{aligned}
|00\rangle &\xrightarrow{H_A} \frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle\Big)|0\rangle \\
&\xrightarrow{P_A} \frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big)|0\rangle = \frac{1}{\sqrt{2}}\Big(|00\rangle - |10\rangle\Big) \\
&\xrightarrow{O} \frac{1}{\sqrt{2}}\Big(|00\rangle - |11\rangle\Big) \\
&\xrightarrow{N_B} \frac{1}{\sqrt{2}}\Big(|01\rangle - |10\rangle\Big) = |\Psi^-\rangle
\end{aligned}$$

Donc, le circuit permet de créer l'état intriqué $|\Psi^-\rangle$ à partir de l'état séparable de départ $|00\rangle$. Ceci démontre que le ou-exclusif est bel et bien une porte non séparable (si on pouvait créer l'état $|\Psi^-\rangle$ de façon indépendante sur $\mathcal{H}_A$ et $\mathcal{H}_B$, alors l'état serait séparable). Notons qu'il existe un circuit plus simple pour produire $|\Psi^-\rangle$ à partir de $|00\rangle$.

L'autre opération que l'on peut avoir dans un circuit est celle de la mesure. L'opérateur de mesure dans la base standard, noté $\mathcal{M}$, s'effectue en plaçant le diagramme suivant sur le fil correspondant.

$$\alpha|0\rangle + \beta|1\rangle \quad \text{—}\mathcal{M}\text{—} \quad \begin{cases} |0\rangle & \text{avec prob. } |\alpha|^2 \\ |1\rangle & \text{avec prob. } |\beta|^2 \end{cases}$$

L'état quantique qui sort à la droite du fil est $|0\rangle$ ou $|1\rangle$, selon le résultat de la mesure. Il est facile de mesurer dans la base diagonale si on a un appareil qui mesure dans la base standard. En effet, si on note par $\mathcal{M}'$ l'opérateur qui mesure dans la base diagonale, alors



2.5 Les mélanges statistiques

L'état quantique le plus général ne peut pas être représenté par un état pur. Par exemple, supposons qu'on reçoive le qubit $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ juste après qu'il ait été mesuré dans la base standard. Ne sachant pas le résultat de la mesure, comment peut-on décrire le qubit ? On sait que la mesure a produit l'état $|0\rangle$ ou $|1\rangle$ avec probabilité $1/2$. Cependant, on ne peut pas représenter cet état par $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ (car il n'est plus en superposition). On dit alors que le qubit est un mélange statistique de $|0\rangle$ et $|1\rangle$.

Définition 2.16 Soit $|\psi_1\rangle, \ldots, |\psi_k\rangle \in \mathcal{H}^n$. Un système quantique est un *mélange statistique* des états $|\psi_1\rangle, \ldots, |\psi_k\rangle$ si le système est dans l'état $|\psi_i\rangle$ avec probabilité p_i, où $\sum_{i=1}^{k} p_i = 1$. On note le système par $\{\psi\} = \{(p_i, |\psi_i\rangle)\}$. Si $k = 1$, alors le système est un état *pur*, sinon on dit que l'état est *mélangé*. ◇

Pour représenter les états mélangés de façon plus concise, on utilise souvent un objet mathématique défini comme suit.

Définition 2.17 Une *matrice* (ou *opérateur*) *de densité* ρ sur $\mathcal{H}^n$ est une matrice $2^n \times 2^n$ hermitienne, définie semi-positive de trace 1. C'est-à-dire que

1) $\rho = \rho^\dagger$,

2) $\langle\psi|\rho|\psi\rangle \geq 0$, pour tout $|\psi\rangle \in \mathcal{H}^n$, et

3) $\mathrm{tr}(\rho) = 1$.

La matrice de densité associée à $\{\psi\} = \{(p_i, |\psi_i\rangle)\}$ est

$$\rho_\psi = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{2.22}$$

◇

Ainsi, la matrice de densité d'un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ est

$$\rho = |\psi\rangle\langle\psi| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \begin{pmatrix} \alpha^\star & \beta^\star \end{pmatrix} = \begin{pmatrix} \alpha\alpha^\star & \alpha\beta^\star \\ \alpha^\star\beta & \beta\beta^\star \end{pmatrix} \tag{2.23}$$

Exemple 2.18 Soit les états $|\nearrow\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, $|\nwarrow\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$, $\{\psi\} = \{(\frac{1}{2}, |0\rangle), (\frac{1}{2}, |1\rangle)\}$ et $\{\phi\} = \{(\frac{1}{2}, |\nwarrow\rangle), (\frac{1}{2}, |\nearrow\rangle)\}$ (notons que $\{\psi\}$ est l'état du système présenté au début de la section) ; alors

$$|0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \; |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

$$|\nearrow\rangle\langle\nearrow| = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \; |\nwarrow\rangle\langle\nwarrow| = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \text{ et}$$

$$\rho_\psi = \tfrac{1}{2}|0\rangle\langle 0| + \tfrac{1}{2}|1\rangle\langle 1| = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} = \tfrac{1}{2}|\nearrow\rangle\langle\nearrow| + \tfrac{1}{2}|\nwarrow\rangle\langle\nwarrow| = \rho_\phi$$

◇

On voit donc que deux mélanges statistiques peuvent produire la même matrice de densité. Lorsque la matrice de densité d'un état sur $\mathcal{H}^n$ est $\frac{1}{2^n}\mathrm{I}_n$, on dit que l'état est *maximalement mélangé*. Voyons maintenant comment trouver la matrice de densité associée à un sous-système quantique.

Définition 2.19 Soit un système bipartite (produit tensoriel de deux systèmes) $|\psi_{\mathrm{AB}}\rangle \in \mathcal{H}_{\mathrm{A}}^m \otimes \mathcal{H}_{\mathrm{B}}^n$. Soit $\{|i_{\mathrm{A}}\rangle\}$ une base orthonormée pour $\mathcal{H}_{\mathrm{A}}^m$ et $\{|k_{\mathrm{B}}\rangle\}$, une pour $\mathcal{H}_{\mathrm{B}}^m$. On peut donc exprimer l'état par

$$|\psi_{\mathrm{AB}}\rangle = \sum_{\substack{i\in\{0,1\}^m \\ k\in\{0,1\}^n}} \alpha_{ik}|i_{\mathrm{A}}\rangle \otimes |k_{\mathrm{B}}\rangle$$

où $\sum_{ik} |\alpha_{ik}|^2 = 1$. La matrice de densité représentant le sous-système A est obtenue en effectuant la *trace partielle* du sous-système B définie par

$$\rho_{\mathrm{A}} = \mathrm{tr}_{\mathrm{B}} |\psi_{\mathrm{AB}}\rangle\langle\psi_{\mathrm{AB}}| = \sum_{\substack{i,j\in\{0,1\}^m \\ k\in\{0,1\}^n}} \alpha_{ik}\alpha^{\star}_{jk} |i_{\mathrm{A}}\rangle\langle j_{\mathrm{A}}| \tag{2.24}$$

◇

En d'autres mots, la trace partielle permet d'abstraire une partie d'un système et nous donne la matrice de densité associée au sous-système qui nous intéresse. Par exemple, prenons l'état $|\Psi^-\rangle$. Soit $\mathcal{H}_{\mathrm{A}} = \mathcal{H}_{\mathrm{B}} = \mathcal{H}$, $\mathcal{H}_{\mathrm{A}}$ représentant l'espace du premier qubit et $\mathcal{H}_{\mathrm{B}}$, celui du deuxième. La base pour les deux espaces est $\{|0\rangle, |1\rangle\}$. Les valeurs des α_j sont donc : $\alpha_{00} = \alpha_{11} = 0$, $\alpha_{01} = 1/\sqrt{2}$ et $\alpha_{10} = -1/\sqrt{2}$. Par l'équation 2.24, on obtient que

$$\begin{aligned}
\rho_{\mathrm{A}} &= \sum_{k=0,1} \alpha_{0k}\alpha^{\star}_{0k} |0_{\mathrm{A}}\rangle\langle 0_{\mathrm{A}}| + \alpha_{0k}\alpha^{\star}_{1k} |0_{\mathrm{A}}\rangle\langle 1_{\mathrm{A}}| + \alpha_{1k}\alpha^{\star}_{0k} |1_{\mathrm{A}}\rangle\langle 0_{\mathrm{A}}| + \alpha_{1k}\alpha^{\star}_{1k} |1_{\mathrm{A}}\rangle\langle 1_{\mathrm{A}}| \\
&= \alpha_{01}\alpha^{\star}_{01} |0_{\mathrm{A}}\rangle\langle 0_{\mathrm{A}}| + \alpha_{10}\alpha^{\star}_{10} |1_{\mathrm{A}}\rangle\langle 1_{\mathrm{A}}| \qquad (\text{car } \alpha_{kk} = 0) \\
&= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

Donc, si on prépare l'état $|\Psi^-\rangle$ et qu'on envoie le premier qubit à Alice et le second à Bob, alors Alice aura entre ses mains un état maximalement mélangé. Ceci est cohérent avec la situation présentée au début de la section. Si Bob mesure sa particule, il sait dans quel état le système d'Alice se trouve. Par contre, pour elle, l'état est maximalement mélangé. Donc, la matrice de densité dépend de l'information classique que l'on a sur un système.

Exemple 2.20 Soit $|\psi_{\mathrm{A}}\rangle = \sum_i \gamma_i |i\rangle \in \mathcal{H}_{\mathrm{A}}^m$ et $|\phi_{\mathrm{B}}\rangle = \sum_k \delta_k |k\rangle \in \mathcal{H}_{\mathrm{B}}^n$ et soit l'état bipartite $|\Gamma_{\mathrm{AB}}\rangle = |\psi_{\mathrm{A}}\rangle \otimes |\phi_{\mathrm{B}}\rangle = \sum_{ik} \alpha_{ik} |i\rangle \otimes |k\rangle$ où $\alpha_{ik} = \gamma_i\delta_k$. Alors la trace partielle

du sous-système B donne

$$\begin{aligned}
\rho_{\mathrm{A}} &= \mathrm{tr}_{\mathrm{B}}\big(|\Gamma_{\mathrm{AB}}\rangle\langle\Gamma_{\mathrm{AB}}|\big) \\
&= \sum_{i,j,k} \alpha_{ik}\alpha^{\star}_{jk}|i_{\mathrm{A}}\rangle\langle j_{\mathrm{A}}| \\
&= \sum_{i,j,k} \gamma_i\delta_k\gamma^{\star}_j\delta^{\star}_k|i_{\mathrm{A}}\rangle\langle j_{\mathrm{A}}| \\
&= \sum_{i,j} \gamma_i\gamma^{\star}_j|i_{\mathrm{A}}\rangle\langle j_{\mathrm{A}}| \underbrace{\sum_{k} \delta_k\delta^{\star}_k}_{1} \\
&= \sum_{i,j} \gamma_i\gamma^{\star}_j|i_{\mathrm{A}}\rangle\langle j_{\mathrm{A}}| \\
&= |\psi_{\mathrm{A}}\rangle\langle\psi_{\mathrm{A}}|.
\end{aligned}$$

Donc, *tracer* un sous-système qui est en produit tensoriel avec le reste du système consiste simplement à se débarrasser des qubits correspondants. ◇

2.5.1 Les opérations sur les états mélangés

Il reste maintenant à décrire comment les transformations et les mesures affectent les états mélangés. Si on applique la transformation U sur un état mélangé ρ décrit par l'équation 2.22, alors l'état évolue à

$$\rho \xrightarrow{\mathsf{U}} \rho' = \sum_i p_i \mathsf{U}|\psi_i\rangle\langle\psi_i|\mathsf{U}^{\dagger}, \tag{2.25}$$

ce qui correspond bien[4] à l'état $\{\psi'\} = \{(p_i, \mathsf{U}|\psi_i\rangle)\}$.

Les transformations ne sont pas nécessairement unitaires si on considère l'évolution d'un sous-système. En effet, si $|\psi\rangle \in \mathcal{H}_{\mathrm{A}} \otimes \mathcal{H}_{\mathrm{B}}$ et que U agisse sur cet espace, alors l'évolution

$$\rho_{\mathrm{A}} = \mathrm{tr}_{\mathrm{B}}\big(|\psi\rangle\langle\psi|\big) \xrightarrow{\mathsf{U}} \mathrm{tr}_{\mathrm{B}}\big(\mathsf{U}|\psi\rangle\langle\psi|\mathsf{U}^{\dagger}\big) = \rho'_{\mathrm{A}}$$

n'est pas, en général, unitaire. Pour décrire l'évolution d'un sous-système, il faut définir la notion de *superopérateur.*

[4] On se rappelle que pour deux matrices A et B, $(AB)^{\dagger} = B^{\dagger}A^{\dagger}$.

Définition 2.21 Un *superopérateur* \$ est une transformation linéaire des matrices de densité vers les matrices de densité. Tout superopérateur peut être décrit par une somme d'opérateurs A_i sur $\mathcal{H}^n$

$$\$(\rho) = \sum_i A_i \rho A_i^\dagger$$

tels que $\sum_i A_i^\dagger A_i = \mathrm{I}_n$. ◇

Exemple 2.22 Soit le superopérateur \$ représenté par

$$\$(\rho) = \frac{1}{4}\Big(\rho + \mathsf{N}\rho\mathsf{N}^\dagger + \mathsf{P}\rho\mathsf{P}^\dagger + \mathsf{NP}\rho\mathsf{P}^\dagger\mathsf{N}^\dagger\Big).$$

Ici, $A_0 = \frac{1}{2}\mathrm{I}$, $A_1 = \frac{1}{2}\mathsf{N}$, $A_2 = \frac{1}{2}\mathsf{P}$ et $A_3 = \frac{1}{2}\mathsf{NP}$. On a bien que $\sum_{i=0}^{3} A_i^\dagger A_i = \mathrm{I}$. L'effet du superopérateur sur un qubit est de le mélanger maximalement. En effet, si $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, alors

$$\begin{aligned}
\$\big(|\psi\rangle\langle\psi|\big) &= \frac{1}{4}\left[\begin{pmatrix} \alpha\alpha^\star & \alpha\beta^\star \\ \alpha^\star\beta & \beta\beta^\star \end{pmatrix} + \begin{pmatrix} \beta\beta^\star & \alpha^\star\beta \\ \alpha\beta^\star & \alpha\alpha^\star \end{pmatrix}\right. \\
&\quad \left. + \begin{pmatrix} \alpha\alpha^\star & -\alpha\beta^\star \\ -\alpha^\star\beta & \beta\beta^\star \end{pmatrix} + \begin{pmatrix} \beta\beta^\star & -\alpha^\star\beta \\ -\alpha\beta^\star & \alpha\alpha^\star \end{pmatrix}\right] \\
&= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \qquad (\text{car } \alpha\alpha^\star + \beta\beta^\star = 1)
\end{aligned}$$

◇

Voyons maintenant comment on mesure les états mélangés. Soit $\{P_i\}$ un ensemble de projecteurs hermitiens (i.e. $P_i^\dagger = P_i$, $P_i^2 = P_i$, $P_i \neq P_j$ pour $i \neq j$ et $\sum_i P_i = \mathrm{I}$). Alors la mesure d'un état ρ par rapport aux P_i retournera i avec probabilité

$$\mathrm{tr}(P_i\rho). \tag{2.26}$$

Après la mesure, l'état devient

$$\rho' = \frac{1}{\mathrm{tr}(P_i\rho)} P_i \rho P_i. \tag{2.27}$$

Exemple 2.23 Si $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, alors la mesure de $\rho = |\psi\rangle\langle\psi|$ qui correspond à la mesure dans la base standard est décrite par $\{P_0, P_1\}$ où

$$P_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \text{ et } P_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

On obtient bien que la mesure donne 0 avec probabilité $\mathrm{tr}(P_0\rho) = \alpha\alpha^\star$ et qu'elle donne 1 avec probabilité $\mathrm{tr}(P_1\rho) = \beta\beta^\star$. De plus, si la mesure donne 0, l'état devient

$$\rho' = \frac{1}{\alpha\alpha^\star}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \alpha\alpha^\star & \alpha\beta^\star \\ \alpha^\star\beta & \beta\beta^\star \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = |0\rangle\langle 0|.$$

Si elle donne 1, l'état devient $|1\rangle\langle 1|$. ◇

Avant de terminer la section, introduisons une notion importante qui nous permettra de déterminer la distance entre deux états quantiques, un peu comme la distance de Hamming permet de le faire dans le cas classique.

Définition 2.24 Soit $|\psi\rangle$ un état pur et soit ρ un état mélangé, tous deux de $\mathcal{H}^n$. Alors la *fidélité* de ρ par rapport à $\psi = |\psi\rangle\langle\psi|$ est la valeur réelle

$$F(\psi, \rho) = \langle\psi|\rho|\psi\rangle. \tag{2.28}$$

◇

La fidélité est un nombre réel entre 0 et 1 (inclusivement). Plus ce nombre est près de 1, plus les états sont "proches". On note que si ρ est aussi un état pur $|\phi\rangle\langle\phi|$, alors l'équation 2.28 se réduit à

$$F(\psi, \rho) = \langle\psi|\phi\rangle\langle\phi|\psi\rangle = |\langle\psi|\phi\rangle|^2.$$

Nous n'utiliserons pas la fidélité dans le cas où les deux états sont mélangés (dans ce cas, la définition de la fidélité est plus complexe). Donnons une dernière propriété de la fidélité.

Proposition 2.25 Soit $\psi = |\psi\rangle\langle\psi|$ et $\phi = |\phi\rangle\langle\phi|$ des états purs de $\mathcal{H}^n$. Alors

1) $F(\psi, \phi) = 1 \iff |\psi\rangle = |\phi\rangle$
2) $F(\psi, \phi) = 0 \iff |\psi\rangle$ est orthogonal à $|\phi\rangle$.

◇

2.6 Généralisation à $\mathcal{H}_q^n$

Toute la théorie de l'information quantique peut se généraliser aux systèmes quantiques avec plus de deux niveaux. On considère ces systèmes pour les mêmes raisons qu'on considère les corps à plus de deux éléments : ils permettent de construire des codes plus efficaces.

Définition 2.26 Un *quqit* est un vecteur normalisé de $\mathcal{H}_q$. La base standard sur $\mathcal{H}_q$ est $\{|0\rangle, |1\rangle, \ldots, |q-1\rangle\}$. Tout comme les qubits, un qu*q*it peut-être exprimé comme une combinaison linéaire des vecteurs de base :

$$|\psi\rangle = \sum_{i=0}^{q-1} \alpha_i |i\rangle \qquad\qquad |\psi\rangle = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{q-1} \end{pmatrix}$$

où les $\alpha_i \in \mathbb{C}$ et $\sum_i |\alpha_i|^2 = 1$. Ici, le vecteur $|i\rangle$ est représenté par le vecteur colonne avec un 1 en position $i+1$ et des 0 partout ailleurs. ◇

Voici comment les portes que nous avons vues se généralisent à $\mathcal{H}_q$. Soit $w = e^{\frac{2\pi\sqrt{-1}}{q}}$ et $i, j \in \mathbb{F}_q$, alors

$$\begin{aligned}
\mathsf{N}_q : \ & |i\rangle & \longrightarrow \ & |i+1\rangle \\
\mathsf{P}_q : \ & |i\rangle & \longrightarrow \ & w^i |i\rangle \\
\mathsf{H}_q : \ & |i\rangle & \longrightarrow \ & \tfrac{1}{\sqrt{q}} \textstyle\sum_{j\in\mathbb{F}_q} w^{ij} |j\rangle \\
\mathsf{O}_q : \ & |i\rangle \otimes |j\rangle & \longrightarrow \ & |i\rangle \otimes |i+j\rangle
\end{aligned}$$

où les additions et multiplications se font sur $\mathbb{F}_q$. Notons que H_q se nomme une transformation de Fourier si q n'est pas une puissance de 2. On utilisera cependant le même symbole pour décrire la transformation.

Exemple 2.27 Un *qutrit* est un qu*q*it avec $q = 3$. Soit un qutrit arbitraire

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle.$$

Analysons l'effet du circuit 2.2 sur $|\psi\rangle$.

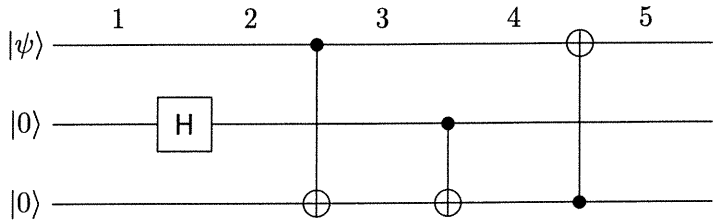


FIG. 2.2: Exemple de circuit pour un qutrit

L'état de départ est, en (1), $\alpha|000\rangle + \beta|100\rangle + \gamma|200\rangle$. La transformation de Fourier de $\mathcal{H}_3$, appliqué sur $|0\rangle$, produit une superposition égale des états de base :

$$\mathsf{H}_3|0\rangle = \frac{1}{\sqrt{3}}\Big(|0\rangle + |1\rangle + |2\rangle\Big).$$

Donc, à l'étape (2), le registre devient

$$\begin{aligned} \xrightarrow{(2)} \quad & \tfrac{\alpha}{\sqrt{3}}\Big(|000\rangle + |010\rangle + |020\rangle\Big) + \\ & \tfrac{\beta}{\sqrt{3}}\Big(|100\rangle + |110\rangle + |120\rangle\Big) + \\ & \tfrac{\gamma}{\sqrt{3}}\Big(|200\rangle + |210\rangle + |220\rangle\Big) \end{aligned}$$

Dans $\mathcal{H}_3$, le O_3 est défini par

$$\begin{aligned} |00\rangle &\xrightarrow{\mathsf{O}_3} |00\rangle \\ |01\rangle &\xrightarrow{\mathsf{O}_3} |01\rangle \\ |02\rangle &\xrightarrow{\mathsf{O}_3} |02\rangle \\ |10\rangle &\xrightarrow{\mathsf{O}_3} |11\rangle \\ |11\rangle &\xrightarrow{\mathsf{O}_3} |12\rangle \\ |12\rangle &\xrightarrow{\mathsf{O}_3} |10\rangle \\ |20\rangle &\xrightarrow{\mathsf{O}_3} |22\rangle \\ |21\rangle &\xrightarrow{\mathsf{O}_3} |20\rangle \\ |22\rangle &\xrightarrow{\mathsf{O}_3} |21\rangle \end{aligned} \qquad \mathsf{O}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Donc, suite aux ou-exclusifs, l'état évolue à

$$\begin{aligned}
&\xrightarrow{(3)} \quad \tfrac{\alpha}{\sqrt{3}}\Big(|000\rangle + |010\rangle + |020\rangle\Big) + \\
&\qquad\quad \tfrac{\beta}{\sqrt{3}}\Big(|101\rangle + |111\rangle + |121\rangle\Big) + \\
&\qquad\quad \tfrac{\gamma}{\sqrt{3}}\Big(|202\rangle + |212\rangle + |222\rangle\Big) \\
&\xrightarrow{(4)} \quad \tfrac{\alpha}{\sqrt{3}}\Big(|000\rangle + |011\rangle + |022\rangle\Big) + \\
&\qquad\quad \tfrac{\beta}{\sqrt{3}}\Big(|101\rangle + |112\rangle + |120\rangle\Big) + \\
&\qquad\quad \tfrac{\gamma}{\sqrt{3}}\Big(|202\rangle + |210\rangle + |221\rangle\Big) \\
&\xrightarrow{(5)} \quad \tfrac{\alpha}{\sqrt{3}}\Big(|000\rangle + |111\rangle + |222\rangle\Big) + \\
&\qquad\quad \tfrac{\beta}{\sqrt{3}}\Big(|201\rangle + |012\rangle + |120\rangle\Big) + \\
&\qquad\quad \tfrac{\gamma}{\sqrt{3}}\Big(|102\rangle + |210\rangle + |021\rangle\Big)
\end{aligned}$$

Donc, le circuit produit la superposition suivante :

$$\frac{\alpha}{\sqrt{3}}\Big(|000\rangle + |111\rangle + |222\rangle\Big) + \frac{\beta}{\sqrt{3}}\Big(|201\rangle + |012\rangle + |120\rangle\Big) + \frac{\gamma}{\sqrt{3}}\Big(|102\rangle + |210\rangle + |021\rangle\Big)$$

Nous utiliserons cette superposition dans la section 5.2. ◇

Chapitre 3

Codes correcteurs d'erreurs quantiques

Nous développons maintenant la théorie des codes correcteurs d'erreurs quantiques. Contrairement au cas classique, les états quantiques sont extrêmement fragiles. Les superpositions quantiques ne restent pas stables très longtemps. Elles perdent leur cohérence, car les systèmes quantiques interagissent avec leur entourage (qu'on appelle environnement). On ne connaît pas de système quantique qui puisse rester intact en mémoire très longtemps, encore moins un qui peut résister à un canal de transmission bruyant.

On a vu au chapitre 1 des codes classiques qui permettent de corriger des erreurs en ajoutant de la redondance aux messages, ce qui permet de déduire les erreurs survenues lors de la transmission. Si on essaie de faire une chose semblable dans le cas quantique, on se bute à des questions inquiétantes. En effet, comment peut-on introduire de la *redondance quantique* sachant que le théorème 2.9 de non-clonage nous dit qu'on ne peut pas copier les qubits ? De plus, comment peut-on déterminer les erreurs en examinant (i.e. mesurant) les *mots de codes quantiques* sachant qu'en mesurant un état, on risque de le détruire ? Malgré ces restrictions importantes, Peter Shor [Sho95] et Andrew Steane [Ste96b] ont trouvé une façon de réduire la décohérence

d'un système quantique. Par la suite, plusieurs constructions de codes correcteurs efficaces ont vu le jour [CS96, CL95, BDSW97].

Dans ce mémoire, nous faisons la supposition importante que nos opérations quantiques (mesures et transformations) sont parfaites, i.e. elles n'introduisent pas d'erreur. La réalité est bien différente. Non seulement les qubits sont modifiés par l'environnement, ils le sont aussi par les opérations. Par exemple, si on veut appliquer une transformation U sur $|\psi\rangle$, notre appareillage implante en fait une transformation U_ε proche de U. Donc, l'état résultant est $|\psi_\varepsilon\rangle$ qui est près de $|\psi\rangle$. Sur un long calcul quantique, ces petites erreurs s'accumulent et risquent de ruiner le calcul. L'ordinateur quantique peut quand même fonctionner même en présence de *portes bruyantes* (ou *imparfaites*). Les procédures de protection dans ce contexte sont étudiées dans le domaine du *calcul tolérant aux fautes* [Sho96, ABO99, Pre97, Got98].

