Protocole de mise en gage de bit relativiste

Rémi Bricout

Rapport de Master 2
Protocole de mise en gage de bit relativiste

Rémi Bricout, encadré par André Chailloux
Inria Paris, équipe SEcurité, CRyptoglogie Et Transmissions

Le contexte général

Actuellement, la sécurité de la plupart des protocoles cryptographiques repose sur des hypothèses calculatoires (factorisation pour RSA, problèmes NP-complets pour les réseaux, logarithme discret pour les courbes elliptiques...). Si ces problèmes ne peuvent pas aujourd'hui être résolus efficacement, il n'existe pas non plus de garantie concernant leur difficulté. Face aux progrès algorithmiques et à l'arrivée potentielle d'un ordinateur quantique, plusieurs pistes sont étudiées. L'une d'elles consiste à s'affranchir des hypothèses calculatoires et à les remplacer par des lois physiques. L'objectif de cette approche est de proposer des protocoles sûrs contre tout attaquant, classique ou quantique, même s'il disposait d'une puissance de calcul infinie. Lunghi *et al.* ont proposé un protocole de mise en gage de bit pour lequel la seule condition de sécurité contre un attaquant classique est l'hypothèse (nécessaire à la cohérence des modèles physiques actuels) selon laquelle il n'est pas possible de transmettre de l'information à une vitesse supérieure à celle de la lumière [LKB+15]. On parle alors de protocole relativiste. La borne de sécurité de leur protocole a récemment été améliorée (simultanément [FF16] et [CCL15]).

Le problème étudié

Le protocole de mise en gage de bit proposé dans [LKB+15] est prouvé sûr contre n'importe quel attaquant classique pour de larges plages de tailles des paramètres (qui sont tout à fait satisfaisants pour une application pratique). Cependant, on ne savait pas si cette borne pouvait encore être améliorée ou non. [PPP16] affirmait avoir trouvé une meilleure borne, mais étaient peu convaincants. Il n'existait pas non plus de résultat concernant sa sécurité face à un attaquant disposant d'une puissance de calcul quantique (c'est-à-dire ni attaque, ni preuve de sécurité). La mise en gage de bit étant une primitive élémentaire qui peut servir de brique dans l'élaboration de constructions plus complexes, si ce protocole pouvait être sûr également face à toute attaque quantique, cela aiderait grandement à la construction de schémas cryptographiques très fiables. J'ai donc cherché à obtenir une borne quelconque (preuve de sécurité ou attaque) concernant la fiabilité du protocole de [LKB+15] contre un attaquant qui dispose d'un ordinateur quantique.

La contribution proposée

Pour ce faire, j'ai soigneusement étudié les articles donnant des preuves de la sécurité du protocole contre un attaquant classique (notamment [LKB+15], [CCL15]) en essayant de les adapter au cas d'un attaquant quantique. Un jeu, $CHSH$, apparaissant naturellement lors de l'étude du protocole, j'ai aussi travaillé sur des démonstrations de la valeur de ce jeu ([BS15] et [PW12]). En particulier, [LKB+15] en utilise une généralisation, appelée "nombre sur le front". J'ai exhibé une stratégie quantique pour ce jeu dans le cas binaire, qui est très largement meilleure que ce qui est possible classiquement. J'ai choisi de ne présenter ce résultat en annexe pour deux raisons. D'une part, il s'agit d'un résultat qui, bien que très joli, n'est que d'un intérêt limité (il sert à montrer que je n'avait aucune chance d'adapter la démonstration de [LKB+15] au cas quantique). D'autre part, sa description ainsi que la démonstration de sa validité nécessitent l'utilisation d'états quantiques, et les calculs sont assez simples, mais requièrent des bases d'informatique quantique (présentées aussi en annexe).

Malheureusement, je n'ai pas pu trouver de résultat direct concernant le principal problème, à savoir le comportement du protocole de [LKB+15] contre un adversaire quantique. En revanche, j'ai proposé la première attaque (classique) contre le protocole, qui montre que la borne démontrée par [CCL15] et [FF16] ne peut essentiellement plus être améliorée. Cela clos donc la question sur la sécurité contre un attaquant ne disposant que d'un ordinateur classique, aussi puissant soit-il, en donnant l'ordre le grandeur du temps durant lequel de protocole est sûr (ou de façon équivalente, la taille des messages nécessaire pour garantir la sécurité pour une durée donnée). Cette attaque pouvant évidemment être réalisée par un attaquant quantique, cela donne également une borne supérieure (très mauvaise) de la sécurité face à un attaquant quantique.

Les arguments en faveur de sa validité

L'attaque que j'ai trouvée a fait l'objet d'un papier, [BC16]. De plus il a permis de mettre fin à un débat concernant un autre papier, [PPP16], qui, après avoir montré plusieurs résultats intéressants, affirmait avoir encore amélioré la borne de sécurité du protocole de mise en gage de bit. Cependant, une démonstration assez délicate était contestée. Mon attaque fournit un contre-exemple qui prouve que leur borne était fausse. Après un échange (cordial) de courriels, ils ont modifié leur papier pour enlever ce résultat.

Le bilan et les perspectives

Le résultat de mon attaque permet de connaitre maintenant avec un encadrement assez précis la sécurité du protocole de mise en gage de bit de [LKB+15] dans le cas d'une attaque classique, alors qu'auparavant seule une borne inférieure était connue. De plus la valeur donnée par mon attaque étant du même ordre de grandeur que les bornes inférieures connues, cela montre que celles-ci, ainsi que l'attaque, sont très proches de la valeur réelle de la sécurité du protocole.

Cependant, si cela clos la question de la sécurité contre un attaquant classique, on ne sait pas ce qu'il en est contre un attaquant disposant d'une puissance de calcul quantique. L'attaque que j'ai proposée peut évidemment être utilisée par un attaquant quantique. Cependant, la brique élémentaire de l'attaque, le jeu $CHSH$, ne profite que très peu de l'utilisation de ressources quantiques. On ne sait donc pas actuellement s'il existe une meilleure attaque (donc la structure serait complètement différente), ou si le protocole est asymptotiquement aussi sûr contre un attaquant quantique qu'il ne l'est contre un attaquant classique (à une constante près).

1 Présentation du problème

Dans cette section, j'introduirai le protocole de engagement de bit étudié, ainsi que le jeu $CHSH$ et sa généralisation $CHSH_Q$, indispensables à l'étude du protocole précédemment mentionné.

1.1 Mise en gage de bit

La mise en gage de bit est une primitive cryptographique faisant intervenir deux protagonistes, qu'on nommera Alice et Bob, qui ne se font pas confiance. Dans un premier temps, la phase d'*engagement*, Alice et Bob ont une communication au cours de laquelle Alice met en gage un bit d'information d. À la fin de cette phase, Bob ne doit pas connaitre d'information quant à la valeur de d. C'est-à-dire que s'il devait parier sur la valeur de d à ce moment, il ne gagnerait qu'avec probabilité $\frac{1}{2}$. Ensuite, après une durée arbitraire, dans la phase de *dévoilement*, Alice et Bob ont de nouveau un échange, dans lequel Alice indique la valeur d qu'elle avait mise en gage. Bob doit pouvoir s'assurer qu'Alice n'a pas changé d'avis entre les deux phases. Cela signifie que si elle a elle a mis en gage d, elle ne peut pas réussir à dévoiler $1-d$, ou plus généralement, pour une stratégie d'engagement quelconque, les chances de dévoiler 0 et 1 ne peuvent pas être grandes toutes les deux.

Idéalement, on aimerait se représenter cette primitive de la façon suivante : lors de la phase d'*engagement*, Alice choisit 0 ou 1, l'écrit sur un papier, l'enferme dans un coffre, et envoie ce coffre à Bob. À ce moment, Bob, bien qu'ayant le coffre en se possession, ne peut obtenir aucune information concernant d car il ne sait pas ouvrir le coffre. Plus tard, lors de la phase de *dévoilement*, Alice envoie à Bob la clé du coffre. De cette façon, Bob peut découvrir le choix d'Alice, sans que celle-ci ne puisse agir car elle n'a plus accès au coffre.

Malheureusement, il s'agit du cas idéal et en pratique il est difficile de garantir qu'aucun des deux participants ne peut tricher. Par exemple, Alice peut essayer de tricher, c'est-à-dire choisir d après la phase d'*engagement*. Pour cela, elle utilise une stratégie $(S_e, S_r(d))$ où S_e est la stratégie pour la phase d'engagement et S_r celle de la phase de *dévoilement*. Alice ne déterminant la valeur de d qui l'arrange seulement entre les deux phases, S_e ne dépend pas de d.

Définition *Protocole contraignant pour Alice*

- *Pour une stratégie $S^A = (S_e, S_r)$, la probabilité de succès d'Alice est*

$$P_A(S^A) = \frac{1}{2}\,\mathbb{P}[\textit{Alice parvient à révéler } 0|(S_e, S_r(0))] + \frac{1}{2}\,\mathbb{P}[\textit{Alice parvient à révéler } 1|(S_e, S_r(1))]$$

- *la probabilité optimale de succès d'Alice est $P_A = \max_{S^A} P_A(S^A)$;*
- *un protocole est ε-contraignant s'il assure $P_A \leqslant \frac{1}{2} + \varepsilon$;*
- *il est parfaitement contraignant s'il est 0-contraignant.*

P_A est donc la probabilité pour Alice de réussir à révéler une variable aléatoire de Bernoulli de paramètre $\frac{1}{2}$ à laquelle elle n'accède qu'entre les deux phases. Cette définition, qui provient de la littérature, est peu intuitive : ce qui est important pour la valeur de gain d'une stratégie est l'écart à $\frac{1}{2}$. En effet, une stratégie honnête permet de "tricher" avec une probabilité de succès de $\frac{1}{2}$.

Si Alice n'est pas digne de confiance, elle n'est pas la seule, Bob peut également vouloir tricher et chercher à deviner la valeur d. On notera S^B sa stratégie.

Définition *Protocole occultant pour Bob*

- *Pour une stratégie S^B, la probabilité de succès de Bob est*

$$P_B(S^B) = \mathbb{P}[\textit{Bob parvient à deviner } d|S^B]$$

- *la probabilité optimale de succès de Bob est $P_B = \max_{S^B} P_B(S^B)$;*
- *un protocole est ε-occultant s'il assure $P_B \leqslant \frac{1}{2} + \varepsilon$;*
- *il est parfaitement occultant s'il est 0-occultant.*

1.2 Le protocole étudié

J'ai présenté ce qu'était la mise en gage de bit en général. Ce sur quoi j'ai travaillé est un protocole possible pour réaliser cette primitive. La particularité de ce protocole est de ne reposer sur aucune hypothèse calculatoire.

