Université de Montréal

Variations sur le protocole BB84 avec bases de polarisation secrètes

par

Shany Xiye Gazaille

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

8 février 2023

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Variations sur le protocole BB84 avec bases de polarisation secrètes

présenté par

Shany Xiye Gazaille

a été évalué par un jury composé des personnes suivantes :

Michel Boyer
(président-rapporteur)

Gilles Brassard
(directeur de recherche)

Louis Salvail
(codirecteur)

Alain Tapp
(membre du jury)

Sommaire

Nous naviguons présentement sur la vague de la deuxième révolution quantique qui nous dirige vers un océan de possibilités. L'approche tant attendue de l'ordinateur quantique affecte notre société, notamment la sécurité mondiale actuelle. C'est la course pour mettre à jour nos réseaux de communication pour maintenir le droit à la vie privée. En cryptographie, bien que le chiffrement de message soit crucial pour des échanges privés, la sécurité générale de toute communication repose majoritairement sur la sécurité d'une clé. C'est pourquoi l'établissement quantique de clé ou QKD (de *quantum key distribution* en anglais) est une importante tâche cryptographique qui se doit d'être résistante aux adversaires quantiques.

Beaucoup d'avancées ont déjà été faites dans le domaine, en l'occurrence l'usage de la fibre optique qui a mené à l'implémentation réelle de protocoles QKD. Par contre, l'obstacle qui continue de limiter tout progrès est la distance. Celle-ci hausse exponentiellement les erreurs introduites dans l'échange dépassant facilement les taux maximum tolérés actuels après quelques centaines de kilomètres seulement. De ce fait, bien que la théorie semble prometteuse, la mise en pratique de protocoles quantiques demeure un défi. Pour viser l'application mondiale, nous nous devons de prioriser l'efficacité.

Ce mémoire présente une variation du fameux protocole BB84 pour maximiser la performance des applications de QKD en augmentant le taux d'erreurs toléré et, en l'occurrence, la distance entre les partis. Un satellite sera introduit comme troisième parti. Il aidera Alice et Bob à partager une chaine secrète. Celle-ci leur permettra de rouler le protocole BB84 sans dévoiler les bases. De plus, deux techniques seront définies, soient le filtrage et la concentration. Ces dernières serviront lors de la communication classique interactive pour diminuer l'erreur entre nos deux individus tout en limitant le gain d'information de leur adversaire. Les bénéfices de cette modification sont la possibilité de recycler les bases secrètes du protocole ainsi que la possibilité d'étendre d'avantage la longueur du canal atteignant ainsi l'objectif de pousser les limites pratiques de QKD.

Mots clés: informatique quantique, cryptographie quantique, établissement quantique de clé, QKD, BB84, taux d'erreurs maximum, filtrage, concentration, base de Breidbart.

Summary

We are currently sailing on the second quantum revolution wave towards an ocean of possibilities. The long awaited quantum computer is near and it will affect global security as we know it. It is a race against the clock to update our entire communication network to maintain the right to personal privacy. An important cryptographic task is key establishment. While communicating privately, the entire security lies mainly in the security of the key used. Therefore, it is crucial that future protocols for key establishment be resistant against quantum adversaries.

Over the years, there has been great progress in the field like the practical use of optical fibre leading to quantum key distribution (QKD) protocols implemented in real life. Despite this, a specific obstacle still remains. Distance poses a serious problem as it increases exponentially the amount of errors introduced in the protocol, meaning we easily exceed the maximum rate that we can currently tolerate after only a few hundred kilometers. Hence, what we do in theory may sound promising, but the actual application in reality remains a challenge. To aim for global use, we need to prioritize efficiency.

This thesis suggests an alternative to the renowned BB84 protocol to help maximize applications of quantum key distribution by increasing the tolerated error rate and thus, the distance between two parties. A satellite will be introduced as a third party to help Alice and Bob share a secret bit sequence. This bit string will allow them to run a BB84 protocol without revealing the bases. Then, two techniques will be defined: filtering and concentration. They will serve in the classical communication phase to help lower the error rate between our two parties while also limiting the amount of information gained by the adversary. Benefits from this approach are the recycling of the secret bases of the protocol as well as the possible extension of the length of the channel, thus achieving the end goal of pushing the limits of practical implementation of QKD.

Key words: quantum theory, quantum cryptography, quantum key distribution, QKD, BB84, maximum error rate, filtering, concentration, Breidbart basis.

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et des abréviations

EPR	Einstein–Podolsky–Rosen
i/r	intercepte et renvoie
QKD	Établissement quantique de clé, de l'anglais *Quantum Key Distibution*
QUESS	Quantum Experiment at Space Scale

Remerciements

Je tiens à remercier Gilles Brassard et Louis Salvail sans qui je ne me serais jamais rendue jusqu'ici. Louis, merci pour les excellentes idées, les longs discours de 2h par semaine et tes notes de cours qui m'ont grandement servie [**23**]. Gilles, merci pour avoir partagé ta vaste connaissance, de m'avoir appris à ne pas tourner les coins ronds et de toujours m'avoir écoutée malgré mes explications confuses. Ma façon de penser et mon jugement plus rigoureux ont grandement été influencés par vos apports.

Évidemment, merci à ma famille et à tous mes proches qui m'ont encouragée et soutenue pendant si longtemps malgré la fin qui semblait s'éterniser. Merci à ceux qui ont pris le temps de lire (et essayer de comprendre) ce mémoire.

Enfin, un dernier remerciement à mes collègues qui ont contribué à mon apprentissage et qui ont fait de ce chapitre une belle aventure.

Chapitre 1

Introduction

L'implémentation de protocole QKD a fort progressé depuis l'arrivée de BB84 en 1984 [**4**]. En 1989, peu après sa création, ses créateurs ont réussi à implémenter leur protocole dans une unique pièce avec Alice et Bob séparés par environ 32 cm seulement [**3**]. Étirer la distance entre Alice et Bob demeure un obstacle. Avec l'usage plus répandu de la fibre optique, nous avons poussé la limite jusqu'à atteindre des centaines de kilomètres malgré la diminution exponentielle de la transmission entre Alice et Bob. Plus récemment, nous voyons des expériences avec un satellite. Le voyagement d'information quantique en espace libre n'est pas nouveau. Envoyer un qubit en espace libre signifie l'envoi de photons dans l'air (ou le vide) sans guide d'onde comme la fibre optique. Comme la fibre optique, la transmission de ce canal empire avec la distance. Certains désavantages de l'envoi en espace libre sont aussi de mise au satellite. Par exemple, il est beaucoup plus difficile d'utiliser le canal le jour à cause de la lumière du soleil. De plus, l'expéditeur et le destinataire doivent tous deux être en ligne de vue. La lumière voyage en ligne droite à moins d'être interrompue par des facteurs extérieurs comme des miroirs, des filtres, des trous noirs, etc. Il est donc possible de transmettre des qubits à un satellite en espace libre sous condition que l'expérience soit effectuée la nuit et que le satellite soit assez bas pour maximiser la transmission. La communication avec un satellite reste tout de même pertinente, puisqu'il orbite autour de la Terre. Il se déplace, permettant ainsi de communiquer à diverse endroit sur Terre sans trop varier la distance à l'espace libre entre lui et les stations.

Le satellite chinois *Micius* est le premier de son genre [**16, 29**]. Lancé par l'Académie des sciences chinoise pour leur projet QUESS (de *Quantum Experiment at Space Scale*) en 2016, *Micius* a déjà été l'outil de nombreuses expériences quantiques. Il a pu implémenter avec succès un protocole BB84 entre lui et une station de l'Institut d'optique quantique et d'information quantique à Vienne, en Autriche. Il a fait de même avec une station à Pékin, en Chine. Ensuite, *Micius* a transmis à l'une de ces deux stations le XOR des deux clés. Alors, les deux stations, séparées par 7500km, ont fini par partager la même clé qui

est sure contre tout adversaire *excepté* le satellite [**15**]. Cela pourrait résoudre le problème de la distance, mais cela veut dire qu'aucun message chiffré avec cette clé ne serait sûr contre le satellite. *Micius* a aussi relié deux stations terrestres pour qu'elles complètent un QKD par intrication [**30**]. Cela veut dire que le satellite a réussi à distribuer des demi-paires EPR à chaque station pour qu'elles puissent rouler un QKD sur Terre. Comme il est possible de mesurer et d'appliquer des transformations sur des qubits intriqués, peu importe la distance entre les registres, le vrai défi de cette méthode fut d'envoyer les demi-paires EPR à leur destination en conservant leur intrication. Il importe de souligner que, lors de cet accomplissement, le satellite ne pouvait qu'envoyer ces qubits simultanément, la raison étant qu'il est extrêmement difficile de conserver des qubits à un même endroit. Les deux stations étaient en Chine, voulant dire que nous sommes loin de la distance entre Pékin et Vienne. Nous travaillons généralement avec des photons, donc ceux-ci doivent constamment être en mouvement, ce qui est difficile à réaliser lorsqu'ils sont pris dans un satellite. Pour d'autres types de qubits, les états sont extrêmement instables et nécessitent des conditions particulières et beaucoup de préparation. Le Canada détient présentement le record de la plus longue période de conservation de qubits. L'université Simon Fraser à Vancouver a réussi à maintenir des qubits à température ambiante pendant plus de 30 minutes et jusqu'à 3 heures à température proche du zéro absolu [**22**]. Par contre, leur approche est difficilement réalisable. La préservation d'information quantique n'est donc pas impossible, mais est présentement peu utile en pratique. Il faut aussi considérer les conditions strictes pour le positionnement du satellite. En effet, il doit rester suffisamment proche de la Terre, pour favoriser la qualité de la transmission, tout en ayant les deux stations dans son champ de vision pour envoyer les demi-paires EPR simultanément. Cela veut dire que nous limitons la possibilité d'augmenter la distance maximale actuelle pour toutes actions nécessitant une distribution de qubits intriqués. L'avantage d'utiliser un satellite pour aider nos partis terrestres à rouler un QKD par intrication est le secret de la clé obtenue. Le satellite ne connait pas la clé établie, contrairement à la stratégie décrite plus haut entre Vienne et Pékin.

Mais qu'en est-il si nous utilisions la clé obtenue par les deux instances de QKD de type *prépare et mesure* pour générer d'autres clés entre deux stations terrestres? Le satellite canadien QEYSSat 1.0 prévoit aussi pouvoir rouler des protocoles QKD entre lui et des stations terrestres. Nous ne basons donc pas ce mémoire sur l'existence unique de *Micius*. Notre but ultime est toujours d'augmenter la distance entre nos deux partis terrestres que nous nommerons Alice et Bob. Supposons qu'un satellite s'entend sur une clé sure θ[1] avec Alice en effectuant un protocole QKD de leur choix. Ensuite, ce satellite s'entend sur une autre clé k avec Bob par QKD. Il envoie $\theta \oplus k$ (chiffrement *one time pad*) à Bob. Alice

[1]Pour des raisons qui ne sont pas évidentes pour le moment, mais qui seront plus apparentes sous peu, nous allons appeler la chaine θ et non k.

et Bob se retrouvent donc avec une clé commune θ sure contre tout adversaire excluant le satellite. Cette clé pourrait être utilisée pour chiffrer un message, mais ce ne serait pas nouveau. Nous ne voulons pas que le satellite connaisse la clé qui servira à chiffrer les messages entre Alice et Bob. De plus, le satellite est un troisième parti qui orbite autour de la Terre. Nous ne voulons pas lui faire appel à chaque chiffrement. Nous voulons donc réutiliser θ astucieusement pour générer plusieurs clés. Nous l'utiliserons dans une version modifiée de BB84 qui nous permettra de le recycler et de créer des clés plus sures contre le satellite. Nous espérons aussi augmenter le taux d'erreurs maximal toléré par un QKD pour allonger la distance possible entre nos deux partis terrestres.

En utilisant θ comme séquence de bases, Alice prépare une chaine x en états BB84. Puisque θ est déjà connu par Alice et Bob, ce dernier peut faire la bonne mesure *à tous les coups*. Les bases ne sont jamais divulguées, donnant ainsi une tournure intéressante au traditionnel protocole BB84. Ève peut tout de même attaquer lors de la communication quantique. Des erreurs sont introduites du côté de Bob et d'Ève. L'établissement de clé est possible même avec la présence d'erreurs chez les trois partis. Le wiretap channel introduit par Wyner [**28**] suit ce concept en supposant que l'adversaire a un canal plus bruité que celui de Bob. Il propose un modèle où l'adversaire *wiretap* la communication reçue par Bob, ajoutant ainsi un second canal qui peut introduire des erreurs supplémentaires. Alors, Bob reçoit l'information d'Alice avec moins d'erreurs qu'Ève. Par contre, cela force une dépendance entre le canal de Bob et d'Ève. Csiszár et Körner [**11**] ont allégé les conditions de Wyner en présentant un modèle où le canal d'Ève ne doit pas nécessairement découler du canal bruité de Bob. Ils généralisent la condition, en assurant la transmission d'un secret de façon sécuritaire si le canal d'Ève est plus bruité que celui de Bob. Nous suivons plutôt Maurer qui a montré l'existence de protocoles d'établissement de clé sûrs où Ève peut avoir moins d'erreurs qu'Alice et Bob [**17**]. Il est possible de s'entendre sur une clé finale grâce à de la communication interactive via un canal public classique authentifié pour diminuer l'erreur entre eux tout en limitant l'information divulguée à Ève. Cela nous permet d'alléger les hypothèses irréalistes où Bob doit avoir un canal plus fidèle qu'Ève pour qu'il s'entende sur une clé finale avec Alice. Des protocoles QKD comme BB84 utilisent déjà de la communication classique interactive pour générer une clé sure [**4**]. Il est possible de prouver que le protocole BB84 traditionnel reste sûr malgré un taux d'erreurs allant jusqu'à environ 11%. Nous ne savons pas si ce taux peut être augmenté, mais il ne peut dépasser 25%. Dans notre version modifiée, nous gardons les bases de mesure secrètes. Avec cet avantage, nous dépasserons naturellement la borne du 11%. Ce qui nous intéresse est plutôt de dépasser la borne de Gottesman et Lo, soit 18.9%, qui est présentement la plus haute borne à notre connaissance pour des protocoles QKD avec qubits BB84 suivi d'une correction interactive [**13**]. Nous utilisons deux techniques. Nous appelons la première, le *filtrage*. Cette méthode commence, tout d'abord, en permutant les chaines d'Alice et Bob publiquement. Ensuite,

elle a pour but de diminuer le taux d'erreurs présent dans la chaine obtenue chez Bob en annonçant les parités de chaque deux bits successif et en ne conservant qu'un seul bit parmi les paires dont la parité concorde. Puis, nos deux partis procèdent à l'étape de *concentration*[2]. Ils permutent encore leurs chaines, effectuent un XOR sur chaque paire de bits et conservent les résultats. Ainsi, l'incertitude d'Ève augmente. Suite à ces deux méthodes, ils font un échantillonnage d'erreurs qui détermine s'il est possible de corriger le taux d'erreurs. Si c'est le cas, supposons que la chaine corrigée est $\hat{x}$. Alice et Bob s'entendent sur une fonction de hachage g qui extrait une clé K de $\hat{x}$. La fonction g est tirée d'une classe de hachage universelle$_2$. Ce que nous espérons, c'est que malgré un taux d'erreurs plus élevé que 18.9%, nous pouvons tout de même établir une clé K d'au moins 1 bit. Par conséquent, la distance possible entre nos deux partis augmente.