3.1 Le modèle d'erreurs

Comme dans le cas classique, nous définirons la notion de canal quantique pour modéliser les erreurs qui affectent les qubits. Nous supposons que la seule source d'erreurs est la transmission à travers le canal. Ce modèle permet aussi de décrire les *mémoires quantiques* où les qubits ne sont pas transmis dans l'espace mais stockés dans le temps. La canal représente alors une unité de temps.

Un canal sera représenté par un superopérateur. L'action du canal est unitaire si on considère le produit tensoriel du registre et de l'environnement, mais puisque ce dernier nous est inaccessible, l'évolution du registre doit être décrite par un superopérateur.

Définition 3.1 Un *canal quantique inconscient* est un superopérateur $\mathcal{X}$ qui prend un état $\rho \in \mathcal{H}$ retourne une matrice de densité

$$\rho_{\mathcal{X}} = \sum_i A_i \rho A_i^\dagger$$

où $\sum_i A_i^\dagger A_i = \mathrm{I}$. L'application du canal sur un registre se fait indépendamment sur

chaque qubit. Si $\rho = \rho_1 \otimes \ldots \otimes \rho_n \in \mathcal{H}^n$, alors

$$\mathcal{X}(\rho_1 \otimes \ldots \otimes \rho_n) = \mathcal{X}(\rho_1) \otimes \ldots \otimes \mathcal{X}(\rho_n)$$

◇

Dans ce travail, nous faisons la supposition importante que le canal agit de façon indépendante sur chaque qubit. Il n'y a pas de corrélations temporelles (erreurs sur le même qubit mais à des temps différents) et spatiales (erreurs sur des qubits voisins). Si on considère un modèle d'erreurs plus particulier, on peut développer des codes correcteurs mieux adaptés que ceux présentés dans ce chapitre (par exemple voir [PVK96]).

Essayons maintenant de mieux comprendre le processus d'erreur. Un des obstacles apparents à la correction d'erreurs quantiques est que l'on doit faire face à une infinité d'erreurs. Classiquement, la seule erreur sur un bit est la négation. Quantiquement, le qubit peut être modifié d'une infinité de façons. Comment peut-on découvrir quelle erreur est survenue pour pouvoir la corriger ? Heureusement, le théorème suivant nous donnera un moyen de *numériser* les erreurs, i.e. de décrire tout processus d'erreur avec des erreurs discrètes. Les matrices I, N, P et NP forment un ensemble bien connu qu'on appelle les matrices de Pauli. Dans la littérature, on les note[1] souvent par

$$\sigma_0 = \mathsf{I},\ \sigma_1 = \mathsf{N},\ \sigma_2 = \mathsf{P},\ \sigma_3 = \mathsf{NP}.$$

On notera l'ensemble des matrices de Pauli par $\mathcal{P} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$ et $\mathcal{P}^{\otimes n}$ représentera l'ensemble des chaînes de n matrices de Pauli en produit tensoriel (par exemple, $\sigma_2 \otimes \sigma_0 \otimes \sigma_1 \otimes \sigma_2 \in \mathcal{P}^{\otimes 4}$).

Théorème 3.2 Soit $\mathcal{X} : \mathcal{H} \to \mathcal{H}$ un canal quantique. L'effet de $\mathcal{X}$ sur un état $|\psi\rangle \in \mathcal{H}$ peut être exprimé par :

$$\mathcal{X}(|\psi\rangle) = \sum_{i=0}^{3} \sigma_i |\psi\rangle \otimes |\widetilde{e_{\sigma_i}}\rangle, \tag{3.1}$$

où les $|\widetilde{e_{\sigma_i}}\rangle$ sont des états de l'environnement. ◇

[1] Les physiciens notent ces matrices par $\sigma_1 = \sigma_x$, $\sigma_3 = \sigma_y$ et $\sigma_2 = \sigma_z$. En réalité, $\sigma_3 = i\mathsf{NP}$ ($i = \sqrt{-1}$), mais on laisse souvent tomber le i, car il n'a pas d'influence sur les calculs. De plus, les matrices de Pauli ne comprennent pas vraiment σ_0, mais on l'ajoute ici pour simplifier les discussions.

Preuve On peut représenter le canal $\mathcal{X}$ par une transformation unitaire $\mathsf{U}_\mathcal{X}$ agissant sur l'espace du qubit et sur $\mathcal{H}_\mathrm{C}$, l'espace du canal (l'environnement). On peut supposer que l'état initial de l'environnement est un état pur $|e_{init}\rangle \in \mathcal{H}_\mathrm{C}$ de dimension arbitraire. On peut donc représenter $\mathsf{U}_\mathcal{X}$ par

$$\begin{aligned} |0\rangle \otimes |e_{init}\rangle &\xrightarrow{\mathsf{U}_\mathcal{X}} |0\rangle \otimes |\widetilde{e_{00}}\rangle + |1\rangle \otimes |\widetilde{e_{01}}\rangle \\ |1\rangle \otimes |e_{init}\rangle &\xrightarrow{\mathsf{U}_\mathcal{X}} |0\rangle \otimes |\widetilde{e_{10}}\rangle + |1\rangle \otimes |\widetilde{e_{11}}\rangle \end{aligned}$$

où les $|\widetilde{e_{ij}}\rangle$ sont les états de l'environnement. Ils ne sont, en général, ni normalisés ni orthogonaux[2], mais ils respectent l'unitarité de $\mathsf{U}_\mathcal{X}$. Donc, l'action du canal sur un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ est

$$\begin{aligned} |\psi\rangle \xrightarrow{\mathcal{X}} & \mathsf{U}_\mathcal{X}\Big((\alpha|0\rangle + \beta|1\rangle) \otimes |e_{init}\rangle\Big) \\ = & \alpha\Big(|0\rangle \otimes |\widetilde{e_{00}}\rangle + |1\rangle \otimes |\widetilde{e_{01}}\rangle\Big) + \beta\Big(|0\rangle \otimes |\widetilde{e_{10}}\rangle + |1\rangle \otimes |\widetilde{e_{11}}\rangle\Big) \\ = & \frac{1}{2}\Big[\ (\alpha|0\rangle + \beta|1\rangle) \otimes (|\widetilde{e_{00}}\rangle + |\widetilde{e_{11}}\rangle) + (\alpha|0\rangle - \beta|1\rangle) \otimes (|\widetilde{e_{00}}\rangle - |\widetilde{e_{11}}\rangle) \\ & + \ (\beta|0\rangle + \alpha|1\rangle) \otimes (|\widetilde{e_{01}}\rangle + |\widetilde{e_{10}}\rangle) - (\beta|0\rangle - \alpha|1\rangle) \otimes (|\widetilde{e_{01}}\rangle - |\widetilde{e_{10}}\rangle)\ \Big] \\ = & \sigma_0|\psi\rangle \otimes |\widetilde{e_{\sigma_0}}\rangle + \sigma_1|\psi\rangle \otimes |\widetilde{e_{\sigma_1}}\rangle + \sigma_2|\psi\rangle \otimes |\widetilde{e_{\sigma_2}}\rangle + \sigma_3|\psi\rangle \otimes |\widetilde{e_{\sigma_3}}\rangle \end{aligned}$$

□

Ce théorème a une importance capitale. Il signifie que toute erreur sur un registre est une superposition de quatre erreurs de base (en fait trois erreurs et l'identité). On peut l'interpréter de la façon suivante : si on transmet un qubit à travers le canal, soit il reste intact ou il subit une erreur soit de négation, soit de changement de phase, ou soit les deux. En réalité, ces événements sont en superposition, le qubit les subit tous. Mais, si on arrive à interagir avec l'état modifié afin de lui faire choisir une des alternatives (en mesurant pour que la superposition se réduise) et qu'on apprenne laquelle est survenue, on pourra alors la corriger. C'est de cette façon que les codes correcteurs quantiques fonctionneront, en numérisant les erreurs.

[2]Pour clarifier, supposons que $\mathcal{H}_\mathrm{C} = \mathcal{H}$ et que $|e_{init}\rangle = |0\rangle$. Donc

$$\begin{aligned} |0\rangle \otimes |0\rangle \xrightarrow{\mathsf{U}_\mathcal{X}} & \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \\ = & |0\rangle \otimes (\alpha|0\rangle + \beta|1\rangle) + |1\rangle \otimes (\gamma|0\rangle + \delta|1\rangle) \\ = & |0\rangle \otimes |\widetilde{e_{00}}\rangle + |1\rangle \otimes |\widetilde{e_{01}}\rangle. \end{aligned}$$

Les $|\widetilde{e_{ij}}\rangle$ englobent les amplitudes du premier qubit, et on voit bien qu'ils ne sont en général ni normalisés, ni orthogonaux (par exemple, si $\alpha = \gamma = 1/\sqrt{2}$, $\beta = \delta = 0$).

Puisque les erreurs sur différents qubits sont indépendantes, tout opérateur d'erreur agissant sur un registre de n qubits sera composé de produits tensoriels d'opérateurs sur un qubit.

Définition 3.3 Un *opérateur de Pauli* E sur $\mathcal{H}^n$ est un produit tensoriel de n matrices de Pauli

$$E = E_1 \otimes \ldots \otimes E_n$$

où $E_i \in \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$. Notons que E est une transformation unitaire et que la transformation inverse est

$$E^\dagger = E_1^\dagger \otimes \ldots \otimes E_n^\dagger.$$

◇

Le théorème 3.2 se généralise aux superopérateurs sur $\mathcal{H}^n$. On peut exprimer l'action du canal sur le registre par une superposition des 4^n opérateurs de Pauli agissant sur le registre. En effet, toute transformation unitaire sur le registre et son environnement peut être exprimée par

$$|\psi\rangle \otimes |e_{init}\rangle \longrightarrow \sum_{E \in \mathcal{P}^{\otimes n}} E|\psi\rangle \otimes |\widetilde{e_E}\rangle,$$

où $|\psi\rangle \in \mathcal{H}^n$ et les $|\widetilde{e_E}\rangle$ sont des vecteurs non orthogonaux et non normalisés représentant l'état de l'environnement pour l'opérateur de Pauli E.

Définition 3.4 Soit E un opérateur de Pauli sur $\mathcal{H}^n$. Le *poids* de E, noté $w(E)$, est le nombre de positions où E_i diffère de l'identité, i.e.

$$w(E) = \Big|\{\, i \,:\, E_i \neq \sigma_0,\, 1 \leq i \leq n \,\}\Big|$$

◇

On peut interpréter le poids d'un opérateur comme étant le nombre d'erreurs que l'opérateur introduit dans le registre (même si σ_3 introduit une combinaison de deux erreurs).

Un code quantique ne pourra pas corriger toutes les erreurs pouvant survenir sur un registre. Le code déterminera un sous-ensemble d'erreurs $\mathcal{E} \subseteq \mathcal{P}^{\otimes n}$ (où $\mathcal{P} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$) qu'il pourra corriger (en général, $\mathcal{E}$ sera l'ensemble de tous les opérateurs de Pauli de poids inférieur à un certain t). Par la suite, une procédure de calcul de syndrome agissant sur le registre et sur un système ancillaire (consistant entre autre d'une mesure collective sur tout le registre) permettre de déterminer quel $E \in \mathcal{E}$ est survenu et appliquer $E^\dagger$ sur le registre pour corriger les erreurs.

Les capacités de corrections d'un code dépendent du canal que nous utilisons. Nous ne verrons que deux canaux qui sont les versions quantiques des canaux présentés au chapitre 1. Le canal que nous présentons maintenant est l'équivalent quantique du canal binaire symétrique. C'est le canal le plus étudié et le plus important.

Définition 3.5 Le *canal dépolarisant*, noté $\mathcal{QD}_p$, laisse passer le qubit intact avec probabilité $1-p$ et applique, avec probabilité $p/3$, soit N, P ou NP. Sa représentation en somme d'opérateurs est :

$$\begin{aligned} \mathcal{QD}_p\big(|\psi\rangle\big) &= (1-p)|\psi\rangle\langle\psi| + \tfrac{p}{3}\mathsf{N}|\psi\rangle\langle\psi|\mathsf{N}^\dagger + \tfrac{p}{3}\mathsf{P}|\psi\rangle\langle\psi|\mathsf{P}^\dagger + \tfrac{p}{3}\mathsf{NP}|\psi\rangle\langle\psi|\mathsf{P}^\dagger\mathsf{N}^\dagger \\ &= (1-p)|\psi\rangle\langle\psi| + \textstyle\sum_{i=1}^{3} \frac{p}{3}\,\sigma_i|\psi\rangle\langle\psi|\sigma_i^\dagger. \end{aligned}$$

◇

L'autre canal que nous verrons est la contrepartie quantique du canal d'effacement. Nous l'utiliserons surtout au chapitre 5.

Définition 3.6 Le *canal d'effacement quantique*, noté $\mathcal{Q}\Sigma_p$, est représenté par

$$\mathcal{Q}\Sigma_p\big(|\psi\rangle\big) = (1-p)|\psi\rangle\langle\psi| + p|\varepsilon\rangle\langle\varepsilon|$$

où $|\varepsilon\rangle$ est un état orthogonal aux états de base. En d'autres termes, le canal laisse le qubit intact avec probabilité $1-p$ et l'efface avec probabilité p. Si on envoie un qubit à travers ce canal, on sait s'il a été effacé ou non. Le canal augmente l'espace de Hilbert de $\mathcal{H}_2$ à $\mathcal{H}_3$ par l'ajout de $|\varepsilon\rangle = |2\rangle$. ◇

3.2 La correction d'erreurs quantiques

Maintenant que nous connaissons l'ennemi du registre quantique, voyons comment on peut l'en protéger. Le but d'un code correcteur quantique est d'encoder un registre de k qubits sur $n > k$ qubits, en répartissant l'information du registre original sur les n qubits. Autrement dit, l'état original est encodé de façon non-locale, dans l'intrication des n qubits. Ainsi, si le canal affecte quelques qubits du registre, on peut tout de même récupérer l'état de départ qui est caché dans les corrélations entre les autres qubits.

Définition 3.7 Un *code quantique binaire*[3] $[[n, k]]$ est un sous-espace $Q \subseteq \mathcal{H}^n$ de dimension 2^k (on utilise des crochets doubles pour distinguer les codes quantiques des codes classiques). La procédure d'encodage amènera les états $|\phi\rangle \in \mathcal{H}^k$ de k qubits sur des états de $\mathcal{H}^n$ de n qubits. L'image des états de base de $\mathcal{H}^k$ sont les mots de base et ils forment une base orthonormée pour Q. Les états $|\psi\rangle \in Q$ sont appelé des *mots de code*[4] de Q. ◇

Définition 3.8 Un code Q *détecte* t erreurs si, en appliquant sur $|\psi\rangle \in Q$ un opérateur de Pauli de poids non supérieur à t, la procédure de détection du code permet de déterminer qu'il y a eu erreur. Similairement, Q *corrige* t erreurs si la procédure de correction permet de corriger les opérateurs de Pauli de poids inférieur ou égal à t. ◇

Définition 3.9 Un code Q est *dégénéré* s'il existe deux opérateurs de Pauli E et E' ainsi qu'un état $|\psi\rangle \in Q$ tels que la procédure de détection d'erreurs ne distingue pas $E|\psi\rangle$ et $E'|\psi\rangle$. Un code qui n'est pas dégénéré est *non-dégénéré*. ◇

Certains codes sont dégénérés (comme le code à 9 qubits de Shor [Sho95]). Ceci signifie que certains groupes d'erreurs (produisant le même effet) sont corrigeables de la même façon. Nous ne verrons dans cet ouvrage que des codes non-dégénérés.

[3] Un code binaire utilise des qubits. On verra à la section 3.7 comment la théorie se généralise aux états à plus de deux dimensions.

[4] Certains auteurs nomment mots de code ce qu'on appele mots de base. Il ne faut pas confondre les deux notions dans cet ouvrage.

On peut définir la distance minimale d'un code quantique de façon similaire à celle d'un code classique.

Définition 3.10 La *distance minimale* d d'un code quantique Q est définie par

$$d = \min_{\substack{|c_1\rangle,|c_2\rangle \in Q \\ |c_1\rangle \neq |c_2\rangle \\ E \in \mathcal{P}^{\otimes n}}} \{ w(E) \, : \, |c_2\rangle = E|c_1\rangle \}.$$

En d'autres mots, la distance minimale est le poids minimal d'un opérateur de Pauli pour transformer un mot de code en un autre mot de code. Si Q encode k qubits sur n qubits, on dit que Q est un code $[[n, k, d]]$. ◇

Le théorème suivant [KL96b] établit des liens entre les théories des codes correcteurs classiques et quantiques.

Théorème 3.11 Comme dans le cas classique, un code quantique avec distance minimale d peut corriger $(d-1)/2$ erreurs. De plus, le code peut détecter $d-1$ erreurs. Si les erreurs sont localisées, i.e. si on connaît leur positions, le code peut corriger $d-1$ erreurs. En particulier, le code peut corriger $d-1$ erreurs d'effacement (il suffit de remplacer les qubits effacé par des $|0\rangle$, on obtient alors des erreurs localisées). ◇

Lorsqu'un code encode un qubit sur n qubits, on peut parler du registre encodé comme étant un état logique encodé.

Définition 3.12 Soit Q un code $[[n, 1]]$ et soit $\mathcal{C}$ la transformation d'encodage. Alors le *zéro logique* et le *un logique* sont, respectivement :

$$\begin{aligned} |\overline{0}\rangle &= \mathcal{C}(|0\rangle) \in \mathcal{H}^n \\ |\overline{1}\rangle &= \mathcal{C}(|1\rangle) \in \mathcal{H}^n \end{aligned}$$

et pour $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathcal{H}$, le $|\psi\rangle$ *logique* est

$$|\overline{\psi}\rangle = \mathcal{C}(|\psi\rangle) = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \in \mathcal{H}^n$$

◇

3.3 Exemple : le code de Steane

Le code [[7,1,3]] de Steane (introduit dans [Ste96a]) est l'un des premiers exemples de code correcteur quantique. Il permet d'encoder un qubit sur sept qubits et peut corriger une erreur arbitraire sur un des qubits. Ce code est basé sur le code classique de Hamming H, que nous avons vu à la section 1.3. La figure 3.1 rappelle les mots de codes de Hamming, ainsi que leur parité (la première sous-colonne contient les quatre bits qui forment le message non-encodé).

$c \in H$		Parité
0000	000	0
0001	111	0
0010	110	1
0011	001	1
0100	101	1
0101	010	1
0110	011	0
0111	100	0

$c \in H$		Parité
1000	011	1
1001	100	1
1010	101	0
1011	010	0
1100	110	0
1101	001	0
1110	000	1
1111	111	1

FIG. 3.1: Mots de code de Hamming et leur parité

Notons $\pi(b)$ la parité d'une chaîne de bits b, i.e. $\pi(b_1 \dots b_n) = \sum_{i=1}^{n} b_i \pmod 2$. Dans le code de Steane, le zéro logique $|\overline{0}\rangle$ est la superposition de tous les mots de code de Hamming de parité 0, et le un logique $|\overline{1}\rangle$, de tous ceux de parité 1.

$$
\begin{aligned}
|\overline{0}\rangle &= \frac{1}{\sqrt{8}}\Big(\sum_{\substack{c\in H \\ \pi(c)=0}} |c\rangle \Big) \\
&= \frac{1}{\sqrt{8}}\Big(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + \\
&\qquad |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle \Big) \qquad (3.2)
\end{aligned}
$$

$$
\begin{aligned}
|\overline{1}\rangle &= \frac{1}{\sqrt{8}}\Big(\sum_{\substack{c\in H \\ \pi(c)=1}} |c\rangle \Big) \\
&= \frac{1}{\sqrt{8}}\Big(|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + \\
&\qquad |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle \Big) \qquad (3.3)
\end{aligned}
$$

Donc, le qubit $|\psi\rangle$ est encodé par

$$|\overline{\psi}\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \qquad (3.4)$$

La distance minimale de 3 du code de Steane est directement liée au fait que le code de Hamming ait la même distance minimale. En effet, chaque mot de base de Steane est un mot de code de Hamming, et pour faire passer de l'un à l'autre, il faut au moins trois erreurs de négation.

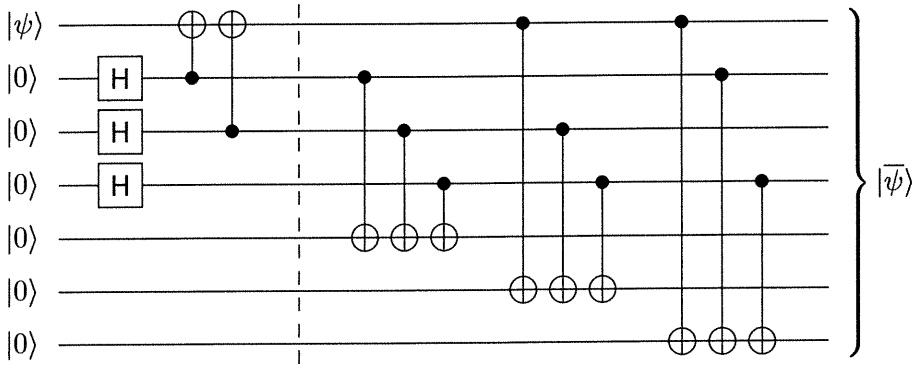


FIG. 3.2: Circuit de la matrice génératrice du code de Steane

Le circuit de la figure 3.2 permet créer la superposition (3.4) à partir d'un qubit arbitraire $|\psi\rangle$. L'état de départ est $|\psi\rangle \otimes |000000\rangle$. Après les trois Walsh-Hadamard, l'état devient

$$\Big(\tfrac{\alpha}{\sqrt{8}}\big(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |0110\rangle + |0111\rangle\big) + \tfrac{\beta}{\sqrt{8}}\big(|1000\rangle + |1001\rangle + |1010\rangle + |1011\rangle + |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle\big)\Big) \otimes |000\rangle$$

qui, après les deux ou-exclusifs, évolue à (les états modifiés sont soulignés)

$$\Big(\tfrac{\alpha}{\sqrt{8}}\big(|0000\rangle + |0001\rangle + |\underline{1010}\rangle + |\underline{1011}\rangle + |\underline{1100}\rangle + |\underline{1101}\rangle + |0110\rangle + |0111\rangle\big) + \tfrac{\beta}{\sqrt{8}}\big(|1000\rangle + |1001\rangle + |\underline{0010}\rangle + |\underline{0011}\rangle + |\underline{0100}\rangle + |\underline{0101}\rangle + |1110\rangle + |1111\rangle\big)\Big) \otimes |000\rangle$$

Pour les états d'amplitude $\frac{\alpha}{\sqrt{8}}$, les quatre premiers qubits représentent un mot de $\mathbb{Z}_2^4$ qui deviendra un mot de code de Hamming de parité 0. Il suffit de modifier les trois derniers qubits ($|000\rangle$) de la bonne façon. De même, les états d'amplitude $\frac{\beta}{\sqrt{8}}$ deviendront des mots de code de parité 1. La modification des trois derniers qubits est effectuée par les ou-exclusifs qui suivent la ligne pointillée. Puisque la matrice génératrice G du code de Hamming est en forme standard et que chacune des trois dernières colonnes contient trois 1, il suffit pour y arriver d'appliquer trois ou-exclusifs sur chacun des trois derniers qubits, comme illustré dans le circuit.

Comme souligné dans la section 3.1, il suffit de pouvoir corriger une erreur de négation, de phase et de phase-négation pour pouvoir corriger une erreur arbitraire sur un qubit. Le code de Steane corrige facilement une erreur de négation en utilisant les propriétés de correction du code de Hamming. En effet, si on note N_i l'opérateur qui applique N sur le i^{e} qubit et l'identité partout ailleurs, alors

$$\mathsf{N}_i|\overline{\psi}\rangle = \frac{\alpha}{\sqrt{8}}\Big(\sum_{\substack{c\in H\\ \pi(c)=0}} \mathsf{N}_i|c\rangle\Big) + \frac{\beta}{\sqrt{8}}\Big(\sum_{\substack{c\in H\\ \pi(c)=1}} \mathsf{N}_i|c\rangle\Big).$$

Ainsi, chaque état de la superposition devient un mot de code de Hamming avec une erreur à la i^{e} position. Si on effectuait une mesure complète du registre, i.e. si on mesurait

les sept qubits, on pourrait apprendre quelle erreur est survenue en appliquant la matrice de parité P sur la chaîne aléatoirement obtenue. Cependant, une telle procédure détruirait la superposition et on perdrait l'état initial $|\psi\rangle$. Pour éviter cela, on applique la version quantique de P, définie par

$$|x\rangle \otimes |000\rangle \rightarrow |x\rangle \otimes |xP^{\mathsf{T}}\rangle$$

sur l'état $\mathsf{N}_i|\overline{\psi}\rangle$. Par linéarité, P sera appliquée sur chaque mot de code (qui sont tous erronés de la même manière). Le système ancillaire contiendra donc, pour chacun, la même information : la représentation binaire de la position de l'erreur. La figure 3.3 donne le circuit qui implante cette opération. On le comprend facilement en étudiant la matrice de parité du code de Hamming. En mesurant les trois qubits ancillaires, on obtient trois bits classiques tels que $i = 4s_1 + 2s_2 + s_3$. Pour corriger l'erreur, on applique N sur le i^{e} qubit. On remarque que le calcul du syndrome d'erreur ne nous apprend rien sur l'état encodé, ce qui est nécessaire pour pouvoir le corriger (sinon, on détrurait la superposition). Donc, l'état redevient $|\overline{\psi}\rangle$ après la correction.