1.2.1 Protocole à un tour

Le protocole de mise en gage de bit relativiste a été introduit par [CSST11]. Pour pouvoir exploiter le fait que la vitesse de déplacement de l'information est bornée, Alice et Bob disposent chacun de deux agents ($\mathcal{A}_1$ et $\mathcal{A}_2$ pour Alice, et $\mathcal{B}_1$ et $\mathcal{B}_2$ pour Bob). $\mathcal{A}_1$ et $\mathcal{B}_1$ sont situés à une position 1, tandis que $\mathcal{A}_2$ et $\mathcal{B}_2$ sont en 2. Le protocole dans lequel Alice engage de bit d est donc le suivant :

- phase de *préparation* : les agents d'Alice (resp. de Bob) échangent un secret $a \in \mathbb{F}_Q$ (resp. $x \in \mathbb{F}_Q$)
- phase d'*engagement* : $\mathcal{B}_1$ envoie x à $\mathcal{A}_1$, qui lui répond $y := a + d \cdot x$
- phase d'*entretien* : tous les agents attendent une durée τ
- phase de *dévoilement* : $\mathcal{A}_2$ envoie d et a à $\mathcal{B}_2$, qui peut alors vérifier que $y = a + d \cdot x$

Ce protocole est parfaitement occultant, c'est-à-dire que Bob, qui ne connait que x et y, ne peut en tirer aucune information concernant d. En effet, la dernière opération d'Alice avant d'envoyer y à Bob est l'ajout d'un élément aléatoire de $\mathbb{F}_Q$.

Supposons qu'Alice veuille tricher. Lors de la phase de *dévoilement*, $\mathcal{A}_2$ doit donc pouvoir fournir $(0, a_0)$ pour révéler 0, ou $(1, a_1)$ pour révéler 1. Les relations $y = a_0 + 0 \cdot x$ et $y = a_1 + 1 \cdot x$ doivent donc être vérifiées. On remarque alors que $x = a_0 - a_1$. Pour tricher, $\mathcal{A}_2$ doit donc connaitre ou deviner x. Or $\mathcal{A}_2$ ne peut avoir accès à x que lorsque l'information a eu le temps de se propager de $\mathcal{B}_1$ à $\mathcal{A}_2$. Ensuite le message de $\mathcal{A}_2$ doit se propager de $\mathcal{A}_2$ à $\mathcal{B}_2$. Par inégalité triangulaire, cela requiert au moins autant de temps que de transmettre un message de $\mathcal{B}_1$ à $\mathcal{B}_2$. Ainsi, si les deux agents de Bob sont situés à une distance D l'un de l'autre, et qu'il s'écoule un temps τ entre l'envoi de x et la réception de d, a, Bob vérifie que $\tau \geq \frac{D}{c}$ (Figure 1). Alice ne peut donc tricher qu'en devinant x, ce qui se produit avec probabilité $\frac{1}{Q}$. Ce protocole est donc $\frac{1}{2Q}$-contraignant. On remarque que Bob n'a pas à faire confiance à Alice quant à la position de ses agents : il lui suffit de connaitre la localisation des siens, en plus précisément la distance D qui les sépare.

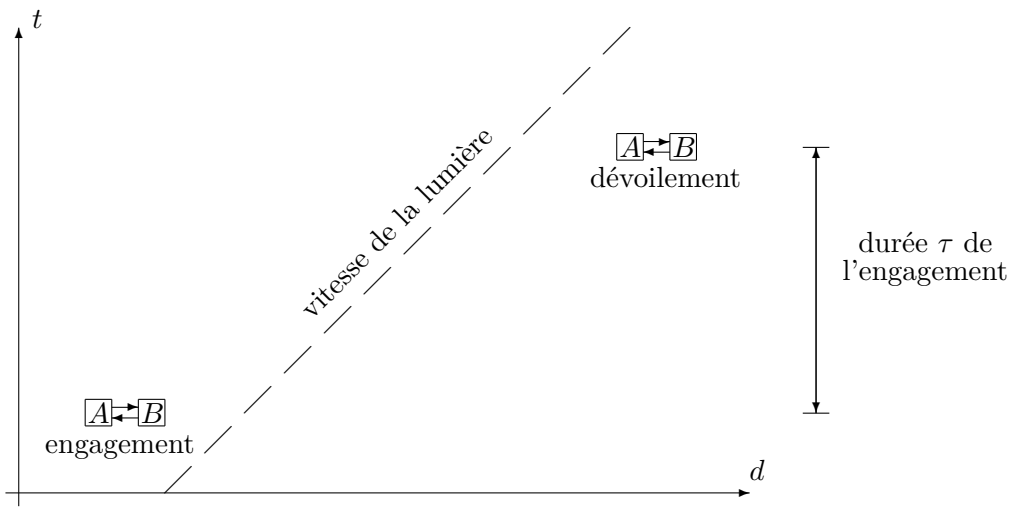

FIGURE 1 – La lumière n'a pas le temps d'aller du lieu 1 au lieu 2 sur la durée de l'engagement

1.2.2 Extension à plusieurs tours

Sur un tour, le protocole est très sûr (Bob ne peut pas tricher, et Alice seulement avec une probabilité infime). Cependant, son utilisation est délicate, car la durée de l'engagement est très faible. [LKB+15] propose une extension du protocole pour en allonger la durée. L'idée est qu'au lieu de révéler a lors de la phase de *dévoilement*, $\mathcal{A}_2$ va engager ce nombre a à l'aide d'un nouveau nombre x_2 envoyé par $\mathcal{B}_2$. La phase de *entretien* consiste donc en des aller-retours entre les lieux

1 et 2. Le protocole de déroule de la façon suivante :

- phase de *préparation* : $\mathcal{A}_1$ et $\mathcal{A}_2$ (resp. $\mathcal{B}_1$ et $\mathcal{B}_2$) échangent $a_1, ..., a_m$ (resp. $x_1, ..., x_m$) $\in \mathbb{F}_Q$
- phase d'*engagement* : $\mathcal{B}_1$ envoie x_1 à $\mathcal{A}_1$, qui lui répond $y_1 := a_1 + d \cdot x_1$
- phase d'*entretien* : à chaque tour k, où $2 \leqslant k < m$, $\mathcal{B}_{k \bmod 2}$ envoie x_k à $\mathcal{A}_{k \bmod 2}$, qui lui répond $y_k := a_k + a_{k-1} \cdot x_k$
- phase de *dévoilement* : $\mathcal{A}_{m \bmod 2}$ révèle d et $y_m := a_m$ à $\mathcal{B}_{m \bmod 2}$. Connaissant a_m, Bob peut alors dépiler tous les messages d'Alice en calculant successivement les a_k, et vérifier la cohérence d'Alice.

On notera $\mathcal{P}_m$ ce protocole lorsqu'il dure m tours.

Ici, le sécurité pour Bob vient du fait qu'à chaque tour k, $\mathcal{A}_{k \bmod 2}$ ne doit pas pouvoir accéder à x_{k-1}. Pour cela, il lui suffit de vérifier qu'il reçoit y_k avant que x_{k-1} n'ait eu le temps de parcourir la distance D séparant ses deux agents (Figure 2). On peut vérifier que ce protocole est à nouveau parfaitement occultant puisqu'Alice utilise un aléa neuf pour cacher chacun de ses messages. De plus [FF16] et [CCL15] ont démontré que ce protocole est ε-contraignant où $\varepsilon = \mathcal{O}(\frac{m}{\sqrt{Q}})$ dans le cas d'une Alice classique, ce qui signifie que pour que la probabilité qu'Alice puisse tricher soit négligeable, il suffit que le carré du nombre de tours soit petit devant la taille du corps $\mathbb{F}_Q$. Ce résultat est une amélioration considérable de la borne exhibée par [LKB+15]. Cependant, rien n'était connu dans les cas où m approche ou dépasse $\sqrt{Q}$.

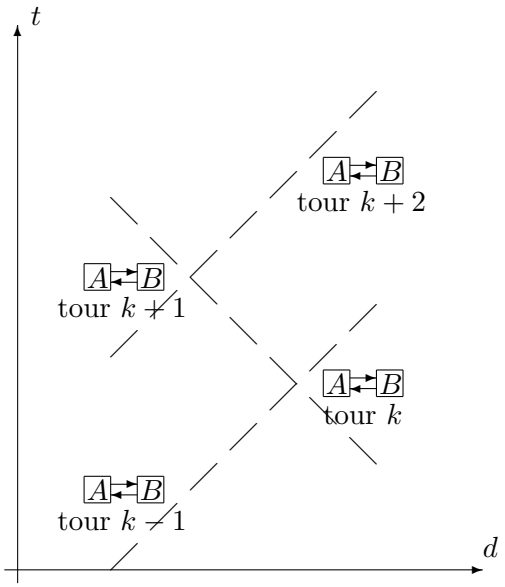

FIGURE 2 – Chaque tour doit se finir hors du cone du futur du précédent

1.3 Les jeux $CHSH$ et $CHSH_Q$

$CHSH$ (Clauser, Horne, Shimony, Holt) est un jeu de coopération entre deux joueurs qui ne peuvent pas communiquer entre eux (ils peuvent seulement s'accorder auparavant sur une stratégie). Chacun des joueurs reçoit un bit ($x \in \{0,1\}$ et $y \in \{0,1\}$) et ils doivent produire des réponses a et b tels que $a \oplus b = x \cdot y$. Classiquement, les joueurs peuvent gagner avec probabilité $\frac{3}{4}$. $CHSH_Q$ est une généralisation de $CHSH$ dans $\mathbb{F}_Q$: les entrées des joueurs sont $x, y \in \mathbb{F}_Q$ (uniformément distribués et indépendants), et ils doivent produire $a, b \in \mathbb{F}_Q$ tels que $a + b = x \cdot y$. Par la suite on notera $\omega(CHSH_Q)$ la valeur du jeu $CHSH$, c'est-à-dire la probabilité qu'ont les joueurs de gagner en suivant une stratégie optimale. Dans [BS15], il est montré que :

$$\omega(CHSH_Q) = \begin{cases} \Omega\left(\sqrt{\frac{1}{Q}}\right) & \text{si } Q = p^{2n} \\ \mathcal{O}(Q^{-\frac{1}{2}-\varepsilon_0}) & \text{si } Q = p^{2n-1} \end{cases}$$

où ε_0 est une constant strictement positive dont la valeur n'est pas connue précisément.

1.4 La variante $CHQH_Q^\gamma$

Pour étudier mon attaque dans le cas le plus général, j'ai introduit une nouvelle variante de $CHSH_Q$. Dans ce cas, les entrées des deux joueurs ne sont plus parfaitement uniformes, mais une entrée (ici 0) est plus probable que les autres (les entrées restent indépendantes). Plus précisément :

Définition *Dans le jeu $CHSH_Q^\gamma$, le premier joueur reçoit $x := 0$ avec probabilité γ, ou n'importe quel autre nombre $x \in \mathbb{F}_Q^*$ avec probabilité $\frac{1-\gamma}{Q-1}$. L'autre joueur reçoit y distribué de la même façon et indépendant de x. Ils doivent produire a et b tels que $a + b = x \cdot y$.*

Lemme 1 *Pour tout $\gamma \in [0, 1]$, $\omega(CHSH_Q^\gamma) \geqslant \omega(CHSH_Q)$.*

Preuve : Toute stratégie aléatoire peut se décomposer comme combinaison linéaire de stratégies déterministes. La probabilité de gagner en suivant une telle stratégie est alors donnée en considérant la combinaison linéaire des valeurs de ces même stratégies déterministes, avec les mêmes coefficients. Ainsi, on considèrera sans perte de généralité que, quand on choisit une stratégie optimale sans plus de contraintes, on la suppose de plus déterministe.