Ce mémoire analyse ce protocole en restreignant l'adversaire. Nous supposons qu'Ève monte des attaques *intercepte et renvoie* dans la base de Breidbart seulement. Premièrement, nous choisissons cette attaque spécifique puisqu'elle simplifie énormément notre analyse de sécurité du protocole. Deuxièmement, la base de Breidbart offre un meilleur avantage que la base rectilinéaire ou diagonale même si l'information est probabiliste. En effet, puisqu'Ève ne connait pas la base utilisée par Alice et Bob, la mesure de Breidbart sur un qubit indépendant offre plus d'information à Ève que n'importe quelles autres bases de mesure. Elle lui permet donc d'en apprendre davantage sur la chaine d'Alice sans obtenir de l'information sur les bases. Nous ne prouvons cependant pas que les attaques du type intercepte et renvoie sont les meilleures attaques qu'Ève peut faire.

Grâce à cette attaque, notre adversaire gagne un maximum d'information sur chaque bit de la chaine d'Alice. Cependant, elle reste ignorante de la séquence de bases et ce, peu importe l'information qu'elle accumule avec la mesure de Breidbart. Puisque la min-entropie sur la séquence de bases reste toujours maximale, Alice et Bob peuvent rouler plusieurs fois le protocole en conservant le même θ à chaque fois. Il en découle que nos partis n'ont pas à dépendre du satellite à chaque établissement de clé sur Terre.

La présence d'un troisième parti allié nous force à définir des hypothèses de sécurité supplémentaires. Il n'est pas nécessaire de faire parfaitement confiance au satellite. Nous supposons seulement qu'il n'intervienne pas lors de la communication quantique entre Alice et Bob. Il est libre d'écouter tout échange sur le canal classique puisque celui-ci est public. La première restriction rend sa connaissance de θ inutile. La partie interactive classique ne dévoile que les parités des paires de bits. Ainsi, savoir les bases exactes de chaque qubit ne lui permet pas de déduire quoi que ce soit sur la chaine de bits chez Alice ou Bob. Il est aussi pertinent de souligner que si l'information du satellite est compromise, que ce soit

[2]La concentration n'est pas une méthode interactive, puisqu'elle nécessite seulement l'annonce des permutations par Alice ou Bob. Nous utilisons souvent le terme *correction interactive* pour parler du filtrage et de la concentration. C'est une façon de simplifier le texte.

volontaire ou non, toutes clés établies antérieurement resteront sures. Puisque nous limitons les attaques dans la base de Breidbart, divulger θ n'affecte que la sécurité des clés ultérieures si Alice et Bob conservent le même θ.

Sous ces différentes conditions, nous analyserons la sécurité du protocole proposé. Tout d'abord, nous montrerons que, lorsque les bases restent secrètes, nous pouvons trouver une meilleure borne sur le taux d'erreurs toléré par BB84 (11%), soit 13.72%. Ensuite, nous varierons la partie interactive classique pour pousser encore plus la distance maximale entre Alice et Bob. Notre borne est de 19.03%, ce qui dépasse la borne de Gottesman et Lo (18.9%). Nous verrons aussi qu'il serait possible d'atteindre la borne maximale de 25% avec un peu plus de travail.

Chapitre 2

Notions de base

2.1. Informatique quantique

2.1.1. Qubit

Toute information peut être représentée par des 0 et des 1 que nous appelons des bits. C'est ce que nous pouvons appeler *l'information classique*. Les ordinateurs sont des machines capables de traiter cette information. Par contre, la vitesse à laquelle cette information est traitée limite nos possibilités en pratique. Avec l'arrivée de la théorie quantique, le monde a vu apparaître la notion de qubit, présentant ainsi une nouvelle perception de l'information. Après la théorie vient la mise en pratique. Bien que les ordinateurs quantiques actuels sont loin d'être de calibre à effectuer ce que nous espérons d'eux, la simple possibilité de ce qu'ils pourront faire a mené à découvrir des solutions à des problèmes qui nous semblaient pendant longtemps impossibles. Mais qu'est-ce que la théorie quantique? Elle repose tout d'abord et avant tout sur une unité spectaculaire, le qubit. Le qubit est une unité qui n'est pas nécessairement limitée à une des deux valeurs binaires. On le définit par une superposition de deux états, ou encore comme un vecteur d'un espace de Hilbert de dimension 2 sur les complexes.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \text{où } \alpha, \beta \in \mathbb{C},\ |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ et } |\alpha|^2 + |\beta|^2 = 1.$$

Un qubit est donc une superposition linéaire des deux vecteurs $|0\rangle$ et $|1\rangle$ normalisés aux composantes complexes. Nous appelons ces deux coefficients les amplitudes. Nous utiliserons souvent les qubits $|0\rangle$ et $|1\rangle$ qui ont une seule amplitude de 1. Une autre paire de qubit qui sera fréquemment utilisée est

$$|+\rangle = \frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle \ \text{ et } \ |-\rangle = \frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle\ .$$

Dans le cadre de ce mémoire, les amplitudes seront toujours des coefficients réels. Comme un qubit est un vecteur normalisé, nous pouvons le visualiser comme un point sur le périmètre d'un cercle de rayon 1 (voir fig. 2.1). Une autre façon de définir un qubit est donc par son angle θ. Ses amplitudes seraient alors $\alpha = \cos(\theta)$ et $\beta = \sin(\theta)$.

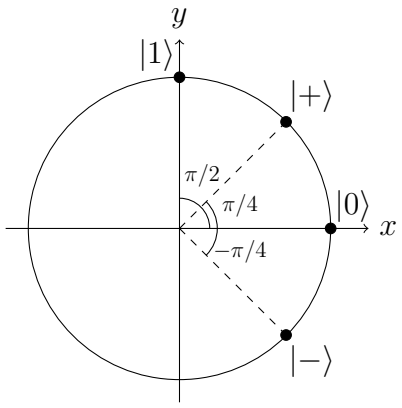


Fig. 2.1. États principaux.

2.1.2. Mesure et opération unitaire

Définition 2.1.1. *Une* base orthonormée $\{v_1, v_2, ..., v_n\}$ *d'un espace euclidien de dimension* n *est un sous-ensemble de* n *vecteurs tel que*

$$\forall\ 1 \leq i \leq n,\ 1 \leq j \leq n,\ \langle v_i, v_j \rangle = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases},$$

où $\langle ., . \rangle$ *est le produit scalaire.*

Lorsque vient le temps de manipuler des qubits, les propriétés quantiques de ceux-ci deviennent extrêmement importantes. Deux actions peuvent être effectuées: une *mesure* ou une *opération unitaire*. Dans le premier cas, nous perturbons le qubit. Il en découle une perte irréversible de l'état quantique dans lequel le qubit se trouvait. C'est ce que nous appelons la *réduction* d'un qubit. Bien que cette caractéristique semble défavorable, il y a une certaine structure qui suit ce phénomène. Nous devons tout d'abord définir la base de mesure. C'est une base orthonormée. Les vecteurs d'une base de mesure sont les états possibles résultant de cette mesure. Ainsi, lorsque nous mesurons un qubit dans une certaine base, le qubit se réduit à l'un de ces états. De ce fait, l'état original, en l'occurrence ses amplitudes, est perdu. Par exemple, si nous mesurons $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ dans la base $\{|0\rangle, |1\rangle\}$, nous aurons $|0\rangle$ ou $|1\rangle$ comme résultat. Dans ce cas précis, nous dirons que la mesure retourne un résultat classique, soit 0 ou 1. Puisqu'un qubit est une superposition d'états, nous perdons cet ensemble d'information lors d'une mesure. Ce phénomène est appliqué dans de nombreux protocoles cryptographiques puisqu'on nous garantit que l'information quantique ne peut pas

être lue avec certitude sans perturber irréversiblement le système. Comme mentionné plus haut, il y a une certaine règle à laquelle les qubits adhèrent lorsqu'ils sont mesurés. Si nous mesurons dans la base $\{|0\rangle, |1\rangle\}$, les amplitudes α et β caractérisent le comportement du qubit. Il est donc possible d'établir une distribution de probabilité des résultats possibles. Effectivement, un qubit se réduit à l'état 0 avec probabilité α^2 et à l'état 1 avec probabilité β^2. Bien que cette distribution nous offre de l'information sur le résultat d'une mesure, on ne peut en dire autant lorsque nous y allons de l'autre sens. Lorsque nous n'avons que le résultat classique d'une mesure, il nous est très difficile d'obtenir de l'information sur la superposition initiale. Par exemple, savoir qu'un qubit se réduit à 1 peut vouloir dire que la superposition quantique était $|1\rangle$, mais aussi qu'elle pouvait être $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ pour n'en nommer que deux seules. La seule information sure que nous pouvons en tirer est que la superposition ne pouvait être $|0\rangle$ puisque sa probabilité de donner 1 est de $\beta^2 = 0$. Il reste toutefois une infinité de possibilités de superpositions. Si nous n'avons aucune information a priori sur α et β, alors ces états sont tous possibles du point de vue de l'observateur.

Il est possible de manipuler des qubits sans perdre leur information. La deuxième action possible est l'application d'une transformation unitaire. C'est une opération sur un état qui est réversible. Nous ne gagnons pas de l'information sur des qubits en les transformant. Leurs états initial et résultant restent inconnus.

2.1.3. Base de mesure

Nous avons introduit la notion de mesure, mais seulement pour la base $\{|0\rangle, |1\rangle\}$ que nous appelons la *base rectilinéaire*. Une autre base très utilisée est la *base diagonale*, soit $\{|+\rangle, |-\rangle\}$. Comme mentionné plus haut, le résultat d'une mesure est l'un des états de la base choisie. Donc si nous mesurons $|0\rangle$ dans la base diagonale, nous aurons $|+\rangle$ ou $|-\rangle$. Rappelons que les probabilités de se réduire à l'un des deux états de base $|0\rangle$ et $|1\rangle$ sont $\alpha^2 = \cos^2(\theta)$ et $\beta^2 = \sin^2(\theta)$ respectivement. Si nous mesurons dans une base autre que la rectilinéaire, la probabilité d'obtenir l'un des états de la base dépendra de la différence entre l'angle θ du qubit et celui de l'état de la base. Nous pouvons donc généraliser pour une base et un qubit arbitraires. Si nous mesurons un qubit $|\psi\rangle = \cos(\theta)|0\rangle + \sin(\theta)|1\rangle$ dans la base $\{b_1, b_2\} = \{\cos(\phi)|0\rangle + \sin(\phi)|1\rangle, \cos(\phi + \pi/2)|0\rangle + \sin(\phi + \pi/2)|1\rangle\}$, nous obtiendrons

$$b_1 \text{ avec probabilité } \cos^2(\theta - \phi) \ ,$$
$$b_2 \text{ avec probabilité } \cos^2(\theta - (\phi + \pi/2)) = \sin^2(\theta - \phi) \ .$$

Par exemple, si nous mesurons $|1\rangle$ dans la base $\{|+\rangle, |-\rangle\}$, le résultat sera $|+\rangle$ avec probabilité $\cos^2(\pi/2 - \pi/4) = 1/2$ et $|-\rangle$ avec probabilité $\sin^2(\pi/2 - \pi/4) = 1/2$.

2.1.4. Notation de Dirac

Le symbole $|.\rangle$ est ce que nous appelons *ket*. Soit,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \ .$$

Son conjugé transposé s'appelle *bra* et est défini comme suit,

$$|\psi\rangle^\dagger = \langle\psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}$$

où l'étoile sur les amplitudes indique le conjugé des nombres complexes. Bien entendu, nous n'en tiendrons pas compte, puisque nos qubits auront des amplitudes réelles. Lorsqu'on ajoute un qubit, nous obtenons un vecteur deux fois plus long. Par exemple,

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

forment une base orthonormée pour l'espace de dimension 4. Deux qubits peuvent donc être écrits comme suit,

$$|\Psi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \ .$$

Un seul qubit est la superposition des deux bits classiques 0 et 1 et de même, deux qubits sont la superposition de toutes les paires de bits classiques possibles (00, 01, 10 et 11). Bien entendu, pour que la superposition soit bien définie, les amplitudes doivent satisfaire $a^2 + b^2 + c^2 + d^2 = 1$. La notation se généralise pour n qubits. Les vecteurs sont alors dans un espace de Hilbert de dimension 2^n.

$$|\Gamma\rangle = \sum_x \alpha_x\,|x\rangle\,, \text{ où } x \in \{0,1\}^n \text{ et } \alpha_x \in \mathbb{C}. \tag{2.1.1}$$

Chaque amplitude permet de définir la probabilité que les n qubits se réduisent à l'état associé s'ils sont mesurés dans la base rectilinéaire (plus précisément, α_x^2). De plus,

$$\|\,|\Gamma\rangle\,\|^2 = 1\ ,$$

où $\|.\|$ est la norme euclidienne. Un *registre quantique* est un espace mémoire qui peut être modélisé par un espace de Hilbert. Il peut donc contenir n qubits formant un espace de dimension 2^n.

Nous dénotons par $\mathcal{L}_n(\mathbb{C})$ l'espace des opérateurs de dimensions $n \times n$ sur les complexes et par $\mathcal{L}_{m,n}(\mathbb{C})$ l'espace des opérateurs de dimensions $m \times n$ sur les complexes. L'ensemble $\mathbb{C}$ sera parfois remplacé par $\mathbb{R}$ ou par $GF(2)$ pour définir l'espace des opérateurs sur ces ensembles. L'ensemble $GF(2)$ est le corps de Galois composé des deux éléments 0 et 1 dont l'addition est le "$\oplus$" et la multiplication est le "$\wedge$" logique. De plus, 0 est l'identité additive et 1 est l'identité multiplicative.

Définition 2.1.2. *Soit $A = (a_{ij})_{i,j} \in \mathcal{L}_{m,n}(\mathbb{C})$ et $B \in \mathcal{L}_{p,q}(\mathbb{C})$. Le* produit tensoriel *de A et B est défini par*

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} .$$

Le produit tensoriel est une opération sur les registres vers un plus gros registre. Par exemple:

$$\begin{aligned} |\psi\rangle \otimes |\varphi\rangle = |\psi\rangle |\varphi\rangle = |\psi\varphi\rangle &= (\alpha_1 |0\rangle + \beta_1 |1\rangle)(\alpha_2 |0\rangle + \beta_2 |1\rangle) \\ &= \alpha_1\alpha_2 |00\rangle + \alpha_1\beta_2 |01\rangle + \beta_1\alpha_2 |10\rangle + \beta_1\beta_2 |11\rangle \ . \end{aligned}$$

Il est parfois possible de séparer un registre en plusieurs registres de plus petite dimension comme suit:

$$\bigotimes_{i=0}^{n} |\varphi_i\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle \otimes ... \otimes |\varphi_n\rangle \ . \tag{2.1.2}$$

Lorsque nous avons plusieurs registres quantiques, nous les identifions clairement par des lettres calligraphique comme $\mathcal{X}$. Cela est utile lorsque nous effectuons des calculs sur des qubits d'un registre spécifique, telles que des mesures ou des transformations unitaires, ou lorsque nous voulons distribuer des qubits à différents partis dans un protocole cryptographique. À titre d'illustration, nous pouvons séparer les n qubits décrits à (2.1.2) en deux registres $\mathcal{A}$ et $\mathcal{B}$ où $\mathcal{A}$ est le registre contenant les états de 1 à i et où $\mathcal{B}$ contient les états de $i+1$ à n. Il est donc possible d'agir uniquement sur le registre $\mathcal{A}$ en faisant, par exemple, une mesure. Lorsque ce sera pertinent, nous indiquerons en indice le registre auquel un état appartient.