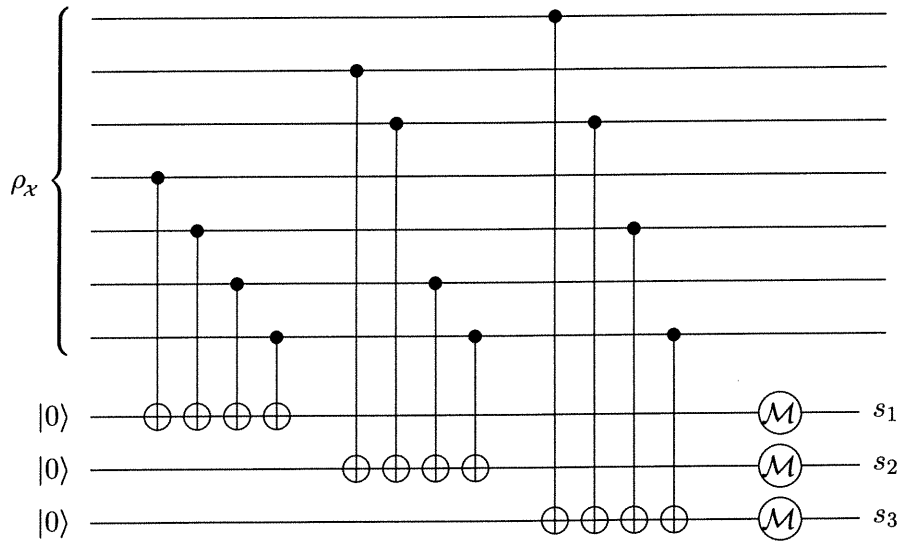


FIG. 3.3: Circuit pour calculer le syndrome du code de Steane

Voyons maintenant comment corriger une erreur de phase. Puisqu'on a les relations suivantes

$$\mathsf{HPH} = \mathsf{N} \text{ et } \mathsf{HNH} = \mathsf{P}, \tag{3.5}$$

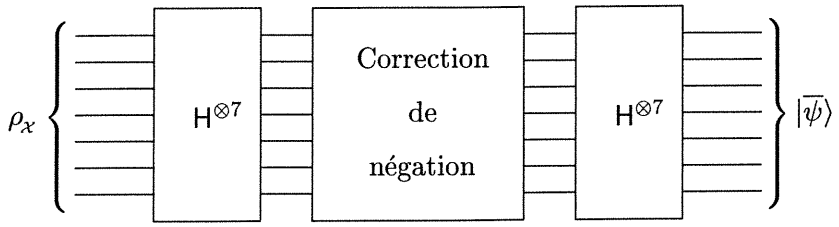


FIG. 3.4: Circuit de correction des erreurs de phase du code de Steane

on remarque que les erreurs de phase sont des erreurs de négation dans la base diagonale. Donc, la correction des erreurs de phase dans la base standard se réduit à la correction des erreurs de négation dans la base diagonale. Heureusement, le circuit 3.3 permet de détecter les erreurs de négation dans la base diagonale. En effet, dans la base diagonale, les états logiques deviennent :

$$\begin{aligned} \mathsf{H}^{\otimes 7}|\overline{0}\rangle &= \tfrac{1}{4}\Big(\textstyle\sum_{c\in H} |c\rangle \Big) \\ &= \tfrac{1}{\sqrt{2}}|\overline{0}\rangle + \tfrac{1}{\sqrt{2}}|\overline{1}\rangle \\ &= |\overline{\nearrow}\rangle \end{aligned}$$

$$\begin{aligned} \mathsf{H}^{\otimes 7}|\overline{1}\rangle &= \tfrac{1}{4}\Big(\textstyle\sum_{c\in H} (-1)^{w(c)}|c\rangle \Big) \\ &= \tfrac{1}{\sqrt{2}}|\overline{0}\rangle - \tfrac{1}{\sqrt{2}}|\overline{1}\rangle \\ &= |\overline{\nwarrow}\rangle \end{aligned}$$

Donc, puisque les états logiques diagonaux sont aussi formés de mots de code de Hamming, la procédure de correction des erreurs de négation permet aussi de corriger une erreur de phase. Pour y arriver, on applique une série de Walsh-Hadamard sur le registre avant d'appliquer la procédure de correction d'erreurs de négation (pour changer de base), puis on réapplique une même série de Walsh-Hadamard pour ramener le registre en base standard. Ceci est schématisé dans la figure 3.4.

La procédure de correction du code de Steane consiste donc à appliquer le circuit 3.3 et à renverser l'erreur de négation détectée s'il y a lieu, puis d'appliquer le circuit 3.4 et de corriger l'erreur de phase si on en trouve une. Notons que si un qubit subit une erreur de phase-négation, cette procédure permet de la corriger automatiquement.

En effet, la première partie corrigera la négation et la seconde, la phase.

Si on veut retrouver l'état $|\psi\rangle$ original, il suffit d'appliquer le circuit 3.2 d'encodage à l'envers, i.e. de droite à gauche. On obtiendra alors un produit tensoriel de $|\psi\rangle$ avec six $|0\rangle$ qu'on peut jeter. Si U_C représente le circuit d'encodage, cette opération consiste à appliquer $U_C^\dagger$ à l'état logique.

Le théorème 3.2 permet de conclure que le code de Steane corrige une erreur arbitraire sur l'un des qubits. Qu'arrive-t-il si plus d'un qubit est atteint ? Si deux erreurs de négation affectent deux qubits différents, l'opération de décodage ne fonctionnera pas. Les mots de base de parité 0 seront corrigés (incorrectement) à des mots de base de parité 1, et vice versa (ceci est une propriété du code de Hamming). Donc, le décodage introduira une erreur de négation logique, i.e.

$$\begin{aligned} |\overline{0}\rangle &\xrightarrow{\mathcal{D}} |\overline{1}\rangle \\ |\overline{1}\rangle &\xrightarrow{\mathcal{D}} |\overline{0}\rangle \end{aligned}$$

De plus, si deux erreurs de phase affectent deux qubits, ils agissent comme des négations dans la base diagonale, ce qui correspond à une erreur de négation logique et agit donc comme une erreur de phase logique dans la base standard.

$$\begin{aligned} |\overline{0}\rangle &\xrightarrow{\mathcal{D}} |\overline{0}\rangle \\ |\overline{1}\rangle &\xrightarrow{\mathcal{D}} -|\overline{1}\rangle \end{aligned}$$

Si une erreur de négation et une erreur de phase affectent deux qubits (ou le même), alors les erreurs seront corrigées correctement par la procédure décrite.

Si, pour un opérateur de Pauli $E = E_1 \otimes \ldots \otimes E_n$, n_i^E (pour $0 \leq i \leq 3$) est le nombre de E_j (pour $1 \leq j \leq n$) tel que $E_j = \sigma_i$, alors pour le code de Steane, les erreurs corrigeables sont

$$\mathcal{E}' = \{\, E \in \mathcal{P}^{\otimes n} \,:\, \max(n_1^E + n_3^E, n_2^E + n_3^E) \leq 1 \,\}$$

(i.e. il peut corriger une erreur de phase et une erreur de négation, peu importe sur quel qubit) mais, pour simplifier, on se contente de dire que

$$\mathcal{E} = \{\, E \in \mathcal{P}^{\otimes n} \,:\, w(E) \leq 1 \,\} \subseteq \mathcal{E}' \tag{3.6}$$

Donc, le code de Steane permet de corriger une erreur arbitraire, tel que prédit par sa distance minimale de 3.

Est-ce que le code de Steane nous est utile, i.e. permet-il de protéger l'information quantique ? Pour étudier cette question, nous devons considérer un canal en particulier. Si on utilise le canal $\mathcal{QD}_p$, alors la probabilité qu'un qubit transmis directement à travers le canal soit erroné est $p_{\text{direct}} = p$. Si on utilise le code de Steane et si on considère l'ensemble d'erreurs de l'équation 3.6 alors on obtient une erreur de décodage si au moins deux erreurs affectent le registre. Ceci survient avec probabilité

$$p_{\text{Steane}} = 1 - \text{prob(aucune ou une seule erreur)} = 1 - \Big((1-p)^7 + 7p(1-p)^6\Big).$$

Donc, si $0 < p < 0,0578$, alors $p_{\text{Steane}} < p_{\text{direct}} = p$. Par exemple, si $p = p_{\text{direct}} = 0,05$, alors $p_{\text{Steane}} \approx 0,0444$. On voit que le code de Steane ne nous est utile que si le canal est très fiable (ce qui n'est pas très utile en pratique). Heureusement, il existe d'autres codes qui sont plus efficaces (nous en verrons quelques uns dans la section 3.6).

3.4 Codes CSS

Nous décrirons maintenant une famille importante de codes quantiques, les codes CSS, découverts par Calderbank et Shor [CS96] et indépendemment par Steane [Ste96b].

Soit C_1 un code $[n, k_1, d_1]$ linéaire et soit $C_2 \subset C_1$ un code $[n, k_2]$ avec $k_2 < k_1$. Le sous-code C_2 induit une relation d'équivalence sur les mots de code de C_1. On dit que $u, v \in C_1$ sont équivalents s'il existe un $w \in C_2$ tel que $u = v + w$. Les classes d'équivalence sont appelés des cosets. Il y a $k_1 - k_2$ cosets et on note l'ensemble des cosets par C_1/C_2. Donc, pour $u \in C_1$,

$$\text{Coset}(u) = \{\, v \in C_1 \,:\, \exists_{w \in C_2}\, u = v + w \,\}.$$

On note que si $u, v \in C_1$ et que $u - v \in C_2$, alors $u \in \text{Coset}(v)$.

On forme le code CSS $Q\ [[n, k_1 - k_2]]$ à partir C_1 et C_2 de la façon suivante. On associe un mot de base à chaque coset de C_2 dans C_1. Soit v un représentant d'un des

cosets, alors le mot de base associé est

$$|\overline{v}\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w \in C_2} |v + w\rangle.$$

Donc, pour un $v \in C_1$, $|\overline{v}\rangle$ est une superposition égale de tous les mots de code dans le Coset(w). Les $|\overline{v}\rangle$ sont bien définis, car si $u \in$ Coset(v), alors soit $w \in C_2$ tel que $u = v + w$. On a que

$$\begin{aligned} |\overline{u}\rangle &= \tfrac{1}{\sqrt{2^{k_2}}} \textstyle\sum_{y \in C_2} |v + y\rangle \\ &= \tfrac{1}{\sqrt{2^{k_2}}} \textstyle\sum_{y \in C_2} |u + w + y\rangle \\ &= \tfrac{1}{\sqrt{2^{k_2}}} \textstyle\sum_{y' \in C_2} |v + y'\rangle \\ &= |\overline{v}\rangle \end{aligned}$$

De plus si, pour $u, v \in C_1$, $u \notin$ Coset(v), alors pour tout $w \in C_2$, $u \neq v + w$. Puisque pour $w, w' \in C_2$,

$$\langle u + w | v + w' \rangle = 1 \iff u + w = v + w' \iff u - v = w' - w \in C_2,$$

alors $\langle u + w | v + w' \rangle = 0$ et donc $\langle \overline{u} | \overline{v} \rangle = 0$. Les mots de base définissent donc une base pour Q qui est un sous-espace de dimension $2^{k_1 - k_2}$ de $\mathcal{H}^n$. Tel qu'annoncé, Q est donc bien un code $[[n, k_1 - k_2]]$.

Le circuit qui génère le code Q n'est, en général, pas facile à trouver. Cleve et Gottesman ont développé une méthode pour construire le circuit d'encodage de façon efficace [CG96].

Le procédure de correction d'erreurs du code Q utilise celles des codes C_1 et $C_2^{\perp}$. Soit d_1 la distance minimale de C_1, alors le code Q peut corriger $\lfloor \frac{d_1 - 1}{2} \rfloor$ erreurs de négation. En effet, pour $e \in \{0,1\}^n$ de poids inférieur ou égal à $\lfloor \frac{d_1 - 1}{2} \rfloor$, soit $E_e^{\sigma_1} = E_1 \otimes \ldots \otimes E_n$ un opérateur de Pauli où $E_i = \sigma_0$ si $e_i = 0$ et $E_i = \sigma_1$ sinon. L'effet de $E_e^{\sigma_1}$ sur un état $|u\rangle$ $(u \in \{0,1\}^n)$ est

$$|u\rangle \xrightarrow{E_e^{\sigma_1}} |u + e\rangle.$$

Donc, si un mot de base $|\overline{v}\rangle \in Q$ est affecté par $E_e^{\sigma_1}$, alors $|\overline{v}\rangle$ (qui était une superposition de mots de code de C_1) devient une superposition de mots de codes de C_1 avec des

négations aux positions dictées par e. On se sert de la matrice de parité P_1 du code C_1 pour diagnostiquer l'erreur. Pour ce faire, on ajoute une ancille $|0^{n-k_1}\rangle$ à l'état et on applique la transformation unitaire suivante :

$$|v\rangle \otimes |0^{n-k_1}\rangle \longrightarrow |v\rangle \otimes |vP_1^{\mathsf{T}}\rangle. \tag{3.7}$$

Appliquée sur un état erroné $|v+e\rangle$, la transformation 3.7 donne

$$|v+e\rangle \otimes |0^{n-k_1}\rangle \longrightarrow |v+e\rangle \otimes |eP_1^{\mathsf{T}}\rangle. \tag{3.8}$$

En mesurant l'ancille, on obtient le syndrome d'erreur qu'on analyse classiquement pour déterminer la position des erreurs. On applique alors des négations aux qubits correspondant pour retrouver $|\overline{v}\rangle$.

Le circuit qui implante la transformation 3.7 est facile à trouver. Comme dans le code de Steane, il suffit de faire des ou-exclusifs sur les qubits ancillaires selon la matrice de parité. Supposons qu'on veuille calculer le i^{e} bit du syndrome, on applique un ou-exclusif du j^{e} qubit vers la i^{e} ancille si $(P_{ij})^{\mathsf{T}} = 1$ (pour $1 \leq j \leq n$).

La correction des erreurs de phases est légèrement plus complexe. Notons d'abord ce qui survient lorsqu'on applique $\mathsf{H}^{\otimes n}$ à un vecteur de base $|\overline{v}\rangle$.

$$\begin{aligned}
|\overline{v}\rangle &= \frac{1}{\sqrt{2^{k_2}}} \sum_{w \in C_2} |v+w\rangle \\
&\xrightarrow{\mathsf{H}^{\otimes n}} \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{2^{k_2}}} \sum_{u \in \{0,1\}^n} \sum_{w \in C_2} (-1)^{u\cdot(v+w)} |u\rangle \quad \text{(par l'équation 2.17)} \\
&= \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{2^{k_2}}} \left[\sum_{u \in C_2^\perp} \sum_{w \in C_2} (-1)^{u\cdot v} (-1)^{u\cdot w} |u\rangle + \sum_{u \notin C_2^\perp} \sum_{w \in C_2} (-1)^{u\cdot v} (-1)^{u\cdot w} |u\rangle \right] \\
&= \frac{1}{\sqrt{2^{n-k_2}}} \sum_{u \in C_2^\perp} (-1)^{u\cdot v} |u\rangle,
\end{aligned} \tag{3.9}$$

où la dernière égalité utilise le lemme 1.27. L'équation 3.9 est une superposition de mots de code de $C_2^\perp$.

Soit $d_2^\perp$ la distance minimale de $C_2^\perp$. Alors on pourra corriger $\lfloor \frac{d_2^\perp - 1}{2} \rfloor$ erreurs de phase. Définissons $E_e^{\sigma_2}$ de façon semblable à $E_e^{\sigma_1}$, sauf que $E_i = \sigma_2$ si $e_i = 1$. De plus,

considérons seulement les e tels que $w(e) \leq \lfloor \frac{d_2^{\perp}-1}{2} \rfloor$. Alors, l'opérateur $E_e^{\sigma_2}$ agit comme suit sur un état $|u\rangle$ $(u \in \{0,1\})$:

$$|u\rangle \xrightarrow{E_e^{\sigma_2}} (-1)^{u.e}|u\rangle,$$

ce qui est équivalent à (par l'équation 3.5) $\mathsf{H}^{\otimes n} E_e^{\sigma_1} \mathsf{H}^{\otimes n}|u\rangle$. Ainsi, les erreurs de phase deviennent des erreurs de négation dans la base diagonale.

Puisque dans la base diagonale les mots de code $|\overline{v}\rangle$ sont des superpositions de mots de code de $C_2^{\perp}$ et que les erreurs de phase deviennent des erreurs de négation, on peut se servir de la matrice de parité G_2 (qui est la matrice génératrice de C_2) pour les diagnostiquer. En appliquant d'abord une série de Walsh-Hadamard sur le registre puis en appliquant ensuite la transformation unitaire

$$|v\rangle \otimes |0^{k_2}\rangle \longrightarrow |v\rangle \otimes |vG_2^{\mathsf{T}}\rangle \tag{3.10}$$

sur le registre et l'ancille, et en mesurant par la suite l'ancille, on obtient le syndrome d'erreur de phase. On corrige ensuite les erreurs en appliquant des négations aux qubits concernés (déterminés par le code $C_2^{\perp}$). Pour terminer, on transpose le registre dans la base standard en appliquant de nouveau une porte de Walsh-Hadamard sur chaque qubit.

Pour corriger les erreurs de phase-négation NP, il suffit d'appliquer séparément les procédures de correction de négation et de phase. L'erreur affectant le qubit sera corrigée en deux étapes. Ceci est une propriété importante des codes CSS. Les procédures de correction des erreurs de phases et des erreurs de négation sont indépendantes et l'ordre dans laquelle on les effectue n'a pas d'importance.

La distance minimale d d'un code CSS est directement reliée aux distances minimales des codes C_1 et $C_2^{\perp}$. En effet, on a la relation suivante :

$$d \geq \min(d_1, d_2^{\perp}) \tag{3.11}$$

Remarque 3.13 Les procédures d'encodage et de décodage fonctionnent de la même façon si on utilise des états mélangés. Si on a un état mélangé $\rho = \left\{(p_i, |\psi_i\rangle)\right\} \in \mathcal{H}^k$

alors, pour un code Q $[[n,k]]$, l'état sera encodé à

$$\mathcal{C}_Q(\rho) = \Big\{(p_i, \mathcal{C}_Q(|\psi_i\rangle)\Big\}$$

◇

Remarque 3.14 Un cas particulier survient lorsqu'on construit un code CSS à partir d'un code linéaire C_1 tel que $C_1^\perp \subset C_1$. Si on choisit $C_2 = C_1^\perp$, alors $C_2^\perp = C_1$. Donc on applique deux fois la même procédure de correction, une fois dans la base standard, une fois dans la base diagonale. ◇

Exemple 3.15 Le code de Steane est un code CSS. Ici, $C_1 = H$ et $C_2 = H^\perp \subset H$ tel que mentionné dans l'exemple 1.31. Le code de Steane illustre bien la remarque précédente. En effet, dans ce cas, le même circuit est utilisé pour corriger les erreurs de phase et les erreurs de négation. ◇

3.5 Code d'effacement sur quatre qubits

Dans cette section, nous donnerons un exemple de code d'effacement. Le code permettra d'encoder un qubit $|\psi\rangle \in \mathcal{H}$ sur quatre et de corriger une erreur d'effacement. Le code ne peut cependant corriger aucune erreur arbitraire.

Pour le code, le zéro et le un logique sont

$$\begin{aligned} |\overline{0}\rangle &= \tfrac{1}{\sqrt{2}}\Big(|0000\rangle + |1111\rangle\Big) \\ |\overline{1}\rangle &= \tfrac{1}{\sqrt{2}}\Big(|0011\rangle + |1100\rangle\Big) \end{aligned}$$

et donc, un état $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ sera encodé par

$$\begin{aligned} |\overline{\psi}\rangle &= \frac{\alpha}{\sqrt{2}}|\overline{0}\rangle + \frac{\beta}{\sqrt{2}}|\overline{1}\rangle \\ &= \frac{\alpha}{\sqrt{2}}\Big(|0000\rangle + |1111\rangle\Big) + \frac{\beta}{\sqrt{2}}\Big(|0011\rangle + |1100\rangle\Big) \end{aligned} \tag{3.12}$$

La figure 3.5 permet de créer cette superposition.

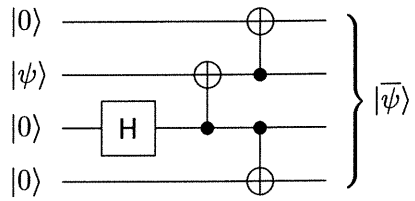


FIG. 3.5: Cicuit d'encodage du code sur quatre qubits

Supposons que le registre est transmis à travers le canal d'effacement et que l'on reçoive $\rho = \mathcal{Q}\Sigma_p(|\overline{\psi}\rangle)$. Supposons que l'un des qubits ait été effacé (on sait quel qubit est affecté car un $|\varepsilon\rangle$ est orthogonal à $|0\rangle$ et $|1\rangle$). On peut corriger l'erreur et reconstituer l'état $|\overline{\psi}\rangle$ en appliquant un des circuits de la figure 3.6. En effet, si on isole chaque qubit (notés A, B, C et D) de

$$|\overline{\psi}\rangle = \frac{\alpha}{\sqrt{2}}\Big(|0_A 0_B 0_C 0_D\rangle + |1_A 1_B 1_C 1_D\rangle\Big) + \frac{\beta}{\sqrt{2}}\Big(|0_a 0_B 1_C 1_D\rangle + |1_a 1_B 0_C 0_D\rangle\Big)$$

on obtient que

$$\begin{aligned}
|\overline{\psi}\rangle &= \frac{1}{\sqrt{2}}|0_A\rangle \otimes \Big(\alpha|0_B 0_C 0_D\rangle + \beta|0_B 1_C 1_D\rangle\Big) + \frac{1}{\sqrt{2}}|1_A\rangle \otimes \Big(\alpha|1_B 1_C 1_D\rangle + \beta|1_B 0_C 0_D\rangle\Big) \\
&= \frac{1}{\sqrt{2}}|0_B\rangle \otimes \Big(\alpha|0_A 0_C 0_D\rangle + \beta|0_A 1_C 1_D\rangle\Big) + \frac{1}{\sqrt{2}}|1_B\rangle \otimes \Big(\alpha|1_A 1_C 1_D\rangle + \beta|1_A 0_C 0_D\rangle\Big) \\
&= \frac{1}{\sqrt{2}}|0_C\rangle \otimes \Big(\alpha|0_A 0_B 0_D\rangle + \beta|1_A 1_B 0_D\rangle\Big) + \frac{1}{\sqrt{2}}|1_C\rangle \otimes \Big(\alpha|1_A 1_B 1_D\rangle + \beta|0_A 0_B 1_D\rangle\Big) \\
&= \frac{1}{\sqrt{2}}|0_D\rangle \otimes \Big(\alpha|0_A 0_B 0_C\rangle + \beta|1_A 1_B 0_C\rangle\Big) + \frac{1}{\sqrt{2}}|1_D\rangle \otimes \Big(\alpha|1_A 1_B 1_C\rangle + \beta|0_A 0_B 1_C\rangle\Big)
\end{aligned}$$

On remarque que l'état des qubits restant est le même si on isole le qubit A ou B, de même pour les qubits C et D (ce qui explique pourquoi les circuits de correction sont identiques pour la perte du qubit A ou B et pour C ou D). Par exemple, si le premier qubit est effacé, on obtient l'état

$$|\varepsilon_A\rangle \otimes \left[\frac{\alpha}{\sqrt{2}}\Big(|0_B 0_C 0_D\rangle + |1_B 1_C 1_D\rangle\Big) + \frac{\beta}{\sqrt{2}}\Big(|0_B 1_C 1_D\rangle + |1_B 0_C 0_D\rangle\Big)\right]$$

Après les deux ou-exclusifs du premier circuit de la figure 3.6, l'état devient

$$\begin{aligned}
\xrightarrow{O^{C,D}} \quad & |\varepsilon_A\rangle \otimes \left[\frac{\alpha}{\sqrt{2}}\Big(|0_B0_C0_D\rangle + |1_B1_C0_D\rangle\Big) + \frac{\beta}{\sqrt{2}}\Big(|0_B1_C0_D\rangle + |1_B0_C0_D\rangle\Big)\right] \\
\xrightarrow{O^{B,C}} \quad & |\varepsilon_A\rangle \otimes \left[\frac{\alpha}{\sqrt{2}}\Big(|0_B0_C0_D\rangle + |1_B0_C0_D\rangle\Big) + \frac{\beta}{\sqrt{2}}\Big(|0_B1_C0_D\rangle + |1_B1_C0_D\rangle\Big)\right] \\
= \quad & |\varepsilon_A\rangle \otimes \Big(\tfrac{1}{\sqrt{2}}|0_B\rangle + \tfrac{1}{\sqrt{2}}|1_B\rangle\Big) \otimes \Big(\alpha|0_C\rangle + \beta|1_C\rangle\Big) \otimes |0_D\rangle \\
= \quad & |\varepsilon_A\rangle \otimes |\nearrow\swarrow_B\rangle \otimes |\psi_C\rangle \otimes |0_D\rangle
\end{aligned}$$

Le qubit $|\psi\rangle$ est donc reconstitué sur le troisième fil.

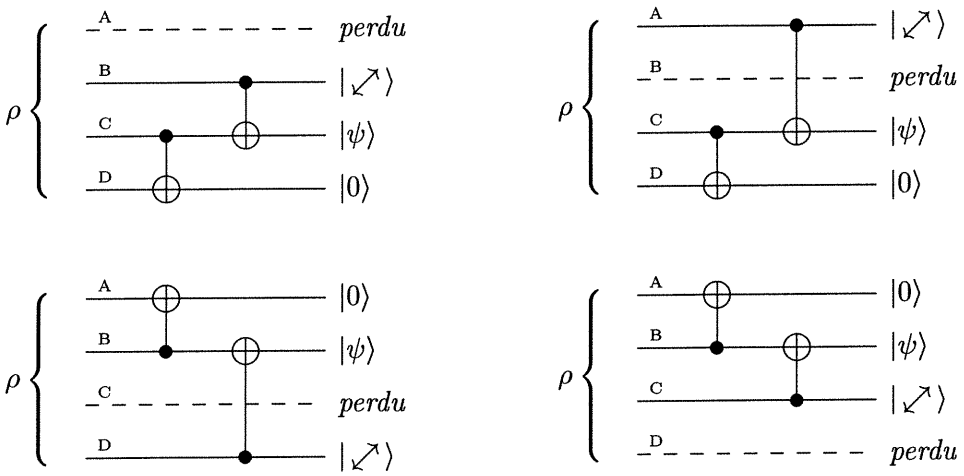


FIG. 3.6: Circuits pour reconstituer $|\psi\rangle$ selon le qubit effacé

Le code ne fonctionne pas si plus d'une erreur d'effacement affectent le registre, ce qui survient avec probabilité

$$p_4 = 1 - \text{prob(aucune ou une seule erreur)} = 1 - \Big((1-p)^4 + 4p(1-p)^3\Big).$$

Donc, si $0 < p < 0,232$, alors $p_4 < p_{\text{direct}} = p$. Par exemple, si $p = p_{\text{direct}} = 0,2$, alors $p_4 \approx 0,181$.