On considère une stratégie, déterministe en vertu de la remarque précédente, (s_1, s_2), optimale pour $CHSH_Q$, c'est-à-dire des fonctions $s_1, s_2 : \mathbb{F}_Q \to \mathbb{F}_Q$ telles que $\Pr_{x,y}[s_1(x) + s_2(y) = xy] = \omega(CHSH_Q)$, où la probabilité est calculée pour x et y indépendants et uniformément distribués. Cette stratégie étant déterministe, on définit $p_{x,y} := 1$ si $s_1(x) + s_2(y) = xy$ et 0 sinon. On a donc $\mathbb{E}_{xy} p_{xy} = \omega(CHSH_Q)$. Soit alors

$$Z_{u,v} := \gamma^2 p_{u,v} + \frac{\gamma(1-\gamma)}{Q-1}\left(\sum_{x\in\mathbb{F}_Q-\{u\}} p_{xv} + \sum_{y\in\mathbb{F}_Q-\{v\}} p_{uy}\right) + \frac{(1-\gamma)^2}{(Q-1)^2}\sum_{\substack{x\in\mathbb{F}_Q-\{u\}\\ y\in\mathbb{F}_Q-\{v\}}} p_{xy}.$$

$Z_{u,v}$ correspond à la probabilité de gagner $CHSH_Q$ si les entrées ont un biais γ vers (u, v). En particulier, $Z_{0,0}$ est la probabilité de gagner le jeu $CHSH_Q^\gamma$ si on utilise la stratégie S. On vérifie facilement que $\mathbb{E}_{u,v}[Z_{u,v}] = \omega(CHSH_Q)$. On peut donc fixer un couple (u, v) tel que $Z_{u,v} \geqslant \omega(CHSH_Q)$.

On considère alors la stratégie $S' = (s'_1, s'_2)$ où $s'_1(x) = s_1(x + u) - xw$ et $s'_2(y) = s_2(y + v) - yu - uv$. En utilisant S' on gagne sur l'entrée (x, y) exactement quand S gagne sur l'entrée $(x + u, y + v)$. En effet :

$$\begin{aligned} s'_1(x) + s'_2(y) = xy &\Leftrightarrow s_1(x+u) - xv + s_2(y+u) - yu - uv = xy \\ &\Leftrightarrow s_1(x+u) + s_2(y+v) = (x+u)(y+v) \end{aligned}$$

On définit alors de la même façon $p'_{xy} = 1$ si $s'_1(x) + s'_2(y) = x \cdot y$ et 0 dans le cas contraire. L'équivalence précédente se traduit donc par $p'_{x,y} = p_{(x+u),(y+v)}$. On définit aussi

$$Z'_{u,v} := \gamma^2 p'_{u,v} + \frac{\gamma(1-\gamma)}{Q-1}\left(\sum_{x\in\mathbb{F}_Q-\{u\}} p'_{xv} + \sum_{y\in\mathbb{F}_Q-\{v\}} p'_{uy}\right) + \frac{(1-\gamma)^2}{(Q-1)^2}\sum_{\substack{x\in\mathbb{F}_Q-\{u\}\\ y\in\mathbb{F}_Q-\{v\}}} p'_{xy}.$$

On remarque que $Z'_{0,0}$ est la probabilité de gagner $CHSH_Q^\gamma$ avec la stratégie S'. De plus, pour tous (x, y), on a $Z'_{x,y} = Z_{x+u,y+v}$. On peut donc conclure

$$\omega(CHSH_Q^\gamma) \geq Z'_{0,0} = Z_{u,v} \geq \omega(CHSH_Q),$$

ce qui montre le résultat annoncé.

□

2 Attaque du protocole relativiste de mise en gage de bit

Nous avons vu que la sécurité pour Bob repose sur le fait qu'à chaque tour k, l'agent actif d'Alice connait la valeur courante x_k, mais pas x_{k-1}. En revanche, cet agent connait x_{k-2} (puisque ce nombre à été révélé à ce même agent deux tours plus tôt), et peut connaitre x_{k-3} ainsi que toutes les valeurs antécédentes. Cependant, ce n'est pas aisé pour Alice de transmettre les x_k suffisamment rapidement pour avoir accès à x_{k-3}. En effet, cela nécessite d'envoyer de l'information à une vitesse supérieure à $\frac{c}{3}$.

Par ailleurs, on suppose qu'Alice ne sait pas quelle valeur d elle veut engager, elle ne fait son choix que plus tard. Le tour durant lequel elle fait son choix n'est pas connu et dépend essentiellement de l'utilisation du protocole, mais il peut être possible qu'Alice connaisse d dès le tour 2.

Dans un premier temps, on considèrera une attaque pour laquelle Alice est dans les meilleures conditions possibles. Il s'agit de la meilleure attaque connue face à laquelle Bob peut être confronté. Dans ce cas, on supposera qu'Alice peut avoir accès à tous les x_k, à l'exception de x_{k-1}. De plus, Alice sait dès le tour 2 si elle doit essayer de révéler $d = 0$ ou $d = 1$. Ces conditions seront affaiblies par la suite, pour pouvoir généraliser le cadre de l'attaque à des conditions plus réalistes.

2.1 Conditions parfaites pour Alice

Cette section est consacrée à la construction et à l'étude d'une attaque du protocole de mise en gage de bit relativiste dans lequel Alice est dans les meilleures conditions possibles. On rappelle que le protocole étant parfaitement occultant, Bob est dans l'incapacité totale de tricher, c'est pourquoi on s'intéresse uniquement à une Alice malhonnête.

Le principe de fonctionnement de l'attaque est le suivant. Tous les trois tours, les agents d'Alice peuvent jouer à un jeu $CHSH_Q$. S'ils gagnent ce jeu, ce qu'ils peuvent faire avec probabilité $\omega(CHSH_Q)$, il leur est alors facile de révéler la valeur d choisie. Si par contre ils ne parviennent pas à gagner, ce n'est aucunement grave : il leur suffit d'essayer de nouveau trois tours plus tard. Plus précisément, lors de chaque étape de trois tours, les deux derniers sont utilisée pour jouer au jeu $CHSH_Q$. Le premier tour, quant à lui, est ajouté pour donner à $\mathcal{A}_1$ et $\mathcal{A}_2$ le temps de déterminer s'ils ont gagné lors de l'étape précédente, ou s'ils n'ont pas gagné, de calcule un facteur correctif η. De cette façon, lors d'un protocole qui dure m tours, les agents d'Alice ont le temps de jouer environ $\frac{m}{3}$ jeux $CHSH_Q$. Puisque chacun est gagné avec probabilité $\omega(CHSH_Q)$ et qu'il leur suffit d'en gagner un, on comprend que la probabilité de parvenir à tricher augmente de façon exponentielle en la durée du protocole.

On suppose ici qu'Alice peut rapidement propager les x_k, et qu'elle détermine la valeur d à révéler dès le tour 2. Ainsi, une stratégie déterministe de triche S pour Alice est donc un m-uplet $(s_1, ..., s_m)$, où s_k est la stratégie adoptée par Alice au tour s_k. Par hypothèses, s_1 est fonction uniquement de x_1, puis s_k est une fonction de $d, x_1, ..., x_{k-2}, x_k$.

Lors du déroulement du protocole, on note y_k la valeur envoyée par Alice au tour k. À la fin du protocole, Bob dispose de d et d'un système de m équation à m inconnues (les a_k et d) :

- $y_1 = a_1 + d \cdot x_1$
- $y_2 = a_2 + a_1 \cdot x_2$
- $y_3 = a_3 + a_2 \cdot x_3$
- ...
- $y_{m-1} = a_{m-1} + a_{m-2} \cdot x_{m-1}$
- $y_m = a_{m-1}$

Bob peut donc résoudre le système en partant de la fin. Il obtient ainsi une relation entre d et les y_k, qui sera notée $\mathcal{C}_m$:

$$y_m = y_{m-1} - x_{m-1}\Big(y_{m-2} - x_{m-2}\Big(... \; ... - x_2(y_1 - d \cdot x_1)...\Big)\Big).$$

Si cette inégalité n'est pas vérifiée, alors Bob est certain qu'Alice a triché. En revanche, si cette égalité est vérifiée, les y_k correspondent à une communication valide et Bob n'a aucun moyen de savoir si Alice a été honnête ou non (résoudre le système donne des valeurs acceptables pour les a_k, et il est possible qu'Alice ait été honnête en utilisant ces a_k). L'objectif pour Alice est donc de forger des messages $y_2, ..., y_m$ tels que la condition $\mathcal{C}_m$ est vérifiée.

Pour une stratégie S d'Alice donnée, on notera $g_m(S)$ la probabilité qu'a Alice de parvenir à tromper Bob en utilisant S :

$$g_m(S) := \mathop{\mathbb{P}}_{d,x_1,...,x_{m-1}}[\mathcal{C}_m(S, d, x_1, ..., , x_{m-1})].$$

Cette probabilité est calculée pour d uniforme dans $\{0, 1\}$, chaque x_k uniforme dans $\mathbb{F}_Q$, et toutes ces variables sont indépendantes dans leur ensemble.

On définit alors $g_m := \max_S(g_m(S))$ la probabilité maximale de triche pour Alice, c'est-à-dire la probabilité qu'elle a de parvenir à tricher dans le protocole $\mathcal{P}_m$ en utilisant une stratégie optimale.

2.1.1 Symétrisation du protocole

Dans la version précédente du protocole, le dernier tour (phase de *dévoilement*) est trop différent du reste des tours, notamment parce que Bob n'envoie pas de nouveau nombre à Alice. Cette différence est gênante dès que l'on cherche à ajouter des tours, pour pouvoir étudier une stratégie sur $m + 3$ tours en fonction d'une stratégie sur m tours. Or c'est précisément ce que l'on voudrait faire.

J'ai donc réduit le problème en introduisant une légère variante du protocole, $\mathcal{P}'_m$. Pour Alice, il est légèrement plus facile de tricher dans le protocole de $\mathcal{P}'_m$ que dans $\mathcal{P}_m$. Cependant, il est encore plus facile de tricher dans $\mathcal{P}_{m+1}$. Ainsi, une bonne stratégie pour le protocole $\mathcal{P}'_m$ donnera une attaque contre $\mathcal{P}_{m+1}$. La modification est la suivante : lors de la phase de *dévoilement*, $\mathcal{B}_{m \bmod 2}$ envoie une nouveau nombre x_m, et $\mathcal{A}_{m \bmod 2}$ renvoie $y_m := a_{m-1} \cdot x_m$ au lieu de $y_m := a_{m-1}$.

Comme pour le protocole $\mathcal{P}_m$, une stratégie S' pour Alice sera un m-uplet $(s'_1, ..., s'_m)$ de fonctions. La condition $\mathcal{C}_m$ à vérifier se transforme naturellement en $\mathcal{C}'_m$:

$$\mathcal{C}'_m(S, d, x_1, \ldots, x_m) \Leftrightarrow y_m = x_m\Big(y_{m-1} - x_{m-1}\Big(y_{m-2} - x_{m-2}\Big(... \; ... - x_2(y_1 - d \cdot x_1)...\Big)\Big)\Big)$$

De même, une stratégie S' donnée fonctionne avec probabilité $g'_m(S')$, et on définit $g'_m := \max_{S'}(g'_m(S'))$.