Il y a des cas spéciaux où n qubits ne sont pas séparables, c'est-à-dire il n'est pas possible de réécrire l'état comme un produit tensoriel de n états. C'est ce que nous appelons l'intrication. Les quatre états de Bell sont de fameux exemples. Nous les appelons aussi *paires*

EPR.

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \ .$$

L'intrication est souvent utilisée en cryptographie puisqu'elle promet une corrélation parfaite entre deux partis qui ne peut être reproduite par des moyens classiques. Rappelons que les amplitudes aident à calculer la probabilité qu'un qubit se réduise sur l'état correspondant. Un registre de n qubits est décrit de la même façon. Par exemple, si nous prenons le registre $|+\rangle \otimes |+\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$, nous avons deux qubits qui se réduiront à 00, 01, 10 ou 11 avec la même probabilité. On note aussi que l'état classique résultant du premier qubit ne prédit pas l'état classique résultant du deuxième. En d'autres mots, même si le premier qubit se réduit à 0, le deuxième peut tout de même se réduire à 0 ou 1. L'intrication, quant à elle, ajoute une corrélation entre les différents bits classiques résultant d'une mesure d'un registre de qubits intriqués. Prenons l'état de Bell $|\Phi^+\rangle$, un registre de deux qubits parfaitement intriqués. L'état se réduira à 00 avec probabilité 1/2 et à 11 avec probabilité 1/2. L'intrication en soit ne nous permet pas de mieux prédire les résultats d'une mesure. Elle nous garantit plutôt un lien entre les résultats des qubits. Si nous savons que le premier qubit s'est réduit à 0, alors il est absolument certain que le deuxième s'est aussi réduit à 0 puisque les seules possibilités de résultats sont 00 et 11.

Pour mesurer des qubits, nous avons ce que nous appelons des *mesures projectives.*

Définition 2.1.3. *Une* mesure projective *est un ensemble d'opérateurs de mesure* $\{M_i\}$*, où* $M_i \in \mathcal{L}_n(\mathbb{C})$*, satisfaisant les conditions suivantes:*

- $\forall i,\ M_i = M_i^\dagger = M_i^2$*;*
- $\forall i, j$*, si* $i \neq j$*,* $\mathrm{tr}(M_i M_j) = 0$*;*
- $\sum M_i = 1_n$.

Si on mesure l'état $|\varphi\rangle$, la probabilité d'obtenir le résultat i est

$$p_i = \langle\varphi| M_i |\varphi\rangle$$

et le résultat de la mesure est

$$\frac{M_i |\varphi\rangle\langle\varphi| M_i}{p_i} \ .$$

Une mesure projective est l'équivalent de la base de mesure expliquée plus haut et les résultats de mesures sont les matrices densité des états de la base. Définissons les mesures

projectives pour la base rectilinéaire $\{M_0, M_1\}$ et la base diagonale $\{M_+, M_-\}$.

$$M_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad M_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} ,$$

$$M_+ = |+\rangle\langle +| = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad M_- = |-\rangle\langle -| = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} .$$

Si nous mesurons le qubit $|\varphi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ avec chaque mesure projective, nous avons:

$\{M_0, M_1\}$:

Probabilités : $p_0 = \langle\varphi|\, M_0\, |\varphi\rangle = \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha^2$,

$$p_1 = \langle\varphi|\, M_1\, |\varphi\rangle = \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \beta^2 .$$

Résultats : $\dfrac{M_0\,|\varphi\rangle\langle\varphi|\, M_0}{p_0} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \Big/ \alpha^2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = |0\rangle\langle 0|$,

$$\frac{M_1\,|\varphi\rangle\langle\varphi|\, M_1}{p_1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \Big/ \beta^2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = |1\rangle\langle 1| .$$

$\{M_+, M_-\}$:

Probabilités : $p_+ = \langle\varphi|\, M_+\, |\varphi\rangle = \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \dfrac{1}{2} + \alpha\beta$,

$$p_- = \langle\varphi|\, M_-\, |\varphi\rangle = \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{1}{2} - \alpha\beta .$$

Résultats : $\dfrac{M_+\,|\varphi\rangle\langle\varphi|\, M_+}{p_+} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \Big/ p_+ = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} = |+\rangle\langle +|$,

$$\frac{M_-\,|\varphi\rangle\langle\varphi|\, M_-}{p_-} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \Big/ p_- = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = |-\rangle\langle -| .$$

Nous retrouvons les mêmes résultats et probabilités présentés dans la sous-section 2.1.3. Rappelons que les mesures sont irréversibles et imprévisibles. Lorsqu'on mesure un état, si au moins deux des p_i de la mesure projective est non nul, alors on ne peut savoir quel sera l'état résultant. Une mesure projective ne nous donne que la liste des états résultants possibles ainsi que la distribution de probabilité d'obtenir chacun d'entre eux.

Des états peuvent être parfaitement distinguables ou non entre eux. Lorsque deux états sont parfaitement distinguables, ils sont orthogonaux. Cela veut dire qu'il existe une mesure projective capable de les différencier à tous les coups. Par exemple, $|+\rangle$ et $|-\rangle$ sont parfaitement distinguables. Si nous avons un des deux, mais que nous ne savons pas lequel, il suffit d'appliquer la mesure projective $\{M_+, M_-\}$ qui retournera un résultat sans perdre d'information sur son état initial. En effet, si $|\varphi\rangle = |+\rangle$, alors $p_+ = 1$ et $p_- = 0$ (vice versa pour $|-\rangle$). Naturellement, deux états non-orthogonaux ne peuvent être systématiquement distingués. Par exemple, les états $|0\rangle$ et $|+\rangle$, ne peuvent pas être distingués parfaitement. Il y aura toujours une probabilité d'erreurs lors d'une mesure. Il va de même pour $|1\rangle$ avec $|+\rangle$ et $|-\rangle$. Ces propriétés sont très utiles pour des protocoles comme BB84.

2.1.5. Distance de trace et trace partielle

Définition 2.1.4. *La* trace-norme *d'un opérateur hermitien $A \in \mathcal{L}_n(\mathbb{C})$ est*

$$\|A\|_1 = \sum_{i=1}^{n} |\lambda_i| \ ,$$

où $\lambda_1, \lambda_2, ..., \lambda_n$ sont les valeurs propres de A.

Définition 2.1.5. *Soit ρ et σ, deux opérateurs hermitiens. La* distance de trace *entre les deux est*

$$\Delta(\rho,\ \sigma) := \frac{\|\rho - \sigma\|_1}{2} \ .$$

Pour montrer qu'un état est très *proche* d'un autre, nous bornerons supérieurement leur distance de trace.

Définition 2.1.6. *Soit $\mathcal{X}$ et $\mathcal{Y}$, deux registres quantiques et l'état $\rho_{\mathcal{XY}} \in \mathcal{X} \otimes \mathcal{Y}$. La* trace partielle *de $\rho_{\mathcal{XY}}$ sur le registre $\mathcal{Y}$ est défini par:*

$$\mathrm{tr}_{\mathcal{Y}}(\rho_{\mathcal{XY}}) := \sum_i \langle v_i| \, \rho_{\mathcal{XY}} \, |v_i\rangle$$

où $\{|v_i\rangle\}_i$ est une base orthonormée sur le registre $\mathcal{Y}$.

2.1.6. Impossibilité de clonage

L'une des plus importantes propriétés de l'information quantique est l'impossibilité de copier des qubits [**18, 27**]. Elle nous est particulièrement utile dans certains protocoles QKD de type *prépare et mesure* comme BB84.

Théorème 2.1.7 (Théorème d'impossibilité de clonage quantique)**.** *Il n'existe aucune opération permise par la théorie quantique qui, pour un $|\varphi\rangle$ arbitraire, puisse transformer*

$|\varphi\rangle \otimes |0\rangle$ *en* $|\varphi\rangle \otimes |\varphi\rangle$.

Il ne faut pas confondre ce principe avec la *téléportation quantique* qui nous permet de *transférer* l'état d'un registre à un autre registre grâce à l'intrication de qubits [**5**]. Cela veut dire que nous ne retrouvons plus l'état initial dans le registre d'où il provient une fois la téléportation complétée.

Il existe des machines de clonage [**8, 7**]. Celles-ci ne peuvent que produire des copies imparfaites de qubits. Le théorème de non-clonage tient toujours. Ces machines peuvent être pertinente dans les analyses de sécurité de certains protocoles cryptographique comme BB84.

2.1.7. Interprétation physique

La nature d'un qubit n'est pas affectée par l'ajout d'un facteur complexe η de norme 1. Cela veut dire qu'il n'existe pas de mesure qui permettrait de détecter la présence d'un tel facteur. En l'occurrence, $|\psi\rangle$ est indiscernable de $\eta\,|\psi\rangle$ et nous appelons η la *phase globale sans importance* (*irrelevant global phase factor* en anglais). Par exemple, le qubit $|0\rangle$ est le même que $-\,|0\rangle$, c'est-à-dire le qubit représenté par l'angle 0 est le même que celui représenté par l'angle π. Un exemple très commun d'un qubit en pratique est un photon polarisé. La polarisation de la lumière est analogue aux états de qubits. Par exemple, $|0\rangle$ représente un photon polarisé horizontalement (voir fig. 2.2).

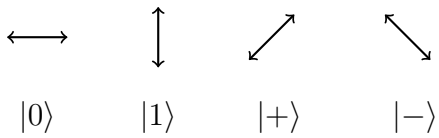


Fig. 2.2. Photon polarisé horizontalement, verticalement, diagonale positive et diagonale négative.

Un exemple d'appareil de mesure pour les photons est un filtre de polarisation suivi d'un détecteur. Un photon passant par ce filtre traversera avec certitude s'il a la même polarisation que le filtre. Sinon, il passera selon les lois de la physique quantique décrites plus haut.

En théorie, nous utilisons le terme *canal* pour représenter la "route" sur laquelle l'information circule. Les canaux classiques transmettent de l'information classique. En pratique, les possibilités sont nombreuses comme un courant électrique dans un fil de cuivre ou même le son de votre voix qui voyage dans l'air. Les canaux quantiques sont plus limités puisqu'ils doivent pouvoir faire circuler des particules aux propriétés quantiques comme des photons. Un exemple largement utilisé de nos jours est la fibre optique puisque celle-ci transmet la

lumière. Par contre, ses applications sont limitées. Les photons sont fragiles, résultant en des pertes d'information ou même des résultats erronés plus la distance de voyagement augmente. Un autre type de canal quantique est *l'air libre* (ou *free space* en anglais). La lumière n'a pas nécessairement besoin d'un canal physique pour se déplacer. Le désavantage de ce type de canal est la hausse d'exposition aux facteurs environnementaux comme la lumière du soleil et les particules dans l'air. De plus, la lumière voyage en ligne droite seulement.

Outre les photons, il est possible d'utiliser des particules pour décrire des qubits tels que des électrons ou des noyaux d'atomes par leur nombre quantique magnétique de spins. Il est cependant plus facile de visualiser des qubits par des photons et leur polarisation dans le cadre de ce travail.

En pratique, des registres peuvent être physiquement distancés. Comme expliqué plus haut, n qubits peuvent être regroupés en registres que nous pouvons ensuite manipuler séparément même s'il y a de l'intrication entre les registres. Par exemple, il est possible d'appliquer une mesure sur la moitié d'un état de Bell à un point A ce qui donne l'illusion d'affecter l'état de l'autre moitié à un point B. Il est d'ailleurs possible de physiquement séparer des registres d'une distance théoriquement infinie sans perdre l'intrication. Certains protocoles QKD se servent de cette propriété pour générer de l'information commune entre deux partis éloignés. Le gros avantage de ce scénario est que l'information obtenue n'existe pas lors de la distanciation ou le déplacement des registres, qui est le moment le plus à risque pour des attaques d'adversaires.

Notons qu'en réalité, les appareils utilisés sont imparfaits. La génération et la détection de photons comportent des erreurs. À titre d'exemple, il est possible qu'un détecteur de photons signale faussement une réception ou encore qu'un générateur de photons en envoie deux par erreur. Nous ne tiendrons pas compte de ce genre d'imperfections dans ce travail.

2.1.8. États classiques

Nous pouvons ranger de l'information classique dans un registre. Nous dirons alors que c'est un *registre classique* et le noterons par une lettre linéale (par exemple: X, Y ou Z). Un registre classique est représenté par une matrice densité diagonale dans laquelle est rangée la distribution de probabilité d'une variable aléatoire X.

Définition 2.1.8. *Un* registre classique X *contenant la variable aléatoire* $X \in \mathbf{X}$ *est définie comme suit*

$$\rho_{\mathsf{X}} = \sum_{x \in \mathbf{X}} P_X(x) \, |x\rangle\langle x|_{\mathsf{X}} \ .$$

Nous pouvons avoir des registres de différents types dans un même état. Un état cq comporte un registre classique et un registre quantique. Un état ccq comporte deux registres classiques et un registre quantique, etc.

2.2. Cryptographie

2.2.1. Établissement de clé

La cryptographie s'intéresse aux façons de garantir la confidentialité et l'intégrité des communications. Les noms Alice et Bob sont utilisés pour identifier les deux individus qui tentent de communiquer entre eux en toute sécurité malgré la présence d'un adversaire nommé Ève (venant de *Eavesdropper* en anglais). En cryptographie, une méthode de chiffrement est un protocole ou un algorithme qui prend un message quelconque et le transforme en "texte chiffré" à l'aide d'une clé. Il existe deux types, soient les chiffrements symétriques et les chiffrements asymétriques. Ceux du premier type sont aussi appelés chiffrements à clé privée. Ils chiffrent des messages selon une clé secrète partagée par Alice et Bob. Il est aussi possible de déchiffrer le message chiffré avec la même clé. Les chiffrements asymétriques, aussi appelés chiffrements à clé publique, utilisent deux clés, soient une privée et une connue de tous.

Les chiffrements à clé privée transforment un message d'Alice en un texte chiffré grâce à une clé. Ces méthodes de chiffrement sont réversibles et aident à cacher le contenu des messages d'un adversaire quelconque. En d'autres mots, lorsque Bob reçoit un texte chiffré, il peut retrouver le message original avec l'aide de la clé tout en limitant l'information divulgée à Ève. Les chiffrements à clé publique ont une structure différente, mais ne seront pas pertinents pour ce mémoire.