Le théorème suivant, prouvé par [GBP97], indique que le code présenté ici est le plus court pour les erreurs d'effacement.

Théorème 3.16 Il n'existe pas de code correcteur quantique $[[n, 1]]$ pour $n < 4$ qui corrige une erreur d'effacement. ◇

Remarque 3.17 Malgré le fait que le code ne permet pas de corriger deux erreurs d'effacement, on peut cependant tirer un peu d'information à partir de deux qubits seulement. En effet, supposons qu'on possède les qubits un et trois de la superposition 3.12, alors on pourrait différencier les états $|\psi\rangle = |0\rangle$ et $|\psi\rangle = |1\rangle$. Et si $|\psi\rangle$ est en superposition, on pourrait obtenir de l'information statistique sur $|\alpha|$ et $|\beta|$ en mesurant les deux qubits. Cette remarque sera importante dans la section 5.2. ◇

3.6 Bornes et autres constructions

Plusieurs constructions de codes efficaces ont vu le jour depuis la découverte des codes correcteurs quantiques. En général, pour qu'un code soit efficace, il doit pouvoir encoder plus d'un qubit (i.e. $k > 1$).

Comme dans le monde classique, il existe des bornes qui donnent des contraintes sur les paramètres des codes quantiques. Nous les présentons simplement, car certaines preuves demandent des notions que nous n'avons pas présentées. Notons que celles présentées ici sont des généralisations quantiques des bornes de la section 1.5

Théorème 3.18 (Borne de Hamming quantique)
Soit Q un code $[[n, k, 2t+1]]$ non-dégénéré. On a que

$$\sum_{i=0}^{t} 3^i \binom{n}{i} \leq 2^{n-k} \tag{3.13}$$

On obtient cette borne en comparant le nombre de façons de choisir jusqu'à t erreurs et la dimension de l'espace nécessaire pour les corriger. Cette borne nous apprend que le code non-dégénéré le plus court qui encode un qubit ($k = 1$) et qui corrige une erreur ($t = 1$) doit avoir une longueur qui satisfait $1 + 3n \leq 2^{n-1}$. Cette inégalité est satisfaite pour $n \geq 5$. ◇

Théorème 3.19 (Borne de Singleton quantique)
Soit Q un code $[[n, k, d]]$. Alors

$$n - k \geq 2(d-1). \tag{3.14}$$

Cette borne ressemble à la borne classique de Singleton (équation 1.8), mais le facteur 2 est ajouté en raison du théorème de non-clonage. Il est clair que n doit être supérieur à $2(d-1)$ (car Q corrige $d-1$ erreurs d'effacement et si on avait deux blocs disjoints de $d-1$ qubits, on pourrait reconstituer deux copies de l'état original), l'ajout de k dans l'inégalité est plus difficile à dériver. ◇

Théorème 3.20 (Borne de Gilbert-Varshamov quantique)
Il existe un code Q $[[n,k,d]]$ tel que

$$\sum_{i=0}^{d-1} 3^i \binom{n}{i} \leq 2^{n-k} \tag{3.15}$$

Cette borne à été dérivée dans [CRSS97]. ◇

Théorème 3.21 (Borne de Gilbert-Varshamov quantique asymptotique)
Pour de grand n, et pour $R = k/n$ et $p = d/2n$ fixé, les meilleurs codes quantiques non-dégénérés satisfont

$$1 - 2p \lg 3 - \mathbf{H}(2p) \leq R \leq 1 - p \lg 3 - \mathbf{H}(p), \tag{3.16}$$

où $\mathbf{H}(x)$ est la fonction d'entropie definie dans l'équation A.2. L'existence de tels codes (des codes CSS, entre autres) a été démontré dans [CS96]. ◇

Voici quelques exemples de codes quantiques importants.

Steane [Ste97] à découvert plusieurs codes en utilisant la construction CSS. Certains de ses codes ont des paramètres tels $[[13,5,3]]$, $[[14,6,3]]$, $[[17,7,3]]$ et $[[20,9,3]]$. De plus, il a développé une méthode pour améliorer certains code CSS [Ste98].

Le code parfait le plus court a été découvert par Laflamme, Miquel, Paz et Zurek [LMPZ96] et indépendamment par Bennett, DiVincenzo, Smolin et Wooters [BDSW97]. Il s'agit d'un code $[[5,1,3]]$ qui permet d'encoder un qubit sur cinq qubits et qui corrige une erreur arbitraire sur l'un des qubits. La longueur $n = 5$ satisfait la borne de Hamming décrite par l'équation 3.13.

Certains codes CSS plus efficaces ont été développés en utilisant comme base de bons codes classiques. Par exemple, on peut utiliser le code de Golay G_{23} (qui est semi-dual)

pour construire un code CSS de paramètres $[[23, 1, 7]]$. Steane [Ste96c] a utilisé les codes semi-duaux de Reed-Muller pour former des codes CSS. Le code de Steane peut être vu comme un cas particulier de cette famille quantique. On peut aussi construire des codes CSS avec des codes de Reed-Salomon [GGB99] et des codes BCH [GB99].

Knill et Laflamme [KL96a] ont développé la méthode de concaténation de codes quantiques. Cette technique, très utile et efficace dans le calcul tolérant aux fautes, encode un registre de façon récursive sur plusieurs niveaux. Par exemple, le code de Steane encode un qubit sur sept. Leur technique consiste à encoder chacun des sept qubits de façon récursive avec le code de Steane. Ainsi, le qubit logique sera encodé avec 49 qubits physiques. Pour corriger les erreurs, on corrige les blocs de qubits qui forment une couche récurvise avant de corriger la suivante.

Finalement, Gottesman [Got97] a développé la structure des codes stabilisateurs qui permet de décrire la majorité les codes connus à ce jour (les codes CSS peuvent être décrits par des stabilisateurs). Les codes stabilisateurs sont les plus étudiés et les plus répandus dans la littérature.

3.7 Codes polynomiaux

On peut généraliser la théorie des codes quantiques aux états de plus de deux dimensions, comme on le fait classiquement. On cherche maintenant à encoder k qu$q$its sur n qu$q$its.

Définition 3.22 Un *code quantique q-aire* $[[n, k, d]]_q$ est un sous-espace $Q \subseteq \mathcal{H}_q^n$ de dimension q^k. Le code encode k qu$q$its sur n qu$q$its. L'image des états de base de $\mathcal{H}_q^k$ sont appelés les mots de base et ils forment une base orthonormée pour Q. Les états de Q sont appelés les *mots de code* de Q. ◇

La théorie développée dans les sections précédentes est aussi valide pour les codes q-aires. La construction des codes CSS sur $\mathcal{H}_q^n$ permet de corriger des erreurs de négation et de phase telles que définies dans la section 2.6.

Nous décrirons maintenant les codes polynomiaux quantiques, introduits par Aharonov et Ben-Or [ABO99]. Ce sont des codes q-aires CSS sur des qu$q$its que nous utiliserons au chapitre 5, dans le contexte de partage de secrets quantiques.

Nous travaillerons avec des polynômes de $\mathbb{F}_q[x]$ de degré non supérieur à d. Soit $a = (a_0, a_1, \ldots, a_d) \in \mathbb{F}_q^{d+1}$, alors le polynôme $f_a \in \mathbb{F}_q[x]$ est décrit, pour $x \in \mathbb{F}_q$, par

$$f_a(x) = a_0 + a_1 x + \ldots + a_d x^d.$$

Notons que les vecteurs décrivant les polynômes sont indexés de 0 à d. Dans ce qui suit, nous utiliserons les lemmes suivants.

Lemme 3.23 Soit f_a, $f_{a'} \in \mathbb{F}_q[x]$ tels que $\deg(f_a) = d$ et $\deg(f_{a'}) \leq d$, alors

$$\Big|\{\, x \in \mathbb{F}_q \;:\; f_a(x) = f_{a'}(x) \,\}\Big| \leq d.$$

En d'autres mots, le nombre de points où les polynômes s'intersectent est inférieur ou égal au degré des polynômes. ◇

Lemme 3.24 Pour des $x_1, \ldots, x_d \in \mathbb{F}_q$ distincts et non-nuls, on a que

$$\Big\{ \Big(f_a(x_1), \ldots, f_a(x_d)\Big) \;:\; a \in \mathbb{F}_q^{d+1} \Big\} = \mathbb{F}_q^d.$$

Autrement dit, pour toute chaîne $(y_1, \ldots, y_d) \in \mathbb{F}_q^d$, il existe un $a \in \mathbb{F}_q^{d+1}$ tel que $f_a(x_i) = y_i$ (pour $1 \leq i \leq d$). ◇

Soit $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$. Le code CSS polynomial Q_x^d de degré $d < n$ est construit avec les codes linéaires suivants :

$$\begin{aligned} C_1 &= \Big\{ \Big(f_a(x_1), \ldots, f_a(x_n)\Big) \;:\; a \in \mathbb{F}_q^{d+1} \Big\} \\ C_2 &= \Big\{ \Big(f_a(x_1), \ldots, f_a(x_n)\Big) \;:\; a \in \mathbb{F}_q^{d+1} \;\wedge\; a_0 = 0 \Big\} \end{aligned}$$

On remarque que $C_2 \subset C_1 \subset \mathbb{F}_q^n$. De plus, pour $u_a = \Big(f_a(x_1), \ldots, f_a(x_n)\Big) \in C_1$,

$$\mathrm{Coset}(u_a) = \{\, u_b \in C_1 \;:\; u_a - u_b \in C_2 \,\} = \{\, u_b \in C_1 \;:\; a_0 = b_0 \,\}$$

On voit qu'il y a q cosets de C_2 dans C_1. Le code Q_x^d protège donc un espace de dimension q, i.e. il encode un qu$q$it sur n qu$q$its. Pour chaque $u_a = \Big(f_a(x_1), \ldots, f_a(x_n)\Big) \in$

C_1, par la construction CSS, on a que,

$$|\overline{u_a}\rangle = \frac{1}{\sqrt{q^d}} \sum_{u_b \in C_2} |u_a + u_b\rangle = \frac{1}{\sqrt{q^d}} \sum_{\substack{c \in \mathbb{F}_q^{d+1} \\ c_0 = a_0}} |u_c\rangle$$

Donc, pour un état de base $|b\rangle \in \mathcal{H}_q$ (i.e. $b \in \mathbb{F}_q$), l'encodage par Q_x^d est

$$|b\rangle \longrightarrow \sum_{\substack{a \in \mathbb{F}^{d+1} \\ a_0 = b}} |f_a(x_1) \dots f_a(x_n)\rangle.$$

Par le lemme 3.23, la distance minimale d_1 du code C_1 est au moins $n - d$. En effet, puisque deux mots de code distincts $u, v \in C_1$ ont au maximum d positions identiques, alors $\Delta(u, v) \geq n - d$. De plus, la distance minimale $d_2^\perp$ du code $C_2^\perp$ est au moins $d+1$. On peut montrer cette affirmation en supposant que la distance minimale est inférieure ou égale à d. Il existerait donc un mot $u \in C_2^\perp$ non-nul de poids $d' \leq d$ tel que pour tout $v \in C_2$, $u \cdot v = 0$. Si u' est la chaîne de longueur d' obtenue en ôtant les 0 de u, et si, pour un $v \in C_2$, v' est la chaîne obtenue de v en gardant les mêmes positions que dans u', on aurait que

$$u \cdot v = u' \cdot v' = 0.$$

Puisque cette équation doit être vraie pour tout $v \in C_2$, on obtient par le lemme 3.24 que

$$u' \cdot w = 0 \quad \text{pour tout } w \in \mathbb{F}_q^{d'}.$$

Le seul $u' \in \mathbb{F}_q^{d'}$ qui satisfait cette dernière équation est le vecteur nul $(0^{d'})$, ce qui contredit l'hypothèse initiale. Donc, la distance minimale du code Q_x^d est, par l'équation 3.11, $d_Q \geq \min(d_1, d_2^\perp)$. Le code permet donc de corriger $\lfloor \frac{d_Q - 1}{2} \rfloor$ erreurs arbitraires et $d_Q - 1$ erreurs d'effacement.

Deuxième partie

Les applications cryptographiques

Chapitre 4

La distribution quantique de clés

En 1979, les chemins de la cryptographie et de l'informatique quantique se sont croisés lorsque Charles Bennett et Gilles Brassard ont fait connaissance, ce qui a mené au développement du premier[1] protocole de distribution quantique de clés (DQC) [BB84]. Le protocole permet à deux participants (qu'on nommera Alice et Bob) d'échanger une clé secrète aléatoire (i.e. une série de bits) pouvant être utilisée pour communiquer de façon parfaitement sécuritaire (par exemple, à l'aide du système de chiffrement du masque jetable[2][Sti95]). La DQC résout le problème important de distribution de clés qui limite l'utilisation de systèmes à clés secrètes dans certaines applications. Classiquement, l'échange de clés entre deux personnes devait se faire soit en privé (ce qui n'est souvent pas possible), soit par des systèmes à clés publiques (ce qui est vulnérable à un attaqueur très puissant[3]). Cependant, dans la DQC, un adversaire arbitrairement puissant qui tente d'espionner le canal de communication pour apprendre la clé sera détecté avec très grande probabilité. Autrement dit, le protocole ne camoufle pas la clé de façon mathématique, mais permet de détecter un adversaire qui tenterait d'espionner le canal durant l'échange de la clé, peu importe sa puissance de calcul.

[1] Leur idée est basée sur le travail de Wiesner [Wie83].

[2] Mieux connu sous son nom anglais de *one-time pad.*

[3] Les problèmes RSA et de log discret (voir [Sti95]), à la base des systèmes à clés publiques les plus populaires, sont "brisés" en temps sous-exponentiel par un ordinateur classique, et en temps polynomial par un ordinateur quantique [Sho97].

Plusieurs chercheurs ont proposé des preuves formelles de sécurité pour la DQC [May98, BBB+99, LC99]. Elles sont complexes (la plus simple ne s'applique pas au protocole BB84) et aucune ne satisfait la communauté scientifique dans son ensemble. Etonnamment, la théorie des codes correcteurs quantiques a été utilisée récemment par Shor et Preskill [SP00] pour donner une preuve relativement simple de la sécurité du protocole BB84. Nous présenterons cette preuve à la section 4.2.1.

La DQC a déjà été implantée en pratique. Le premier prototype, développé en 1989 par Bennett, Brassard, et certains de leurs élèves [BBB+92], a permis à Alice et Bob d'échanger une clé secrète à un taux de 10 bits/seconde sur une distance de 32,5 cm ! Un petit pas pour Alice et Bob, mais un grand pas pour la cryptographie quantique. L'appareil, comme la plupart de ses successeurs, utilisait des photons comme média de transport. Aujourd'hui, on retrouve des prototypes qui fonctionnent sur de longues distances et avec de meilleurs taux. Par exemple, une expérience tenue à Los Alamos a permis d'échanger des clés sur une distance de 48 km [HMP99]. Une autre a effectué le protocole sur une fibre optique de 23 km installée sous le lac de Genève [MZG95]. Une dernière expérience digne de mention a réussi la DQC en plein air (i.e. sans fibre optique) et en plein jour sur une distance de 1,6 km [BHL+00], ce qui donne bon espoir que le protocole pourrait un jour être utilisé pour les communications par satellites.

Malgré les succès expérimentaux, plusieurs obstacles ralentissent le déploiement de cette technologie hors des laboratoires. Un de ces facteurs, selon Lo [Lo99], sont les trop courtes distances atteintes par les prototypes ; ils devront permettre la communication sur des milliers de kilomètres pour que la DQC soit intéressante commercialement. Les erreurs survenant sur les canaux limitent les distances atteintes expérimentalement, car les photons transmis subissent les effets de la décohérence. Contrairement à une transmission classique, un signal quantique ne peut être amplifié par des répéteurs (i.e. des stations relais distribuées tout au long du canal). On ne peut pas simplement analyser (i.e. mesurer) l'état d'un photon pour le retransmettre. Cependant, on peut utiliser des répéteurs logiques qui, avec l'aide de codes correcteurs quantiques, permettent en principe la DQC sur des distances arbitraires. Avant de décrire le fonctionnement d'un répéteur quantique, nous expliquerons le protocole d'échange quantique de clés et prou-

verons sa sécurité.

4.1 La distribution quantique de clés

Il existe différents protocoles de DQC [Ben92, Eke91]. Nous présentons ici le protocole original connu sous le nom de BB84 [BB84]. Le protocole permettra à Alice et Bob de partager une série de bits aléatoires. Ils disposent d'un canal quantique unidirectionnel non sécuritaire (i.e. un adversaire, qu'on nommera Ève, peut écouter et modifier les messages transmis) d'Alice vers Bob et d'un canal classique bidirectionnel authentifié (i.e. Ève peut écouter les messages, mais ne peut pas les modifier[4]).

Le protocole utilise des photons[5] comme support quantique. Ces derniers seront transmis sur le canal quantique (par exemple, sur une fibre optique). L'état du photon qui nous intéresse est sa polarisation, i.e. l'angle $0° \leq \theta < 180°$ du plan, par rapport à l'horizontale, dans lequel le photon oscille en se propageant vers l'avant[6]. Nous utiliserons des photons polarisés à 0°, à 45°, à 90° et à 135° (représentés par $\leftrightarrow$, $\nearrow$, $\updownarrow$ et $\nwarrow$, respectivement). La polarisation d'un photon est un état quantique, un photon peut donc être en superposition de plusieurs polarisations. On peut utiliser le langage de l'informatique quantique pour représenter les différentes polarisations. Les notations suivantes sont équivalentes :

$$\begin{aligned} |\leftrightarrow\rangle &\equiv |0\rangle \\ |\updownarrow\rangle &\equiv |1\rangle \\ |\nearrow\rangle &\equiv \tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle \\ |\nwarrow\rangle &\equiv \tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle \end{aligned}$$

[4] Si Alice et Bob partagent au départ une courte clé secrète, ils peuvent construire un canal authentifié sécuritaire [WC81, BBB+92]. La DQC permettra alors de renouveler cette clé d'authentification en plus de créer la clé de communication.

[5] En principe, la DQC peut être implantée avec n'importe quel type de support quantique, mais puisque la plupart des prototypes utilisent des photons, nous décrirons le protocole en utilisant ces derniers.

[6] Ceci est une simplification de la situation physique réelle. Il existe d'autres sortes de polarisation, par exemple la polarisation circulaire.

	résultats de mesure			
	$\mathcal{M}_+$		$\mathcal{M}_+$	
	$\leftrightarrow$	$\updownarrow$	$\nearrow$	$\nwarrow$
$\lvert\leftrightarrow\rangle$	1	0	1/2	1/2
$\lvert\updownarrow\rangle$	0	1	1/2	1/2
$\lvert\nearrow\rangle$	1/2	1/2	1	0
$\lvert\nwarrow\rangle$	1/2	1/2	0	1

FIG. 4.1: Probabilités des résultats des deux types de mesures

on a donc les relations suivantes :

$$\langle\leftrightarrow|\updownarrow\rangle = \langle\nearrow|\nwarrow\rangle = 0 \quad \text{et}$$
$$\langle\leftrightarrow|\nearrow\rangle = \langle\leftrightarrow|\nwarrow\rangle = \langle\updownarrow|\nearrow\rangle = \langle\updownarrow|\nwarrow\rangle = 1/2.$$

Les états $\{|\leftrightarrow\rangle,|\updownarrow\rangle\}$ forment la base standard ou rectilinéaire (notée par $+$) et $\{|\nearrow\rangle,|\nwarrow\rangle\}$, la base diagonale (notée par $\times$).

Nous utiliserons deux appareils de mesures dans le protocole de DQC. Le premier, noté $\mathcal{M}_+$, permettra de mesurer le photon dans la base standard, le second, noté $\mathcal{M}_\times$, dans la base diagonale. La table 4.1 donne les différentes probabilités des résultats des deux types de mesures.

Par exemple, la mesure $\mathcal{M}_+$ d'un photon dans l'état $|\nearrow\rangle$ retourne $\leftrightarrow$ ou $\updownarrow$ avec probabilité 1/2. Par contre, la mesure $\mathcal{M}_\times$ du même photon retournerait $\nearrow$ avec certitude. Rappelons que la polarisation du photon change après la mesure : le photon est repolarisé pour être cohérent avec le résultat de la mesure.

La sécurité du protocole est basée sur le fait que si Ève tente d'obtenir de l'information en mesurant les états quantiques transmis entre Alice et Bob, elle risque de perturber les photons, ce qui sera remarqué par Alice et Bob. Le protocole de DQC BB84 comprend trois phases : 1) l'échange quantique, 2) l'estimation d'erreurs, et 3) la correction d'erreurs et l'amplification de la confidentialité. L'échange quantique est l'étape où les photons sont transmis d'Alice vers Bob et où une première chaîne de bits

est créée. Dans un monde sans espions et sans erreur de transmission, cette chaîne serait secrète et identique pour Alice et Bob. Cependant, puisque le monde n'est pas parfait, ils devront sacrifier la moitié des bits de cette chaîne pour estimer le taux d'erreurs du canal. S'il y a trop d'erreurs, ils arrêtent le protocole. Pour terminer, ils utiliseront un code correcteur d'erreurs pour extraire une clé identique et secrète avec grande probabilité de succès. Nous utiliserons[7] pour cette dernière phase deux codes linéaires C_1 et C_2, avec paramètre $[n, k_1]$ et $[n, k_2]$ respectivement, tels que $C_2 \subset C_1$ et que C_1 et $C_2^{\perp}$ peuvent corriger t erreurs (bref, deux codes qui pourraient former un code CSS). Nous verrons dans la section 4.2.1 comment choisir les codes C_1 et C_2. On utilise le théorème 1.26 en supposant que la matrice de parité P_2 du code C_2 est formée de P_1 (la matrice de parité du code C_1) et de K (une matrice de dimension $(k_1 - k_2) \times n$), i.e. $P_2 = \begin{pmatrix} P_1 \\ K \end{pmatrix}$. La clé finale sera de longueur $k = k_1 - k_2$.

Avant de présenter le protocole, introduisons un peu de notation. L'expression $a \in_R A$ signifie que l'élément a est choisi aléatoirement et uniformément dans l'ensemble A. Semblablement, $A \subset_R B$ signifie que le sous-ensemble A est choisi aléatoirement et uniformément parmi tous les sous-ensembles de B (si la cardinalité de A est fixée, le choix se fait parmi tous les sous-ensembles de même cardinalité). Soit un ensemble fini $A \subset \mathbb{N}$, A_i est le i^{e} élément de l'ensemble lorsqu'il est trié (pour $1 \leq i \leq |A|$). Soit $x = x_1 \dots x_n$ une chaîne de bits et $A \subset \mathbb{N}$ un ensemble fini tel que $A_{|A|} < n$, $x^{(A)}$ représente la sous-chaîne de x indexée par les éléments de A, i.e. $x^{(A)} = x_{\mathrm{A}_1} \dots x_{\mathrm{A}_{|A|}}$. Pour $|\psi\rangle \in \mathcal{H}$ et $h \in \{+, \times\}$, la fonction $\mathcal{M}_h^{(0,1)}(|\psi\rangle)$ retourne 0 si la mesure de $|\psi\rangle$ dans la base h retourne $\leftrightarrow$ ou $\swarrow\!\!\nearrow$, et retourne 1 si la mesure donne $\updownarrow$ ou $\nwarrow\!\!\searrow$.

La figure 4.2 décrit le protocole BB84. Au tout début, Alice et Bob s'entendent sur certains paramètres : $0 \leq \delta \leq 1$ désignera une proportion de photons supplémentaires à envoyer pour que le protocole fonctionne bien et $0 \leq p \leq 1$ est le taux d'erreurs acceptable pour que $pn \leq t$ avec grande probabilité. Alice transmettra $N = \lceil (4 + \delta)n \rceil$ photons à Bob. Les photons transmis par Alice à l'étape 1 sont aléatoirement

[7] Notre phase de correction d'erreurs et d'amplification de confidentialité est présentée de façon légèrement différente de ce qu'on voit d'habitude, mais le protocole est équivalent. Ceci facilitera la preuve de sécurité.

Paramètres : $0 \leq \delta \leq 1$, $0 \leq p \leq 1$, $N = \lceil (4+\delta)n \rceil$. C_1 un code $[n, k_1]$ et C_2 un code $[n, k_2]$ tels que C_1 et $C_2^{\perp}$ corrigent t erreurs et $C_2 \subset C_1$. La matrice de parité P_2 du code C_2 est sous forme $P_2 = \left(\frac{P_1}{K}\right)$ où P_1 est la matrice de parité de C_1.

Échange quantique

1) Alice choisit deux chaînes aléatoires : $b \in_R \{0,1\}^N$ et $h \in_R \{+,\times\}^N$ représentant les bits à transmettre et les bases correspondantes, respectivement. Alice prépare N photons $|\psi_i\rangle$ en encodant, pour le i^{e} photon, le bit b_i dans la base h_i, i.e. elle prépare $|\leftrightarrow\rangle$ si $b_i = 0$ et $h_i = +$, $|\updownarrow\rangle$ si $b_i = 1$ et $h_i = +$, $|\swarrow\!\!\nearrow\rangle$ si $b_i = 0$ et $h_i = \times$, et $|\nwarrow\!\!\searrow\rangle$ si $b_i = 1$ et $h_i = \times$. Elle transmet ensuite les photons à Bob sur le canal quantique $\mathcal{X}$.