Lemme 2 *$\forall m \geqslant 2$, on a : $g_m \leqslant g'_m \leqslant g_{m+1}$.*

Preuve :

- Pour la première inégalité : on considère une stratégie optimale $S = (s_1, ..., s_m)$ contre $\mathcal{P}_m$, où s_k est la stratégie adoptée par Alice au tour k (i.e. une fonction qui renvoie y_k quand on lui donne comme entrée toutes les connaissances dont dispose Alice au tour k). Par définition, cette stratégie permet à Alice de tricher dans $\mathcal{P}_m$ avec probabilité $g_m(S)$. On considère maintenant la stratégie suivante : $S' := (s_1, ..., s_{m-1}, s'_m)$, contre $\mathcal{P}'_m$, avec $s'_m(d, x_1, \ldots, x_{m-2}, x_m) := x_m \cdot s_m(d, x_1, \ldots, x_{m-2})$. S' permet à Alice de gagner dans $\mathcal{P}'_m$ au moins aussi efficacement qu'avec S dans $\mathcal{P}_m$, puisque S' gagne chaque fois que S gagnait. En effet, supposons que S fasse gagner dans la situation $(d, x_1, ..., x_{m-1})$. Cela signifie que $\mathcal{C}_m(S, d, x_1, \ldots, x_{m-1})$ est satisfaite, ou de façon équivalente :

  $$s_m(d, x_1, \ldots, x_{m-2}) = y_{m-1} - x_{m-1}\Big(y_{m-2} - ... \; ... - x_2(y_1 - d \cdot x_1)...\Big)$$

  Comme $s'_m(d, x_1, \ldots, x_{m-2}, x_m) = x_m \cdot s_m(d, x_1, \ldots, x_{m-2})$, on obtient immédiatement

  $$s'_m(d, x_1, \ldots, x_{m-2}, x_m) = x_m\Big(y_{m-1} - x_{m-1}\Big(y_{m-2} - ... \; ... - x_2(y_1 - d \cdot x_1)...\Big)\Big)$$

ce qui est exactement $\mathcal{C}'_m(S', d, x_1, \ldots, x_m)$, et ce quel que soit x_m. Ainsi, on a donc :

$$g_m = g_m(S) \leq g'_m(S') \leq g'_m.$$

– Pour l'autre inégalité, on fixe une stratégie $S' = (s_1, ..., s_m)$ contre $\mathcal{P}'_m$. On considère alors la stratégie $S := (s_1, ..., s_m, \overline{0})$ contre $\mathcal{P}_{m+1}$, où $\overline{0}$ est la fonction qui renvoie toujours zero 0, quelle que soit son entrée. Cela signifie qu'en appliquant S, on a toujours $y_{m+1} = 0$. Cette stratégie S est au moins aussi efficace pour gagner contre $\mathcal{P}_{m+1}$ que S' ne l'est contre $\mathcal{P}'_m$. De fait, pour des conditions $(d, x_1, ..., x_m)$, supposons que S' gagne $\mathcal{P}'_m$, alors $\mathcal{C}'(S', d, x_1, \ldots, x_m)$ est vérifiée, ou de façon équivalente :

$$y_m = x_m\Big(y_{m-1} - x_{m-1}\Big(y_{m-2} - x_{m-2}\Big(... \; ... - x_2(y_1 - d \cdot x_1)...\Big)\Big)\Big)$$

De là, on en déduit immédiatement

$$y_{m+1} = 0 = y_m - x_m\Big(y_{m-1} - x_{m-1}\Big(y_{m-2} - x_{m-2}\Big(... \; ... - x_2(y_1 - d \cdot x_1)...\Big)\Big)\Big)$$

ce qui est précisément $\mathcal{C}_{m+1}(S, d, x_1, \ldots, x_m)$. Cela montre donc que :

$$g'_m = g'_m(S') \leq g_{m+1}(S) \leq g_{m+1}.$$

□

Ainsi, dans toute la suite, on travaillera donc sur le protocole $\mathcal{P}'$, puis on invoquera ce lemme pour conclure.

On cherche donc à trouver des fonctions $(s'_1, ..., s'_m)$ qui permettent de vérifier $\mathcal{C}'_m(d, x_1, ..., x_m)$ avec la meilleure probabilité possible. Cette condition $\mathcal{C}'_m$ étant particulièrement peu agréable à lire, on réalise quelques modification. D'une part, on peut développer $\mathcal{C}'_m$. On obtient alors

$$\begin{array}{rcl} y_m & = & x_m \cdot y_{m-1} \\ & - & x_m \cdot x_{m-1} \cdot y_{m-2} \\ & + & x_m \cdot x_{m-1} \cdot x_{m-2} \cdot y_{m-3} \\ & \vdots & \vdots \\ & - & (-1)^m x_m \cdot x_{m-1} \cdot x_{m-2} \cdot ... \cdot x_1 \cdot d \end{array}$$

Ce qui peut s'écrire de la façon suivante :

$$\mathcal{C}'_m(S', d, x_1, \ldots, x_m) \Leftrightarrow y_m = \sum_{i=1}^{m-1} \left((-1)^{m-i} y_i \cdot \prod_{j=i+1}^{m} x_i \right) - (-1)^m d \cdot \prod_{j=1}^{m} x_j$$

Pour clarifier les calculs, on utilisera $\tilde{y_i} := (-1)^{i-1} y_i$. Cela permet d'éliminer des équations tous les facteurs (-1). De plus cela ne change absolument rien pour le protocole, puisqu'il est exactement aussi difficile pour Alice de forger les y_k ou les $\tilde{y_k}$. Avec les $\tilde{y_k}$, on peut donc écrire $\mathcal{C}'_m$ sous la forme :

$$\sum_{i=1}^{m} \left(\tilde{y_i} \prod_{j=i+1}^{m} x_j \right) = d \prod_{j=1}^{m} x_j$$

2.1.2 Description de l'attaque

Je vais ici détailler la construction par récurrence de mon attaque. L'initialisation se fait en considérant la stratégie S' suivante pour $\mathcal{P}'_3$:

– Pour le tour 1, $\mathcal{A}_1$ renvoie 0

- $\mathcal{A}_2$ connait x_2 au tour 2, tandis qu'au tour 3, $\mathcal{A}_1$ connait x_1, ils jouent donc un $CHSH_Q$ et obtiennent respectivement a et b (et espèrent que $a+b = x_2 \cdot x_1$)
- pour le tour 2, $\mathcal{A}_2$ renvoie $a \cdot d$
- pour le tour 3, $\mathcal{A}_1$ renvoie $x_3 \cdot d \cdot b$

Pour cette stratégie, $\mathcal{C}'3$ devient $x_3 \cdot d \cdot (a+b-x_1 \cdot x_2) = 0$. Alice gagne donc si $x_3 = 0$, si $d = 0$, ainsi que si $a + b = x_1 \cdot x_2$. Ces trois événements sont indépendants, et on obtient donc :

$$g'_3 \geqslant g'_3(S') = 1 - \frac{1}{2}\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))$$

La stratégie est ensuite définie par récurrence par pas de trois tours : la stratégie S'_{k+3} est définie grâce à S'_k :

Description récursive de la stratégie d'attaque

- tours 1 à k : Alice exécute la stratégie S'_k en produit $\tilde{y}_1, \ldots, \tilde{y}_k$.
- tour $k+1$: Alice envoie toujours $\tilde{y}_{k+1} = 0$.
- tours $k+2$ et $k+3$: à compter du tour $k+2$, $\mathcal{A}_1$ et $\mathcal{A}_2$ connaissent $d, x_1, \ldots, x_k$. On note alors
$$\eta := d \prod_{j=1}^{k} x_j - \sum_{i=1}^{k} (\tilde{y}_i \prod_{j=i+1}^{k} x_j).$$
$\mathcal{A}_{(k+2) \bmod 2}$ et $\mathcal{A}_{(k+3) \bmod 2}$ jouent à $CHSH_Q$ avec les entrées x_{k+2} et x_{k+1} et obtiennent les valeurs a et b. Ils renvoient alors pour le protocole $\tilde{y}_{k+2} = \eta \cdot a$ et $\tilde{y}_{k+3} = \eta \cdot b \cdot x_{k+3}$. On remarque que si $\eta = 0$, ce qui correspond au cas où S'_k a déjà satisfait $\mathcal{C}'_k$, Alice renvoie $\tilde{y}_{k+2}, \tilde{y}_{k+3} = 0$ quelles que soient les valeurs de a et b.

2.1.3 Analyse de l'attaque

Lemme 3 *$\forall k \geq 2$, g'_k vérifie :*

$$1 - g'_{k+3}(S'_{k+3}) \leq (1 - \frac{1}{Q})(1 - \omega(CHSH_Q))(1 - g'_k(S'_k)).$$

Preuve : On considère $\mathcal{P}'_{k+3}$. La condition de victoire d'Alice $\mathcal{C}'_{k+3}$ est :

$$\sum_{i=1}^{k+3} \left(\tilde{y}_i \prod_{j=i+1}^{k+3} x_j \right) = d \prod_{j=1}^{k+3} x_j$$

ou encore, en séparant les trois derniers termes :

$$\begin{aligned} & \tilde{y}_{k+3} \\ + \; & x_{k+3} \cdot \tilde{y}_{k+2} \\ + \; & x_{k+3} \cdot x_{k+2} \cdot \tilde{y}_{k+1} \\ + \; & x_{k+3} \cdot x_{k+2} \cdot x_{k+1} \cdot \sum_{i=1}^{k} \left(\tilde{y}_i \prod_{j=i+1}^{k} x_j \right) \\ = \; & x_{k+3} \cdot x_{k+2} \cdot x_{k+1} \cdot d \prod_{j=1}^{k} x_j \end{aligned}$$

En se souvenant que $\eta := d \prod_{j=1}^{k} x_j - \sum_{i=1}^{k} (\tilde{y}_i \prod_{j=i+1}^{k} x_j) \in \mathbb{F}_q$, on peut simplifier $\mathbf{C}'_{k+3}$ en :

$$\mathcal{C}'_{k+3} \Leftrightarrow \tilde{y}_{k+3} + x_{k+3} \cdot \tilde{y}_{k+2} + x_{k+3} \cdot x_{k+2} \cdot \tilde{y}_{k+1} = x_{k+3} \cdot x_{k+2} \cdot x_{k+1} \cdot \eta$$

En utilisant la description de l'attaque $\tilde{y}_{k+2} = \eta \cdot a$ et $\tilde{y}_{k+3} = \eta \cdot b \cdot x_{k+3}$, où a et b sont les résultat qu'Alice obtient dans le jeu $CHSH_Q$, on peut expliciter $\mathcal{C}'_{k+3}$:

$$\begin{aligned} \mathcal{C}'_{k+3} &\Leftrightarrow \tilde{y}_{k+3} + x_{k+3} \cdot \tilde{y}_{k+2} + x_{k+3} \cdot x_{k+2} \cdot \tilde{y}_{k+1} = x_{k+3} \cdot x_{k+2} \cdot x_{k+1} \cdot \eta \\ &\Leftrightarrow x_{k+3} \cdot b \cdot \eta + x_{k+3} \cdot a \cdot \eta + 0 = x_{k+3} \cdot x_{k+2} \cdot x_{k+1} \cdot \eta \\ &\Leftrightarrow \eta \cdot x_{k+3} \cdot (a + b - x_{k+1} \cdot x_{k+2}) = 0 \\ &\Leftrightarrow (x_{k+3} = 0) \vee (\eta = 0) \vee (a + b = x_{k+1} \cdot x_{k+2}) \end{aligned}$$

Ces trois événements sont indépendants puisque :

- $(x_{k+3} = 0)$ ne dépend que de x_{k+3}, et se produit avec probabilité $\frac{1}{Q}$.
- $(\eta = 0)$ ne dépend que de $d, x_1, ..., x_k$, et se produit avec probabilité $g'_k(S_k)$.
- $(a + b = x_{k+1} \cdot x_{k+2})$ ne dépendent que de x_{k+1} et x_{k+2} ($\mathcal{A}_1$ et $\mathcal{A}_2$ utilisent une stratégie optimale pour le jeu $CHSH_Q$ avec les entrées x_{k+1}, x_{k+2}, en ignorant toutes les informations superflues). Cela se produit avec probabilité $\omega(CHSH_Q)$.

