Informatique quantique IFT6155

La sécurité inconditionnel de QKD

Motivation

Nous allons dans ce chapitre démontrer la sécurité du protocole d'échange de clefs inventé par Gilles Brassard et Charles Bennett.

La sécurité semble découler facilement du principe d'insertiture. Tout gain d'information s'obtient en perturbant un système. Il a fallu attendre plusieurs années avant d'obtenir une preuve formelle de sécurité. La première preuve à été obtenur par Dominic Mayers en 1998. Cette preuve est technique et difficile.

Il existe de nombreux modèles pour QKD. Premièrement, plusieurs protocoles d'échange de clefs existent. Différentes hypothèses peuvent être faites au sujet des participants. Par exemple, est-ce que les appareils d'Alice et Bob sont parfaits. On t'il accès a un ordinateur quantique. Sont-ils capables (ou non) de produire des photons isolés. Il existe aussi plusieurs hypothèses sur la puissance de l'espion. Tout cela mis ensemble donne une multitude de scénarios et il existe donc une quantité impressionnante de publication traitant de la sécurité de la cryptographie quantique.

Nous allons ici voir la preuve de sécurité présentée par John Preskill et Peter Shor en 2000.

Protocole EPR

$$|\Psi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle) \quad |\Phi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$$

Si Alice et Bob partagent l'état, $|\Phi^{+}\rangle^{\otimes n}$ ils peuvent obtenir une clef secrète inconditionnellement sécuritaire en mesurant chaque paire dans la base de calcul.

Plus formellement.

Théorème: (Lo et Chau) Si Alice et Bob partagent un état ayant fidelité $1 - 2^{-s}$ avec $|\Phi^{+}\rangle^{\otimes n}$ l'incertitude de Eve au sujet de la clef est au moins $n - 2^{-c} + 2^{O(-2s)}$ ou $c = s - \log(2m + s + 1/\log_e(2))$.

Famille de code CSS

Les codes CSS sont construits à partir de deux codes linéaires C_1 et C_2 tel que C_1 et $C_2^{\perp}$ corrige t erreur et $C_2 \subset C_1$.

Les mots de code de Q sont associé aux mots de code de C_1

$$v \leftarrow \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v \oplus w\rangle.$$

On remarque que si $v_1 \oplus v_2 \in C_2$ alors les mots de code associé à v_1 et v_2 sont identique.

Il est clair que si on choisit une transformation unitaire et qu'on l'applique à tous les mots de code on obtient un code absolument équivalent.

Nous allons a partir de Q définir une famille de code equivalent $Q_{x,z}$. Les mots de code de $Q_{x,z}$ associé au mot de code $v \in C_1$ sont:

$$v \leftarrow \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{z \cdot w} |v \oplus w \oplus x\rangle.$$

Purification et code CSS

Pour corriger un état encodé par un code $Q_{x,z}$ codant m qubits dans n qubits et corigant t erreur il suffit de calculer le syndrome dans la base σ_x et dans la base σ_z. Si les syndromes sont zéro alors il n'y a pas d'erreur et sinon, on corrige, de façon appropriée.

Si Alice et Bob partagent des paires $|\Phi^+\rangle$ ils peuvent individuellement mesurer les deux syndromes. Comme chacun d'eux possède, de son point de vue, un état complètement mélangé, ils obtiendront des syndromes aléatoires. Il n'est pas difficile de montrer que si les paires étais parfaites ils obtiendraient exactement les mêmes syndromes. Plus précisément le XOR de leurs syndromes leurs permettent de corriger t erreur sur les paires qu'il partagent. Une fois les syndromes mesurés et les erreurs corrigées ils obtiennent m état $|\Phi^+\rangle$ encoder avec le code $Q_{x,z}$.

Protocole de Lo-Chau

- Alice crée $|\Phi^+\rangle^{\otimes 2n}$.
- Alice choisit $b \in_R \{0,1\}^{2n}$ et applique $I \otimes H^{b_i} |\Phi^+\rangle_i$.
- Alice transmet le deuxième qubit de chaque paire à Bob.
- Bob annonce qu'il à recu les $2n$ qubit.
- Alice choisit n positionne, et transmet b_i à Bob pour ces positions.
- Bob applique H sur son qubit pour les positions ou $b_i = 1$.
- Alice et Bob mesurent les positions sélectionnées dans la base de calculs et ils vérifient qu'il obtient le même résultat. S'il y a trop d'erreurs, il avorte le protocole.
- Alice et Bob purifient leurs paires a l'aide d'un code CSS.
- Alice et Bob mesurent les paires purifiées pour obtenir une clef secrète.

Sécurité du protocole de Lo-Chau

Pour montrer la sécurité du protocole précédent il suffir de montrer que si le taux d'erreur dans les positions mesuré est suffisaient faible alors le protocole de purification utilisant le code CSS donnera un état dont la fidélité à $|\Phi^+\rangle^{\otimes m}$ est très bonne.

Intuitivement si une petite perturbation est faite aux paires originales alors le purification fonctionnera et donne des bonnes paires. Par contre si une grosse perturbation est faite à l'état alors le taux d'erreur sera trop grand et le protocole avortera.