Par le principle de Kerckhoff, toutes les méthodes de chiffrement doivent être sures même si tous les détails concernant la procédure, excepté la clé, sont connus publiquement [**14**]. La sécurité du protocole entier repose donc sur la clé et la façon dont le protocole utilise son secret, ce qui rend l'établissement de clé une tâche cruciale pour maximiser l'ignorance de l'adversaire. L'établissement de clé est une tâche cryptographique qui précéde une méthode de chiffrement. C'est un protocole à part qui génère la clé privée qui sert à compléter le chiffrement des messages. Une clé est définie par une chaine qui entre en paramètre dans une méthode de chiffrement. C'est sur le secret de ce paramètre qu'est basée la sécurité du protocole puisque les codes sont supposément connus des adversaires. Un exemple de protocole de chiffrement est le *masque jetable* (*one time pad* en anglais). La méthode de chiffrement est simplement l'addition bit par bit du message m avec une chaine aléatoire k, c'est-à-dire $c := m \oplus k$. Le résultat est donc une chaine c qui semble parfaitement aléatoire aux yeux d'Ève assurant ainsi la sécurité du protocole. Ce type de chiffrement a été prouvé sure par Claude Shannon dans la mesure où connaître la chaine c ne dévoile aucune information supplémentaire sur le message m sauf sa longueur [**25**]. Il est facile pour Bob de retrouver le message original $c \oplus k = (m \oplus k) \oplus k = m$. Dans ce contexte, le message chiffré est "$m \oplus k$" et k est la clé. Il est important de comprendre que seule la clé est inconnue d'Ève.

Elle sait comment le message est chiffré, pourtant cela ne l'aide pas à violer la sécurité du protocole. Elle pourrait essayer de faire un XOR de la chaine c avec toutes les clés possibles. Cependant, même si elle réussit à retrouver le message en clair, elle n'aura aucun moyen de savoir si c'est bien le message envoyé par Alice, puisque tous les autres messages clairs de même longueur peuvent s'expliquer par une clé appropriée. Cela veut dire qu'elle ne peut faire mieux que choisir une chaine aléatoire de même longueur que le message pour deviner la chaine d'Alice.

Ce mémoire sera centré sur l'établissement quantique de clé, soit l'étape précédant un chiffrement de message et utilisant les propriétés de la physique quantique. Comme nous venons de le voir, il est simple de chiffrer sécuritairement un message en autant que la clé même soit sure vis-à-vis un adversaire. La sécurité d'un algorithme de chiffrement n'est pas la même chose que la sécurité de la clé de chiffrement. Nous définissons la sécurité d'un protocole d'établissement de clé par son indistinguabilité d'un protocole idéal. Bien entendu, nous définissons le tout dans un contexte où Alice et Bob sont contre un adversaire quantique. La cryptographie post-quantique base sa sécurité sur l'existence d'adversaire possédant un ordinateur quantique.

Définition 2.2.1. *Soit $\Pi_{n,k}$, un protocole d'établissement de clé avec paramètre de sécurité n qui génère des clés de longueur k (Si Alice et Bob doivent avorter, Alice choisit une clé aléatoire de longueur k). Soit $\rho_{\mathsf{AB}\mathcal{E}}$, l'état ccq qui contient l'information classique d'Alice dans le registre A, l'information classique de Bob dans le registre B et l'information quantique qu'Ève a accumulé lors des échanges quantiques et classiques dans le registres $\mathcal{E}$. $\Pi_{n,k}$ est* statistiquement indiscernable d'un protocole idéal *si pour n suffisamment grand, $\alpha > 0$ et pour chaque adversaire, il existe un état idéal $\rho^{+}_{\mathsf{A}\mathcal{E}} := 2^{-k}\mathbb{1}_{\mathsf{A}} \otimes \rho^{+}_{\mathcal{E}}$ satisfaisant*

$$\Delta(\rho_{\mathsf{A}\mathcal{E}}, \rho^{+}_{\mathsf{A}\mathcal{E}}) \leq 2^{-\alpha n} \ .$$

Nous pouvons obtenir $\rho_{\mathsf{A}\mathcal{E}}$ grâce à la trace partielle de $\rho_{\mathsf{AB}\mathcal{E}}$ sur le registre B.

2.2.2. Codes correcteurs

Une caractéristique importante d'une chaine de bits est son nombre de bits non nul.

Définition 2.2.2. *Le* poids de Hamming *d'une chaine $x \in \{0,1\}^n$ est*

$$w(x) = \Big|\{i \mid x_i = 1, 1 \leq i \leq n\}\Big| \ .$$

Si x' est la chaine x avec des erreurs, alors $w(x \oplus x')$ retourne le nombre d'erreurs. Les étapes préparatoires à l'obtention d'une clé sure entre Alice et Bob sont principalement de la communication qui se fait sur des canaux. Ceux-ci peuvent introduire des erreurs dans la communication dues à des facteurs extérieurs que nous appelons *bruit*. Des erreurs introduites par l'intervention d'Ève sont aussi considérées comme du bruit. Lors de l'évaluation

du taux d'erreurs maximum toléré par un protocole, nous considérons tous types de bruit. Lorsque nous demandons qu'un canal ne soit pas touché par Ève, nous dirons qu'il est authentifié, c'est-à-dire qu'un adversaire voit tout passer, mais ne peut interférer. Le canal classique du protocole BB84 qui sera présenté au prochain chapitre est authentifié. Nous l'appelerons le canal publique.

Lorsqu'il y a du bruit dans la communication entre Alice et Bob, ils doivent pouvoir corriger les erreurs pour partager la même information. Rappelons que $GF(2)$ est le corps de Galois composé des deux éléments 0 et 1 dont l'addition est le "$\oplus$" et la multiplication est le "$\wedge$" logique.

Définition 2.2.3. *Un* $[n, k, 2t+1]$-code linéaire classique $C \subset \{0,1\}^n$ *est défini comme suit*

$$C := \{c \in \{0,1\}^n \mid cH^T = 0_{n-k}\} \ ,$$

où $H \in \mathcal{L}_{n-k,n}(GF(2))$ *et tel que* $\forall c_1, c_2 \in C$ *et* $c_1 \neq c_2, w(c_1 \oplus c_2) \geq 2t+1$.

Les éléments de C sont appelés *mots de code* et sont tous à une distance minimale de $2t+1$ entre eux. La matrice H est sa *matrice de parité*. Les codes correcteurs sont utilisés pour retrouver l'état initial d'une chaine lorsque des erreurs sont introduites. Pour corriger les erreurs, Alice et Bob utilisent un *syndrome* spécifique à leur chaine définie à l'aide de la matrice de parité. La longueur du syndrome est $n-k$. La correction d'erreurs ne fonctionne que si le nombre d'erreurs est borné supérieurement par t.

Étapes:

1. Alice envoie $x \in \{0,1\}^n$ à Bob via un canal bruité.
2. Bob reçoit $x' = x \oplus e$, la chaine d'Alice avec des erreurs.
3. Alice annonce publiquement le syndrome $s = xH^T$ à Bob.
4. Bob retrouve le $e \in \{0,1\}^n$ au plus petit poids de Hamming satisfaisant $eH^T = s \oplus x'H^T$.
5. Bob corrige les erreurs $x' \oplus e = x$.

Il est garanti que e soit, en effet, la bonne chaine d'erreurs.

$$s \oplus x'H^T = xH^T \oplus x'H^T = (x' \oplus x)H^T = (x \oplus e \oplus x)H^T = eH^T \ .$$

La distance minimale de C nous assure que seulement un e satisfait cette égalité. S'il existait un second $e' \neq e$ tel que $e'H^T = eH^T$, alors $0 = e'H^T \oplus eH^T = (e' \oplus e)H^T$. Ces égalités tiennent seulement si $e' \oplus e \in C$ par définition du code linéaire. De plus, e et e' ont tous les deux un poids de Hamming borné supérieurement par t. Il en découle que $w(e' \oplus e) \leq 2t$. Puisque 0^n fait toujours partie de C, alors $w(0^n, e' \oplus e) \leq 2t$, ce qui contredit la condition que tous les mots de code de C doivent avoir une distance minimale de $2t+1$ entre eux. Ainsi, Alice et Bob corrigent parfaitement les erreurs sous condition qu'il y ait au plus t erreurs dans la chaine de Bob. La facilité à laquelle Bob retrouve e à l'étape 4 dépend des codes utilisés.

Nous utiliserons régulièrement le taux d'erreurs et non le nombre d'erreurs. La définition suivante sera donc plus appropriée.

Définition 2.2.4. *Le* poids de Hamming relatif *d'une chaine* $x \in \{0,1\}^n$ *est*

$$\omega(x) = \frac{w(x)}{n} \ .$$

2.2.3. Classe de fonctions de hachage

Les fonctions de hachage sont des fonctions dont l'ensemble image est plus petit que le domaine. Ces fonctions sont nécessairement non injectives causant ce que nous appelons des *collisions* entre les différents éléments du domaine. D'un point de vu cryptographique, il faut que deux chaines de bits différentes de longueur n aient la même image de longueur m avec très faible probabilité (où $n > m$). Soit $\{0,1\}^n \to \{0,1\}^m$, l'ensemble de toutes les fonctions qui prennent en entrée une chaine de bits de longueur n et retourne une image de longueur m. Si Alice et Bob possèdent deux chaines différentes $x_1, x_2 \in \{0,1\}^n$ et pigent aléatoirement une fonction f de $\{0,1\}^n \to \{0,1\}^m$, alors la probabilité que $f(x_1) = f(x_2)$ est 2^{-m}. Ainsi, si Ève possède une chaine contenant de l'information partielle sur x, il sera peu probable que l'image de sa chaine après l'application de f soit la même que $f(x)$. Elle aurait autant de chance de trouver l'image en pigeant une chaine aléatoire $y \in_R \{0,1\}^m$.

Par contre, il faut un nombre exponentiel de bits pour décrire une telle fonction aléatoire. Nous introduisons donc les classes de fonction universelles$_2$ [**9**].

Définition 2.2.5. *Soit* $\mathcal{G} \subset \{0,1\}^n \to \{0,1\}^m$*, un ensemble de fonctions de hachage. On dit que* $\mathcal{G}$ *est une* classe de fonctions universelles$_2$ *si* $\forall x_1 \neq x_2 \in \{0,1\}^n$*, l'inégalité suivante est satisfaite:*

$$\underset{g \in_R \mathcal{G}}{P}[g(x_1) = g(x_2)] \leq 2^{-m} \ .$$

Ainsi, pour deux chaines distinctes x_1 et x_2 de longueur n, si une fonction g est pigée aléatoirement dans une classe de fonctions universelles$_2$, la probabilité de collision de $g(x_1)$ et $g(x_2)$ est la même que si g avait été pigé dans l'ensemble de toutes les fonctions. Il est possible d'avoir une classe $\mathcal{G} \subset \{0,1\}^n \to \{0,1\}^m$ dont les fonctions peuvent être décrites avec moins de bits, mais qui a tout de même les mêmes propriétés par rapport aux collisions.

2.2.4. Entropie

L'entropie est une mesure d'incertitude d'une expérience aléatoire. Supposons que nous avons une variable aléatoire X qui suit une distribution de probabilité $\{p_i\}_{i=1}^n$. L'entropie de X doit être positive. Une entropie de 0 signifie que X a aucune surprise, c'est-à-dire nous savons à 100% quel sera le résultat de l'expérience. Si l'entropie est de $\lg(n)$, alors la distribution de X est uniforme. Il existe différentes mesures d'entropie. Dans ce mémoire, nous utiliserons l'entropie binaire, l'entropie de Hartley et la min-entropie. Nous commençons

par introduire l'entropie de Shannon qui calcule l'incertitude d'une variable aléatoire X en moyenne [**24**]. Nous dénotons le logarithme en base 2 par lg.

Définition 2.2.6. *Soit X, une variable aléatoire suivant la distribution de probabilités $\{p_i\}_{i=1}^n$. Son entropie de Shannon est*

$$H_n(X) := -\sum_{i=1}^{n} p_i \lg(p_i) \ .$$

Lorsque $n = 2$, nous l'appelons l'entropie binaire et la dénoterons h. Puisqu'il n'y a que deux résultats possibles, la probabilité p d'un évènement implique que la probabilité du second est $1-p$. Ainsi, pour une variable aléatoire X suivant la distribution $\{p, 1-p\}$, nous écrivons

$$h(p) := -(p \lg(p) + (1-p) \lg(1-p)) \tag{2.2.1}$$

pour représenter $H_2(X)$. L'entropie de Hartley de X est le nombre de bits de mémoire nécessaire pour ranger la variable sans perte.

Définition 2.2.7. *Soit X, une variable aléatoire suivant la distribution de probabilités $\{p_i\}_{i=1}^n$. Son entropie de Hartley est*

$$H_0(X) := \lg \left(\sum_{\substack{1 \leq i \leq n \\ p_i \neq 0}} 1 \right) \ .$$

La min-entropie calcule l'incertitude en pire cas. Elle tient seulement compte de la plus grande probabilité de la distribution.

Définition 2.2.8. *Soit X, une variable aléatoire suivant la distribution de probabilités $\{p_i\}_{i=1}^n$. Sa min-entropie est*

$$H_{min}(X) := -\lg \left(\max_{1 \leq i \leq n} \{p_i\} \right) \ .$$

Nous travaillerons avec des chaines de bits. Une variable aléatoire d'une chaine de longueur n a donc 2^n résultats possibles (ou impossibles si certains ont une probabilité de 0) et l'incertitude maximale est $-\lg(2^{-n}) = n$. Ainsi, du point de vue d'un adversaire, la meilleure probabilité de deviner la chaine $X = x$ est $2^{-H_{min}(X)}$. Supposons qu'il accumule de l'information et qu'elle est contenue dans une variable aléatoire Y. Si Y et X sont dépendantes, alors son incertitude va diminuer.

Définition 2.2.9. *Soit X et Y deux variables aléatoires. La min-entropie conditionnelle de X étant donné Y est*

$$H_{min}(X \mid Y) := -\lg \left(\sum_{y} P(y) \cdot \left(\max_{x} P(x \mid Y = y) \right) \right) \ .$$

Chapitre 3

L'établissement de clé BB84

3.1. Protocole

Il est impossible d'être dans le monde de la cryptographie sans avoir entendu le nom BB84 [**4**]. Le premier protocole QKD ayant vu le jour en 1984 est né de deux pères fondateurs du milieu, Charles H. Bennett et Gilles Brassard. Ce protocole utilise astucieusement les propriétés uniques à l'information quantique, dont le non-clonage, pour générer sécuritairement une clé entre deux partis. BB84 est du type *prépare et mesure*. Il existe aussi des protocoles QKD qui usent plutôt de l'intrication pour générer une clé.

Le protocole BB84 utilise deux types de canaux pour les échanges entre nos deux partis. Alice commence par envoyer de l'information à Bob via un canal quantique qui peut être attaqué par Ève. Le but du protocole est donc de détecter lorsque l'adversaire interfère et de s'entendre sur une clé sure malgré tout. Ensuite, Alice et Bob communiquent tous les deux sur des canaux classiques publics. Nous utilisons des canaux classiques certifiés authentiques. Cela signifie que toute information reçue par Bob provient bien d'Alice et vice-versa. Un adversaire peut donc voir clairement tout ce qui circule, mais par contre ne peut y toucher.

Nous allons présenter le protocole non interactif qui utilise des codes correcteurs linéaires. L'analyse de sécurité nous sera utile pour forger l'analyse de notre version modifiée au chapitre 5.