Ainsi, cette stratégie particulière donne

$$g'_{k+3}(S'_{k+3}) = \mathbb{P}[\mathcal{C}'_{k+3}] = 1 - (1 - g'_k(S'_k))(1 - \frac{1}{Q})(1 - \omega(CHSH_Q))$$

ou de façon équivalente :

$$1 - g'_{k+3}(S'_{k+3}) \leq (1 - \frac{1}{Q})\,(1 - \omega(CHSH_Q))\,(1 - g'_k(S'_k)).$$

□

Nous avons maintenant tous les outils requis pour montrer le résultat principal :

Théorème 2 *$\forall m \geq 3$, on a :*

$$g_m \geq 1 - \frac{1}{2}\left(\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\right)^{\lfloor \frac{m-1}{3} \rfloor}$$

Preuve : En itérant le lemme précédent, on obtient

$$1 - g'_{3k}(S'_{3k}) \leq \left(\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\right)^{k-1}(1 - g'_3(S'_3))$$

Puis à l'aide de l'initialisation pour $\mathcal{P}'_3$: $g'_3(S_3) \geq 1 - \frac{1}{2}(1 - \frac{1}{Q})(1 - \omega(CHSH_Q))$ on déduit

$$g'_{3k} \geq g'_{3k}(S_{3k}) \geq 1 - \frac{1}{2}\left(\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\right)^{k}.$$

En revenant au protocole original grâce au lemme 2, on a :

$$g_{3k+1} \geq g'_{3k} \geq 1 - \frac{1}{2}\left(\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\right)^{k}.$$

Ainsi, si m peut se mettre sous la forme $m = 3k + 1$ pour un certain entier k, on a

$$g_m \geq 1 - \frac{1}{2}\left(\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\right)^{\frac{m-1}{3}}$$

et puisque g_m est une fonction croissante (par exemple par le lemme 2), on conclut que, pour $m \geq 3$:

$$g_m \geq 1 - \frac{1}{2}\left(\left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\right)^{\lfloor \frac{m-1}{3} \rfloor}$$

□

2.2 Généralisation

Dans la partie précédente, nous avons étudié une attaque réalisable par Alice si :
- elle est capable de communiquer suffisamment efficacement pour qu'à chaque tour k, il ne lui manque que l'information x_{k-1}
- l'utilisation du protocole lui permet de connaitre la valeur d qu'elle a intérêt à révéler dès le tour 2.

En pratique, rien n'assure que ces conditions puissent être réunies. Il est possible d'affaiblir ces hypothèses en les remplaçant par :
- Alice requiert un temps de propagation ρ (ce temps est exprimé en nombre de tours) pour transmettre de l'information entre $\mathcal{A}_1$ et $\mathcal{A}_2$. Au tour k, l'agent actif d'Alice connait donc $x_1, x_2, ..., x_{k-\rho}$. Il a aussi connaissance de $x_{k-2}, x_{k-4}, ...$ puisque ces tours se sont déroulés au même endroit. Puisqu'un agent a accès à tout ce qui c'est passé un nombre pair de tours auparavant, on peut supposer sans perte de généralité que ρ est pair.
- Alice ne parvient à se décider qu'au tour k_0 concernant la valeur qu'il lui plait de révéler. Sa stratégie doit donc être indépendante de d jusqu'au tour k_0.

La seconde modification n'a qu'une influence très faible vu qu'il suffit d'ajouter un décalage dans la stratégie pour tenir compte de k_0. En revanche, l'augmentation du temps de propagation ρ va conduire à une modification plus profonde de la structure de l'attaque. En effet, les agents $\mathcal{A}_1$ et $\mathcal{A}_2$ vont avoir besoin de plus d'un tour pour analyser ce qu'ils ont fait précédemment (déterminer s'ils ont déjà gagné, et calculer le facteur correctif η). Il va donc s'écouler $\rho - 1$ tours entre deux occasion pour $\mathcal{A}_1$ et $\mathcal{A}_2$ de jouer un $CHSH_Q$. De plus le fait d'attendre ces tours additionnels introduit un biais supplémentaire. Il ne s'agira donc pas de jeux $CHSH_Q$, mais $CHSH_Q^\gamma$.

On peut remarquer que le cas étudié précédemment correspond aux valeurs minimales $\rho = 2$ et $k_0 = 2$. Si $k_0 = 1$, c'est qu'Alice est honnête et estime qu'elle n'a pas besoin de tricher, et si $\rho = 1$, c'est qu'elle sait téléporter de l'information.

2.2.1 Description de l'attaque

Pour $k \geqslant k_0$, la stratégie $S'_{k+\rho+1}$ d'Alice pour attaquer le protocole qui dure $k+\rho+1$ tours est définie à partir de S'_k :

Description récursive de la stratégie d'attaque $S'_{k+\rho+1}$

- tours 1 à k : Alice utilise la stratégie S'_k pour produire $\tilde{y}_1, \ldots, \tilde{y}_k$.
- tours $k+1$ à $k+\rho-1$: Alice envoie $\tilde{y}_{k+1}, \ldots, \tilde{y}_{k+\rho-1} = 0$.
- tours $k+\rho$ et $k+\rho+1$: à partir du tour $k+\rho$, $\mathcal{A}_1$ et $\mathcal{A}_2$ connaissent $d, x_1, \ldots, x_k$. On définit alors

$$\eta := d \prod_{j=1}^{k} x_j - \sum_{i=1}^{k} (\tilde{y}_i \prod_{j=i+1}^{k} x_j).$$

Par ailleurs, $\mathcal{A}_1$ connait aussi $X = \prod_{j \text{ impair} \,:\, k+1 \leq j \leq k+\rho} x_j$, et $\mathcal{A}_2$ connait $Y = \prod_{j \text{ pair} \,:\, k+1 \leq j \leq k+\rho} x_j$. $\mathcal{A}_{(k+\rho) \bmod 2}$ et $\mathcal{A}_{(k+\rho+1) \bmod 2}$ utilisent donc une stratégie optimale pour $CHSH_Q^\gamma$ avec $\gamma := 1 - (1 - \frac{1}{Q})^{\frac{\rho}{2}}$ et les entrées X et Y pour obtenir a et b. Pour le protocole de engagement de bit, ils envoient donc à Bob : $\tilde{y}_{k+\rho} = \eta \cdot a$ et $\tilde{y}_{k+\rho+1} = \eta \cdot b \cdot x_{k+\rho+1}$.

2.2.2 Analyse

Lemme 4 *$\forall k \geq k_0$, on a*

$$g'_{k+\rho+1} \geq \left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\, g'_k + 1 - \left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))$$

$$1 - g'_{k+\rho+1} \leqslant \left(1 - \frac{1}{Q}\right)(1 - \omega(CHSH_Q))\,(1 - g'_k)$$

Preuve : Cette démonstration ressemble fort à celle du 3. On considère une stratégie définie comme ci-dessus. La condition pour qu'Alice gagne est $\mathcal{C}'_{k+\rho+1}$:

$$\sum_{i=1}^{k+\rho+1} \left(\tilde{y_i} \prod_{j=i+1}^{k+\rho+1} x_j \right) = d \prod_{j=1}^{k+\rho+1} x_j$$

c'est-à-dire, en séparant les $\rho+1$ derniers termes du reste de la somme :

$$\begin{array}{rl} & \tilde{y}_{k+\rho+1} \\ + & x_{k+\rho+1} \cdot \tilde{y}_{k+\rho} \\ + & 0 \\ \vdots & \\ + & 0 \\ + & \left(\prod_{j=k+1}^{k+\rho+1} x_j \right) \sum_{i=1}^{k} \left(\tilde{y_i} \prod_{j=i+1}^{k} x_j \right) \\ = & \left(\prod_{j=k+1}^{k+\rho+1} x_j \right) d \prod_{j=1}^{k} x_j \end{array}$$

En utilisant $\eta := d \prod_{j=1}^{k} x_j - \sum_{i=1}^{k} (\tilde{y_i} \prod_{j=i+1}^{k} x_j)$, on peut simplifier $\mathcal{C}'_{k+\rho+1}$ en :

$$\begin{aligned} \mathcal{C}'_{k+\rho+1} &\Leftrightarrow \tilde{y}_{k+\rho+1} + x_{k+\rho+1} \cdot \tilde{y}_{k+\rho} = \left(\prod_{j=k+1}^{k+\rho+1} x_j \right) \cdot \eta \\ &\Leftrightarrow \tilde{y}_{k+\rho+1} + x_{k+\rho+1} \cdot \tilde{y}_{k+\rho} = X \cdot Y \cdot \eta. \end{aligned}$$

Dans cette stratégie, Alice utilise $\tilde{y}_{k+\rho} = a \cdot \eta$ et $\tilde{y}_{k+\rho+1} = x_{k+\rho+1} \cdot b \cdot \eta$, où a et b sont les résultat d'une instance de $CHSH_Q^\gamma$, joué avec les entrées X et Y. En effet, les x_k étant indépendants et uniformément distribués dans $\mathbb{F}_Q$, X et Y ne sont pas uniformément distribués, mais ont un biais γ vers 0. On obtient donc :

$$\mathcal{C}'_{k+\rho+1} \Leftrightarrow (x_{k+\rho+1} = 0) \vee (\eta = 0) \vee (a + b = XY)$$

Or ces trois événements sont indépendants. Le premier se produit avec probabilité $\frac{1}{Q}$, le second avec probabilité g'_k. Pour le troisième, on remarque que X est le produit de $\frac{\rho}{2}$ nombres aléatoires, indépendants et uniformes dans $\mathbb{F}_Q$. Ainsi :

- $\mathbb{P}[X = 0] = 1 - (1 - \frac{1}{Q})^{\frac{\rho}{2}} = \gamma$
- $\forall z \in \mathbb{F}_Q^*$, $\mathbb{P}[X = z] = \frac{1-\gamma}{Q-1}$

Y est distribué de la même façon que X, et ces deux variables sont indépendantes. ainsi, $\mathbb{P}[a+b = XY]$ est exactement la probabilité de gagner un jeu $CHSH_Q^\gamma$ en utilisant la stratégie optimale. Cela donne :

$$g'_{k+\rho+1} \geq 1 - \left(1 - \frac{1}{Q}\right) (1 - g'_k) \left(1 - \omega(CHSH_Q^\gamma)\right)$$