Nous allons maintenant décrire un protocole tous à fait équivalent. Les mesures d'Alice sur ses paires peuvent en fait être exécutées avant même que les paires soient transmises sans changer les statistiques du protocole. On obtient le protocole suivant. Les syndrome x et z obtenue par Alice dicte l'état transmit.

Protocole CSS

- Alice choisit de façon aléatoire et uniforme n bit de vérification, une clef k de m bit, une chaine b de $2n$ bit, une chaine x de n bit et une chaine z de n bit.
- Alice encode k en utilisant un code CSS $Q_{x,z}$.
- Alice choisit n position parmis $2n$ et place les bits de vérification à ces positions et les bits de k encoder dans les autres positions.
- Alice applique H aux positions ou $b_i = 1$.
- Alice transmet les $2n$ qubits à Bob et Bob confirme les avoir recu.
- Alice anonce b, x, z ainsi que la position et la valeur des bits de vérification.
- Bob applique H aux position ou $b_i = 1$, mesure les bits de vérification et vérifie qu'il ont la bonne valeur. Si ça n'est pas le cas, il avorte le protocole.
- Bob décode (et corrige) la clef secrète k en utilisant le code $Q_{x,z}$ et confirme à Alice que le protocole a réussi.

Nous allons maintenant voir que le protocole CSS est parfaitement équivalent à un protocole standard BB84 avec correction d'erreur utilisant C_1 et amplification de confidentialité utilisant C_2.

Premièrement, on remarque que z, l'information permettant de corriger les erreurs de phase, n'est pas nécessaire à Bob. Bob finit par mesurer dans la base de calcul.

Soit $k' \in C_1$ le mot de code choisi par Alice. Elle transmet donc

$$\frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{z \cdot w} |k' \oplus w \oplus x\rangle.$$

Comme z reste inconnu de Bob de son point de vue l'état est

$$\frac{1}{|C_2|} \sum_{w \in C_2} |k' \oplus w \oplus x\rangle \langle k' \oplus w \oplus x|$$

Dans le protocole Alice transmet $k' \oplus w \oplus x$ et x uniformément distribué tel que $(k' \oplus w \oplus x) \oplus x = k' \oplus w \in C_1$. Une fois que Bob a reçu $k' \oplus w \oplus x \oplus \epsilon$ il XOR x et obtient $k' \oplus w \oplus \epsilon$. Bob utilisa la capacité de correction de C_1 pour obtenir le mot de code $k' \oplus w \in C_1$. La clef secrète est maintenant le coset de $k' \oplus w$ dans C_2.

Prenons $v = k' \oplus w \oplus x$ et $u = k' \oplus w$. On à que v est uniformément distribué et u est un mot de code aléatoire de C_1. On obtient le même protocole si Alice transmet $|v\rangle$ puis $u + v$ comme information pour corriger les erreurs.

Protocole CSS

- Alice et Bob sont entendu sur deux code C_1 et C_2 avec les bonnes propriétés.
- Alice choisit de façon aléatoire et uniforme a et b deux chaînes de $2n$ bits.
- Alice applique H aux positions ou $b_i = 1$ sur $|a\rangle$.
- Alice transmet les $2n$ subits à Bob et Bob confirme les avoir reçu.
- Alice anonce b ainsi que a_i pour $b_i = 1$.
- Bob applique H aux position ou $b_i = 1$, mesure les bit de vérification et vérifie qu'il ont la bonne valeur. Si ça n'est pas le cas, il avorte le protocole.
- Soit v la chaîne correspondante aux positions restantes. Alice choisit uniformément $u \in C_1$ et transmet à Bob $u \oplus v$.
- Bob utilise les propriétés de C_1 pour récupéré v.
- Alice et Bob partage k le coset de $u + C_2$.

Nous avons maintenant un protocole proche de BB84.

Il ne nous reste qu'à modifier le protocole, sans changer sa sécurité, pour que Bob n'ait pas besoin d'une mémoire quantique.

Pour ce faire, Bob n'attendra pas de conaitre b et mesurera dans une base aléatoire. Cela implique qu'il fera le bon choix la moitié du temps.

En commençant avec $(4+\sigma)n$ il restera avec bonne probabilité $2n$ positions ou le bon choix aura été fait.

On obtient le protocole suivant.

Protocole BB84

- Alice choisit de façon uniformément $(4+\sigma)n$ bits aléatoires et $(4+\sigma)n$ bases aléatoires (base de calcul ou H).
- Alice transmet les bit choisit dans la base correspondante.
- Bob mesure les qubits dans une base alléatoire (base de calcul ou H).
- Alice et Bob annoncent les bases utilisées. Ils rejettent les positions ou une base différente a été utilisée. S'il reste moins de $2n$ position, le protocole avorte sinon ils continuent avec les $2n$ premières positions qui correspondent.
- Alice choisi et annonce n position ainsi que la valeur du bit à ces positions.
- Bob vérifie que le taux d'erreur est acceptable et avorte le protocole sinon.
- Bob vérifie que le taux d'erreur est acceptable et avorte le protocole sinon.
- Soit v la chaîne des bits aux positions restante et u un mot de code aléatoire de C_1, Alice anonce $u \oplus v$.
- Bob utilise C_1 pour récupérer v.
- Alice et Bob utilisent le coset de $u + C_2$ comme clef secrète.