Chaque étape sera expliquée en détail après la description du protocole. Pour une chaine $x \in \{0,1\}^N$ et un ensemble de position $P = (i_j)_{1 \leq j \leq n}$ où $1 \leq i_1 < i_2 < ... < i_n \leq N$, nous définissons $x_P = x_{i_1} x_{i_2} ... x_{i_n}$. Alice et Bob doivent partager un $[n, k, 2t+1]$-code correcteur C où t est le nombre maximum d'erreurs que C peut corriger et une classe de fonctions universelle$_2$ $\mathcal{G}$. La matrice de parité de C est $H \in \mathcal{L}_{n-k,n}(GF(2))$. La longueur d'un syndrome associé à un mot de code est $\ell = n - k$.

Protocole:

1. **Préparation**: Soit $N := (4+\varepsilon)n$. Alice génère deux chaines $x, \theta \in_R \{0,1\}^N$. Elle prépare les registres $|\varphi\rangle_{\mathcal{B}} = \bigotimes_{i=1}^{N} |\varphi_{x_i\theta_i}\rangle_{\mathcal{B}_i}$ (voir fig. 3.1). Elle envoie $|\varphi\rangle_{\mathcal{B}}$ à Bob via le canal quantique.
2. **Mesure**: Bob génère $\theta' \in_R \{0,1\}^N$. Il mesure le i-ème registre de $|\varphi\rangle_{\mathcal{B}}$ avec la mesure projective $\{M_0, M_1\}$ si $\theta'_i = 0$ et avec $\{M_+, M_-\}$ si $\theta'_i = 1$ pour $1 \leq i \leq N$. Il obtient la chaine des résultats $x' \in \{0,1\}^N$. Bob envoie θ' à Alice via le canal classique public. S'il n'a rien reçu à l'étape 2 et/ou Alice ne reçoit pas de confirmation après un certain laps de temps, ils avortent. Si significativement moins que la moitié de θ' satisfait $\theta'_i = \theta_i$, ils avortent.
3. **Estimation des erreurs**: Alice choisit $L \subseteq \{i \mid 1 \leq i \leq N, \theta'_i = \theta_i\}$ tel que $|L| = 2n$ et pige $T = \{i_1, i_2, ...i_n\} \subset_R L$. Elle envoie L, T et $x_T \in \{0,1\}^n$ à Bob via le canal classique. Bob détermine $x'_T \in \{0,1\}^n$. Si $\omega(x_T \oplus x'_T) + \varepsilon > \frac{t}{n}$, ils avortent.
4. **Correction d'erreurs**: Alice envoie le syndrome $s := x_{L\setminus T} H^T \in \{0,1\}^\ell$ à Bob. Il détermine $x'_{L\setminus T} \in \{0,1\}^n$. Soit $e \in \{0,1\}^n$, les erreurs introduites dans la chaine d'Alice, c'est-à-dire $x'_{L\setminus T} = x_{L\setminus T} \oplus e$. Bob trouve le $e' \in \{0,1\}^n$ ayant le plus petit poids de Hamming satisfaisant $e'H^T = s \oplus x'_{L\setminus T} H^T$. Bob calcule $\hat{x} := x'_{L\setminus T} \oplus e'$.
5. **Amplification de secret**: Alice pige $g \in_R \mathcal{G}$ et envoie g à Bob via le canal classique. Alice et Bob établissent leur clé finale, soit $K := g(x_{L\setminus T})$ et $K' := g(\hat{x})$ respectivement.

3.1.1. Préparation

Le paramètre $\varepsilon > 0$ est l'erreur que nous tolérons dans les étapes suivantes. Nous avons posé $N = (4+\varepsilon)n$ puisque la chaine sera coupée en deux à deux reprises. Il s'agit donc de finir avec une clé brute de longueur n.

Alice utilise quatre états différents. Les deux paires d'états doivent être distinguables dans la paire, mais non-distinguables entre eux. BB84 utilise habituellement $|0\rangle$, $|1\rangle$, $|+\rangle$ et $|-\rangle$, puisque $|0\rangle$ et $|1\rangle$ sont distinguables ainsi que $|+\rangle$ et $|-\rangle$. Cependant, il ne suffit pas que $|0\rangle$ soit non-distinguable de $|+\rangle$. Il faut que la mesure de $|0\rangle$ ou de $|1\rangle$ avec la mesure projective $\{M_+, M_-\}$ donne une distribution de probabilités uniforme pour les deux résultats. De la même façon, mesurer $|+\rangle$ ou $|-\rangle$ avec la mesure projective $\{M_0, M_1\}$ doit donner une distribution uniforme. Ainsi mesurer n'importe quel des quatre qubits dans la mauvaise mesure projective fait perdre à l'observateur toute information sur l'état initial. L'étape de préparation des qubits est ce que nous appelons *préparation de qubits BB84* puisque cette idée fût introduite par ce protocole même.

$$|\varphi_{00}\rangle = |0\rangle \qquad |\varphi_{01}\rangle = \tfrac{1}{2}|0\rangle + \tfrac{1}{2}|1\rangle$$

$$|\varphi_{10}\rangle = |1\rangle \qquad |\varphi_{11}\rangle = \tfrac{1}{2}|0\rangle - \tfrac{1}{2}|1\rangle$$

Fig. 3.1. États BB84.

3.1.2. Mesure

Les quatre états BB84 et leurs relations entre eux sont à l'origine de la sécurité du protocole. Chaque registre $\mathcal{B}_i$ contient nécessairement un de ses quatre états. Bien entendu, pour retrouver les bits classiques de la chaine x, il faut mesurer chaque registre avec la mesure projective qui retournera le bon résultat x_i avec certitude. Il faut donc mesurer dans la base rectilinéaire ou diagonale dépendant de la base θ_i de chaque qubit. Mesurer dans la bonne base assure l'obtention du bon résultat x_i, tandis que mesurer dans la mauvaise base fait perdre toute information sur x_i. En d'autres mots, si un registre $\mathcal{B}_i$ est mesuré avec la mauvaise mesure projective, les résultats classiques 0 et 1 seront tous les deux équiprobables. Ce n'est donc pas plus avantageux que tout simplement deviner aléatoirement $x_i \in \{0, 1\}$. Comme θ est inconnu de Bob au moment de la communication quantique, il devine chaque base. Pour un gros n, il réussit environ la moitié du temps. En annonçant par la suite son θ' contenant toutes ses bases, Alice et lui peuvent voir les registres qui sont supposés donner les mêmes résultats. Tous les qubits qui sont mesurés par la mauvaise mesure projective deviennent inutiles pour le reste du protocole et sont donc jetés.

Il ne faut pas oublier que tout adversaire peut interférer avec la communication quantique allant d'Alice vers Bob. Ève ne peut parfaitement copier de qubit par le principe du non-clonage quantique. Des copies imparfaites introduiraient nécessairement du bruit. Par contre, elle peut mesurer, elle aussi, les différents registres envoyés par Alice avant de les transmettre à Bob. Elle ne connait pas θ au moment de la communication quantique, donc elle peut seulement deviner les bases comme Bob. Mesurer un registre $\mathcal{B}_i$ avec la bonne mesure projective θ_i lui donne l'information classique de x_i avec certitude, mais elle ne le sait pas encore. Elle envoie ensuite à Bob le qubit $|\varphi_{x_i\theta_i}\rangle$. Dans ce cas, il est impossible pour Bob de savoir si Ève a attaqué le registre. Heureusement, elle a autant de chance de deviner la mauvaise base. Dans ce cas, elle n'obtient aucune information pertinente sur x_i. Ensuite, soit elle obtient le bon résultat et envoie à Bob x_i dans la mauvaise base, soit elle obtient le mauvais résultat et envoie $\bar{x}_i$ dans la mauvaise base. Nous nous intéressons seulement au cas où Bob mesure dans la bonne base (sinon le résultat est jeté). Alors, si Bob mesure selon le bon θ_i, il obtient x_i avec probabilité 1/2. Il est donc possible d'avoir des erreurs dans la chaine classique de Bob même s'il conserve que ceux obtenus avec les bonnes mesures projectives. Si ce nombre d'erreurs est assez petit, il est possible pour Alice et Bob de corriger ces erreurs. Il est important de noter qu'Ève gagne toute l'information sur la chaine d'Alice

lorsqu'elle choisit la bonne base. Corriger les erreurs permet à nos deux partis de s'entendre sur une même clé, mais cela n'assure en rien sa sécurité.

3.1.3. Estimation des erreurs

Alice et Bob doivent savoir le nombre d'erreurs chez Bob pour savoir quel code utiliser. La façon la plus directe serait de comparer toute leur chaine classique. Par contre, le canal classique est public. De l'information en clair serait 100% compromise. Leur autre option est de sacrifier des bits de leur chaine pour une estimation du nombre d'erreurs. Puisqu'ils doivent faire une estimation, il est important qu'ils choisissent les bits de l'échantillon de sorte à resprésenter avec justesse les bits restants. Ils doivent s'entendre sur ce que nous appelons une stratégie d'échantillonnage [**23**] pour estimer avec un haut taux de précision la quantité d'erreurs dans le reste de leur chaine non dévoilée. Il est possible de prendre seulement $\sqrt{L}$ bits pour un bon échantillonnage, mais pour simplifier la notation nous allons en choisir $L/2 = n$. Ceci n'affectera pas le calcul de notre borne.

Les stratégies d'échantillonnage estiment le poids de Hamming relatif d'une chaine. Elles laissent une marge d'erreurs ε, c'est-à-dire l'estimation du poids de Hamming relatif est, au plus, le véritable poids de Hamming relatif de la chaine restante plus ε. Nous choisissons donc le ε au début du protocole en fonction de l'erreur que nous sommes prêts à accepter lors de l'étape d'échantillonnage. Lorsque nous parlons de probabilité de réussite d'une stratégie d'échantillonnage, cela veut dire la probabilité qu'elle ne dépasse pas la marge d'erreur allouée. Cette probabilité de succès est habituellement bornée inférieurement par une fonction de n et de ε.

Ce que nous cherchons à estimer est un taux d'erreurs entre Alice et Bob. Il faudrait alors échantillonner de la chaine $x \oplus x'$. Au lieu de cela, Alice et Bob vont prendre leur échantillon de x et x'. Puisque nous savons qu'une stratégie d'échantillonnage laisse une marge d'erreur de ε, nous l'ajoutons au taux d'erreurs estimé. Alice et Bob peuvent ensuite procéder à la correction d'erreurs si le taux estimé ne dépasse pas la capacité de correction possible. Ils peuvent s'entendre sur un code de correction à cette étape. L'analyse de sécurité considère le plus gros taux d'erreurs possible, cependant, en pratique, il se peut qu'il y ait moins d'erreurs. Ainsi, il peut être plus avantageux pour eux de choisir un code avec un plus gros taux de transmission pour dévoiler moins d'information via le syndrome.

3.1.4. Correction des erreurs

Cette étape permet à Alice et Bob de corriger les erreurs introduites par Ève. Nous appelons la clé résultante, la *clé brute*. Rappelons que si le nombre d'erreurs est plus petit que t, alors Bob est assuré de trouver le bon e' pour corriger sa chaine. En d'autres mots, e'

sera égal à e.

$$e'H^T = s \oplus x'_{L\setminus T}H^T = (x_{L\setminus T} \oplus x'_{L\setminus T})H^T = eH^T \ .$$

Par contre, nous ne savons pas le nombre exact d'erreurs dans la chaine de Bob. Nos deux partis n'ont qu'une estimation, voulant dire qu'il se peut qu'ils continuent le protocole même s'il y a un nombre d'erreurs plus grand que t. Dans ce cas, il n'est pas garanti que $e' = e$. La borne sur le succès de la stratégie d'échantillonnage est donc très importante.

De plus, l'annonce du syndrome doit se faire lors de la communication classique, signifiant qu'Ève gagne un peu plus d'information sur la chaine d'Alice. Nous baisserons donc son incertitude en fonction du syndrome. La longueur du syndrome sera primordiale pour évaluer l'avantage de l'adversaire à cette étape. Les codes correcteurs ont tous leurs avantages et désavantages. Il faut tenir compte de la facilité d'implémentation en pratique, mais aussi de leur taux de transmission k/n. La longueur du syndrome ℓ est donnée par la différence entre n et k. Il en découle que pour un n fixe, plus le taux de transmission est grand, plus la longueur du syndrome est petite. En l'occurrence, cela limite la quantité d'information dévoilée à Ève.

Le protocole BB84 utilise les codes linéaires pour la correction d'erreurs, mais il existe d'autres types. Puisque les codes spécifiques sont rarement précisés dans les protocoles, nous basons les résultats sur des bornes sur les taux de transmission qui promettent l'existence de codes correcteurs satisfaisant différentes conditions. Il suffit de prendre une borne pertinente pour un protocole spécifique. La capacité du canal nous donne une borne supérieure pour tout taux de transmission sans toutefois garantir l'existence de codes qui atteignent l'égalité.

$$\frac{k}{n} \leq C \ ,$$

où C est la capacité du canal. Pour un canal symétrique binaire, $C_{bin} = 1 - h(t/n)$. La fonction h est l'entropie binaire pour le taux d'erreurs t/n (voir 2.2.1).

Shannon a montré l'existence de codes aléatoires (mais non linéaires) se rapprochant de la borne, mais qui ne corrigent pas parfaitement toutes les erreurs [**24**]. La borne de Gilbert-Varshamov semble à priori beaucoup plus pratique puisqu'elle garantit l'existence de codes linéaires pouvant parfaitement tout corriger si l'erreur est moins que 1/4 [**12, 26**]. Par contre, elle offre une égalité pour un taux de transmission $1 - h(2t/n)$, ce qui est plus petit que la capacité du canal. Encore plus intéressants sont les codes polaires. En 2008, Arikan [**1**] a démontré l'existence de familles de codes polaires à efficacité pratique pouvant atteindre la capacité du canal. Ce travail est basé sur l'existence de ces codes. Nous pouvons alors espérer qu'un taux de transmission puisse être aussi près que voulue de $1 - h(t/n)$. Il nous sera possible de déterminer la longueur du syndrome minimale nécessaire lorsque nous connaitrons le taux d'erreurs.

3.1.5. Amplification de secret

Suite à la correction d'erreurs entre Alice et Bob, la chaine obtenue est très faible en termes de sécurité. La réconciliation n'augmente pas l'incertitude chez Ève, mais plutôt, la diminue. Alice et Bob doivent *amplifier* cette incertitude pour générer une clé statistiquement sure. Nous appelons cette clé, la *clé finale* et nous appelons cette étape, l'*amplification de secret*. Nous utilisons des fonctions de hachage pour obtenir une chaine plus courte, mais sure. Si Ève connaissait la fonction à l'avance, elle pourrait planifier ses attaques astucieusement afin de gagner plus d'information. Nous devons donc avoir une classe de fonctions universelles$_2$ dans laquelle Alice et Bob pourront facilement piocher une fonction g rendue à l'étape d'amplification de secret. Rappelons que cette classe de fonction agit comme l'ensemble de toutes les fonctions $\{0,1\}^n \to \{0,1\}^m$ pour les collisions, sans toutefois avoir des fonctions qui prennent un nombre exponentiel de bits pour être décrites.