Puis, le lemme 1 permettant de se ramener à $CHSH_Q$:

$$g'_{k+\rho+1} \geq 1 - \left(1 - \frac{1}{Q}\right) (1 - g'_k) (1 - \omega(CHSH_Q))$$

c'est-à-dire :

$$1 - g'_{k+\rho+1} \leqslant \left(1 - \frac{1}{Q}\right) (1 - \omega(CHSH_Q)) (1 - g'_k)$$

□

Théorème 3 *Pour tous $k_0 \geqslant 2$ et $\rho \geqslant 2$, quel que soit $m \geqslant k_0 + \rho + 1$, on a :*

$$g_m \geq 1 - \frac{1}{2}\left((1-\frac{1}{Q})\,(1-\omega(CHSH_Q))\right)^{\frac{m-k_0-1}{\rho+1}}$$

Preuve : On utilise récursivement le lemme précédent (Lemme 4), en initialisant trivialement par $g'_{k_0} \geqslant \frac{1}{2}$. Cela amène : $\forall k \geq k_0$

$$g'_{k_0+k(\rho+1)} \geq 1 - \frac{1}{2}\left((1-\frac{1}{Q})\,(1-\omega(CHSH_Q))\right)^{k},$$

puis par le lemme 2

$$g_{k_0+k(\rho+1)+1} \geq 1 - \frac{1}{2}\left((1-\frac{1}{Q})\,(1-\omega(CHSH_Q))\right)^{k}.$$

Ainsi, si m peut se mettre sous la forme $m = k_0 + k(\rho+1) + 1$, on a $k = \frac{m-k_0-1}{\rho+1}$ et

$$g_m \geq 1 - \frac{1}{2}\left((1-\frac{1}{Q})\,(1-\omega(CHSH_Q))\right)^{\frac{m-k_0-1}{\rho+1}}.$$

Enfin, g_m étant une fonction croissante en m. On peut en conclure que :

$$g_m \geq 1 - \frac{1}{2}\left((1-\frac{1}{Q})\,(1-\omega(CHSH_Q))\right)^{\lfloor\frac{m-k_0-1}{\rho+1}\rfloor}.$$

□

Si on remplace ρ et k_0 par 2 pour se placer dans le cas des conditions parfaites étudiées précédemment, on obtient une borne très légèrement plus faible que ce qui avait été prouve alors. C'est du à l'initialisation qui est plus grossière ici.

2.3 Récapitulatif et ordres de grandeur

Grâce à [BS15] et [PP16], on sait que :

- Si $Q = p^{2k}$, alors $\omega(CHSH_Q) = \Omega\left(\frac{1}{\sqrt{Q}}\right)$
- Si $Q = p^{2k+1}$, alors $\omega(CHSH_Q) = \Omega\left(Q^{-2/3}\right)$

Pour le cas d'une puissance paire, on sait qu'on ne peut pas faire mieux. Par contre, dans le cas d'une puissance impaire, actuellement, on sait seulement qu'on ne pourra pas faire mieux que $\mathcal{O}(Q^{-\frac{1}{2}-\frac{1}{700}})$.

Dans le cas où le corps du protocole est une puissance paire d'un nombre premier, si Alice utilise une de ces stratégies dans le cas où les conditions sont parfaites pour elle, elle dispose donc d'une stratégie qui lui permet de gagner avec probabilité $1 - \frac{1}{2}\left(1-\Omega\left(\frac{1}{\sqrt{Q}}\right)\right)^{\lfloor\frac{k-1}{3}\rfloor}$. Or [CCL15] et [FF16] ont montré qu'Alice ne peut pas gagner avec une probabilité supérieure à $\frac{1}{2} + 2k\sqrt{\frac{2}{q}}$. Cela signifie que :

- le protocole est sûr si le nombre k de tours est petit devant $\sqrt{Q}$. En effet, dans ce cas, l'écart entre une Alice honnête et une Alice malhonnête n'est que $\Theta\left(\frac{k}{\sqrt{Q}}\right)$, ce qui est petit. (Ce fait était déjà connu)
- si le nombre k de tours est du même ordre de grandeur que $\sqrt{Q}$ (c'est-à-dire $k = t\sqrt{Q}$), alors Alice peut gagne avec probabilité $1 - \frac{1}{2}e^{-t/3}$, ce qui converge rapidement vers 1. Ce résultat n'était pas connu avant la découverte de mon attaque.

Cela justifie l'affirmation selon laquelle la borne de sécurité précédente ainsi que mon attaque sont toutes les deux presque optimales. L'ordre de grandeur est connu, il reste possible de chercher à affiner les constantes, mais ce serait d'un intérêt limité.

Dans le cas d'une puissance impaire, en utilisant les meilleures stratégies connues à l'heure actuelle, on ne pourra pas faire mieux que $1 - \frac{1}{2}\left(1 - \Omega(Q^{-2/3})\right)^{\lfloor \frac{k-1}{3} \rfloor}$. Ce résultat est évidemment moins efficace que ce qui se faisait dans le cas d'une puissance paire. Cependant, la convergence vers 1 est encore en exponentielle : si $k = tQ^{2/3}$, alors Alice gagne encore avec probabilité $1 - \frac{1}{2}e^{-t/3}$. Il faut donc que le protocole dure un temps $Q^{1/6}$ fois plus long pour qu'Alice soit certaine de tricher avec une probabilité aussi bonne que dans le cas où Q est une puissance paire.

Enfin, dans le cas plus général (et plus réaliste), il ne faut plus 3 tours pour chaque occurrence d'un jeu $CHSH$, mais $\rho + 1$. Ainsi, pour qu'Alice puisse avoir une aussi bonne chance de tricher que dans le cas où l'agent actif du tour k n'ignore que x_{k-1}, il lui suffit d'un protocole qui dure $\frac{\rho+1}{3}$ fois plus longtemps.

En reprenant ici l'exemple d'application numérique de [CCL15]. Si Bob exige une sécurité de 128 bits, que ses agents se situent à une distance $d = 100\ km$, et qu'il utilise un corps de taille $Q = 2^{340}$, alors les communications consistent en des messages de taille 340 *bits* (ce qui est parfaitement raisonnable), et le protocole resterait sûr pendant 30 ans. Cela signifie qu'au bout de ces 30 ans, Alice pourrait enfin tricher suffisamment pour gagner avec probabilité $\frac{1}{2} + 2^{-128}$. Mon attaque est donc intéressante en théorie, mais en pratique il serait très facile d'utiliser des tailles de messages suffisantes pour que l'attaque soit insignifiante.

Si ce protocole n'est pas très pratique à réaliser (il nécessite une très bonnes synchronisation, et ne tolère pas les pertes ni les retards de paquets), il a cependant été réalisé [VMH+16]. Ils ont en effet fait tourner ce protocole durant 24 heures, avec des agents éloignée de 7 km. Le protocole a donc demandé 5×10^9 tours. Avec une taille de corps de $Q = 2^{128}$, Alice pouvait donc gagner avec une probabilité d'au plus $\frac{1}{2} + 7.8 \times 10^{-10}$.

Conclusion

J'ai travaillé sur le protocole de [LKB+15] pour la mise en gage de bit, un protocole crée pour résister à des attaquant disposant d'une puissance de calcul arbitrairement grande. Il était auparavant connu que, pour que le protocole soit sûr durant k tour, il suffisait que les clefs soient choisies dans un corps de taille grande devant k^2. J'ai montré que si le cardinal du corps est une puissance paire d'un nombre premier, alors c'est aussi nécessaire. Si c'est une puissance impaire, ma borne est un peu plus faible. Ce résultat est intéressant car il s'agit de la première attaque proposée contre ce protocole. La proximité de la borne donnée par cette attaque et la borne de sécurité connue précédemment ([CCL15] et [FF16]) permet d'estimer que la question est close. Cette attaque à par ailleurs donné lieu à un papier [BC16]. Une autre conséquence de cette attaque est d'avoir permis de réfuter avec certitude un résultat affirmé par [PPP16].

Références

[BC16] Rémi Bricout and André Chailloux. Recursive cheating strategies for the relativistic F_Q bit commitment protocol. *arXiv preprint quant-ph :1608.03820*, 2016. 2, 15

[BS15] Mohammad Bavarian and Peter W. Shor. Information causality, szemerédi-trotter and algebraic variants of CHSH. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 123–132, 2015. 2, 5, 14

[CCL15] Kaushik Chakraborty, André Chailloux, and Anthony Leverrier. Arbitrarily long relativistic bit commitment. *Phys. Rev. Lett.*, 115 :250501, Dec 2015. 1, 2, 5, 14, 15

[CSST11] Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two provers in isolation. In *Advances in Cryptology–ASIACRYPT 2011*, pages 407–430. Springer, 2011. 4

[FF16] Serge Fehr and Max Fillinger. On the composition of two-prover commitments, and applications to multi-round relativistic commitments. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 477–496, 2016. 1, 2, 5, 14, 15

[LKB+15] T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, S. Wehner, and H. Zbinden. Practical relativistic bit commitment. *Phys. Rev. Lett.*, 115 :030502, Jul 2015. 1, 2, 4, 5, 15, 18

[PP16] Matej Pivoluska and Martin Plesch. An explicit classical strategy for winning a CHSH_q game. *New Journal of Physics*, 18(2) :025013, 2016. 14

[PPP16] Matej Pivoluska, Marcin Pawlowski, and Martin Plesch. Experimentally secure relativistic bit commitment. *arXiv preprint quant-ph :1601.08095*, 2016. 1, 2, 15

[PW12] M. Pawlowski and A. Winter. Hyperbits : The information quasiparticles. *Physical Review A, 85(2)*, 2012. 2

[VMH+16] E. Verbanis, A. Martin, R. Houlmann, G. Boso, F. Bussières, and H. Zbinden. 24-hour relativistic bit commitment. *arXiv preprint arXiv :1605.07442*, 2016. 15

Annexes

Annexe A : Règles de calcul sur les états quantiques

Durant mon stage, il m'est arrivé de travaillé sur des états quantiques. En particulier, la validité de la stratégie que j'ai trouvée pour gagner quantiquement au jeu du "nombre sur le front" requiert de savoir calculer sur des états quantiques. L'objectif ici n'est pas d'écrire un manuel de calcul des états quantiques, mais de donner suffisamment d'éléments pour comprendre les calculs quantiques présents dans ce rapport. Cette section ne contiendra donc que les bases, ainsi que tout ce qui est nécessaire à la compréhension de la stratégie du "nombre sur le front". En particulier, on se limitera au états purs.

Q-bits

Un q-bit (équivalent quantique du bit) est l'unité d'information de base d'un calcul quantique. Il peut être dans l'état 0 (on notera $|0\rangle$), soit dans l'état 1 ($|1\rangle$), soit plus généralement dans une combinaison linéaire des cas précédents ($\alpha|0\rangle + \beta|1\rangle$, avec $|\alpha|^2 + |\beta|^2 = 1$). On parle dans ce cas de superposition d'états.