2) Bob choisit une chaîne aléatoire $\hat{h} \in_R \{0,1\}^N$ et mesure chaque photon $\rho_i = \mathcal{X}\big(|\psi_i\rangle\big)$ reçu dans la base dictée par $\hat{h}_i$, i.e. pour $1 \leq i \leq N$, $\hat{b}_i = \mathcal{M}_{\hat{h}_i}^{(0,1)}(\rho_i)$. Bob annonce ensuite $\hat{h}$ à Alice.

3) Si $\Delta(h, \hat{h}) > N - 2n$, ils arrêtent le protocole. Sinon, Alice choisit aléatoirement $2n$ positions $J = \{j_1, \ldots, j_{2n}\} \subset_R \{1, \ldots, N\}$ telles que $h_{j_i} = \hat{h}_{j_i}$ pour $1 \leq i \leq 2n$. Alice choisit aléatoirement n positions $\tau = \{\tau_1, \ldots, \tau_n\} \subset_R J$ pour tester les erreurs et annonce τ à Bob.

Estimation des erreurs

4) Bob annonce à Alice ses mesures aux positions tests $\hat{b}^{(\tau)} = (\hat{b}_{\tau_1}, \ldots, \hat{b}_{\tau_n})$.

5) Alice vérifie que $\Delta(b^{(\tau)}, \hat{b}^{(\tau)}) \leq pn$, sinon ils arrêtent le protocole.

Correction d'erreurs et amplification de confidentialité

6) Alice choisit un mot de code aléatoire $v \in_R C_1$ et annonce $v + b^{(\bar{\tau})}$, où $b^{(\bar{\tau})}$ sont les n bits restants de b qui n'ont pas servi dans le test d'erreurs.

7) Bob calcule $\hat{b}^{(\bar{\tau})} - (v + b^{(\bar{\tau})}) = (b^{(\bar{\tau})} + e) - (v + b^{(\bar{\tau})}) = v + e$, où e est un vecteur d'erreurs. Il corrige ensuite cette chaîne à un vecteur $\hat{v} \in C_1$.

8) La clé dérivée d'Alice est $\kappa = vK^{\mathsf{T}}$ et celle de Bob est $\hat{\kappa} = \hat{v}K^{\mathsf{T}}$. Si $w(e) < \lfloor (d_1 - 1)/2 \rfloor$, alors $\kappa = \hat{\kappa}$.

FIG. 4.2: Le protocole BB84 de DQC

choisis dans l'ensemble $\{|\updownarrow\rangle, |\leftrightarrow\rangle, |\swarrow\nearrow\rangle, |\nwarrow\searrow\rangle\}$. Si on suppose qu'il n'y a pas d'erreur de transmission et que, pour le i^{e} photon, le choix de la base d'envoi d'Alice coïncide avec le choix de la base de mesure de Bob (i.e. $h_i = \hat{h}_i$), alors le résultat de la mesure sera identique au bit envoyé (i.e. $\hat{b}_i = b_i$). Par contre, si les bases sont différentes, les bits seront égaux avec probabilité $1/2$ (comme le montre la figure 4.1), on laissera donc tomber ces positions à l'étape 3. Les bases coïncideront sur un nombre espéré de $N/2$ positions et la probabilité qu'il ne reste pas $2n$ positions pour le choix de J sera exponentiellement faible[8]. Parmi les positions où les bases sont identiques, Alice choisira n positions tests τ et les positions complémentaires $\bar{\tau} = J - \tau$ permettront de dériver la clé. Par la suite, ils procéderont à l'estimation des erreurs causées par le canal $\mathcal{X}$ (c'est pourquoi Bob reçoit une série d'états mélangés ρ_i). Ils comparent la chaîne d'Alice b et celle des mesures de Bob $\hat{b}$ aux positions tests. Si trop (plus de pn) d'erreurs sont présentes, ils arrêtent le protocole en concluant que les erreurs sont causées par un espion (nous discuterons de l'effet des observations d'un espion dans la section 4.2). La dernière phase du protocole permet à Alice et Bob de transformer leurs chaînes presque identiques de longueur n (sur lesquelles Ève connaît peut-être quelques bits) en deux chaînes de longueur $k = k_1 - k_2$ qui seront identiques avec très grande probabilité et sur lesquelles Ève n'aura aucune information. Ce sera le coset de C_2 dans C_1 du mot de code v d'Alice qui permettra de dériver la clé finale. Si on note par e le vecteur d'erreurs entre la chaîne d'Alice et celle de Bob (i.e. $e_i = 1$ si les bits sont différents à la position i, et 0 sinon), Bob obtiendra, en corrigeant (avec le code C_1) le vecteur $v + e$ (notons que $v - e = v + e$ dans $\mathbb{Z}_2^n$), un vecteur $\hat{v} \in C_1$ qui sera égal à v si $w(e) < t$, ce qui survient avec grande probabilité pour un choix de p opportun. Pour terminer, en supposant que $v = \hat{v}$, ils utiliseront la clé $\kappa = vK^{\mathsf{T}}$ de taille k. Nous verrons dans la prochaine section que si le test d'estimation d'erreurs de l'étape 5 passe, alors l'information qu'Ève pourrait avoir sur la clé est exponentiellement faible.

Exemple 4.1 La figure 4.3 donne un exemple d'exécution des deux premières phases du protocole. Le premier photon envoyé ($\swarrow\nearrow$) est l'encodage de 0 dans la base diagonale, mais puisque Bob le mesure dans la base standard, le photon sera jeté à l'étape 3. En

[8] Selon la loi des grands nombre de Bernshtein (voir théorème A.11), cette probabilité est inférieure à $e^{-(n\delta^2)/4(4+\delta)}$.

Paramètres : $n = 4$, $\delta = 1/4$, $p = 1/4$, $N = 17$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1)																	
b :	0	0	1	0	1	1	1	0	0	1	0	1	1	0	1	0	0
h :	×	+	+	×	+	×	×	+	×	×	+	×	+	×	+	+	×
ψ_i :	↙↗	↔	↕	↙↗	↕	↖↘	↖↘	↔	↙↗	↖↘	↔	↖↘	↕	↙↗	↕	↔	↙↗
2)																	
ρ_i :	↙↗	↔	↔	↙↗	↕	↖↘	↖↘	↔	↙↗	↕	↔	↙↗	↕	↙↗	↖↘	↔	↔
$\hat{h}$:	+	×	+	×	×	×	+	+	+	×	+	+	×	×	+	×	+
$\mathcal{M}_{\hat{h}_i}$:	↕	↖↘	↔	↙↗	↙↗	↖↘	↕	↔	↔	↖↘	↔	↔	↖↘	↙↗	↔	↖↘	↔
$\hat{b}$:	1	1	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0
3)																	
J :			√	√		√		√		√	√			√	√		
τ :				√		√					√				√		
4)																	
$\hat{b}^{(\tau)}$:				0		1					0				0		
5)																	
err :															√		
$b^{(\bar{\tau})}$:			1					0		1				0			
$\hat{b}^{(\bar{\tau})}$:			0					0		1				0			
err :			√														

FIG. 4.3: Exemple d'exécution du protocole de DQC

effet, les bases étant différentes, le résultat de la mesure diffère du bit original avec probabilité 1/2, comme c'est le cas ici. Le quatrième photon est transmis et mesuré dans la même base, le bit d'Alice et celui de Bob concordent donc. Même si Alice et Bob utilisent la même base pour le photon 3, une erreur de transmission ayant affectée le photon introduit une erreur dans la chaîne de Bob. Par contre, le photon 10 qui subit le même sort n'introduira pas d'erreur car la chance a voulu que le résultat de la mesure concorde au bit d'Alice. On remarque qu'il y a eu cinq erreurs de transmissions (sur les photons 3, 10, 12, 15 et 17), ce qui est légèrement plus élevé que le taux d'erreurs espéré de 1/4. Ceci est soit dû à la malchance, soit dû à un espion. À l'étape 3, Alice n'a pas le choix pour sélectionner son ensemble J, car exactement $8 = 2n$ positions ont utilisé les mêmes bases. Cependant, le choix de τ est aléatoire. Puisqu'il n'y a qu'une erreur sur les quatre bits tests et que cette proportion n'est pas supérieure à $1 = pn$, alors le protocole continue. À la fin, Alice et Bob partagent une clé de quatre bits presque identique (il y a une erreur sur le premier bit, i.e. sur le 3$^{\text{e}}$ photon). Ils pourront ensuite distiller une clé secrète plus petite en effectuant la suite du protocole. À ce point, la chaîne commune n'est pas forcément secrète. Par exemple, supposons qu'Ève décide de mesurer le huitième photon dans la base standard lorsqu'il passe sur le canal. Puisque sa mesure se fait dans la même base que celle d'Alice et Bob, son action ne modifiera pas le photon et elle obtiendra le même résultat que Bob. À la fin de protocole, puisque ce photon aura permis de dériver un bit de la chaîne finale, Ève connaîtra un bit de celle-ci. ◇

Exemple 4.2 Pour illustrer la phase de correction d'erreurs et d'amplification de la confidentialité, supposons qu'on utilise le code de Hamming et son dual comme codes sous-tendants, i.e. $C_1 = H$ et $C_2 = H^{\perp}$ (comme dans le code de Steane). Donc, ici, $n = 7$ et $k = 2^{k_1 - k_2} = 2^{4-3} = 1$, le protocole permettra donc à Alice et Bob de s'entendre sur une clé de un bit. De plus, supposons qu'avant l'étape 6, les chaînes d'Alice et Bob sont, respectivement, $b^{(\bar{\tau})} = 0110110$ et $\hat{b}^{(\bar{\tau})} = 0110010$ (une erreur étant présente sur le 5$^{\text{e}}$ bit de Bob). À l'étape 6, Alice choisit un mot de code de Hamming aléatoire, disons $v = 1010101$, et transmet $v + b^{(\bar{\tau})} = 1100011$ à Bob. Ce dernier calcule $\hat{b}^{(\bar{\tau})} - (v + b^{(\bar{\tau})}) = 0110010 - 1100011 = 1010001$, qui est bien égal à $v + 0000100$. En

appliquant la matrice de parité P_1 à cette chaîne, il obtient $1010001P_1^{\mathsf{T}} = 101$. Ceci lui indique qu'il y a une erreur sur le 5^{e} bit et il corrige donc ce dernier pour obtenir $\hat{v} = v$. Pour obtenir la clé finale, ils calculent tous deux la valeur $\kappa = vK^{\mathsf{T}}$ où K est une matrice de dimension 1×7 avec des 1 à chaque position (voir la matrice de l'équation 1.6). Ici, $\kappa = 0$. ◇

4.2 La sécurité de la DQC

Essayons de voir comment Ève pourrait obtenir de l'information sur la clé secrète en observant ou en participant au protocole. On suppose qu'elle voit tout se qui ce passe sur le canal classique et qu'elle peut agir sur le canal de transmission quantique, i.e. elle peut mesurer les qubits qui passent (dans n'importe quelle base) et les modifier (i.e. appliquer une transformation, ou même les effacer). Elle peut même intriquer les photons passant avec des états qu'elle mesurera après le protocole. Bref, elle peut faire subir n'importe quel algorithme quantique au registre des photons[9]. En d'autres mots, pour modéliser l'attaque d'Ève, on suppose qu'Alice prépare tous ses photons et les transmet en bloc, qu'Ève les intercepte, leur applique un algorithme quantique et renvoie le bloc possiblement modifié à Bob.

On peut comprendre intuitivement l'argument de sécurité de la DQC. Plus Ève agit sur les photons (en les mesurant ou les transformant), plus les photons seront modifiés. Donc, la chaîne b d'Alice sera très différente de la chaîne $\hat{b}$ de Bob, ce qui sera détecté à l'étape de vérification avec grande probabilité. Par exemple, supposons qu'Ève se limite à mesurer les photons transmis soit avec $\mathcal{M}_+$, soit avec $\mathcal{M}_\times$ (comme Bob). On ne s'intéresse qu'au cas où Alice et Bob utilisent la même base pour transmettre et pour mesurer (sinon, le photon sera inutilisé). Supposons donc qu'ils utilisent la base standard pour un des photons. Si Ève le mesure aussi dans la base standard, elle ne modifiera pas le photon et elle obtiendra le même résultat de mesure que Bob (elle connaîtra donc un bit de la chaîne de ce dernier). Cependant, si elle mesure dans

[9] On permet à Ève de faire des mesures et des transformations collectives, i.e. sur plusieurs photons, même si en pratique cela est difficile. Cette attaque est plus puissante que si on lui permettait seulement d'agir sur les photons un par un, lorsqu'ils passent.

la base diagonale, le photon sera repolarisé pour être cohérent avec le résultat de la mesure. Bob, à son tour, mesurera le photon modifié et avec probabilité 1/2 (voir la figure 4.1) le résultat de la mesure sera différent du photon transmis par Alice. Puisqu'Alice et Bob utiliseront le photon pour s'entendre sur un bit (car ils utilisent la même base), Ève introduit une erreur dans la chaîne avec probabilité 1/2. Donc, chaque fois qu'Ève mesure un photon qui sera utilisé dans la clé, elle choisit la mauvaise base avec probabilité 1/2 et alors la mesure de Bob est erronée avec probabilité 1/2, elle introduit donc une erreur avec probabilité 1/4.

À haut niveau, on comprend que plus Ève observe le canal quantique, plus elle introduira des erreurs dans la chaîne de Bob. Mais comment peut-on s'assurer que le protocole est sécuritaire ? Ève ne pourrait-elle pas utiliser des bases de mesures qui sont moins dangereuses ? Ne pourrait-elle pas se contenter d'observer le canal que très rarement dans le but d'obtenir seulement un peu d'information sur la clé. Encore pire, qu'arrive-t-il si Ève est assez puissante pour remplacer le canal bruyant par un canal parfait ? Elle pourrait alors se permettre d'observer suffisamment de qubit pour simuler le taux d'erreurs toléré par Alice et Bob.

Malgré toutes ces attaques possibles, Ève ne peut rien apprendre sur la clé secrète d'Alice et Bob. En effet, plusieurs preuves de sécurité ont été présentées pour ce protocole, mais chacune d'entre elles a des inconvénients. Les preuves de Mayers [May98] et de Biham, Boyer, Boykin, Mor et Roychowdhury [BBB+99] sont très difficiles à comprendre, tandis que celle de Lo et Chau [LC99], bien que simple, nécessite l'utilisation d'ordinateurs quantiques pour exécuter le protocole de DQC. Nous présentons une preuve simple de sécurité qui utilise les codes CSS[10] (qui est une utilisation inusitée des codes correcteurs quantiques à la cryptographie).

[10]La preuve n'est pas parfaite elle non plus, car les prototypes expérimentaux n'utilisent pas des appareils parfaits. Nous discuterons de ces points faibles dans la section 4.2.1.

4.2.1 Preuve de sécurité de Shor et Preskill

La preuve de sécurité du protocole BB84 que nous présentons maintenant est due à Shor et Preskill [SP00]. Leur approche consiste à réduire un protocole de DQC, développé et prouvé sécuritaire par Lo et Chau [LC99] basé sur la purification de l'intrication [BDSW97], au protocole BB84. Nous ne donnerons pas tous les détails techniques et les calculs mathématiques, mais la description devrait être suffisante pour que le lecteur soit convaincu de la sécurité du protocole.

Nous utiliserons dans la preuve des codes équivalents aux codes CSS construits avec les deux mêmes codes linéaires C_1 et C_2 ($[n, k_1]$ et $[n, k_2]$ respectivement). Dans le reste de la section, on note par P_1 la matrice de parité du code C_1, par P_2 celle du code C_2 et par G_2, celle du code $C_2^\perp$ (on se rappelle que la matrice génératrice G_2 du code C_2 est une matrice de parité pour le code $C_2^\perp$, voir théorème 1.22).

Définition 4.3 Soit Q un code CSS construit à l'aide des deux codes C_1 et C_2. Soit $x, z \in \{0,1\}^n$. On construit le code $Q_{(x,z)}$ en associant à chaque $v \in C_1$ un mot de base

$$|\overline{v_{(x,z)}}\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w \in C_2} (-1)^{w \cdot z} |v + w + x\rangle.$$

◇

Remarque 4.4 Le code $Q_{(x,z)}$ est en quelque sorte le code Q (équivalent à $Q_{(0^n,0^n)}$) "translaté" par x et z. On note que si $u \in \text{Coset}(v)$, alors $|\overline{u_{(x,z)}}\rangle = |\overline{v_{(x,z)}}\rangle$. Puisqu'il y a $2^{k_1-k_2} = 2^k$ cosets différents, il y a autant d'états de base et $Q_{(x,z)}$ est donc un espace de dimension 2^k.

Cependant, il n'y a que $2^{n-k_1+k_2}$ codes $Q_{(x,z)}$ différents. En effet si, pour $x, x', z, z' \in \{0,1\}^n$, $xP_1^\mathsf{T} = x'P_1^\mathsf{T}$ et $zG_2^\mathsf{T} = z'G_2^\mathsf{T}$, alors $Q_{(x,z)} = Q_{(x',z')}$ (si, en plus, $xP_2^\mathsf{T} = x'P_2^\mathsf{T}$, alors $|\overline{v_{(x,z)}}\rangle = |\overline{v_{(x',z')}}\rangle$ pour un $v \in C_1$; sinon les états de base sont permutés mais l'espace engendré reste le même).

Un code $Q_{(x,z)}$ a les mêmes propriétés qu'un code CSS Q, i.e. c'est un code $[[n, k_1 - k_2]]$ avec distance minimale non inférieure à celle de C_1 et de $C_2^\perp$. Il est facile de modifier les circuits d'encodage et de correction du code Q pour former ceux du code $Q_{(x,z)}$. ◇

Exemple 4.5 Supposons que Q est le code de Steane (i.e. $C_1 = H$ et $C_2 = H^{\perp}$). Alors $Q = Q_{(0^n,0^n)}$ est engendré par les vecteurs $|\overline{0^7_{(0^7,0^7)}}\rangle$ et $|\overline{1^7_{(0^7,0^7)}}\rangle$ (décrits dans les équations 3.2 et 3.3). On a que $|\overline{0001111_{(0^7,0^7)}}\rangle = |\overline{0^7_{(0^7,0^7)}}\rangle$ et que $|\overline{0101010_{0^7,0^7}}\rangle = |\overline{1^7_{0^7,0^7}}\rangle$, car $0001111 \in \text{Coset}(0000000)$ et $0101010 \in \text{Coset}(1111111)$.

De plus, $|\overline{0^7_{(1^7,0^7)}}\rangle = |\overline{1^7_{(0^7,0^7)}}\rangle$ et $|\overline{1^7_{(1^7,0^7)}}\rangle = |\overline{0^7_{(0^7,0^7)}}\rangle$ (i.e. si $x = 1^7$, les états de base sont interchangés. Cependant, l'espace $Q_{(1^7,0^7)}$ est le même que Q, car $1^7 \in C_1$.

Puisque $0000001P_1^{\mathsf{T}} = 0001110P_1^{\mathsf{T}} = 1110001P_1^{\mathsf{T}} = 111$, alors $Q_{(0000001,0^7)} = Q_{(0001110,0^7)} = Q_{(1110001,0^7)}$. De plus, puisque $0000001P_2^{\mathsf{T}} = 0001110P_2^{\mathsf{T}} = 1111$, alors $|\overline{u_{(0000001,0^7)}}\rangle = |\overline{u_{(0001110,0^7)}}\rangle$, pour tout $u \in H$. Cependant, puisque $1110001P_2^{\mathsf{T}} = 1110 \neq 1111$, les états de bases sont interchangés, i.e. $|\overline{u_{(1110001,0^7)}}\rangle = |\overline{(u \oplus 1^7)_{(0000001,0^7)}}\rangle$. ◇

Nous verrons maintenant un protocole qui permet de faire la DQC basé sur la purification de l'intrication. La purification de l'intrication est un protocole qui permet de transformer deux registres de n qubits intriqués en deux registres de $m < n$ qubits qui sont maximalement intriqués en concentrant l'intrication des n paires initiales. Par exemple on pourrait transformer n paires qui sont près de l'état $|\Phi^+\rangle^{\otimes n}$ en deux registres de $m < n$ qubits dans l'état $|\Phi^+\rangle^m$. L'idée du protocole de Lo et Chau est simple. Si Alice et Bob partagent une série de paires $|\Phi^+\rangle$ (i.e. pour une des paires, le qubit d'Alice et celui de Bob sont conjointement dans l'état $|\Phi^+\rangle$), alors s'ils mesurent chaque qubit dans la base standard, ils obtiendront la même chaîne de bits (comme expliqué dans la remarque 2.13). Le but du protocole de Lo et Chau est donc de permettre à Alice et Bob de partager des paires $|\Phi^+\rangle$ pour en tirer une clé secrète. En quelques mots, Alice préparera les paires $|\Phi^+\rangle$ et enverra la moitié de chaque paire à Bob. Ils vérifieront ensuite qu'il n'y a pas eu trop d'erreurs sur le canal quantique et purifieront les états restants afin d'obtenir un nombre plus petit de paires très près de $|\Phi^+\rangle$. En mesurant ces états, ils partageront une clé secrète.

Shor et Preskill ont modifié le protocole original de Lo et Chau pour effectuer la purification avec un code code $Q_{(x,z)}$[11]. Cette modification permettra de réduire

[11] Bennett, DiVincenzo, Smolin et Wooters [BDSW97] ont montré l'équivalence entre les codes cor-

le protocole de Lo et Chau (qui nécessite l'utilisation d'un ordinateur quantique) au protocole BB84 (qu'on peut mettre en œuvre sans ordinateur quantique).

La figure 4.4 décrit le protocole de Lo et Chau modifié. Pour préparer l'état de l'étape 1, Alice prépare d'abord $2n$ paires $|\Phi^+\rangle$ et applique un Walsh-Hadamard sur le deuxième qubit de la i^{e} paire si $h_i = 1$. Elle transmet ensuite les deuxièmes qubits de chaque paire sur le canal quantique. Bob reçoit ainsi des états mélangés $\rho_{\mathcal{X}_i}$. Ils sacrifieront la moitié de leur qubits pour estimer le taux d'erreurs. Bob défait d'abord les Walsh-Hadamard d'Alice en appliquant aussi un Walsh-Hadamard sur les mêmes qubits. Puisque cette porte quantique est auto-inverse, Alice et Bob partageraient à ce moment $2n$ paires $|\Phi^+\rangle$ si le canal était parfait et s'il n'y avait pas d'espion. Pour estimer le taux d'erreurs, Alice et Bob mesurent dans la base standard chaque qubit aux positions tests (dictées par τ) et comparent les résultats à l'aide du canal classique authentifié. Si trop de mesures diffèrent (plus de pn), alors ils arrêtent le protocole, car il y a trop d'erreurs. Par la suite, il purifieront leurs paires pour former $k = k_1 - k_2$ paires $|\Phi^+\rangle$.

Pour y arriver, Alice mesure d'abord les syndromes s_1 et s_2 pour les erreurs de négation et de phase, respectivement (en appliquant le circuit de calcul de syndromes de Q). Supposons qu'il n'y ait pas eu d'erreurs de phase ou de négation, alors Alice et Bob partageraient à ce moment n paires $|\Phi^+\rangle$, i.e.

$$\rho_{\mathrm{A}_{\bar{\tau}}} \otimes \hat{\rho}_{\mathrm{B}_{\bar{\tau}}} = |\Phi^+\rangle^{\otimes n}.$$

Alors, la mesure des syndromes s_1 et s_2 projetterait les registres sur un état dans l'espace $Q_{(x,z)} \otimes Q_{(x,z)}$ pour un x et un z tels que $s_1 = xP_1^{\mathsf{T}}$ et $s_2 = zG_2^{\mathsf{T}}$. La mesure des syndromes de Bob donnerait $\hat{s}_1 = s_1$ et $\hat{s}_2 = s_2$. À ce point, l'état conjoint des deux registres serait $|\Phi^+\rangle^{\otimes k}$ encodé par le code $Q_{(x,z)}$. En effet, la superposition $|\Phi^+\rangle^{\otimes n} = \sum_{u\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|u\rangle \otimes |u\rangle$ s'écroulerait en une superposition des 2^k chaînes qui correspondent aux syndromes mesurés. Chacune d'elles se décoderait à une chaîne différente de $\mathbb{Z}_2^k$. En décodant leur registre chacun de leur côté, ils obtiendraient alors l'état $|\Phi^+\rangle^{\otimes k}$ qu'ils mesureraient localement pour obtenir une clé commune.

recteurs et certains protocoles de purification.

Paramètres : $0 \leq p \leq 1$, deux codes linéaires C_1 $[n, k_1]$ et C_2 $[n, k_2]$ ($k = k_1 - k_2$).

Échange quantique

1) Alice choisit une chaîne aléatoire $h \in_R \{0,1\}^{2n}$ et prépare l'état intriqué

$$\bigotimes_{i=1}^{2n} (\mathrm{I}_\mathrm{A} \otimes \mathsf{H}_\mathrm{B}^{h_i}) |\Phi^+\rangle = \bigotimes_{i=1}^{2n} \rho_{\mathrm{A}_i} \otimes \rho_{\mathrm{B}_i} = \rho_\mathrm{A} \otimes \rho_\mathrm{B}$$

Alice envoie le deuxième qubit de chaque paire sur le canal quantique, i.e. elle transmet ρ_B.

2) Bob confirme à Alice la réception des $2n$ qubits $\rho_{\mathcal{X}_i} = \mathcal{X}(\rho_{\mathrm{B}_i})$.

Estimation des erreurs

3) Alice choisit aléatoirement n positions $\tau = \{\tau_1, \ldots, \tau_n\} \subset_R \{1, \ldots, 2n\}$ pour tester les erreurs et annonce τ et h à Bob.

4) Bob applique H à son registre selon h, i.e. il calcule $\hat{\rho}_\mathrm{B} = \bigotimes_{i=1}^{2n} \mathsf{H}_\mathrm{B}^{h_i} \rho_{\mathcal{X}_i}$ et mesure ensuite les positions tests dans la base standard, i.e. pour $1 \leq i \leq n$, $\hat{b}_i = \mathcal{M}_+^{(0,1)}(\hat{\rho}_{\mathrm{B}_{\tau_i}})$. Il envoie ensuite $\hat{b}$ à Alice.

5) Alice mesure aussi ses qubits aux positions tests, i.e. pour $1 \leq i \leq n$, $b_i = \mathcal{M}_+^{(0,1)}(\rho_{\mathrm{A}_{\tau_i}})$. Si $\Delta(b, \hat{b}) > pn$, ils arrêtent le protocole.