Nous voulons que g, la fonction pigée aléatoirement, puisse retourner une chaine qui semble aléatoire du point de vue d'Ève. Nous présentons le théorème d'amplification de secret de Renner contre les adversaires quantiques [**21, 19**].

Théorème 3.1.1 (Amplification de secret contre les adversaires quantiques)**.** *Soit $\mathscr{G} \subset \{0,1\}^n \to \{0,1\}^m$ une classe de fonctions universelles$_2$. Soit $\rho_{\mathsf{X}\mathcal{E}} = \sum_x P_X(x) \left|x\right\rangle\!\left\langle x\right|_{\mathsf{X}} \otimes \rho^x_{\mathcal{E}}$ un état cq avec un registre classique X et un registre quantique $\mathcal{E}$. Alors,*

$$\mathop{\mathbb{E}}_{g\in\mathscr{G}} \left[\left\| \omega^g_{\mathsf{Y}\mathcal{E}} - \frac{\mathbb{1}_{\mathsf{Y}}}{2^m} \otimes \rho_{\mathcal{E}} \right\|_1 \right] \leq \sqrt{2^{m - H_{min}(\mathsf{X}|\mathcal{E})}} \ ,$$

où $\omega^g_{\mathsf{Y}\mathcal{E}} := \sum_x P_X \left|g(x)\right\rangle\!\left\langle g(x)\right|_{\mathsf{Y}} \otimes \rho^x_{\mathcal{E}}$ est l'état cq avec registre classique Y.

L'état $\omega^g_{\mathsf{Y}\mathcal{E}}$ représente l'état résultant de l'étape d'amplification de secret et $H_{min}(. \mid .)$ est la min-entropie conditionnelle (2.2.9). Le théorème de Renner utilise l'entropie de collision. La version présentée plus haut est comme celle présentée dans [**23**]. Par la définition de la distance de trace (2.1.5), nous avons

$$\mathop{\mathbb{E}}_{g\in\mathscr{G}} \left[2 \cdot \Delta \left(\omega^g_{\mathsf{Y}\mathcal{E}}, \frac{\mathbb{1}_{\mathsf{Y}}}{2^m} \otimes \rho_{\mathcal{E}} \right) \right] \leq \sqrt{2^{m - H_{min}(\mathsf{X}|\mathcal{E})}} \ , \tag{3.1.1}$$

$$\mathop{\mathbb{E}}_{g\in\mathscr{G}} \left[\Delta \left(\omega^g_{\mathsf{Y}\mathcal{E}}, \frac{\mathbb{1}_{\mathsf{Y}}}{2^m} \otimes \rho_{\mathcal{E}} \right) \right] \leq \sqrt{2^{m - H_{min}(\mathsf{X}|\mathcal{E}) - 2}} \ . \tag{3.1.2}$$

3.2. Sécurité

3.2.1. Conditions de sécurité

Pendant l'estimation d'erreurs, il se peut que le taux d'erreurs réel puisse être plus important que celui estimé (considérant aussi la marge d'erreur allouée ε). Cela impliquerait que la chaine corrigée d'Alice ne serait pas nécessairement la même que celle de Bob rendant l'amplification de secret inutile. Cette probabilité d'erreurs peut être bornée supérieurement. Plus précisément, la réussite de l'étape d'échantillonnage peut être bornée inférieurement par $6e^{-\varepsilon^2 n/24}$ avec une tolérance d'erreurs de ε [**23**]. Il en découle que la réussite de l'échantillonnage repose sur le choix de n et ε. Alice et Bob peuvent choisir astucieusement ces deux paramètres de sorte à rendre la probabilité d'échec négligeable.

Selon la définition de sécurité 2.2.1, une clé finale établie entre Alice et Bob est sure si pour n suffisamment grand et contre n'importe quel adversaire, l'état cq contenant l'information classique de la chaine d'Alice et l'information quantique d'Ève est très proche d'un état idéal, c'est-à-dire $\Delta(\rho_{\mathsf{A}\mathcal{E}}, \rho^{+}_{\mathsf{A}\mathcal{E}}) \leq 2^{-\alpha n}$ où $\alpha > 0$. Pour suivre la notation des définitions que nous avons vues plus haut, nous posons X comme étant le registre contenant la clé brute d'Alice et Y comme étant le registre contenant sa clé finale. Nous précisons aussi que le registre $\mathcal{E}$ contient l'information qu'Ève a accumulé pendant la communication quantique ainsi que l'information provenant du dévoilement des bases à la communication classique. Ainsi, nous devons avoir

$$\Delta(\rho_{\mathsf{Y}\mathcal{E}}, \rho^{+}_{\mathsf{Y}\mathcal{E}}) \leq 2^{-\alpha n} \ ,$$

où $\alpha > 0$. Nous pouvons borner supérieurement cette distance de trace à l'aide du théorème d'amplification de secret contre des adversaires quantiques (3.1.1) comme suit:

$$\Delta(\rho_{\mathsf{Y}\mathcal{E}}, \rho^{+}_{\mathsf{Y}\mathcal{E}}) \leq \sqrt{2^{m - H_{min}(\mathsf{X}|\mathcal{E}) - 2}} \ .$$

Puisque nous voulons que la clé finale contienne au moins 1 bit, nous posons $m = 1$. La condition de sécurité devient alors,

$$\begin{aligned}
\sqrt{2^{1 - H_{min}(\mathsf{X}|\mathcal{E}) - 2}} &\leq 2^{-\alpha n} \\
2^{-H_{min}(\mathsf{X}|\mathcal{E}) - 1} &\leq 2^{-2\alpha n} \\
2^{-H_{min}(\mathsf{X}|\mathcal{E})} &\leq 2^{-2\alpha n + 1} \\
H_{min}(\mathsf{X} \mid \mathcal{E}) &\geq 2\alpha n - 1 \ ,
\end{aligned}$$

où $\alpha > 0$. Comme α peut être choisi aussi petit que nécessaire, il suffit que

$$H_{min}(\mathsf{X} \mid \mathcal{E}) > 0 \ , \tag{3.2.1}$$

où $\alpha > 0$. La min-entropie $H_{min}(\mathsf{X} \mid \mathcal{E})$ est calculée après la correction d'erreurs et avant l'amplification de secret. Comme discuté à la section 3.1.4, pour calculer la min-entropie

d'Ève, nous devons tenir compte de l'information divulguée par l'annonce publique du syndrome lors de l'étape de réconciliation. Au lieu de supposer que l'information rangée dans le registre $\mathcal{E}$ contient le syndrome, nous allons ajouter un autre registre. Nous réécrivons la min-entropie d'Ève comme suit:

$$H_{min}(\mathsf{X} \mid \mathcal{E}, \mathsf{Z}) \ ,$$

où $\mathcal{E}$ contient l'information quantique d'Ève et Z contient l'information classique du syndrome.

Théorème 3.2.1. *Soit $\rho_{\mathsf{X}\mathcal{E}\mathsf{Z}}$ un état ccq où X et Z sont les registres classiques et $\mathcal{E}$ est le registre quantique. Si Z est la variable aléatoire du contenu de Z, alors*

$$H_{min}(\mathsf{X} \mid \mathcal{E}, \mathsf{Z}) \geq H_{min}(\mathsf{X} \mid \mathcal{E}) - H_0(Z) \ ,$$

où $H_0(Z)$ est l'entropie de Hartley (2.2.7) de la variable aléatoire Z.

Soit ℓ, la longueur du syndrome. Alors, $Z \in \{0,1\}^\ell$. Il en découle que la min-entropie d'Ève sur la chaine d'Alice ne peut pas diminuer de plus que $H_0(Z) = \ell$. Nous pouvons donc réécrire la condition (3.2.1) comme suit:

$$H_{min}(\mathsf{X} \mid \mathcal{E}) - \ell > 0 \ . \tag{3.2.2}$$

On peut conclure que la min-entropie doit être plus grande que la longueur du syndrome pour qu'Alice et Bob finissent avec une clé finale sure d'au moins 1 bit. La min-entropie d'Ève, $H_{min}(\mathsf{X} \mid \mathcal{E})$, dépend de l'information qu'elle accumule lors de la communication quantique et la longueur du syndrome dépend du taux d'erreurs à corriger.

3.2.2. Attaques

Afin d'évaluer la min-entropie d'Ève, il est important de savoir comment elle obtient de l'information et comment elle optimise cette information recueillie. Il existe trois types d'attaques, soit l'attaque incohérente, l'attaque collective et l'attaque cohérente.

- L'attaque incohérente s'illustre dans la situation où Ève manipule les qubits individuellement et indépendamment pendant la communication quantique.

- L'attaque collective est sensiblement similaire à l'attaque incohérente, excepté le fait qu'Ève peut attendre à la fin de la communication classique pour choisir comment elle manipule les qubits qu'elle a interceptés. Naturellement, à cause du théorème de non-clonage, Ève ne peut copier les qubits. Si elle décide de les intercepter et de faire une mesure, il n'est pas garanti que ces qubits seront transmis

sans erreur à Bob. Elle peut cependant lier chacun des qubits interceptés à un système extérieur qu'elle conserve et qu'elle peut mesurer plus tard. Elle peut par ailleurs utiliser une machine de clonage quantique pour obtenir des copies imparfaites des qubits. Dans BB84, elle peut conserver les copies imparfaites jusqu'à ce que Bob énonce les bases à la communication classique. Ève pourra à ce moment mesurer toutes les imitations dans les bonnes bases. Ce genre d'attaque n'est pas tout puissant puisque la machine copie avec un certain taux d'erreurs.

- L'attaque cohérente est la plus générale. Elle permet à Ève de manipuler le système complet. Elle peut donc lier collectivement tous les états envoyés par Alice à un système qu'elle mesure après avoir espionné la communication publique. L'attaque cohérente est la plus complexe puisqu'elle englobe toutes les attaques possibles. Néanmoins, il est possible de prouver la sécurité d'un protocole QKD seulement en assurant sa sécurité face aux attaques collectives [**20**]. De plus, l'attaque de "clonage" est considérée comme étant la meilleure attaque possible pour qu'Ève obtienne un maximum d'informations sur la chaine d'Alice.

Malgré tout, pour simplifier les analyses, nous nous intéresserons seulement aux attaques incohérentes dans le cadre de ce travail. Un exemple de ce type d'attaque est l'*intercept-resend*, ou *intercepte et renvoie* que nous appelerons *i/r* pour abréger. Lorsqu'elle attaque un qubit, elle l'intercepte pour le mesurer et renvoie par la suite un qubit à Bob selon le résultat de sa mesure. Par exemple, pour attaquer un qubit d'un protocole BB84, elle peut choisir au hasard entre la base rectilinéaire et la base diagonale pour le mesurer. Par la suite, elle peut préparer un état BB84 selon son résultat et sa base choisie (figure 3.1) et l'envoyer à Bob.

Rappelons qu'Alice choisit aléatoirement ses bits ainsi que les bases dans lesquelles elle les prépare. Nous savons que mesurer $|0\rangle$ et $|1\rangle$ dans la base diagonale donnera les bons résultats, c'est-à-dire $|+\rangle$ et $|-\rangle$ respectivement avec probabilité 1/2 chaque. Il en va de même en mesurant $|+\rangle$ et $|-\rangle$ dans la base rectilinéaire. Bien entendu, lorsqu'elle mesure dans la bonne base, elle obtient le bon résultat à tous les coups. Soit p_{succE} ="Ève obtient le bon résultat", alors

$$\begin{aligned} p_{succE} &= P[\text{bonne base}] + P[\text{mauvaise base}] \cdot P[\text{bon résultat} \mid \text{mauvaise base}] \\ &= \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \\ &= 0.75 \ . \end{aligned}$$

Ève réussit donc à obtenir le même bit qu'Alice 75% du temps. De plus, puisque les bases seront annoncées plus tard, Ève saura à ce moment exact quels bits ont été mesurés dans la

bonne base. Elle aura donc environ la moitié qu'elle connaitra parfaitement et l'autre moitié qu'elle ignorât complètent.

3.2.3. Taux d'erreurs maximal

Nous pouvons borner inférieurement la min-entropie d'Ève [**23**]. Soit n, la longueur de la clé avant la correction d'erreurs (qui est la même longueur que sa clé brute), et p, le taux d'erreurs chez Bob après la communication quantique. Alors,

$$H_{min}(\mathsf{X} \mid \mathcal{E}) \geq n\,(1 - h(p))\ .$$

Nous avons conclu qu'Alice et Bob peuvent s'entendre sur une clé finale sure sous la condition que la min-entropie d'Ève soit plus grande que la longueur du syndrome (3.2.2). De ce fait, si $n(1 - h(p)) \geq \ell$, nous garantissons la sécurité du protocole selon la définition 2.2.1.

Soit k, la dimension du code linéaire utilisé pour la correction d'erreurs. Pour un n fixe, plus le taux de transmission k/n du code est grand, plus la longueur du syndrome, $\ell = n - k$, est petite. Comme nous avons vu à la section 3.1.4, k/n peut être, au plus, égal à la capacité du canal grâce aux codes polaires. Ainsi,

$$\begin{aligned} & \frac{k}{n} = 1 - h(p) \\ \Leftrightarrow \quad & k = n - n \cdot h(p) \\ \Leftrightarrow \quad & n - k = n - n + n \cdot h(p) \\ \Leftrightarrow \quad & \ell = n \cdot h(p)\ . \end{aligned}$$

Il en découle que

$$\begin{aligned} n(1 - h(p)) &\geq nh(p) \\ n &\geq 2nh(p) \\ \frac{1}{2} &\geq h(p)\ . \end{aligned}$$

Nous pouvons déterminer le taux d'erreurs maximal toléré par le protocole en trouvant le plus gros p satisfaisant l'égalité. Puisque $h(0.11) \approx 1/2$, nous concluons que le protocole BB84 non interactif peut tolérer un taux d'erreurs allant jusqu'à 11%. Le but des prochains chapitres est de présenter une variante du protocole pour augmenter ce taux d'erreurs et celui de Gottesman et Lo.

Chapitre 4

BB84 modifié

4.1. Protocole

Nous présentons maintenant une version modifiée de BB84. Nous ajoutons l'hypothèse qu'Alice et Bob partagent initialement les bases de mesures. Puisque l'étape du dévoilement des bases est l'étape où Ève peut gagner le plus d'information sur la chaine d'Alice, il est évident que cette modification apporte un avantage à nos deux partis. Nous ajoutons un troisième parti pour pouvoir implémenter cette condition d'une façon réaliste et pratique. Supposons que nous avons accès au satellite canadien QEYSSat 1.0 et qu'il interagit avec Alice et Bob.

Dans ce mémoire nous tentons de pousser la limite du taux d'erreurs toléré par un protocole QKD pour augmenter la distance maximale entre Alice et Bob. Nous voulons donc dépasser le 11%. Nous ajoutons aussi de la correction d'erreurs interactive pour dépasser le 18.9% de Gottesman et Lo [**13**].

Alice et Bob partagent préalablement une classe de fonctions universelle$_2$ $\mathcal{G}$. Rappelons que t est le nombre maximum d'erreurs qu'ils puissent corriger à l'aide d'un code correcteur dont H est la matrice de parité.