Il est possible de mesurer un état $\alpha|0\rangle + \beta|1\rangle$ dans la base $(|0\rangle, |1\rangle)$. Si on fait cette opération, on fait s'écrouler l'état quantique :

- avec probabilité $|\alpha|^2$, on obtiendra $|0\rangle$, et le q-bit sera ensuite dans l'état $|0\rangle$
- avec probabilité $|\beta|^2$, on obtiendra $|1\rangle$, et le q-bit sera ensuite dans l'état $|1\rangle$

En effet, en physique quantique, l'observation perturbe le système, et c'est le cas ici. Mesurer un q-bit détruit la superposition, et fixe le q-bit dans l'état mesuré. En particulier, si on mesurait à nouveau le même q-bit, on obtiendrait le même résultat.

États quantiques et intrication

Plus généralement, on peut utiliser des systèmes mettant en jeu plusieurs q-bits. Un système de n q-bits vit dans l'espace de Hilbert $\mathbb{C}^{2^n}$.

Par exemple, pour un système de deux q-bits, un état quelconque sera une combinaison de la forme $|\varphi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, avec $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. $|00\rangle = |0\rangle \otimes |0\rangle$ est l'état dans lequel les deux q-bits sont dans l'état $|0\rangle$, et ainsi de suite. $\otimes$ représente le produit tensoriel.

Pour l'état $|\varphi\rangle$ d'une telle paire de q-bits, deux cas sont possibles :

- soit $|\varphi\rangle$ peut se mettre sous la forme $(\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle)$, auquel cas l'état est séparable ;
- soit ce n'est pas possible, auquel cas les états sont intriqués : agir sur l'un aura des conséquences sur le comportement de l'autre. C'est par exemple le cas d'une paire EPR (Einstein, Podolsky, Rosen) $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$: cet état ne peut pas se décomposer comme un produit tensoriel.

Mesure partielle

On peut décider de ne mesurer qu'un seul q-bit d'un système quantique. Notons $|\varphi\rangle$ un état de départ, dont on veut mesurer le premier q-bit. Cet état peut se réécrire $|0\rangle \otimes |\varphi_0\rangle + |1\rangle \otimes |\varphi_1\rangle$. On va donc obtenir le résultat suivant :

- avec probabilité $\||\varphi_0\rangle\|_2^2$, la mesure du premier q-bit donnera $|0\rangle$, et l'état s'effondre dans le nouvel état $\frac{1}{\||\varphi_0\rangle\|_2}(|0\rangle \otimes |\varphi_0\rangle)$
- avec probabilité $\||\varphi_1\rangle\|_2^2$, elle donnera $|0\rangle$, et l'état s'effondre dans le nouvel état $\frac{1}{\||\varphi_1\rangle\|_2}(|1\rangle \otimes |\varphi_1\rangle)$

Par exemple ; pour une paire EPR, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Lors de la mesure du premier q-bit,

- avec probabilité $\frac{1}{2}$, on mesure $|0\rangle$, et l'état devient $|00\rangle$;
- avec probabilité $\frac{1}{2}$, on mesure $|1\rangle$, et l'état devient $|11\rangle$

En particulier, si on sépare ces q-bits, et qu'on les mesure simultanément, les deux mesures donnerons la même valeur, bien que cette valeur ne puisse pas être connue par avance.

J'en profite ici pour signaler que partager un état quantique permet à deux personnes (ou plus) d'avoir une certaine corrélation dans leurs mesures, et donc dans leurs décisions. Cependant, cela ne permet en aucun cas de transmette la moindre information. Si deux personnes A et B peuvent partager autant de ressources quantiques qu'ils le souhaitent et se concerter à l'avance, et si on isole A et B et que l'on donne un message m à A, il est impossible à B d'en apprendre ne serait-ce qu'un bit.

Opérations sur les q-bits

Il est possible de modifier l'état d'un système de q-bits, mais seules les opérations unitaires sont autorisées. Cela permet par exemple d'effectuer une mesure dans n'importe quelle base orthonormée. En effet, les carrés des modules des coefficients d'un état quantique sont les probabilités d'obtenir ce résultat lors d'une mesure. La norme $\|\cdot\|_2$ de tout état quantique est donc nécessairement 1. Toute opération sur les états quantiques se doit donc de préserver le norme $\|\cdot\|_2$, ce qui caractérise les matrices unitaires.

Résultats généraux de commutation

Toutes les opérations de mesure réalisées sur des q-bit différents commutent entre elles. Cela se montre facilement dans le cas où l'on mesure tous les q-bits d'un système chacun dans la base $(|0\rangle, |1\rangle)$. En effet, pour un état $a_0|0\rangle + a_1|1\rangle + ... + a_N|N\rangle$ comportant $n = log_2(N)$ q-bits ($|a_0|^2 + ... + |a_N|^2 = 1$), le résultat des mesures donnera l'état $|k\rangle$ avec probabilité $|a_k|^2$ quel que soit

l'ordre des mesures. Ce résultat se généralise à des mesures partielles et des mesures dans d'autres bases.

De façon générale, les opérations réalisées sur un système quantique ne commutent pas (non-commutativité des matrices unitaires). Cependant, dans le cas du "nombre sur le front", les opérations réalisées sont particulières et vont commuter. En effet, dans ma stratégie pour le jeu "nombre sur le front", un système de n q-bits est coupé et chaque q-bit est donné à un joueur différent. Lorsqu'un joueur k modifie l'état du système, il ne peut agir que sur le q-bit qui lui a été confié. La matrice unitaire représentant son action peut donc se décomposer en produit tensoriel : $\underbrace{I \otimes I \otimes ... \otimes I}_{k-1\ fois} \otimes M \otimes \underbrace{I \otimes I \otimes ... \otimes I}_{n-k\ fois}$, où M est une transformation unitaire de $\mathbb{C}^2$. Deux telles opérations commutent donc évidemment dès lors qu'elles sont réalisées par des joueurs différents.

Une autre façon de se convaincre que de telles opérations doivent commuter (qu'il s'agisse du cas des mesures ou des opérations sur les q-bits) est que les joueurs sont isolés, c'est-à-dire infiniment loin les uns des autres. Si le fait d'effectuer une mesure avant ou après une autre joueur pouvait avoir une influence, cela impliquerait la possibilité de téléporter de l'information, ce qui est rejeté dans le modèle de la physique quantique. La localité a été abandonnée (deux systèmes éloignée peuvent avoir des comportements corrélés), mais il reste néanmoins impossible de transmettre de l'information à une vitesse supérieure à celle de la lumière.

Annexe B : Stratégie quantique pour gagner à "Nombre sur le front"

En introduisant le protocole de mise en gage de bit relativiste, [LKB+15] donne une première borne de sécurité. Pour cela, ils réduisent le protocole à un jeu dit "nombre sur le front", qui est une généralisation possible de $CHSH$ à n joueurs. En étudiant les démonstrations classiques de sécurité du protocole de engagement de bit, et en cherchant à m'en inspirer pour obtenir une borne de sécurité contre un adversaire quantique, j'ai cherché à déterminer ce que pouvait devenir ce jeu dans le cas où les joueurs peuvent partager un état quantique.

J'ai réussi à trouver une très jolie stratégie, qui généralise les résultats connus sur le jeu $CHSH$ d'origine. Dans cette stratégie, tous les joueurs suivent le même protocole, et n'ont pas non plus à se soucier de l'ordre des entrées. Le jeu est parfaitement symétrique entre tous les joueurs, mais cela ne garantissait pas de trouver une bonne stratégie qui respecte ces symétries.

Cette stratégie permet à n joueurs de gagner avec probabilité $\cos\left(\frac{\pi}{2^{n+1}}\right)^2$ quelles que soient les entrées, alors que toute stratégie classique ne peut faire mieux que $1 - \frac{1}{2^n}$ en moyenne. La probabilité d'échec des joueurs est donc équivalente à $\frac{\pi^2}{2^{2n+1}}$ au lieu de $\frac{1}{2^n}$ classiquement.

Jeu du "Nombre sur le front"

Le modèle du jeu pour n joueurs est le suivant : n nombres $x_1, ..., x_n$ sont tirés indépendamment et uniformément dans $\{0, 1\}$. Chaque joueur reçoit tous les x_j sauf celui qui prote son numéro (le joueur k reçoit $(x_1, ..., x_{k-1}, x_{k+1}, ..., x_n)$). Tout se passe donc comme si ces n joueurs étaient en cercle, et chacun a un nombre collé sur le front. Il est donc possible de voir les nombres des autres joueurs, mais le sien est hors de vue. Ces joueurs ne peuvent pas communiquer et doivent répondre chacun un y_k. Ils ont gagné si $\prod_{j=1}^{n} x_j = \bigoplus_{j=1}^{n} y_j$.

Démarche de recherche de la stratégie

Pour trouver cette stratégie, j'ai écrit un programme pour en rechercher automatiquement. En partant d'une stratégie aléatoire, le programme calcule la valeur du gain, et modifie aléatoirement la stratégie. La modification est d'autant plus petite que la stratégie est efficace. Je ne suis pas

allé jusqu'à implémenter une descente de gradient car cela n'était pas nécessaire : pour de petites valeurs de n, la convergence était suffisamment rapide.

Cette recherche m'a permis de trouver des stratégies possibles pour de petites valeurs de n. J'ai ensuite extrapolé pour exhiber une stratégie générique. La principale difficulté pour cette extrapolation a été que pour un n donné, de très nombreuses stratégies donnent la même valeur, et il n'y a donc pas unicité de la stratégie optimale. J'ai imposé des contraintes (principalement de symétrie) pour ne conserver qu'un nombre raisonnable de stratégies (tant que ces contraintes restaient compatibles avec la valeur du gain optimal).

Les paramètres pour trouver une stratégie à n joueurs sont :
- les dimensions d_k des états quantiques possédés par chaque joueur k
- les opérations réalisées par ces joueurs en fonction de toutes leurs entrées (c'est-à-dire une matrice unitaire, à coefficients complexes, de taille $2^{d_k} \times 2^{d_k}$, et ce pour chacune des 2^{k-1} entrées possibles).

La base de mesure n'est pas un paramètre car il est toujours possible de mesurer dans la base canonique, quitte à appliquer une opération (unitaire) auparavant. Ensuite, cette opération est fusionnée avec celle déjà réalisée.

Description de la stratégie

Dans la suite, pour un angle α, la notation M_α désignera la matrice (unitaire) $\frac{1}{\sqrt{2}}\begin{bmatrix}1 & e^{i\alpha}\\ 1 & -e^{i\alpha}\end{bmatrix}$.

On utilisera aussi les angles $\alpha_{n,k} := \dfrac{(-1)^k\pi}{(k+1)2^k\binom{n}{k+1}}\sum_{j=0}^{n}\binom{n}{k-j}$.

La stratégie est la suivante :
- avant le début du jeu, les joueurs partagent un état GHZ (i.e. $\frac{1}{\sqrt{2}}(|000..0\rangle+|111..1\rangle)$), chaque joueur disposant d'un q-bit de cet état.
- chaque joueur compte le nombre de 1 parmi ses entrées, nombre que l'on notera k, puis applique $M_{\alpha_{n,k}}$ à son q-bit
- tous les joueurs mesurent leur q-bit et renvoient le résultat de leur mesure.