Purification

6) Alice mesure les syndromes d'erreurs de négation s_1 et de phase s_2 pour son registre et les transmet à Bob.

7) Bob mesure, pour son registre, les syndromes de négation $\hat{s}_1$ et de phase $\hat{s}_2$ et calcule les syndromes d'erreurs $e_1 = s_1 - \hat{s}_1$ et $e_2 = s_2 - \hat{s}_2$. Il applique ensuite le circuit de correction de Q en utilisant les syndromes e_1 et e_2, ce qui donnera le même état qu'Alice.

8) Ils décodent chacun leur registre avec le code $Q_{(x,z)}$ pour obtenir conjointement $|\Phi^+\rangle^{\otimes k} = \phi_\mathrm{A} \otimes \phi_\mathrm{B}$. Ils mesurent chaque qubit dans la base standard pour obtenir la clé, i.e. $\kappa = \mathcal{M}_+^{(0,1)}(\phi_\mathrm{A}) = \mathcal{M}_+^{(0,1)}(\phi_\mathrm{B})$.

FIG. 4.4: Le protocole de Lo et Chau modifié de DQC

Par contre, supposons qu'il y ait au plus t erreurs de négation et t erreurs de phase entre les états $\rho_{A_{\bar{\tau}}}$ et $\hat{\rho}_{B_{\bar{\tau}}}$. Si on applique une erreur de négation, de phase, ou les deux à un qubit d'une paire $|\Phi^+\rangle$, celle-ci devient

$$\begin{aligned} |\Phi^+\rangle &\xleftrightarrow{\sigma_1} |\Psi^+\rangle \\ |\Phi^+\rangle &\xleftrightarrow{\sigma_2} |\Phi^-\rangle \\ |\Phi^+\rangle &\xleftrightarrow{\sigma_3} |\Psi^-\rangle. \end{aligned}$$

Dans cette situation, on aura que $s_1 \neq \hat{s}_1$ ou $s_2 \neq \hat{s}_2$ (i.e. au moins un des deux sera différent). À partir des nouveaux syndromes $e_1 = s_1 - \hat{s}_1$ et $e_2 = s_2 - \hat{s}_2$, Bob peut calculer les positions des erreurs de phase et de négation entre son registre et celui d'Alice et les corriger (en utilisant le circuit de correction de Q). L'état de Bob, qui était dans l'espace $Q_{(x',z')}$ (tels que $\hat{s}_1 = x'P_1^{\mathsf{T}}$ et $\hat{s}_2 = zG_2^{\mathsf{T}}$) sera alors, après cette correction, transformé à un état de $Q_{(x,z)}$ qui, conjointement avec l'état d'Alice, est l'encodage de $|\Phi^+\rangle^{\otimes k}$ par $Q_{(x,z)}$.

Donc, si moins de t erreurs de phase et de négation surviennent dans la transmission ou par l'espionnage d'Ève, les étapes 6 à 8 permettront d'obtenir un état $|\Phi^+\rangle^{\otimes k}$ qu'ils utiliseront pour partager une clé.

Exemple 4.6 Supposons que durant la transmission, une erreur de négation ait affecté le troisième qubit et qu'on utilise le code de Steane Q (i.e. $C_1 = H$ et $C_2 = H^{\perp}$). L'état $\rho_{A_{\bar{\tau}}} \otimes \hat{\rho}_{B_{\bar{\tau}}}$ serait alors

$$\sum_{u \in \{0,1\}^7} \frac{1}{\sqrt{2^7}} |u_1, u_2, u_3, u_4, u_5, u_6, u_7\rangle \otimes |u_1, u_2, (u_3 \oplus 1), u_4, u_5, u_6, u_7\rangle.$$

Les syndromes d'erreurs de négation seraient alors (voir l'équation 1.5 pour la matrice de parité du code de Hamming)

$$s_1 = (u_4 \oplus u_5 \oplus u_6 \oplus u_7, u_2 \oplus u_3 \oplus u_6 \oplus u_7, u_1 \oplus u_3 \oplus u_5 \oplus u_7)$$

et

$$\hat{s}_1 = (u_4 \oplus u_5 \oplus u_6 \oplus u_7, u_2 \oplus (u_3 \oplus 1) \oplus u_6 \oplus u_7, u_1 \oplus (u_3 \oplus 1) \oplus u_5 \oplus u_7).$$

Donc, $e_1 = s_1 - \hat{s}_1 = (0, 1, 1)$ indiquerait bel et bien qu'une erreur de négation se trouve sur le troisième qubit (car $3 = 011_2$). En appliquant le circuit de correction de Q, on corrigerait cette erreur. ◇

La preuve de sécurité de ce protocole est légèrement complexe et ressemble essentiellement à la preuve de [LC99] pour leur protocole Lo et Chau original. Pour donner l'idée générale, on remarque qu'on peut calculer classiquement la probabilité que l'écart entre le nombre d'erreurs dans les bits tests et ceux qui permettront de dériver la clé soit significatif. En effet, puisqu'Ève ne sait pas si un qubit est un qubit test ou non, elle ne peut pas le traiter différemment. La probabilité qu'il y ait plus de θn erreurs sur les positions de la clé et moins de $(\theta - \varepsilon)n$ erreurs sur les positions tests est asymptotiquement moins que $e^{-\frac{\varepsilon^2 n}{4(\theta-\theta^2)}}$ (par la loi des grands nombres de Bernshtein, voir théorème A.11). Si on note par ρ l'état partagé par Alice et Bob avant la mesure de l'étape 8 (i.e. $\rho \approx \phi_{\mathrm{A}} \otimes \phi_{\mathrm{B}}$), alors on peut montrer que

$$\langle (\Phi^+)^{\otimes n} | \rho | (\Phi^+)^{\otimes n} \rangle > 1 - \frac{1}{2^{\mathcal{O}(n)}}. \tag{4.1}$$

[SP00] utilise alors directement un théorème de [LC99] pour conclure que l'information mutuelle entre Ève et la clé dérivée de la mesure de ρ est exponentiellement faible.

En résumé, le protocole permet à Alice et Bob de s'entendre sur une clé de telle façon que la probabilité est exponentiellement faible qu'Ève ait une quantité non-négligeable d'information sur celle-ci.

On remarque qu'on peut changer le protocole de Lo et Chau légèrement sans affecter le résultat et la sécurité. Il n'y a aucune différence si Alice mesure ses qubits tests avant de les envoyer à Bob, i.e. si elle choisit τ (étape 3) et si elle fait ses mesures (étape 5) avant de transmettre les qubits à Bob (étape 1). Elle obtiendra une chaîne de bits b aléatoires. Au lieu de préparer réellement n paires $|\Phi^+\rangle$ tests, elle pourrait elle-même choisir la chaîne b aléatoirement et préparer l'état $|b_i\rangle \otimes |b_i\rangle$ pour la i^{e} paire test. De plus, elle peut aussi mesurer ses syndromes s_1 et s_2 avant de faire la transmission quantique. Ceci est équivalent à encoder k paires $|\Phi^+\rangle$ avec un code $Q_{(x,z)}$ choisit aléatoirement (parmi les x et z tels que $s_1 = xP_1^{\mathsf{T}}$ et $s_2 = zG_2^{\mathsf{T}}$). Finalement, Alice peut aussi mesurer les k paires $|\Phi^+\rangle$ avant de les encoder par $Q_{(x,z)}$. Elle obtiendra ainsi une clé aléatoire de k bits. Comme pour les bits tests, le résultat est le même si elle choisit d'abord la clé κ aléatoirement et qu'elle l'encode avec $Q_{(x,z)}$. On obtient donc ainsi le protocole CSS décrit dans la figure 4.5.

Paramètres : $0 \leq p \leq 1$, deux codes linéaires $C_1\ [n, k_1]$ et $C_2\ [n, k_2]$ ($k = k_1 - k_2$).

Échange quantique

1) Alice choisit aléatoirement $b \in_R \{0,1\}^n$, $h \in_R \{0,1\}^{2n}$, $\tau = \{\tau_1, \ldots, \tau_n\} \subset_R \{1, \ldots, 2n\}$, $\kappa \in_R \{0,1\}^k$, $x \in_R \{0,1\}^n$ et $z \in_R \{0,1\}^n$. Elle prépare ensuite l'état $|\overline{\kappa}\rangle = \mathcal{C}_{Q_{(x,z)}}(|\kappa\rangle)$, l'encodage de κ par le code $Q_{(x,z)}$. Elle réindexe les n qubits de $|\overline{\kappa}\rangle$ avec les éléments de $\bar{\tau} = \{1, \ldots, 2n\} - \tau$, i.e. $|\overline{\kappa}\rangle = \rho_{\bar{\tau}_1} \otimes \ldots \otimes \rho_{\bar{\tau}_n}$. Elle prépare ensuite n qubits, indexés avec les éléments de τ, selon la chaîne b, i.e. $\rho_{\tau_i} = |b_i\rangle$, pour $1 \leq i \leq n$. Elle applique un Walsh-Hadamard aux positions dictées par h, i.e.

$$\rho' = \bigotimes_{i=1}^{2n} \mathsf{H}^{h_i} \rho_i$$

Finalement, elle envoie les $2n$ qubits ρ'_i à Bob.

2) Bob confirme à Alice la réception des $2n$ qubits $\rho_{\mathcal{X}_i} = \mathcal{X}(\rho'_i)$.

3) Alice annonce b, h, x et z à Bob.

4) Bob applique un Walsh-Hadamard aux positions dictées par h pour défaire ceux d'Alice, i.e. il calcule

$$\hat{\rho} = \bigotimes_{i=1}^{2n} \mathsf{H}^{h_i} \rho_i$$

Il mesure ensuite les qubits aux positions tests, il obtient donc $\hat{b}_i = \mathcal{M}_+^{(0,1)}(\hat{\rho}_{\tau_i})$. Il vérifie que $\Delta(b, \hat{b}) < pn$, sinon il arrête le protocole.

Correction d'erreurs

5) Bob utilise le code $Q_{(x,z)}$ pour décoder l'état $\rho^{(\bar{\tau})}$ (pour obtenir $|\kappa\rangle$) et il mesure les k qubits dans la base standard pour retrouver la clé ; i.e.

$$\kappa = \mathcal{M}_+^{(0,1)}\Big(\mathcal{D}_{Q_{(x,z)}}(\rho_{\bar{\tau}_1} \otimes \ldots \otimes \rho_{\bar{\tau}_n})\Big).$$

FIG. 4.5: Le protocole CSS de DQC

La sécurité tient toujours. Intuitivement, on voit que si le taux d'erreur est assez faible pour que le code puisse corriger la clé, alors le théorème 2.10 nous dit qu'Ève ne peut rien apprendre sur la clé.

Voyons maintenant comment on peut réduire le protocole CSS au protocole BB84. Pour cela, il suffit de remarquer que les erreurs de phase n'influencent pas le décodage de Bob. En effet, que l'état décodé soit $|\kappa\rangle$ ou $-|\kappa\rangle$, le résultat de la mesure sera le même. Les codes CSS nous sont ici d'une grande utilité, car l'étape de correction des erreurs de phase et des erreurs de négations sont indépendantes. Bob peut simplement ne pas corriger les erreurs de phase et le protocole fonctionnera tout de même. On peut alors supposer qu'Alice n'envoie pas z. Ceci n'aide aucunement Ève, car si elle pouvait attaquer ce nouveau protocole, elle pourrait attaquer l'ancien en igorant le z transmis.

Si Bob ne connaît pas z, sont état à l'étape 5 est le mélange $\rho^{(\bar{r})} = \left\{\left(\frac{1}{2^n}, |\overline{v_z}\rangle\right)\right\}$ où v_z est le vecteur de $Q_{(x,z)}$ qui encode la clé (pour les différents z). L'équation 4.2 développe cet état. L'équation (1) est simplement le calcul de la matrice de densité $\sum_z \frac{1}{2^n}|\overline{v_z}\rangle\langle\overline{v_z}|$ qu'on développe dans l'équation (2). Si on sépare en deux sommations les cas où $w = w'$ et $w \neq w'$, on obtient l'équation (3). Si $w = w'$, alors $w + w' = 0^n$ et donc pour chaque terme, la phase sera 1 et puisque z n'apparaît pas à l'intérieur de la sommation, on obtiendra 2^n sommations identiques (tel qu'exprimé dans la première partie de l'équation (4)). Dans le cas où $w \neq w'$, alors on a que $w + w'$ ne sera jamais nul. De plus, on peut inverser l'ordre des sommations car z est indépendant du reste. Donc, en utilisant l'équation 1.3, on voit que la somme pour les z donnera toujours 0. Cette sommation disparaîtra et dans l'équation (5), il ne reste que les termes où $w = w'$.

$$
\begin{aligned}
\rho &\overset{(1)}{=} \sum_{z\in\mathbb{Z}_2^n} \frac{1}{2^n}\left[\left(\frac{1}{\sqrt{2^{k_2}}}\sum_{w\in C_2}(-1)^{w\cdot z}\,|v+w+x\rangle\right)\left(\frac{1}{\sqrt{2^{k_2}}}\sum_{w'\in C_2}(-1)^{w'\cdot z}\,\langle v+w'+x|\right)\right] \\
&\overset{(2)}{=} \frac{1}{2^{n+k_2}}\sum_{z\in\mathbb{Z}_2^n}\left[\sum_{w,w'\in C_2}(-1)^{(w+w')\cdot z}\,|v+w+x\rangle\langle v+w'+x|\right] \\
&\overset{(3)}{=} \frac{1}{2^{n+k_2}}\left[\left(\sum_{z\in\mathbb{Z}_2^n}\sum_{w\in C_2}(-1)^{(w+w)\cdot z}\,|v+w+x\rangle\langle v+w+x|\right)\right. \\
&\qquad \left.+\left(\sum_{z\in\mathbb{Z}_2^n}\sum_{w\neq w'\in C_2}(-1)^{(w+w')\cdot z}\,|v+w+x\rangle\langle v+w'+x|\right)\right] \\
&\overset{(4)}{=} \frac{1}{2^{n+k_2}}\left[\left(2^n\sum_{w\in C_2}|v+w+x\rangle\langle v+w+x|\right)\right. \\
&\qquad \left.+\left(\sum_{w\neq w'\in C_2}\sum_{z\in\mathbb{Z}_2^n}(-1)^{(w+w')\cdot z}\,|v+w+x\rangle\langle v+w'+x|\right)\right] \\
&\overset{(5)}{=} \frac{1}{2^{k_2}}\sum_{w\in C_2}|v+w+x\rangle\langle v+w+x| \qquad (4.2)
\end{aligned}
$$

Si Alice avait envoyé l'état $|v+w+x\rangle$ pour un w aléatoirement choisi dans C_2, Bob aurait entre les mains le même état. Donc, Alice pourrait aussi bien envoyer $|v'+x\rangle$ pour un v' aléatoirement choisi dans C_1 (puisque v dépend de κ qui est aléatoire, alors $v'\in C_1$ est aussi aléatoire). Si on ne considère que les erreurs de négation, Bob recevrait $|v'+x+e\rangle$ et grâce au x transmis par Alice, il pourrait classiquement corriger la chaîne $v'+e$ à v' si $w(e)\leq t$ (en utilisant le code C_1). Puisque la clé est retrouvée en décodant $|\overline{v}\rangle$ et que chaque $u\in\text{Coset}(v)$ donne le même résultat, alors on remarque que la clé est en fait identifiée par le coset de v. Ces dernières modifications sont en fait ce qu'on obtient dans le protocole BB84 présenté dans la figure 4.2. L'état effectif transmis par Alice (sans les qubits test et les qubits de bases différentes) est $|b^{(\bar{\tau})}\rangle$ et l'information de décodage (le x) est ici $v+b^{(\bar{\tau})}$ où $v\in C_1$. Bob, après ses mesures, obtient la chaîne $b^{(\bar{\tau})}+e$ à laquelle il soustrait $v+b^{(\bar{\tau})}$ pour obtenir $v+e$, qu'il corrige classiquement à v (avec grande probabilité). Le coset de v (i.e. vK^{T}) permet de dériver la clé κ.

Pour que le protocole soit sécuritaire, il faut choisir nos codes C_1 et C_2 judicieusement. Premièrement, on veut que les distances minimales de C_1 et de $C_2^{\perp}$ soient grandes, afin que la correction de la clé fonctionne avec grande probabilité et pour

tolérer le plus grand taux d'erreurs p possible. La borne de Gilbert-Varshamov quantique (voir théorème 3.21) nous dit que de bons codes existent. On obtient dans [SP00] qu'on peut atteindre des taux $p \approx 0.11$. En pratique, il faut que C_1 soit un code qui se décode efficacement. En citant le travail de [May98], [SP00] affirme qu'en choisissant un sous-code C_2 aléatoire d'un code C_1 efficacement décodable, on obtient avec grande probabilité un code $C_2^{\perp}$ avec une grande distance minimale.

La grande faiblesse de cette preuve est qu'elle ne s'applique pas directement aux cas expérimentaux [BLMS00]. En effet, la preuve suppose que la source de photons d'Alice est parfaite[12] (comme dans les preuves de [May98] et de [BBB$^+$99]). Mayers [May00] a remarqué que la preuve de Shor et Preskill nécessite aussi que les appareils de mesure de Bob soient parfaits (sinon, on ne peut pas dériver la borne 4.1 sur la fidélité de l'état final). Mayers et Yao [MY98] ont proposé une façon de tester la qualité d'une source de photons et des appareils de mesure pour savoir s'ils sont assez sécuritaires pour les utiliser dans le protocole BB84. De plus, Ben-Or [BO] a annoncé qu'il avait développé une preuve de sécurité du protocole si la source utilisée est presque parfaite.

Le protocole BB84 nécessite de laisser tomber en moyenne la moitié des photons envoyés (si Alice et Bob ne choisissent pas la même base). Lo, Chau et Ardehali [LCA00] ont récemment présenté une modification du protocole BB84 qui le rend plus efficace. En quelques mots, les bases de transmission et de mesure sont choisies avec probabilité p et $1-p$ (ici, on avait $p = 1-p = 1/2$). Pour un choix judicieux de p, les bases correspondront sur un nombre élevé de positions, donc moins de photons seront nécessaires pour dériver une clé de longueur k. Cependant, l'estimation du taux d'erreurs doit se faire de façon plus raffinée (sinon le protocole devient non sécuritaire). La preuve de sécurité présentée ici se généralise directement à ce protocole, prouvant donc sa sécurité.

[12] Une source de photons est parfaite si elle permet de créer un photon à la fois. Il est très difficile d'avoir une source parfaite en laboratoire. On se contente de sources (un laser, par exemple) qui permettent d'envoyer un court rayon lumineux (possiblement) constitué de quelques photons polarisés de la même façon. Ceci est très dangereux pour la sécurité du protocole car Ève pourrait simplement garder les photons en trop (avec un séparateur de rayons lumineux) pour pouvoir trouver la clé.

4.3 Les répéteurs quantiques

Depuis que la DQC a vu le jour dans les laboratoires, les chercheurs ont tenté d'augmenter les distances de transmission. Les erreurs survenant sur le canal quantique (par exemple, la fibre optique) limitent les distances sur lesquelles on peut exécuter le protocole. Plus le canal est long, plus la probabilité d'erreurs est grande. Classiquement, on utilise des répéteurs électroniques pour augmenter la distance atteinte en amplifiant le signal électrique transmis. Cependant, les lois de la mécanique quantique nous empêchent d'utiliser cette technique sur des signaux quantiques.

Certains chercheurs [BDCZ98] ont développé des répéteurs quantiques qui utilisent la purification de l'intrication pour augmenter la fiabilité du canal. Leur technique consiste à transmettre des demi-paires de Bell (par exemple, $|\Psi^-\rangle$) sur le canal et de les purifier afin de pouvoir s'en servir pour téléporter [BBC+93] l'état à transmettre. Cependant, il y a des limites sur le bruit toléré par le protocole de purification. Puisque sur un canal de transmission, le taux d'erreurs est proportionnel à la distance, ils peuvent contourner le problème en utilisant des stations relais tout au long du canal afin d'effectuer la purification à plusieurs reprises.

Les protocoles de purification les plus efficaces nécessitent une communication classique bidirectionelle. Bennett, DiVincenzo, Smolin et Wooters [BDSW97] ont montré qu'on pouvait utiliser des codes correcteurs pour effectuer certains protocoles de purification moins efficaces, en évitant cependant l'utilisation du canal de communication classique. On peut alors implanter les répéteurs quantiques en utilisant les codes correcteurs. C'est ce que nous considèrerons dans cette section.

Un répéteur quantique consiste donc en un appareil qui implante la correction d'erreurs sur un canal quantique. Par exemple, au lieu de transmettre directement un état quantique ρ, Alice enverrait plutôt l'état encodé $\bar{\rho} = \mathcal{C}_Q(\rho)$ pour un code quantique Q. Une série de répéteurs, placés à des distances appropriées sur le canal (comme l'illustre la figure 4.6, $\mathcal{Q}$ représentant le circuit de correction du code Q), corrigeraient les erreurs sur des portions indépendantes du canal. Les répéteurs quantiques permettent,

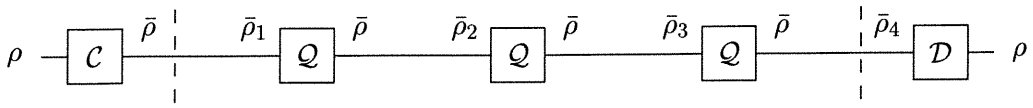


FIG. 4.6: Répéteurs quantiques sur un canal

en principe, de faire des transmissions avec grande fidélité sur des distances arbitraires.

Les répéteurs quantiques pourraient donc nous être utiles pour augmenter les distances atteintes par la DQC. En théorie, les codes correcteurs résolvent le problème, mais ils nécessitent un ordinateur quantique pour être implantés. On cherche donc les solutions les plus simples (conceptuellement et technologiquement) pour pouvoir utiliser les répéteurs dans un avenir rapproché.

Le contexte que nous étudions n'est cependant pas aussi restrictif que dans le cas général de communication. En effet, nous ne transmettons pas ici un état quantique que nous devons protéger à tout prix, mais un photon choisi parmi quatre polarisations qui peut être jeté sans problème. Donc, les répéteurs quantiques peuvent se contenter de détecter les erreurs et de laisser tomber les photons qui sont erronés.

Le code le plus simple pour détecter une erreur est le code à 4 qubits présenté dans la section 3.5. On peut en effet le modifier pour qu'il détecte une erreur arbitraire au lieu de corriger une erreur d'effacement.

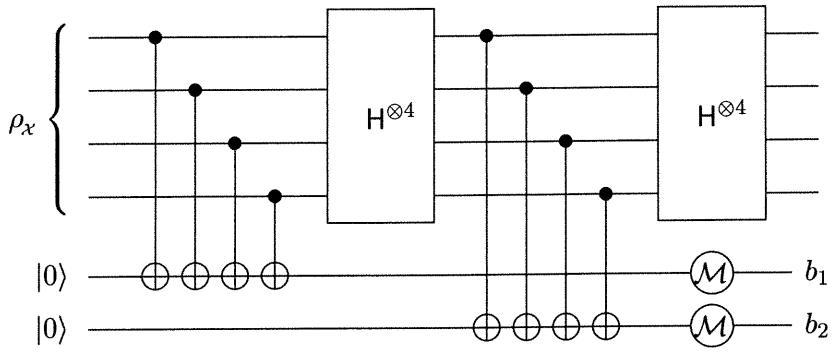


FIG. 4.7: Circuit de détection d'erreurs du code à 4 qubits

On se rappelle que dans le code à quatre qubits, on a que

$$\begin{aligned} |\overline{0}\rangle &= \frac{1}{\sqrt{2}}\Big(|0000\rangle + |1111\rangle\Big) \\ |\overline{1}\rangle &= \frac{1}{\sqrt{2}}\Big(|0011\rangle + |1100\rangle\Big) \end{aligned}$$

Puisque chaque état logique est une superposition de mots de poids pair, on peut détecter une erreur de négation en calculant la parité des quatre qubits. De plus, il est facile de vérifier que

$$\mathsf{H}^{\otimes 4}|\overline{0}\rangle = \frac{1}{\sqrt{8}}\Big(\sum_{\substack{x\in\{0,1\}^4 \\ \pi(x)=0}} |x\rangle\Big) \tag{4.3}$$

$$\mathsf{H}^{\otimes 4}|\overline{1}\rangle = \frac{1}{\sqrt{8}}\Big(\sum_{\substack{x\in\{0,1\}^4 \\ \pi(x)=0}} (-1)^{x_1\oplus x_2}|x\rangle\Big). \tag{4.4}$$

On remarque que les états logiques dans la base diagonale sont aussi des superpositions de mots de parité 0. On peut donc corriger les erreurs de phase en transformant le registre avec des transformation de Walsh-Hadamard et en calculant la parité de chaque position. Par le théorème 3.2, on conclut que le code peut détecter une erreur arbitraire sur l'un des qubits. Le circuit 4.7 permet de faire cette vérification. Si $b_1 = b_2 = 0$, alors on conclut que le registre est intact. Sinon, on sait qu'une erreur a affecté un des qubits. Le circuit de détection est relativement simple et pourrait être implanté avec les technologies expérimentales actuelles (voir [LVZS99] pour un exemple d'implantation d'un circuit de correction d'erreur).

Si on insiste pour corriger absolument les photons, le code le plus petit pour y arriver est le code à 5 qubits (voir section 3.6). Dans ce code, la probabilité que le registre encodé ne puisse être corrigé, après avoir été transmis dans $\mathcal{Q}D_p$, est

$$p_5 = 1 - \text{prob(aucune ou une seule erreur)} = 1 - \Big((1-p)^5 + 5p(1-p)^4\Big).$$

Donc, si $0 < p < 0,131$, alors $p_5 < p_{\text{direct}} = p$. Si on suppose que le canal sur lequel on fait la DQC est représenté par $\mathcal{Q}D_p$ où p varie entre 0 et 1 selon la distance (i.e. si on envoie un photon jusqu'à la moitié du canal, on aurait $p = 1/2$ et si on l'envoie sur le canal au complet, on aurait $p = 1$. Donc, en plaçant des répéteurs à des distances

telles que la probabilité d'erreur soit inférieure à p_5, on transmettrait les photons avec une meilleure fidélité. En ajustant la distance entre les répéteurs, on pourrait obtenir une très haute fidélité de transmission. Ceci est vrai si on suppose que les répéteurs n'introduisent pas d'erreurs sur les états transmis. Les prototypes devront probablement utiliser des techniques de tolérance aux fautes [ABO99, Pre97, Got98] pour éviter cette situation.