Protocole:

1. **Établissement des bases**: Le satellite génère une clé $\theta \in \{0,1\}^N$ avec Alice à l'aide d'un protocole QKD. Il génère une clé $k \in \{0,1\}^N$ avec Bob de la même façon. Il envoie $\theta \oplus k$ à Bob. Bob obtient $\theta = (\theta \oplus k) \oplus k$.
2. **Préparation**: Alice génère $x \in_R \{0,1\}^N$. Elle prépare les registres $|\varphi\rangle_{\mathcal{B}} = \bigotimes_{i=1}^{N} |\varphi_{x_i\theta_i}\rangle_{\mathcal{B}_i}$ (voir fig. 3.1). Elle envoie $|\varphi\rangle_{\mathcal{B}}$ à Bob via le canal quantique.
3. **Mesure**: Bob mesure le i-ème registre de $|\varphi\rangle_{\mathcal{B}}$ avec la mesure projective $\{M_0, M_1\}$ si $\theta_i = 0$ et avec $\{M_+, M_-\}$ si $\theta_i = 1$ pour $1 \leq i \leq N$. Il obtient la chaine des résultats $x' \in \{0,1\}^N$. Bob confirme publiquement la réception des qubits. Si Alice ne reçoit pas de confirmation de Bob après un certain laps de temps, ils avortent.

4. **Filtrage**: Bob permute sa chaine de résultats de bits et envoie la permutation à Alice. Ils apparient chacun leur chaine et annoncent la parité de chaque paire de bits. Ils ne conservent que le 2ème bit de chacune des paires ayant la même parité. Soit $x_f \in \{0,1\}^n$, la chaine résultante chez Alice et $x'_f \in \{0,1\}^n$, celle de Bob.
5. **Concentration**: Bob permute sa chaine de résultats de bits et envoie la permutation à Alice. Alice et Bob apparient chacun leur chaine et ne conservent que le XOR de chaque paire. Soit $x_c \in \{0,1\}^{n/2}$, la chaine résultante chez Alice et $x'_c \in \{0,1\}^{n/2}$, celle de Bob.
6. **Estimation des erreurs**: Alice pige $T = \{i_1, i_2, ...i_{n/4}\} \subset_R \{1, 2, ..., \frac{n}{2}\}$. Elle envoie T et $x_{c,T} \in \{0,1\}^{n/4}$ à Bob via le canal classique. Bob détermine $x'_{c,T} \in \{0,1\}^{n/4}$. Si $\omega(x_{c,T} \oplus x'_{c,T}) + \varepsilon > \frac{t}{n/4}$, ils avortent.
7. **Correction d'erreurs**: Alice envoie et $s := x_{c,\overline{T}} H^T \in \{0,1\}^\ell$ à Bob. Il détermine $x'_{c,\overline{T}} \in \{0,1\}^{n/4}$. Soit $e \in \{0,1\}^{n/4}$, les erreurs introduites dans la chaine d'Alice, c'est-à-dire $x'_{c,\overline{T}} = x_{c,\overline{T}} \oplus e$. Bob trouve le $e' \in \{0,1\}^{n/4}$ ayant le plus petit poids de Hamming satisfaisant $e'H^T = s \oplus x'_{c,\overline{T}} H^T$. Bob calcule $\hat{x} := x'_{c,\overline{T}} \oplus e'$.
8. **Amplification de secret**: Alice pige $g \in_R \mathcal{G}$ et envoie g à Bob via le canal classique. Alice et Bob établissent leur clé finale, soit $K := g(x_{c,\overline{T}})$ et $K' := g(\hat{x})$ respectivement.

4.1.1. Établissement des bases

Le satellite est un troisième parti passif après avoir complété son rôle. Comme c'est un outil gouvernemental, nous supposons qu'il n'intervient pas dans les échanges entre Alice et Bob et ne divulgue pas d'information aux adversaires. Bien entendu, la communication entre le satellite et nos deux partis n'est pas protégée. C'est pourquoi nous utilisons un protocole QKD. La clé θ d'Alice est nécessairement sure contre tout adversaire ainsi que la clé k de Bob. De plus, la chaine $\theta \oplus k$ que le satellite envoie à Bob est un masque jetable qui est une technique de chiffrement sure. Nous avons donc une façon réaliste et pratique d'échanger une chaine de bits entre Alice et Bob qui servira de séquence de bases.

Pourquoi utilisons-nous cette étape pour définir θ et non la clé finale? Cette étape génère une clé sure, mais nous voulons éviter de faire appel au satellite à chaque roulement du protocole. Les étapes suivantes nous permettent de réutiliser θ de sorte à ne plus avoir à utiliser le satellite pour établir des clés futures.

4.1.2. Préparation

La préparation des qubits BB84 reste inchangée. La seule différence est qu'Alice ne doit pas générer une séquence de bases aléatoire.

4.1.3. Mesure

Si Ève fait des attaques i/r dans les bases rectilinéaire et diagonale, alors elle n'aura aucun moyen de confirmer que ces résultats sont certains ou complètement aléatoires, puisque les bases ne sont jamais dévoilées publiquement. Elle ne saura jamais si ses mesures projectives choisies sont les bonnes, alors elle ne possèdera que de l'information probabiliste et non déterministe. Nous verrons plus tard comment elle peut choisir une autre base de mesure pour optimiser la quantité d'information probabiliste qu'elle accumule lors de ses attaques.

Alice et Bob partagent déjà θ. Il en résulte qu'ils peuvent conserver l'entièreté de leur bits pour les étapes suivantes. Dans BB84, ils perdaient la moitié de leurs chaines à cette étape (les bits où Bob avait choisi la mauvaise base de mesure).

Même si Bob est assuré d'avoir mesuré chaque registre dans la bonne base, il est encore possible qu'Ève soit intervenue dans la communication. Bob peut donc avoir des erreurs dans sa chaine.

4.1.4. Filtrage

Cette étape a pour but de filtrer majoritairement des bits erronés. Alice et Bob permutent leur chaine pour empêcher Ève de planifier ses attaques en fonction des positions. Ensuite, ils s'échangent la parité de chaque paire de bits. Ils en déduisent que les paires dont la parité ne concorde pas ont nécessairement une erreur du côté de Bob. Ils les jettent automatiquement. Ainsi, les paires restantes ont soit aucune erreur, soit deux erreurs. La dernière est beaucoup moins probable que la première. Puisque les parités sont annoncées sur le canal classique (et ainsi, annoncées à Ève) Alice et Bob perdent un bit de secret par paire. C'est pour cette raison qu'ils ne conservent qu'un seul bit parmi chaque paire. De plus, la position des bits conservés est connue publiquement. Ceci ne donne aucun avantage à Ève puisque la chaine est permutée avant l'annonce des paires. Si ce n'était pas le cas, pendant l'échange quantique, Ève pourrait cibler les bits qu'elle sait qui seront gardés dans les paires. Alice et Bob perdront une partie de leur chaine lorsqu'ils vont comparer les parités et ils jetteront un bit par paires conservées. La chaine sera coupée d'au moins la moitié, mais cela diminue grandement le nombre d'erreurs chez Bob.

Si la chaine initiale est assez longue, il est possible de réappliquer la technique de filtrage par la suite. Pourvu qu'il reste au moins 2 bits, Alice et Bob peuvent comparer la parité. Puisque ce processus diminue le taux d'erreurs entre Alice et Bob, il semblerait logique de simplement refaire cette méthode jusqu'à ce que le taux d'erreurs ait suffisamment baissé pour pouvoir utiliser un code correcteur (ou jusqu'à ce qu'il ne reste qu'un seul bit). Nous pourrions donc tolérer un taux d'erreurs arbitraire. Cependant, ceci serait difficilement réalisable en pratique. Nous verrons au prochain chapitre (5) que le filtrage diminue non seulement le taux d'erreurs de nos participants, mais de notre adversaire aussi.

4.1.5. Concentration

Le filtrage diminue le taux d'erreurs chez Bob tout en diminuant graduellement celui chez Ève. Pour user des bénéfices de multiples filtrages, on ajoute entre chaque, une étape de concentration. Cette technique a pour but de pousser davantage le taux d'erreurs toléré, mais surtout d'augmenter la différence entre le taux d'erreurs de Bob et celui d'Ève.

Alice et Bob permutent encore leur chaine pour empêcher les attaques ciblées d'Ève. Contrairement au filtrage, Alice et Bob n'annoncent pas les parités des paires, ce qui leur laisse deux bits d'information par paire. Ils peuvent donc garder le résultat du XOR pour augmenter l'incertitude chez Ève. Aussitôt qu'une paire contient un bit inconnu de l'adversaire, la parité lui est aussi inconnue. Cela augmente significativement le taux d'erreurs chez Ève. Nous controns alors les effets secondaires néfastes du filtrage.

Alice et Bob peuvent répéter l'étape de filtrage et de concentration pour continuer à diminuer leur taux d'erreurs et augmenter la min-entropie de la chaine chez Ève. Bien entendu, s'ils répètent ces deux étapes, les longueurs des chaines aux étapes 4, 5 et 6 ne sont plus les mêmes. Les chaines sont coupées en deux après la concentration ainsi qu'à l'échantillonnage. D'une façon plus générale, si les chaines sont de longueur c après l'étape de filtrage, alors il reste $c/4$ des bits à la correction d'erreurs.

4.1.6. Estimation des erreurs

Supposons qu'Alice et Bob ont chacun une chaine de longueur $n/2$. Comme dans BB84, ils pigent un échantillon de chacune des deux chaines selon une stratégie d'échantillonnage avec une marge d'erreurs ε (voir section 3.1.3). Ils prennent la moitié de chaque chaine comme échantillons ($\sqrt{n/2}$ serait suffisant, mais nous voulons simplifier les calculs suivants). Ils prennent le XOR des deux pour estimer leur taux d'erreurs. Il reste $n/4$ bits dans chaque chaine.

Posons $m = n/4$ pour alléger la notation. Il serait difficile de choisir un code correcteur avant le roulement du protocole, puisqu'Alice et Bob ne connaissent ni la longueur résultante de leur chaine après le filtrage et la concentration, ni le taux d'erreurs. Ils peuvent donc choisir un code C à cette étape-ci. Supposons que C a un taux de transmission k/m, une matrice de parité $H \in \mathcal{L}_{m-k,m}(GF(2))$ et une distance minimale de $2t + 1$ (c'est-à-dire C peut corriger au plus t erreurs). Alice et Bob s'assurent que le taux d'erreurs estimé ne dépasse pas ce que C peut corriger. N'oublions pas qu'une stratégie d'échantillonnage laisse une marge d'erreur ε. Il faut alors l'ajouter au taux d'erreurs estimé.

4.1.7. Correction des erreurs et amplification de secret

La correction d'erreurs et l'amplification de secret restent inchangées. Nous arrivons donc à la même conclusion que le protocole standard. Pour que la clé finale soit sure, la min-entropie d'Ève sur la chaine d'Alice avant la correction d'erreurs doit être plus grande que la longueur du syndrome. Lors de l'analyse de notre version modifiée, nous nous concentrerons alors à calculer l'incertitude chez Ève après le filtrage et la concentration ainsi que le taux d'erreurs chez Bob.

Chapitre 5

Analyse de BB84 modifié

Nous cherchons à maximiser le taux d'erreurs qu'Alice et Bob peuvent tolérer dans un roulement de notre version du protocole. Nous notons ce taux p. Le taux d'erreurs p et la min-entropie d'Ève sur la chaine d'Alice dépendent des attaques d'Ève lors de la communication quantique. Puisque la correction d'erreurs non interactive (par code correcteurs linéaires) ne peut corriger plus qu'un certain taux d'erreurs, nous ajoutons le filtrage et la concentration pour former une étape de correction d'erreurs interactive. Le taux résultant de cette communication classique interactive est e. Comme Maurer le proposait, la communication classique interactive entre Alice et Bob après le transfert des qubits est un outil puissant qui nous permet d'établir une clé sure même si l'adversaire a un taux d'erreurs initialement plus petit que celui de Bob. Rappelons que la clé finale est sure selon notre définition (2.2.1) si la condition (3.2.2) est satisfaite avant la correction d'erreurs. Soit

$$H_{min}(\mathsf{X} \mid \mathcal{E}) > \ell \ ,$$

où X est le registre classique contenant la clé brute d'Alice, $\mathcal{E}$ est le registre quantique contenant l'information qu'Ève a accumulée et ℓ est la longueur du syndrome. Plus le taux d'erreurs e à corriger par codes correcteurs est grand, plus le syndrome est long. Pour un haut taux p, nous voulons que le filtrage et la concentration maximisent $H_{min}(\mathsf{X} \mid \mathcal{E})$ tout en minimisant e. Nous verrons comment chacune de ces étapes affectent l'incertitude chez Ève et la longueur du syndrome pour, ainsi, maximiser p.

5.1. Attaques

Dans le cadre de ce travail, nous n'analyserons la sécurité que pour des attaques incohérentes i/r. Il n'y aura pas de preuve assurant que l'analyse de ces attaques spécifiques est suffisante. Nous limitons notre analyse pour le moment et nous laissons la question ouverte pour de futurs travaux. Nous expliquerons pourquoi cette attaque est tout de même pertinente dans notre contexte. Pour le protocole BB84, nous avons présenté les attaques i/r où

Ève mesure soit dans la base rectilinéaire, soit dans la base diagonale. C'est extrêmement avantageux dans ce contexte, mais pas nécessairement dans notre version modifiée où les bases ne sont jamais révélées.