Validité de la stratégie

Commençons par déterminer l'état quantique du système après que chaque joueur a appliqué sa matrice. Pour cela, on montrera le lemme suivant, qui permet d'expliciter l'état obtenu en appliquant des matrices de la forme M_α à un état GHZ :

Lemme 5 *En partant d'un état GHZ à n joueurs, biaisé, $\frac{1}{\sqrt{2}}(|\underbrace{000..0}_{n}\rangle + z|\underbrace{111..1}_{n}\rangle)$ où $|z| = 1$, après que chaque joueur i a appliqué M_{α_i} sur le i^e q-bit, on obtient l'état*

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{x_1,\dots,x_n\in\{0,1\}^n}(1+z(-1)^{x_1+\dots+x_n}e^{i(\alpha_1+\dots+\alpha_n)})|x_1x_2...x_n\rangle$$

Preuve : Prouvons ce résultat par récurrence :
- Pour $n = 1$:
  On part de $\varphi = \frac{1}{\sqrt{2}}(|0\rangle + z|1\rangle)$, on applique $M_\alpha := M_\alpha$. On obtient alors

$$\begin{aligned}\psi &= \frac{1}{2}((|0\rangle+|1\rangle)+z(e^{i\alpha}|0\rangle-e^{i\alpha}|1\rangle))\\ &= \frac{1}{2}((1+ze^{i\alpha})|0\rangle+(1-ze^{i\alpha})|1\rangle)\\ &= \frac{1}{\sqrt{2^{n+1}}}\sum_{x_1\in\{0,1\}}(1+z(-1)^{x_1}e^{i\alpha})|x_1\rangle\end{aligned}$$

– Pour $n \geqslant 1$, supposons la propriété vérifiée au rang n :
On commence avec un état $\varphi = \frac{1}{\sqrt{2}}(|\underbrace{000..0}_{n+1}\rangle + z|\underbrace{111..1}_{n+1}\rangle)$. L'ordre dans lequel les joueurs appliquent leurs opérations n'ayant aucune importance, on supposera que le joueur $n+1$ agit d'abord. Il applique donc sa matrice $M_{\alpha_{n+1}} = M_{\alpha_{n+1}}$. On obtient alors l'état

$$\begin{aligned}\tilde{\varphi} &= \frac{1}{2}(|\underbrace{000..0}_{n}\rangle(|0\rangle + |1\rangle) + z|\underbrace{111..1}_{n}\rangle(e^{i\alpha_{n+1}}|0\rangle - e^{i\alpha_{n+1}}|1\rangle)) \\ &= \frac{1}{2}((|000..0\rangle + ze^{i\alpha_{n+1}}|111..1\rangle)|0\rangle + (|000..0\rangle - ze^{i\alpha_{n+1}}|111..1\rangle)|1\rangle)\end{aligned}$$

Les n premiers joueurs peuvent maintenant exécuter leurs opérations, ce qui donne, par hypothèse de récurrence :

$$\begin{aligned}\psi &= \frac{1}{2}\Big(\frac{1}{\sqrt{2^{n+1}}} \sum_{x_1,...,x_n \in \{0,1\}^n} (1 + ze^{i\alpha_{n+1}}(-1)^{x_1+...+x_n}e^{i(\alpha_1+...+\alpha_n)})|x_1x_2...x_n\rangle|0\rangle \\ &+ \frac{1}{\sqrt{2^{n+1}}} \sum_{x_1,...,x_n \in \{0,1\}^n} (1 + (-ze^{i\alpha_{n+1}})(-1)^{x_1+...+x_n}e^{i(\alpha_1+...+\alpha_n)})|x_1x_2...x_n\rangle|1\rangle\Big) \\ &= \frac{1}{\sqrt{2^{(n+1)+1}}} \sum_{x_1,...,x_n \in \{0,1\}^{n+1}} (1 + z(-1)^{x_1+...+x_{n+1}}e^{i(\alpha_1+...+\alpha_{n+1})})|x_1x_2...x_{n+1}\rangle\end{aligned}$$

Ce qui correspond exactement à la relation au rang $n+1$.

□

On veut maintenant étudier la distribution de probabilité des mesures, montrons pour cela le lemme suivant. Ce lemme permet de relier la distribution de XOR des mesures des joueurs en fonction des angles α_i choisis :

Lemme 6 *En partant d'un état GHZ à n joueurs, si chaque joueur applique une matrice M_{α_i}, puis mesure son q-bit et retourne le résultat de sa mesure, alors le XOR des sorties est* 0 *avec probabilité* $\cos(\frac{1}{2}\sum_{i=1}^{n}\alpha_i)^2$*, et donc* 1 *avec probabilité* $1 - \cos(\frac{1}{2}\sum_{i=1}^{n}\alpha_i)^2$.

Preuve : On se ramène au cas, équivalent, où les joueurs appliquent tous leur matrice, puis seulement réalisent leur mesure. Après les opérations réalisées par les joueurs, en vertu du lemme précédent (Lemme 5), l'état quantique du jeu est :

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x_1,...,x_n \in \{0,1\}^n} (1 + (-1)^{x_1+...+x_n}e^{i(\alpha_1+...+\alpha_n)})|x_1x_2...x_n\rangle$$

Le XOR des sorties des joueurs est 0 si et seulement si $x_1 \oplus ... \oplus x_n = 0$. Or exactement 2^{n-1} uplets $(x_1, ..., x_n)$ vérifient cette condition, et chacun a pour amplitude $\frac{1+e^{i(\alpha_1+...+\alpha_n)}}{\sqrt{2^{n+1}}}$. Ainsi, chacun de ces états est mesuré avec probabilité $\frac{|1+e^{i(\alpha_1+...+\alpha_n)}|^2}{2^{n+1}}$. Or

$$\begin{aligned}|1 + e^{i(\alpha_1+...+\alpha_n)}|^2 &= |e^{\frac{i}{2}(\alpha_1+...+\alpha_n)}(e^{-\frac{i}{2}(\alpha_1+...+\alpha_n)} + e^{\frac{i}{2}(\alpha_1+...+\alpha_n)})|^2 \\ &= 1 \times |2\cos(\tfrac{1}{2}(\alpha_1 + ... + \alpha_n))|^2 \\ &= 4\cos(\tfrac{1}{2}(\alpha_1 + ... + \alpha_n))^2\end{aligned}$$

Ainsi, les joueurs mesurent l'un des états donnant $\bigoplus_{i=1}^{n} x_i = 0$ avec probabilité $\cos(\frac{1}{2}\sum_{i=1}^{n}\alpha_i)^2$

□

Nous savons maintenant avec quelle probabilité le XOR des réponses des joueurs sera 0 ou 1, en fonction de la somme des angles α_i. On va donc montrer que les valeurs proposées ont été bien choisies, en calculant les valeurs possibles que la somme des angles $\sum_{i=1}^{n}\alpha_i$:

Lemme 7 *Dans la stratégie à n joueurs, les $\alpha_{n,k}$ vérifient :*

$$\begin{cases} \alpha_{n,0} = \dfrac{\pi}{n2^n} \\ \alpha_{n,n-1} = \left(1 - \dfrac{1}{2^n}\right) \dfrac{(-1)^{n-1}\pi}{n} \\ (n-k-1)\alpha_{n,k+1} + (k+1)\alpha_{n,k} = \dfrac{(-1)^{k+1}\pi}{2^n}, \quad 0 \leqslant k < n-1 \end{cases}$$

Preuve : On rappel : $\alpha_{n,k} := \dfrac{(-1)^k\pi}{(k+1)2^n \binom{n}{k+1}} \sum_{j=0}^{k} \binom{n}{k-j}$.

- Le cas $\alpha_{n,0}$ est immédiat.
- Pour $\alpha_{n,n-1}$:

$$\begin{aligned} \alpha_{n,n-1} &= \frac{(-1)^{n-1}\pi}{n2^n \binom{n}{n}} \sum_{j=0}^{n-1} \binom{n}{n-1-j} \\ &= \frac{(-1)^{n-1}\pi}{n2^n} \sum_{j=0}^{n-1} \binom{n}{j} && j \leftarrow n-j-1 \\ &= \frac{(-1)^{n-1}\pi}{n2^n} \left(\underbrace{\sum_{j=0}^{n} \binom{n}{j}}_{2^n} - \binom{n}{n} \right) \\ &= \left(1 - \frac{1}{2^n}\right) \frac{(-1)^{n-1}\pi}{n} \end{aligned}$$

- Enfin, pour la relation entre $\alpha_{n,k}$ et $\alpha_{n,k+1}$:

$$\begin{aligned} (k+1)\alpha_{n,k} &= (k+1)\frac{(-1)^k\pi}{(k+1)2^n \binom{n}{k+1}} \sum_{j=0}^{k} \binom{n}{k-j} \\ &= \frac{(-1)^k\pi}{2^n \binom{n}{k+1}} \sum_{j=0}^{k} \binom{n}{k-j} \\ &= \frac{(-1)^k\pi}{2^n \binom{n}{k+1}} \sum_{j=1}^{k+1} \binom{n}{k+1-j} && j \leftarrow j+1 \end{aligned}$$

$$\begin{aligned} (n-k-1)\alpha_{n,k+1} &= (n-k-1)\frac{(-1)^{k+1}\pi}{(k+2)2^n \binom{n}{k+2}} \sum_{j=0}^{k+1} \binom{n}{k+1-j} \\ &= \frac{(-1)^{k+1}\pi}{2^n \binom{n}{k+1}} \sum_{j=0}^{k+1} \binom{n}{k+1-j} \\ &= -(k+1)\alpha_{n,k} + \frac{(-1)^{k+1}\pi}{2^n} \end{aligned}$$

□

Nous avons maintenant tous les outils pour montrer le résultat :

Théorème 4 *La stratégie précédente permet de gagner avec probabilité* $\cos\left(\frac{\pi}{2^{n+1}}\right)^2$ *quelles que soient les entrées.*

Preuve : Il s'agit simplement de la combinaison des lemmes 6 et 7. Trois cas peuvent se présenter :

- Si toutes les entrées sont des 0, alors chaque joueur appliquera $M_{\alpha_{n,0}}$. Le XOR des sorties sera donc 0 (la bonne réponse) avec probabilité $\cos\left(\frac{1}{2}n\alpha_{n,0}\right)^2$, de par le lemme 6. De plus, selon le lemme 7, $n\alpha_{n,0} = \frac{\pi}{2^n}$. Ce qui conduit au résultat attendu.
- Si toutes les entrées sont des 1. Chaque joueur reçoit alors $n-1$ entrées égales à 1, et applique donc $M_{\alpha_{n,n-1}}$. Comme $n\alpha_{n,0} = \left(1-\frac{1}{2^n}\right)(-1)^{n-1}\pi$, le XOR des sorties est donc 0 avec probabilité $\cos\left(\frac{\pi}{2}-\frac{\pi}{2^n}\right)^2$. Les joueurs échouent donc avec probabilité $\sin\left(\frac{\pi}{2^n}\right)^2$
- Enfin, si parmi les entrées $k+1$ sont des 1, et $n-k-1$ sont des 0, alors $n-k-1$ joueurs vont observer $k+1$ entrées égales à 1, tandis que les $k+1$ autres vont n'en voir que k. Ainsi, $n-k-1$ joueurs appliquent $M_{\alpha_{n,k+1}}$ et $k+1$ appliquent $M_{\alpha_{n,k}}$. Comme $(n-k-1)\alpha_{n,k+1}+(k+1)\alpha_{n,k} = \frac{(-1)^{k+1}\pi}{2^n}$, on obtient à nouveau la probabilité recherchée.

□