Alternativement, on préférera peut-être utiliser le code de Steane car son circuit d'encodage et de décodage est plus simple (donc plus facile à implanter en pratique). Preskill [Pre97] a analysé les difficultés relatives pour implanter les circuits de ces deux codes.

On remarque que l'ajout de tels répéteurs n'enlève rien à la sécurité du protocole, car Ève aurait elle-même pu effectuer cette procédure sur le canal.

Chapitre 5

Partage de secret quantique

David, président d'un pays puissant, possède le code de lancement qui active les missiles de l'arsenal militaire. Il désire léguer le contrôle de ces missiles à ses généraux : Patrick, Patrice, Paul, Pierre et Philippe. Cependant, il ne veut pas simplement donner une copie du code à chacun, de peur que l'un d'entre eux abuse du pouvoir qui lui sera accordé. Il décide donc que pour lancer les missiles, au moins trois généraux devront s'être concertés. Ce qu'il voudrait faire est de séparer son code en plusieurs morceaux et de les distribuer de telle sorte que si trois généraux se regroupent, ils pourront reconstituer le code de lancement. Par contre, les parts distribuées à deux généraux ne devraient donner aucun indice sur le code de lancement, i.e. elles ne devraient contenir aucune information.

David peut y arriver en utilisant un protocole de partage de secrets, i.e. un protocole qui permet de distribuer le secret parmi certains participants de telle façon que seuls certains sous-ensembles d'entre eux puissent le reconstituer.

Nous verrons dans ce chapitre un partage de secrets classiques ainsi que sa généralisation qui permet de partager un secret quantique. Nous décrirons le lien entre les codes correcteurs d'erreurs et les protocoles de partage de secrets et nous verrons comment les codes correcteurs permettent de construire ces derniers.

Initialisation

Soit $s \in \mathbb{F}_q$ le secret à distribuer dans le groupe $P = \{P_1, \ldots, P_n\}$ où $q > n$. D choisit n éléments distincts et non-nuls de $\mathbb{F}_q$, notés x_i, et les annonce publiquement.

Distribution

1) D choisit aléatoirement $r-1$ éléments $a_1, \ldots, a_{r-1} \in \mathbb{F}_q$.

2) Il forme ensuite le polynôme $f(x) \in \mathbb{F}_q[x]$ de degré $r-1$.

$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_{r-1} x^{r-1}$$

3) Chaque participant P_i reçoit sa part $y_i = f(x_i)$.

FIG. 5.1: Le PS (r, n) de Shamir

5.1 Partage de secrets : cas classique

Un des premiers exemples de partage de secret est dû à Shamir [Sha79]. Nous n'étudierons qu'un certain type de protocole, défini comme suit.

Définition 5.1 Un *partage de secrets* (PS) est un protocole permettant à un *distributeur* D de distribuer un secret $s \in \mathbb{F}_q$ entre n participants de telle façon que tout groupe de $r \leq n$ participants peut reconstituer le secret, mais tout groupe de moins de r personnes ne peut obtenir aucune information sur s. On note un tel protocole un PS (r, n). On note par $P = \{P_1, \ldots, P_n\}$ l'ensemble des participants et par $R \subseteq P$ l'ensemble des participants qui veulent reconstituer le secret. ◇

Dans la littérature, un tel protocole est appelé un partage de secrets *à seuil*, car le secret peut être retrouvé seulement lorsque $|R|$ dépasse un certain seuil. D'autres protocoles permettent de protéger le secret contre des structures d'adversaires plus complexes. Cependant, nous nous intéresserons seulement au type de PS défini plus haut.

Définition 5.2 Soit $x = (x_1, \ldots, x_k) \in \mathbb{F}_q^k$. La *matrice de Vandermonde* associée à x est la matrice

$$V_k(x) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_k \\ x_1^2 & x_2^2 & \ldots & x_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{k-1} & x_2^{k-1} & \ldots & x_k^{k-1} \end{pmatrix}$$

i.e. $V_k(x)_{ij} = x_j^{i-1}$, pour $1 \leq i, j \leq k$. ◇

Il est bien connu que le déterminant de $V_k(x)$ est $\prod_{i<j}(x_j - x_i)$. Si les x_i sont distincts, le déterminant est non-nul et la matrice est donc inversible.

Le PS de Shamir est défini dans la figure 5.1. Le protocole de Shamir n'est basé sur aucune hypothèse calculatoire. Comme le montre le théorème suivant, le protocole est bien un partage de secrets sécure.

Théorème 5.3 Le protocole de Shamir est un PS (r, n). ◇

Preuve Soit $R = \{P_{i_1}, \ldots, P_{i_t}\}$ les t participants qui veulent reconstituer le secret.

- Si $t \geq r$, alors les participants de R peuvent retrouver le secret. Ensemble, ils connaissent t points (x_i, y_i) du polynôme. Ils n'ont besoin que de r points pour retrouver s. En effet, on a que

$$\left(s, a_1, \ldots, a_{r-1}\right) \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_{i_1} & x_{i_2} & \ldots & x_{i_r} \\ x_{i_1}^2 & x_{i_2}^2 & \ldots & x_{i_r}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{i_1}^{r-1} & x_{i_2}^{r-1} & \ldots & x_{i_r}^{r-1} \end{pmatrix} = \left(y_{i_1}, y_{i_2}, \ldots, y_{i_r}\right)$$

Puisque la matrice de Vandermonde est inversible, il peuvent retrouver le vecteur

des coefficients de f en calculant

$$(y_{i_1}, y_{i_2}, \ldots, y_{i_r}) \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_{i_1} & x_{i_2} & \ldots & x_{i_r} \\ x_{i_1}^2 & x_{i_2}^2 & \ldots & x_{i_r}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{i_1}^{r-1} & x_{i_2}^{r-1} & \ldots & x_{i_r}^{r-1} \end{pmatrix}^{-1} = (s, a_1, \ldots, a_{r-1})$$

Donc, tout groupe d'au moins r participants peut retrouver le secret s.

- Si $t < r$, les participants de R n'apprennent rien sur le secret. En effet, supposons que $t = r - 1$. Les participants de R ont $r - 1$ équations à r inconnues. Pour chaque secret s', il existe une r^{e} équation $f(0) = s'$ cohérente avec les autres. Donc, aucune information n'est révélée par leurs parts.

□

Exemple 5.4 Supposons que David veuille partager le secret $s = 7$ du corps $\mathbb{Z}_{13}$ parmi 5 participants, pour que 3 d'entre eux puissent le retrouver. David, qui utilise le PS $(3, 5)$ de Shamir, choisit et annonce publiquement les $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $x_4 = 4$ et $x_5 = 5$. Il construit ensuite le polynôme aléatoire (en fixant $a_0 = 7$)

$$f(x) = 7 + 10x + 2x^2.$$

Par la suite, David calcule $y_1 = 6$, $y_2 = 9$, $y_3 = 3$, $y_4 = 1$, $y_5 = 3$ et les distribue au P_i correspondant.

La reconstitution du secret par le groupe $R = \{P_2, P_3, P_5\}$ se fait comme suit. Ils forment la matrice de Vandermonde suivante et calculent son inverse (tous les calculs se font dans $\mathbb{Z}_{13}$) :

$$V_3(x_2, x_3, x_5) = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 9 & 12 \end{pmatrix} \qquad V_3(x_2, x_3, x_5)^{-1} = \begin{pmatrix} 5 & 6 & 9 \\ 8 & 10 & 6 \\ 1 & 10 & 11 \end{pmatrix}$$

Finalement, ils calculent

$$(9,3,3)\begin{pmatrix} 5 & 6 & 9 \\ 8 & 10 & 6 \\ 1 & 10 & 11 \end{pmatrix} = (\mathbf{7},10,2)$$

et ils retrouvent le secret $s = 7$. Cependant, P_1 et P_4 n'apprennent rien sur le secret. Leur information commune se résume à

$$\begin{aligned} y_1 &= 6 \equiv s + a_1 + a_2 \pmod{13} \\ y_4 &= 1 \equiv s + 4a_1 + 16a_2 \pmod{13} \end{aligned}$$

Tous les s étant équiprobables, leurs parts ne révèlent aucune information sur s. ◇

Remarque 5.5 Dans le protocole de Shamir, on peut facilement ajouter des participants (sans toutefois changer le seuil r qui permet de reconstituer le secret). Pour ce faire, le distributeur choisit de nouveaux $x_i \neq 0$ distincts des autres et calcule pour chacun $y_i = f(x_i)$. Le nombre total de participants doit cependant être inférieur à q (la cardinalité du corps). ◇

Remarque 5.6 On peut varier le niveau de confiance entre les participants en donnant un nombre différent de parts à chacun. Si on reprend l'exemple du début de la section, supposons que Gérard est le général en chef. En lui donnant deux parts, David lui permet de retrouver le code de lancement avec l'aide d'un seul autre général. ◇

5.2 Partage de secrets : cas quantique

Dans le cas quantique, le secret à partager est un état arbitraire $|\psi\rangle \in \mathcal{H}_q$ inconnu du distributeur et les parts des participants seront des états mélangés de dimension possiblement supérieure à q. Nous verrons qu'il y a certaines contraintes sur les partages de secrets quantiques (PSQ) qui sont imposées par les lois de la mécanique quantique.

Définition 5.7 Un *partage de secrets quantiques* PSQ $((r,n))$ est un protocole permettant à un distributeur D d'encoder et de distribuer un état quantique $|\psi\rangle \in \mathcal{H}_q$ entre

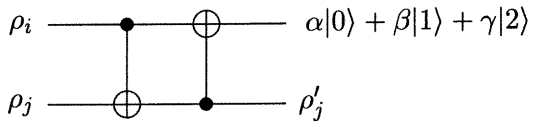
n participants de telle façon que tout groupe de $r \leq n$ peut reconstituer l'état $|\psi\rangle$, mais que tout groupe de moins de r personne ne peut obtenir aucune information sur le secret. $\diamond$

En terme quantique, la deuxième condition stipule que la matrice de densité réduite des parts de moins de r participants (obtenue en prenant la trace partielle de r ou plus parts) doit être indépendante de l'état $|\psi\rangle$.

Exemple 5.8 Comme premier exemple, voici un PSQ $((2,3))$ qui permet de partager un *qutrit* parmi trois participants. Tout d'abord, on encode le qutrit $|\psi\rangle \in \mathcal{H}_3$ par

$$\begin{aligned} |\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle \quad \longrightarrow \quad & \frac{\alpha}{3}\Big(|000\rangle + |111\rangle + |222\rangle\Big) + \\ & \frac{\beta}{3}\Big(|012\rangle + |120\rangle + |201\rangle\Big) + \\ & \frac{\gamma}{3}\Big(|021\rangle + |102\rangle + |210\rangle\Big) \end{aligned} \tag{5.1}$$

Le circuit 2.2, de l'exemple 2.27, permet de créer cette superposition à partir de $|\psi\rangle$ et de deux états ancillaires $|0\rangle$. Chaque participant reçoit un des qutrits, i.e. la sortie d'un des fils. Deux participants peuvent ensemble retrouver le secret $|\psi\rangle$ en utilisant le circuit suivant.



En effet, supposons que les deux premiers participants veulent reconstituer le secret.

Si on applique le circuit sur la superposition 5.1, en prenant $i = 1$ et $j = 2$, on obtient :

$$\begin{aligned}
\xrightarrow{\mathsf{O}_3^{(1,2)}} \quad & \frac{\alpha}{3}\Big(|000\rangle + |121\rangle + |212\rangle\Big) + \\
& \frac{\beta}{3}\Big(|012\rangle + |100\rangle + |221\rangle\Big) + \\
& \frac{\gamma}{3}\Big(|021\rangle + |112\rangle + |200\rangle\Big) \\
\xrightarrow{\mathsf{O}_3^{(2,1)}} \quad & \frac{\alpha}{3}\Big(|000\rangle + |021\rangle + |012\rangle\Big) + \\
& \frac{\beta}{3}\Big(|112\rangle + |100\rangle + |121\rangle\Big) + \\
& \frac{\gamma}{3}\Big(|221\rangle + |212\rangle + |200\rangle\Big) \\
= \quad & \Big(\alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle\Big)\frac{1}{\sqrt{3}}\Big(|00\rangle + |12\rangle + |21\rangle\Big)
\end{aligned}$$

Par contre, l'état de la part d'un seul participant est indépendant de $|\psi\rangle$. En effet, on peut vérifier que pour $1 \leq i < j \leq 3$,

$$\mathrm{tr}_{i,j}(|\psi\rangle\langle\psi|) = \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix}$$

qui est l'état complètement mélangé sur $\mathcal{H}_3$. ◇

Remarque 5.9 Il n'est pas nécessaire d'utiliser toutes les parts créées par un protocole de partage de secrets. On peut construire un PSQ $((r, n-1))$ à partir d'un PSQ $((r, n))$ (où $r < n$) en jetant (i.e. détruisant) une part lors de l'encodage. ◇

Exemple 5.10 Pour illustrer la remarque précédente, on voit qu'on peut construire un PSQ $((2, 2))$ à partir du PSQ $((2, 3))$ présenté dans l'exemple 5.8. Pour ce faire, on détruit le qutrit sortant du troisième fil du circuit 2.2 et on distribue normalement les deux autres parts à P_1 et P_2. Ils devront tous deux se réunir pour reconstruire le secret. On note que l'état encodé sera ici un état mélangé, obtenu en prenant la trace partielle du troisième qutrit dans la superposition 5.1. ◇

Classiquement, la construction de Shamir permet de créer des protocoles de PS (r, n) pour tous r et $n \geq r$. Cependant, les lois de la mécanique quantique nous imposent des restrictions sur les paramètres des PSQ.

Théorème 5.11 Il n'existe pas de PSQ $((r,n))$ pour $n \geq 2r$. ◇

Preuve Supposons que $n \geq 2r$ et soit $R_i = \{P_{i_1}, \ldots, P_{i_r}\}$ et $R_j = \{P_{j_1}, \ldots, P_{j_r}\}$ deux groupes de participants disjoints voulant reconstituer le secret. Puisque $|R_i| = |R_j| = r$, chaque groupe peut reconstruire le secret initial $|\psi\rangle$. Cette procédure permettrait alors de copier des états arbitraires de $\mathcal{H}_q$, ce qui contredit le théorème 2.9 de non-clonage. Donc, un tel protocole n'existe pas. □

Il y a un lien important entre les codes correcteurs quantiques (CCEQ) et les PSQ. En effet, tout protocole PSQ $((r,n))$ est un code correcteur qui permet d'encoder un qu*q*it sur n tout en corrigeant $r-1$ erreurs d'effacement. Cependant, tous les codes ne donnent pas des PSQ. En effet, si un code corrige t erreurs d'effacement, alors moins de $n-t$ qu*q*its ne contiennent pas nécessairement aucune information (la remarque 3.17 donne un exemple de ce fait). Or, les PSQ demandent qu'on ne puisse tirer aucune information d'un nombre insuffisant de parts. Les conditions pour qu'un code puisse servir de PSQ sont données par le théorème suivant :

Théorème 5.12 Tout CCEQ Q de paramètres $[[2d-1,1,d]]_q$ peut être transformé en un PSQ $((d,2d-1))$. ◇

Preuve Si le code corrige $d-1$ erreurs d'effacement, on peut reconstituer le secret encodé par le code en réunissant au moins $(2d-1)-(d-1) = d$ parts. Le théorème 2.10 nous permet de conclure qu'on ne peut tirer aucune information sur l'état de départ si on ne possède que $d-1$ positions. En effet, si on pouvait obtenir de l'information sur le secret avec $d-1$ parts, alors cette interaction modifierait l'information contenue dans les d autres parts. Donc, $d-1$ parts ne contiennent aucune information sur le secret. □

La remarque 5.9 nous permet d'affirmer le corollaire suivant.

Corollaire 5.13 On peut construire un PSQ $((r,n))$ pour tout $r \leq n < 2r$ à partir d'un CCEQ $[[2r-1,1,r]]_q$ en jetant les parts non désirées. ◇

Exemple 5.14 Le CCEQ $[[5,1,3]]$ présenté à la section 3.6 permet de corriger deux erreurs d'effacement. Il peut donc être utilisé comme PSQ $((3,5))$. En jetant des parts,

on peut construire des PSQ $((3,4))$ et $((3,3))$. ◇

Remarque 5.15 On peut montrer qu'il n'existe pas de PSQ $((2,3))$ sur des qubits. En effet, si un tel protocole existait, il pourrait servir de CCEQ qui encode un qubit sur trois et qui corrige une erreur d'effacement. Or, on a vu à la section 3.5 qu'un tel code n'existe pas. Par contre on peut utiliser le protocole de l'exemple 5.8 sans utiliser la troisième dimension du qutrit secret pour encoder un qubit. Les parts seront cependant des qutrits. ◇

5.2.1 Construction de CGL

Cleve, Gottesman et Lo [CGL99] ont montré comment construire des codes pouvant servir de PSQ $((r,n))$ pour tout $r \leq n < 2r$. Leur construction généralise le protocole de Shamir (de la section 5.1) et utilise une variante des codes polynomiaux de la section 3.7. Nous présenterons dans la section 5.2.2 une modification de ce protocole qui utilise directement les codes polynomiaux.

Définition 5.16 Soit $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$. Le code CSS CGL Q_x^d de degré d est construit avec les codes linéaires suivants :

$$\begin{aligned} C_1 &= \left\{ \Big(f_a(x_1), \ldots, f_a(x_n)\Big) : a \in \mathbb{F}_q^{d+1} \right\} \\ C_2 &= \left\{ \Big(f_a(x_1), \ldots, f_a(x_n)\Big) : a \in \mathbb{F}_q^{d+1} \wedge a_{d+1} = 0 \right\} \end{aligned}$$

Comme dans le cas des codes polynomiaux, on peut montrer que la distance minimale de Q_x^d est au moins $\min(n-d, d+1)$. Pour $s \in \mathbb{F}_q$, l'encodage par Q_x^d produit

$$\mathcal{C}_{Q_x^d}(|s\rangle) = \sum_{\substack{a \in \mathbb{F}_q^{d+1} \\ a_d = s}} |f_a(x_1), \ldots, f_a(x_n)\rangle.$$

◇

Le protocole de CGL est décrit dans la figure 5.2. Les parts seront de dimension q et on peut montrer qu'on peut le choisir tel que $q \leq 2\max(q', n)$. L'étape 1) n'est pas très élaborée. Elle consiste seulement à considérer l'état de $\mathcal{H}_{q'}$ comme un état de $\mathcal{H}_q$

Paramètres : r et n tels que $r \leq n < 2r$.

Initialisation

Soit $|\psi'\rangle \in \mathcal{H}_{q'}$ le secret à distribuer dans le groupe $P = \{P_1, \ldots, P_n\}$. D choisit un nombre premier q tel que $\max(q', n) \leq q$. Il choisit aussi n éléments distincts $x_i \in \mathbb{F}_q$ et forme $x = (x_1, \ldots, x_n)$ qu'il annonce publiquement.

Distribution

1) D transforme l'état $|\psi'\rangle \in \mathcal{H}_{q'}$ pour obtenir un état $|\psi\rangle \in \mathcal{H}_q$, i.e.

$$\sum_{i \in \mathbb{F}_{q'}} \alpha_i |i\rangle \longrightarrow \sum_{i \in \mathbb{F}_{q'}} \alpha_i |i\rangle + \sum_{j \in \mathbb{F}_q - \mathbb{F}_{q'}} 0|j\rangle.$$

2) D encode $|\psi\rangle = \sum_{s \in \mathbb{F}_q} \alpha_s |s\rangle$ (où $\sum_{s \in \mathbb{F}_q} |\alpha_s|^2 = 1$) par le code CGL Q_x^{r-1} de degré $r-1$,

$$\mathcal{C}_{Q_x^{r-1}}(|\psi\rangle) = \rho_1 \otimes \ldots \otimes \rho_n$$

où pour $s \in \mathbb{F}_q$,

$$\mathcal{C}_{Q_x^{r-1}}(|s\rangle) = \sum_{\substack{a \in \mathbb{F}_q^r \\ a_{r-1} = s}} |f_a(x_1) \ldots f_a(x_n)\rangle. \tag{5.2}$$

3) Chaque participant P_i reçoit sa part ρ_i.

FIG. 5.2: Le PSQ (r, n) de CGL

(comme si on considérait un vecteur $(x, y) \in \mathbb{R}^2$ comme le vecteur $(x, y, 0) \in \mathbb{R}^3$). Le théorème suivant démontre la sécurité du protocole.

Théorème 5.17 Le protocole de CGL est un PSQ $((r, n))$. ◇

Preuve Supposons que $R = \{\rho_{i_1}, \ldots, \rho_{i_t}\} \subset \{\rho_1, \ldots, \rho_n\}$ avec $t \geq r$. Puisque le code CGL Q_x^{r-1} a les paramètres $[[n, 1, n-r+1]]_q$, il peut donc corriger $n-r$ erreurs d'effacement (la distance minimale est bien $n-r+1$ car $n-r+1 \leq r$ si $n < 2r$ La procédure de décodage de Q_x^{r-1} permettra alors de reconstituer le secret à partir de t parts de R. De plus, l'état obtenu en prenant la trace partielle de l'espace des t parts est complètement indépendant de $|\psi\rangle$ (par le théorème 5.12). Donc le protocole de CGL est bel et bien un PSQ $((r, n))$. □

Exemple 5.18 L'exemple 5.8 utilise cette construction. Ici, $q = 3$, $m = 3$, $r = 2$, $\mathbb{F}_3 = \mathbb{Z}_3$, $x_0 = 0$, $x_1 = 1$ et $x_2 = 2$. ◇

Exemple 5.19 Pour illustrer plus clairement comment fonctionne la reconstitution d'un secret partagé par le PSQ de CGL, décrivons une procédure de décodage pour le cas $n = 2r - 1$. Sans perte de généralité, supposons qu'on veuille reconstituer le secret à partir du registre formé des r premiers quqits, i.e. $\rho = \rho_1 \otimes \ldots \otimes \rho_r$. La procécude de décodage est la suivante :

1) On applique la matrice de Vandermonde inverse $V_r^{-1}(x_1, \ldots, x_r)$ à l'état ρ pour obtenir $\rho^{(1)}$.
2) On déplace chaque quqit cycliquement vers la droite pour obtenir $\rho^{(2)} = \rho_r^{(1)} \otimes \rho_1^{(1)} \otimes \ldots \otimes \rho_{r-1}^{(1)}$ (cette étape permettra de terminer avec le secret $|\psi\rangle$ en position 1).
3) On applique la matrice de Vandermonde $V_{r-1}(x_{r+1}, \ldots, x_n)$ à l'état $\rho_2^{(2)} \otimes \ldots \otimes \rho_r^{(2)}$ pour obtenir $\rho^{(3)}$ ($\rho^{(2)}$ reste intouché à cette étape).
4) On additionne $\left(\rho_1^{(3)} \times x_{r+i}^{r-1}\right)$ à $\rho_i^{(3)}$, pour $2 \leq i \leq r$, ce qui donne $\rho^{(4)}$.
5) On a que $\rho^{(4)} = |\psi\rangle \otimes \rho_2^{(4)} \otimes \ldots \otimes \rho_r^{(4)}$.

Considérons l'effet de cette procédure sur un mot de base du code Q_x^{r-1}. Soit $s \in \mathbb{F}_q$, alors le mot de base associé est décrit par l'équation 5.2. Notons que pour $y_1, \ldots, y_k \in$

$\mathbb{F}_q$, la matrice de Vandermonde $V_k(y_1, \ldots, y_k)$ appliquée à un état $|a_0 \ldots a_{k-1}\rangle$ (où les $a_i \in \mathbb{F}_q$) produit l'état $|f_a(y_1) \ldots f_a(y_k)\rangle$ et que la matrice inverse appliquée à ce dernier état donne l'état original des coefficients. Donc, l'étape 1 produit l'état

$$\rho^{(1)} = \sum_{\substack{a \in \mathbb{F}_r \\ a_{r-1}=s}} |a_0 \ldots a_{r-1}\rangle \otimes |f_a(x_{r+1}) \ldots f_a(x_n)\rangle$$

La deuxième partie en produit tensoriel représente les $r-1$ parts $\rho_{r+1}, \ldots, \rho_n$ qui sont inaccessibles. La deuxième étape amène le registre dans l'état

$$\rho^{(1)} = \sum_{\substack{a \in \mathbb{F}_r \\ a_{r-1}=s}} |a_{r-1} a_0 \ldots a_{r-2}\rangle \otimes |f_a(x_{r+1}) \ldots f_a(x_n)\rangle$$

Le premier qu*q*it contient alors le secret $|s\rangle$. Cependant, pour les $|\psi\rangle$ en superposition de plusieurs états de base, le premier qu*q*it est en général intriqué avec le reste du registre. Les deux dernières étapes permettront de le désintriquer du reste. En appliquant $V_{r-1}(x_{r+1}, \ldots, x_n)$ à l'état $|a_0, \ldots, a_{r-2}\rangle$, on obtient dans la troisième étape l'état $|f_{a'}(x_{r+1}) \ldots f_{a'}(x_n)\rangle$ et en additionnant $\left(\rho_1^{(3)} \times x_{r+i}^{r-1}\right) = s x_{r+i}^{r-1}$ à chaque position, on obtient la superposition suivante à l'étape 4

$$\begin{aligned} \rho^{(4)} &= |s\rangle \sum_{\substack{a \in \mathbb{F}_r \\ a_{r-1}=s}} |f_a(x_{r+1}) \ldots f_a(x_n)\rangle \otimes |f_a(x_{r+1}) \ldots f_a(x_n)\rangle \\ &= |s\rangle \sum_{y \in \mathbb{F}_q^{r-1}} |y_1 \ldots y_{r-1}\rangle \otimes |y_1 \ldots y_{r-1}\rangle \end{aligned} \tag{5.3}$$

On obtient la dernière égalité en remarquant que pour tout $s \in \mathbb{F}_q$ et $y \in \mathbb{F}_q^{r-1}$, il existe un unique $f_a \in \mathbb{F}_q[x]$ avec $a \in \mathbb{F}_q^{r-1}$ et $a_{r-1} = s$ tel que $f_a(x_{r+i}) = y_i$, pour $1 \leq i \leq r-1$. Les qu*q*its 2 à r sont à ce point indépendants du secret. On conclut donc que la reconstitution d'un secret arbitraire (par linéarité) est correcte et on peut récupérer le secret en position 1. ◇

La construction de CGL nous permet donc de construire des PSQ $((r, n))$ pour tout $r \leq n < 2r$. En effet, puisqu'on peut construire des CCEQ CGL $[[2r-1, 1, r]]_q$ pour tout $r > 0$, alors le corollaire 5.13 indique qu'ils peuvent construire des PSQ $((r, n))$ pour tout $r \leq n < 2r$.

5.2.2 Construction de CGL modifié

Nous présentons ici une version du PSQ de CGL qui utilise directement les codes polynomiaux de la section 3.7. Notre construction permet de simplifier le circuit de décodage et est une généralisation plus directe du protocole de Shamir. Cependant, les dimensions des parts seront possiblement plus grandes que dans le protocole CGL.

Le protocole de CGL camoufle le secret $s \in \mathbb{F}_q$ comme le coefficient de x^{r-1}, ce qui permet de choisir des x_i nuls. On peut simplifier la procédure de reconstitution si on utilise les mêmes polynômes que dans le PS de Shamir (i.e. s est le coefficient a_0). On peut alors se servir directement des codes polynomiaux de Aharonov et Ben-Or pour obtenir un PSQ. On doit alors restreindre les x_i à être non-nuls, ce qui peut nécessiter de choisir une dimension q supérieure à ce qu'on aurait choisi avec le protocole CGL (dans le cas où $q' = n$, par exemple). De plus, on a besoin d'un état ancillaire $|0\rangle \in \mathcal{H}_q$ pour permettre la reconstitution.

Le protocole modifié est décrit dans la figure 5.3. Il encode un secret $|s\rangle$ en une superposition de tous les choix possibles de polynômes utilisés dans le PS de Shamir.

Exemple 5.20 La procédure de reconstitution est très semblable à celle décrite dans l'exemple 5.19. Pour $n = 2r - 1$, soit $\rho = \rho_1 \otimes \ldots \otimes \rho_r$ le registre duquel on veut reconstituer $|\psi\rangle$. Voici les étapes à suivre.

1) On applique la matrice de Vandermonde inverse $V_r^{-1}(x_1, \ldots, x_r)$ à l'état ρ pour obtenir $\rho^{(1)}$.
2) On applique la matrice de Vandermonde $V_r(0, x_{r+1}, \ldots, x_n)$ à l'état $|0\rangle \otimes \rho_2^{(1)} \otimes \ldots \otimes \rho_r^{(1)}$ pour obtenir $\rho^{(2)}$ ($|0\rangle$ est un état ancillaire).
3) On additionne $\rho_1^{(2)}$ à $\rho_i^{(2)}$, pour $2 \leq i \leq r$, ce qui donne $\rho^{(3)}$.
4) On a que $\rho^{(3)} = |\psi\rangle \otimes \rho_2^{(3)} \otimes \ldots \otimes \rho_r^{(4)}$.

Cette procédure de décodage est illustrée par le circuit 5.4. L'étape 1 produit l'état

$$\rho^{(1)} = \sum_{\substack{a \in \mathbb{F}_r \\ a_0 = s}} |a_0, \ldots, a_{r-1}\rangle \otimes |f_a(x_{r+1}) \ldots f_a(x_n)\rangle$$

Paramètres : r et n tels que $r \leq n < 2r$.

Initialisation

Soit $|\psi'\rangle \in \mathcal{H}_{q'}$ le secret à distribuer dans le groupe $P = \{P_1, \ldots, P_n\}$. D choisit un nombre premier q tel que $q \geq q'$ et $q > n$. Il choisit aussi n éléments distincts et non-nuls $x_i \in \mathbb{F}_q$ et forme $x = (x_1, \ldots, x_n)$ qu'il annonce publiquement.

Distribution

1) D transforme l'état $|\psi'\rangle \in \mathcal{H}_{q'}$ pour obtenir un état $|\psi\rangle \in \mathcal{H}_q$.

2) D encode $|\psi\rangle$ par le code polynomial Q_x^{r-1} de degré $r-1$,

$$\mathcal{C}_{Q_x^{r-1}}(|\psi\rangle) = \rho_1 \otimes \ldots \otimes \rho_n$$

où pour $s \in \mathbb{F}_q$,

$$\mathcal{C}_{Q_x^{r-1}}(|s\rangle) = \sum_{\substack{a \in \mathbb{F}_q^r \\ a_0 = s}} |f_a(x_1) \ldots f_a(x_n)\rangle. \tag{5.4}$$

3) Chaque participant P_i reçoit sa part ρ_i.

FIG. 5.3: Le PSQ (r, n) de CGL modifié

Le premier qu*q*it contient alors le secret qui sera intriqué avec le reste du registre. On ajoute une ancille dans l'état $|0\rangle$ pour appliquer la matrice de Vandermonde sur r qu*q*its. Ainsi, un état $|0, a_1, \ldots, a_{r-1}\rangle$ sera transformé dans la deuxième étape à un état $|f_{a'}(x_{r+1}) \ldots f_{a'}(x_n)\rangle$ (où $f_{a'}(x) = 0 + a_1 x + \ldots + a_{r-1}x^{r-1}$) et en additionnant le qu*q*it $\rho_1 = |s\rangle$ à chaque position (ce qui se fait en appliquant un ou-exclusif sur $\mathcal{H}_q$), on obtient la superposition suivante à l'étape 3 :

$$\begin{aligned} \rho^{(3)} &= |s\rangle \sum_{\substack{a \in \mathbb{F}_r \\ a_0 = s}} |f_a(x_{r+1}) \ldots f_a(x_n)\rangle \otimes |f_a(x_{r+1}) \ldots f_a(x_n)\rangle \\ &= |s\rangle \sum_{y \in \mathbb{F}_q^{r-1}} |y_1 \ldots y_{r-1}\rangle \otimes |y_1 \ldots y_{r-1}\rangle \end{aligned} \tag{5.5}$$

On laisse tomber l'ancille, qui est encore dans l'état $|0\rangle$, car $f_{0^r}(0) = 0$. Les qu*q*its 2 à r sont à ce point indépendants du secret. On conclut donc que la reconstitution du secret est correcte. ◇

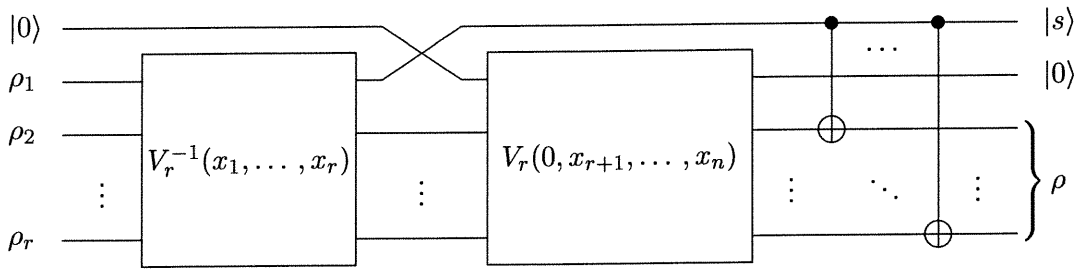


FIG. 5.4: Circuit de décodage de pour le PSQ de CGL modifié.

Conclusion

L'avenir de l'ordinateur quantique est incertain. Peut-être révolutionera-t-il la façon dont on traitera l'information dans le 21^e siècle, ou peut-être restera-t-il à jamais un modèle théorique impossible à construire. Encore plus nébuleux est l'avenir des codes correcteurs d'erreurs quantiques. La majorité des chercheurs du domaine croient que les codes correcteurs seront essentiels à la construction d'un ordinateur quantique (si une telle chose est possible). Mais peut-être qu'une découverte technologique les rendra désuets (comme les transistors ont fait pour les lampes à vide dans les premiers ordinateurs).

Les applications cryptographiques des codes correcteurs quantiques, présentées dans la deuxième partie, laissent croire que les codes correcteurs pourront être utilisés dans un contexte plus large que pour ce qu'ils ont été conçus. Les domaines de la cryptographie quantique et des codes correcteurs quantiques sont tous deux très récent et il est fort à parier que plus ces domaines progresseront, plus de liens les uniront.

Ces deux théories ont beaucoup progressé en tirant des idées des théories classiques respectives. La recherche future bénéficiera sûrement en explorant les liens entres les théories classiques respectives. Par exemple, le système cryptographique à clés publiques de McEliece [Sti95] est basé sur un problème de théorie des codes classiques. Il serait intéressant de voir si on peut en tirer un protocole quantique utilisant les codes quantiques.

Il est intéressant de remarquer que liens entre les codes quantiques et la distribution quantique de clés sont à la fois théoriques et pratiques. D'une part, ils peuvent être uti-

lisés pour prouver la sécurité du protocole et d'une autre, ils permettront probablement de rendre la distribution plus efficace (par l'ajout de répéteurs).

Bien que la preuve de sécurité de BB84 de Shor et Preskill n'est pas parfaite, elle a permis de voir les preuves de Lo et Chau et de Mayers sous un nouvel angle. Elle donne une bonne intuition sur ces dernières et permettra peut-être de construire une preuve qui sera acceptée globalement.

De plus, il est possible que les techniques de calcul avec tolérances aux fautes puissent être utilisées pour ôter l'hypothèse sur les appareils de mesure dans la preuve de Shor et Preskill. En effet, ces techniques permettent (en calcul avec tolérance aux fautes) de faire des mesures exactes même si les appareils sont imparfaits. On peut probablement adapter ces techniques pour les inclure dans la preuve.

Les protocoles de partage de secrets classiques ont permis de développer le domaine riche du calcul distribué. Certains chercheurs tentent d'utiliser le protocole CGL pour développer des protocoles de calculs distribués quantiques. Cependant, la transposition directe des techniques classiques ne fonctionne pas [Smi]. Si on arrive un jour à développer un tel domaine, les techniques présentées au chapitre 5 seront sûrement utilisées pour y arriver.

Bibliographie

[*] Les archives quant-ph se trouvent à l'adresse internet suivante : http ://xxx.lanl.gov/archive/quant-ph.

[ABO99] Dorit Aharonov et Michael Ben-Or. Fault tolerant quantum computation with constant error. Préparution, juin 1999. Archivé à quant-ph/9906129.

[BB84] Charles H. Bennett et Gilles Brassard. Quantum cryptography : Public-key distribution and coin tossing. *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India*, pp. 175 – 179, 1984.

[BBB+92] Charles H. Bennett, François Bessette, Gilles Brassard, Louis Salvail et John Smolin. Experimental quantum cryptography. *Journal of Cryptology*, vol. 5, no 1, pp. 3 – 28, 1992.

[BBB+99] Eli Biham, Michel Boyer, Oscar Boykin, Tal Mor et Vwani Roychowdhury. A proof of the security of quantum key distribution. *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, 1999. Archivé à quant-ph/9912053.

[BBC+93] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres et William K. Wootters. Teleporting an unkown quantum state by dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, vol. 70, pp. 1895 – 1898, 1993.

[BBM92] Charles H. Bennett, Gilles Brassard et N. David Mermin. Quantum cryptography without Bell's theorem. *Phys. Rev. Lett.*, vol. 68, pp. 557 – 559, 1992.

[BCT99] Gilles Brassard, Richard Cleve et Alain Tapp. The cost of exactly simulating quantum entanglement with classical communication. *Phys. Rev. Lett.*, vol. 83, no 9, pp. 1874 – 1878, août 1999. Archivé à quant-ph/9901035.

[BDCZ98] Hans J. Briegel, W. Dür, Juan I. Cirac et Peter Zoller. Quantum repeaters for communication. Préparution, mars 1998. Archivé à quant-ph/9803056.

[BDSW97] Charles H. Bennett, David P. DiVincenzo, John Smolin et William K. Wootters. Mixed state entanglement and quantum error correction. *Phys. Rev. A*, vol. 54, no 5, pp. 3824 – 3851, novembre 1997. Archivé à quant-ph/9604024.

[Ben92] Charles H. Bennett. Quantum cryptography using any two nonorthogonal states. *Phys. Rev. Lett.*, vol. 68, no 21, pp. 2121 – 2124, mai 1992.

[BHL+00] William T. Buttler, Richard J. Hughes, Steve K. Lamoreaux, George L. Morgan, Jane E. Nordholt et Glen Peterson. Daylight quantum key distribution over 1.6 km. *Phys. Rev. Lett.*, vol. 84, 2000. Archivé à quant-ph/0001088.

[BLMS00] Gilles Brassard, Norbert Lütkenhaus, Tal Mor et Barry C. Sanders. Limitations on practical quantum cryptography. *Phys. Rev. Lett.*, vol. 85, no 6, août 2000.

[BO] Michael Ben-Or. Non publié. Annoncé au NEC workshop on Quantum Cryptography, 1999.

[Bra] Gilles Brassard. *Quantum Information Processing for Computer Scientists.* À être publié.

[CG96] Richard Cleve et Daniel Gottesman. Efficient computation of encodings for quantum error correction. Préparution, juillet 1996. Archivé à quant-ph/9607030.

[CGL99] Richard Cleve, Daniel Gottesman et Hoi-Kwong Lo. How to share a quantum secret. Préparution, janvier 1999. Archivé à quant-ph/9901025.

[CL95] Isaac Chuang et Raymond Laflamme. Quantum error correction by coding. Préparution, octobre 1995. Archivé à quant-ph/951003.

[CLSZ95] Isaac Chuang, Raymond Laflamme, Peter W. Shor et Wojciech H. Zurek. Quantum computers, factoring and decoherence. *Science*, vol. 270, pp. 1635 – 1637, 1995. Archivé à quant-ph/9503007.

[CRSS96] Robert Calderbank, Eric M. Rains, Peter W. Shor et Neil Sloane. Quantum error correction via codes over GF(4). Préparution, septembre 1996. Archivé à quant-ph/9608006. Soumis à *IEEE Transactions on Information Theory.*

[CRSS97] Robert Calderbank, Eric M. Rains, Peter W. Shor et Neil Sloane. Quantum error correction and orthogonal geometry. *Phys. Rev. Lett.*, vol. 78, no 3, pp. 405 – 408, 1997. Archivé à quant-ph/9605005.

[CS96] Robert Calderbank et Peter W. Shor. Good quantum error correcting codes exist. *Phys. Rev. A*, vol. 54, no 2, pp. 1098 – 1105, août 1996.

[Eke91] Artur Ekert. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.*, vol. 67, no 6, pp. 661 – 663, août 1991.

[GB99] Markus Grassl et Thomas Beth. Quantum BCH codes. Préparution, octobre 1999. Archivé à quant-ph/9910060.

[GBP97] Markus Grassl, Thomas Beth et Thomas Pellizzari. Codes for the quatum erasure channel. Préparution, janvier 1997. Archivé à quant-ph/9610042 v2.

[GGB99] Markus Grassl, Willi Geiselmann et Thomas Beth. Quantum Reed-Solomon codes. Préparution, octobre 1999. Archivé à quant-ph/9910059.

[Got96] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys. Rev. A*, vol. 54, pp. 1862 – 1868, 1996.

[Got97] Daniel Gottesman. *Stabilizer codes and quantum error correction.* PhD thesis, California Institute of Technology, 1997. Archivé à quant-ph/9608006.

[Got98] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, vol. 57, no 1, pp. 127 – 137, janvier 1998.

[Gri90] Joseph Grifone. *Algèbre linéaire.* Cépaduès-Éditions, 1990.

[Gro97] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, vol. 79, pp. 325 – 328, 1997.

[Gru99] Jozef Gruska. *Quantum Computing.* McGraw Hill, 1999.

[HMP99] Richard J. Hughes, George L. Morgan et Glen Peterson. Practical quantum key distribution over 48-km optical fiber network. Rapport technique, avril 1999. Archivé à quant-ph/9904038.

[HR96] Serge Haroche et Jean-Michel Raymond. Quantum Computing : Dream or nightmare. *Physics Today*, vol. 49, pp. 51 – 52, août 1996.

[KL96a] Emanuel H. Knill et Raymond Laflamme. Concatenated quantum codes. Préparution, août 1996. Archivé à quant-ph/9608012.

[KL96b] Emanuel H. Knill et Raymond Laflamme. Theory of quantum error correcting codes. *Phys. Rev. A*, vol. 55, no 2, pp. 900 – 911, février 1996.

[Kra86] Evangelos Kranakis. *Primality and Cryptography.* Willey-Teubner, 1986.

[Lan95] Rolf Landauer. Is quantum mechanically coherent computation useful? *Proceedings of the Drexel-4 Symposium on Quantum Nonintegrability*, vol. 44, 1995.

[LC99] Hoi-Kwong Lo et Hoi Fung Chau. Unconditional security of quantum key distribution over arbitrarily distances. *Science*, vol. 283, pp. 2050 – 2056, 1999. Archivé à quant-ph/9803006.

[LCA00] Hoi-Kwong Lo, Hoi Fung Chau et M. Ardehali. Efficient quantum key distribution scheme and proof of its unconditional security. Préparution, novembre 2000. Archivé à quant-ph/0011056.

[LMPZ96] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz et Wojciech H. Zurek. Perfect quantum error correction code. *Phys. Rev. Lett.*, vol. 78, p. 405, 1996.

[LN86] Rudolf Lidl et Harald Niederreiter. *Introduction to Finite Fields and their Applications.* Cambridge University Press, 1986.

[Lo99] Hoi-Kwong Lo. Will quantum cryptography ever become a successful technology in the marketplace. Préparution, avril 1999. Archivé à quant-ph/9904091.

[LVZS99] Debbie Leung, Lieven Vandersypen, Xinhan Zhou et Mark Sherwood. Experimental realization of a two qubit phase damping quantum code. Préparution, mai 1999. Archivé à quant-ph/9811068 v2.

[May98] Dominic Mayers. Unconditional security in quantum cryptography. À être publié dans *Journal of the ACM.* Archivé à quant-ph/9802025.

[May00] Dominic Mayers. On Shor's and Preskill's security proof for the BB84 protocol. À être publié.

[McE87] Robert J. McEliece. *Finite Fields for Computer Scientists and Engineers.* Kluwer Academic Publishers, 1987.

[MS77] Jessie MacWilliams et Neil Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1977.

[MY98] Dominic Mayers et Andrew C.-C. Yao. Quantum cryptography with imperfect apparatus. Rapport technique, 1998. Archivé à quant-ph/9809039.

[MZG95] Antoine Muller, Hugo Zbinden et Nicolas Gisin. Underwater quantum coding. *Nature*, vol. 378, 1995.

[NC00] Michael A. Nielsen et Isaac Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[Per93] Asher Peres. *Quantum Theory : Concepts and Methods.* Kluwer academic Press, 1993.

[Pre97] John Preskill. Fault-tolerant quantum computation. Préparution, décembre 1997. Archivé à quant-ph/9712048.

[PVK96] Martin B. Plenio, Vlatko Vedral et Peter L. Knight. Quantum error correction in the presence of spontaneous emission. Préparution, 1996. Archivé à quant-ph/9603022.

[Rom92] Steven Roman. *Coding and Information Theory.* Springer-Verlag, 1992.

[Sha79] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, vol. 22, no 11, pp. 612 – 613, novembre 1979.

[Sho95] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, vol. 52, pp. 2493 – 2496, 1995.

[Sho96] Peter W. Shor. Fault-tolerant quantum computation. *Proceedings of the Symposium on the Foundations of Computer Science*, 1996. Archivé à quant-ph/9605011.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal of Computation*, vol. 26, no 5, pp. 1484 – 1509, octobre 1997.

[Smi] Adam Smith. Communication personnelle.

[SP00] Peter W. Shor et John Preskill. Simple proof of security of the BB84 quantum key distribution scheme. *Phys. Rev. Lett.*, vol. 85, pp. 441 – 444, 2000. Archivé à quant-ph/0003004 v2.

[Ste96a] Andrew M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, vol. 77, no 5, pp. 793 – 797, 1996.

[Ste96b] Andrew M. Steane. Multiple particle interference and quantum error correction. *Proc. Roy. Soc. Lond. A*, vol. 452, p. 2551, 1996.

[Ste96c] Andrew M. Steane. Quantum Reed-Muller codes. Préparution, août 1996. Archivé à quant-ph/9608026 v2.

[Ste97] Andrew M. Steane. Space, time, parallelism and noise requirements for reliable quantum computing. Préparution, août 1997. Archivé à 9708021.

[Ste98] Andrew M. Steane. Enlargement of Calderbank-Shor-Steane quantum codes. Préparution, 1998. Archivé à quant-ph/9802061.

[Sti95] Douglas R. Stinson. *Cryptography, Theory and Practice.* CRC Press, 1995.

[Unr94] William G. Unruh. Decoherence and quantum computers : a problem. *Proceedings of the Workshop on Physics of Computation : PhysComp 94*, 1994.

[WC81] Mark N. Wegman et Larry Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, vol. 22, pp. 265 – 279, 1981.

[Wie83] Stephen Wiesner. Conjugate coding. *Sigact News*, vol. 15, no 1, 1983. Rédigé vers 1970.

[WZ82] William K. Wootters et Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, vol. 299, no 802, 1982.

Annexe A

Outils utiles

Nous présentons dans cette annexe quelques notions générales utilisées dans le mémoire.

A.1 Notions d'algèbre

Nous verrons maintenant la notion de groupe, d'anneau et de corps. Pour un traitement plus détaillé, consultez [McE87, LN86].

Définition A.1 Un *groupe* est un ensemble G muni d'une opération binaire interne $*$ telle que, pour a, b et c dans G,

$$\begin{array}{lll} 1) & a*(b*c) = (a*b)*c & \text{(associativité)} \\ 2) & \exists_{e\in G}\ e*a = a*e = a & \text{(élément neutre)} \\ 3) & \exists_{a'\in G}\ a*a' = a'*a = e & \text{(élément inverse)} \end{array}$$

On note le groupe par $(G,*)$. De plus, le groupe est *commutatif* si pour tout $a, b \in G$, on a que $a*b = b*a$. ◇

Exemple A.2 $(\mathbb{Z},+)$, $(\mathbb{R},+)$, $(\mathbb{C},+)$, $(\mathbb{R}-\{0\},\cdot)$ et $(\mathbb{C}-\{0\},\cdot)$ sont des groupes. ◇

Définition A.3 Un *anneau* est un ensemble A muni de deux opérations binaires internes $+$ et $\cdot$ telles que, pour a, b et c dans A,

1) $(A, +)$ est un groupe commutatif.
2) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (associativité)
3) $a \cdot (b + c) = a \cdot b + a \cdot c$ (distributivité à gauche)
3) $(a + b) \cdot c = a \cdot c + b \cdot c$ (distributivité à droite)

On note l'anneau par $(A, +, \cdot)$. ◇

Exemple A.4 $(\mathbb{Z}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$, $(\mathbb{C}, +, \cdot)$, $(\mathbb{Z}_n, \oplus, \odot)$ sont des anneaux (où $\oplus$ et $\odot$ sont l'addition et la multiplication modulo n respectivement). ◇

Définition A.5 Un *corps* est un ensemble K muni de deux opérations binaires internes $+$ et $\cdot$ telles que, pour a, b et c dans K,

1) $(K, +, \cdot)$ est un anneau.
2) $(K - \{0\}, \cdot)$ est un groupe, où 0 est l'élément neutre du groupe $(K, +)$.

On note le corps par $(K, +, \cdot)$. ◇

Exemple A.6 $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$, $(\mathbb{C}, +, \cdot)$, $(\mathbb{Z}_p, \oplus, \odot)$ sont des corps (où p est un nombre premier). ◇

Remarque A.7 On note par $\mathbb{F}_q$ un corps quelconque à q éléments. On peut montrer qu'il existe des corps de q éléments pour tout $q = p^n$ pour un p premier et $n \in \mathbb{N}$. ◇

A.2 Notions de probabilités et de théorie de l'information

Dans cette section, on considère que X et Y sont deux variables aléatoires qui peuvent prendre les valeurs $x_1, \ldots, x_n$ et $y_1, \ldots, y_m$ respectivement. Pour $1 \leq i \leq n$ et $1 \leq j \leq m$, on note

$$\begin{aligned} p_i &= \text{prob}(X = x_i), \\ q_i &= \text{prob}(Y = y_i), \text{ et} \\ r_{ij} &= \text{prob}(X = x_i \,|\, Y = y_j). \end{aligned}$$

Définition A.8 L'*entropie* de X est

$$\mathbf{H}(X) = -\sum_{i=1}^{n} p_i \lg p_i \tag{A.1}$$

On abuse parfois de la notation en définissant l'entropie d'une valeur réele $0 \leq p \leq 1$ par

$$\mathbf{H}(X) = -p \lg p - (1-p) \lg(1-p). \tag{A.2}$$

◇

Définition A.9 L'*entropie conditionelle* de X par rapport à Y est

$$\mathbf{H}(X \mid Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} q_j r_{ij} \lg r_{ij} \tag{A.3}$$

◇

Définition A.10 L'*information mutuelle* entre X et Y est

$$\mathbf{I}(X\,;\,Y) = \mathbf{H}(X) - \mathbf{H}(X \mid Y) \tag{A.4}$$

◇

Théorème A.11 (Loi des grands nombres de Bernshtein) Soit $X_1, \ldots, X_n$ des variables aléatoires de Bernouilli indépendantes telles que $\text{prob}(X_i = 1) = p$ pour $1 \leq i \leq n$. Alors, pour tout $0 < \varepsilon \leq p(1-p)$, on a que

$$\text{prob}\left(\left| \sum_{i=1}^{n} \frac{X_i}{n} - p \right| \geq \varepsilon \right) \leq 2e^{-\frac{n\varepsilon^2}{4p(1-p)}} \leq 2e^{-n\varepsilon^2} \tag{A.5}$$

◇