5.1.1. Base de Breidbart

Lorsque nous tournons la base rectilinéaire de $\pi/8$, nous obtenons la base de Breidbart. Nous pouvons aussi le voir comme le juste milieu entre la base rectilinéaire et la base diagonale. Soit p_{ab}, la probabilité que l'état préparé par Alice soit $|\varphi_{ab}\rangle$ (voir fig. 3.1). À la sous-section 2.1.3, nous avons vu que pour un θ arbitraire (l'angle d'un état $|\psi\rangle$), la probabilité qu'une mesure projective $\{M_i, M_j\}$ retourne l'état associé à M_i est $\cos^2(\theta - \theta_i)$, où θ_i est l'angle de l'état i. Similairement, la probabilité qu'il retourne l'état associé à M_j est $\sin^2(\theta - \theta_i)$. Ainsi, la probabilité qu'Ève obtienne le même résultat qu'Alice en mesurant dans la base de Breidbart est

$$\begin{aligned}
p_{succE} &= P[\text{bit } 0 \wedge \text{base } 0] \cdot P[\text{résultat } 0 \mid \text{bit } 0 \wedge \text{base } 0] \\
&\quad + P[\text{bit } 1 \wedge \text{base } 0] \cdot P[\text{résultat } 1 \mid \text{bit } 1 \wedge \text{base } 0] \\
&\quad + P[\text{bit } 0 \wedge \text{base } 1] \cdot P[\text{résultat } 0 \mid \text{bit } 0 \wedge \text{base } 1] \\
&\quad + P[\text{bit } 1 \wedge \text{base } 1] \cdot P[\text{résultat } 1 \mid \text{bit } 1 \wedge \text{base } 1] \\
&= p_{00} \cdot \cos^2(0 - \tfrac{\pi}{8}) + p_{10} \cdot \sin^2(\tfrac{\pi}{2} - \tfrac{\pi}{8}) + p_{01} \cdot \cos^2(\tfrac{\pi}{4} - \tfrac{\pi}{8}) + p_{11} \cdot \sin^2(\tfrac{3\pi}{4} - \tfrac{\pi}{8}) \\
&= \frac{1}{4} \cdot 4\cos^2(\tfrac{\pi}{8}) \\
&= \cos^2(\tfrac{\pi}{8}) \\
&\approx 0.85 \ .
\end{aligned}$$

Ève a donc un meilleur taux de succès par qubit en utilisant la base de Breidbart au lieu de la base $\{M_0, M_1\}$ ou $\{M_+, M_-\}$. Par contre, si elle emploie cette attaque contre un protocole BB84 standard, elle ne pourra jamais savoir lesquels de ses bits sont les mêmes que ceux d'Alice. Sa connaissance serait donc probabiliste et non déterministe. De ce fait, apprendre les bases à la communication classique ne lui offre pas d'avantage si elle a tout mesuré dans la base de Breidbart. Lors de l'analyse de BB84, nous supposons qu'Ève attaque en mesurant dans les bases rectilinéaires et diagonales comme Bob pour maximiser l'information sur la chaine d'Alice lors de l'annonce des bases. Cependant, si les bases ne sont pas annoncées, il est impossible pour un adversaire d'obtenir de l'information déterministe. En comparant purement sa probabilité de succès par bit d'obtenir le même résultat qu'Alice, la base de Breidbart est plus avantageuse. Encore mieux, nous savons que la base de Breidbart est la mesure ayant la meilleure probabilité de succès pour un qubit BB84 (voir fig. 5.1). Si Ève

cherche a maximiser l'information sur la chaine d'Alice sachant qu'elle ne saura jamais les base d'Alice et Bob, alors elle doit mesurer dans la base de Breidbart.

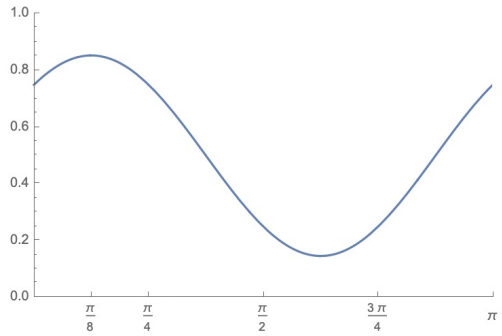


Fig. 5.1. Probabilité de succès d'une mesure selon l'angle de la base.

5.1.2. Erreur chez Bob

Il est aussi important de voir les répercussions d'attaques i/r chez Bob. Le but d'Ève est non seulement de maximiser l'information qu'elle a sur la chaine d'Alice, mais aussi de ne pas se faire détecter trop facilement. Dans un tel cas, Alice et Bob avortent le protocole. Nous présentons un théorème qui décrit le taux d'erreurs chez Bob selon l'attaque d'Ève.

Théorème 5.1.1. *Si Alice envoie un qubit BB84 à Bob et Ève effectue un i/r dans une base arbitraire réelle* $\{M_i, M_j\}$ *où* $M_i, M_j \in \mathcal{L}_n(\mathbb{R})$*, alors Bob obtiendra le bon résultat avec probabilité 3/4 sous condition qu'il connaisse les bases d'Alice.*

Démonstration. Lorsque Ève mesure un état BB84, l'état qu'elle envoie ensuite à Bob est le résultat de sa mesure. Soit θ, l'angle de l'état du projecteur M_i. Nous appellerons cet état, l'état i. L'état j sera l'état associé au projecteur M_j. Calculons la probabilité qu'Alice envoie $|\varphi_{00}\rangle$ et que Bob réussisse à obtenir la valeur 0 même avec l'intervention d'Ève.

$$p_{00} \cdot \Big(P\left[\text{È: état } i \mid |\varphi_{00}\rangle\right] \cdot P\left[\text{B: état } |0\rangle \mid \text{È: état } i\right] + P\left[\text{È: état } j \mid |\varphi_{00}\rangle\right] \cdot P\left[\text{B: état } |0\rangle \mid \text{È: état } j\right] \Big) .$$

Le résultat d'Ève dépend de l'état envoyé par Alice et le résultat de Bob dépend de l'état envoyé par Ève. De plus, même si Ève n'a pas le bon résultat, Bob peut encore retrouver la même valeur qu'Alice. Nous avons alors,

$$\frac{1}{4} \cdot \Big(\cos^2(\theta) \cdot \cos^2(\theta) + \sin^2(\theta) \cdot \sin^2(\theta) \Big) = \frac{1}{4} \cdot \Big(\cos^4(\theta) + \sin^4(\theta) \Big) .$$

En tenant compte de tous les états possibles envoyés par Alice, nous obtenons,

$$
\begin{aligned}
p_{succB} &= p_{00}\cdot\Big(P[\text{È:état } i]\cdot P[\text{B:état } |0\rangle] + P[\text{È:état } j]\cdot P[\text{B:état } |0\rangle]\Big) \\
&\quad + p_{10}\cdot\Big(P[\text{È:état } i]\cdot P[\text{B:état } |1\rangle] + P[\text{È:état } j]\cdot P[\text{B:état } |1\rangle]\Big) \\
&\quad + p_{01}\cdot\Big(P[\text{È:état } i]\cdot P[\text{B:état } |+\rangle] + P[\text{È:état } j]\cdot P[\text{B:état } |+\rangle]\Big) \\
&\quad + p_{11}\cdot\Big(P[\text{È:état } i]\cdot P[\text{B:état } |-\rangle] + P[\text{È:état } j]\cdot P[\text{B:état } |-\rangle]\Big) \\
&= \frac{1}{4}\Big(\cos^4\theta + \sin^4\theta\Big) + \frac{1}{4}\Big(\cos^4\theta + \sin^4\theta\Big) + \frac{1}{4}\Big(\cos^4(\theta-\tfrac{\pi}{4}) + \sin^4(\theta-\tfrac{\pi}{4})\Big) \\
&\quad + \frac{1}{4}\Big(\cos^4(\theta-\tfrac{\pi}{4}) + \sin^4(\theta-\tfrac{\pi}{4})\Big) \\
&= \frac{1}{2}\Big(\cos^4\theta + \sin^4\theta + \cos^4(\theta-\tfrac{\pi}{4}) + \sin^4(\theta-\tfrac{\pi}{4})\Big) \\
&= \frac{1}{2}\Big(\Big(\frac{1+\cos(2\theta)}{2}\Big)^2 + \Big(\frac{1-\cos(2\theta)}{2}\Big)^2 + \Big(\frac{1+\cos\Big(2\theta-\frac{\pi}{2}\Big)}{2}\Big)^2 + \Big(\frac{1-\cos\Big(2\theta-\frac{\pi}{2}\Big)}{2}\Big)^2\Big) \\
&= \frac{1}{2}\Big(\frac{1+2\cos(2\theta)+\cos^2(2\theta)+1-2\cos(2\theta)+\cos^2(2\theta)}{4} \\
&\quad + \frac{1+2\cos\Big(2\theta-\frac{\pi}{2}\Big)+\cos^2(2\theta-\frac{\pi}{2})+1-2\cos\Big(2\theta-\frac{\pi}{2}\Big)+\cos^2(2\theta-\frac{\pi}{2})}{4}\Big) \\
&= \frac{4+2\cos^2(2\theta)+2\cos^2(2\theta-\frac{\pi}{2})}{8} \\
&= \frac{4+2\Big(\frac{1+\cos(4\theta)}{2}\Big)+2\Big(\frac{1+\cos(4\theta-\pi)}{2}\Big)}{8} \\
&= \frac{4+1+\cos(4\theta)+1+\cos(4\theta-\pi)}{8} \\
&= \frac{6+\cos(4\theta)+\cos(4\theta)\cos\pi+\sin(4\theta)\sin\pi}{8} \\
&= \frac{3}{4}\,.
\end{aligned}
$$

□

Les équivalences proviennent des identités trigonométriques $\cos^2(a) = \frac{1+\cos(2a)}{2}$ et $\sin^2(a) = \frac{1-\cos(2a)}{2}$.

Peu importe la base dans laquelle Ève décide de mesurer ses qubits interceptés, le taux d'erreurs chez Bob reste le même. La différence de base affecte donc uniquement les résultats chez Ève. Comme les bases ne sont jamais divulguées publiquement, il est alors plus pertinent pour elle de mesurer chaque qubit dans la base de Breidbart afin d'obtenir un maximum d'information.

De plus, garder les bases secrètes ferait en sorte que l'attaque collective avec une machine de clonage quantique perdrait sa pertinence puisque Ève n'apprendra pas plus d'informations sur les bases pendant la communication classique. Conserver les imitations des qubits d'Alice pour les mesurer plus tard n'est donc plus aussi avantageux. Notre intuition nous laisse croire que l'analyse d'attaques incohérentes serait donc suffisante pour vérifier la sécurité d'un tel protocole.

De ce fait, nous limitons notre analyse de notre protocole BB84 modifié en ne considérant que les attaques du type i/r dans la base Breidbart.

5.2. Secret des bases

Cette version modifiée du protocole BB84 permet à Alice et Bob de recycler la séquence de bases. Contrairement au protocole standard, les bases ne sont jamais dévoilées. Nous pouvons réutiliser θ pour établir plusieurs clés, puisque sa min-entropie est et demeure maximale dans le cas où Ève décide d'attaquer dans la base de Breidbart.

Est-ce qu'on révèle quoi que ce soit sur θ lors du roulement du protocole? Pendant la communication quantique (étapes 1, 2 et 3), les bases restent secrètes. Ève peut seulement intercepter et mesurer des registres envoyés par Alice. Cependant, par les lois de la physique quantique, elle ne peut rien déduire sur la base dans laquelle le bit a été préparé par le résultat de sa mesure (Breidbart ou autre).

Si Ève apprend un bit de la chaine d'Alice en clair, elle ne peut rien savoir sur la base dans laquelle ce bit a été préparé puisqu'elle a mesuré dans la base de Breidbart. Par exemple, si elle apprend qu'un bit est 0, alors il y a 50% de chance que c'était un 0 dans la base rectilinéaire et 50% de chance que c'était un 0 dans la base diagonale. Les états $|0\rangle$ et $|+\rangle$ sont tous les deux à une distance $\pi/8$ (angle) du 0 dans la base de Breidbart. Ceci implique que toute information accumulée lors du filtrage ne compromet pas la confidentialité de θ.

Lors de la correction d'erreurs, θ reste sûr peu importe le code choisi.

Il est aussi intéressant de remarquer qu'après l'établissement de clé, dévoiler de l'information sur les messages clairs ne compromet pas la sécurité de θ. Supposons que, plus tard dans le futur, Ève obtient en clair un des messages qui a été chiffré. Elle peut donc retrouver la clé du chiffrement. Par rétro-ingénierie, elle peut obtenir la clé brute $\hat{x}$ puisqu'elle connait la fonction de hachage. Cependant, connaitre cette chaine ne lui donne pas d'information sur θ comme discuté plus haut. Alors, ce type de problème n'affecte pas le secret des bases.

Une analyse complète devrait tenir compte de tous les types d'attaques. Il ne faudrait pas seulement se préoccuper de l'information qu'Ève peut gagner sur la chaine d'Alice, mais aussi de l'information qu'elle peut obtenir sur les bases. Il est donc possible qu'elle puisse modeler ses attaques afin de diminuer son incertitude maximale sur θ au détriment de perdre de l'information sur la clé du moment. Ce serait des efforts qui porteraient fruits à long terme.

Dans un cas où Ève réussi à violer le secret des bases, nous pouvons mettre θ dans la fonction de hachage avec notre clé brute pour générer non seulement notre clé finale, mais une nouvelle séquence de bases aussi. À ce moment, il faudrait donc s'assurer que notre protocole puisse générer une clé assez longue pour en tirer le nouveau θ. Ce serait une solution intéressante si une analyse complète révèle que θ n'est plus sûr lorsqu'Ève utilise des attaques autres que les i/r dans la base de Breidbart.

L'étape avec le satellite peut sembler fastidieuse, mais elle offre des avantages à long terme. Le secret de θ nous permet non seulement d'augmenter la distance maximale entre deux partis pour un QKD, mais aussi de rouler le protocole à plusieurs reprises sans diminuer la tolérance aux erreurs.

5.3. Sécurité initiale

La chaine de Bob est affectée par le bruit du canal et les attaques d'Ève. Par contre, l'ajout des erreurs provenant du canal de communication augmente les chances qu'Alice et Bob avortent le protocole. Alors, enlever le bruit du canal est plus avantageux pour Ève. Moins d'erreurs chez Bob implique qu'elle peut introduire elle-même plus d'erreurs suite à ses attaques sans que le protocole soit avorté suite à l'estimation d'erreurs. Il lui suffit de transmettre directement les bits sortant de chez Alice à Bob, évitant alors que ceux-ci soient affectés par le bruit du canal. Par conséquent, notre analyse considèrera seulement les erreurs causées par l'adversaire.

Dans l'hypothèse que l'adversaire effectue des attaques de type i/r en mesurant dans la base de Breidbart, nous savons que le taux d'erreurs chez Bob lorsque Ève intercepte un qubit d'Alice est de 1/4. Si Ève intercepte δN des qubits de la chaine d'Alice, le taux d'erreurs du point de vue de Bob sera donc de $\delta/4$. Il en découle que le plus gros taux d'erreurs qui peut être introduit chez Bob est de 25% (si Ève attaque toute la chaine). Notons que ceci n'est qu'un seul type d'attaque et ne généralise pas toutes les possibilités de l'adversaire. Néanmoins, l'efficacité de l'attaque utilisant la base de Breidbart laisse croire qu'il serait difficile de faire mieux sachant que les bases de mesure ne sont jamais révélées. Nous laissons cette porte ouverte pour la prochaine génération (ou les anciennes).

Analysons l'attaque d'Ève et l'information qu'elle en tire. Elle choisit préalablement les δN positions qu'elle va intercepter. Lorsqu'elle attaque, elle mesure directement le qubit envoyé par Alice dans la base de Breidbart et note le résultat. Ensuite elle envoie l'état résultant à Bob. Lorsqu'elle n'attaque pas, elle ne fait que transmettre les qubits d'Alice directement à Bob sans introduire d'erreurs (par de la téléportation quantique par exemple). Ensuite, elle conserve une chaine de bit contenant les valeurs qui sont les plus probables chez Alice (lorsque 0 et 1 sont équiprobables, la valeur dans la chaine est "?").

Nous utilisons la min-entropie pour calculer l'incertitude d'Ève, par contre il est possible de modifier notre approche pour calculer d'autres entropies moins pessimistes comme l'entropie de collision. À ces fins, Ève devrait conserver une distribution de probabilité représentant les probabilités que chaque bit de la chaine d'Alice soit 0. Lorsque Ève mesure un qubit dans la base de Breidbart, elle sait que le bit d'Alice est 0 avec probabilité $\cos^2 \frac{\pi}{8}$ si sa mesure donne 0 et avec probabilité $\sin^2 \frac{\pi}{8}$ si la mesure donne 1. Lorsqu'elle ne mesure pas un qubit, elle ne sait rien sur le bit d'Alice, c'est-à-dire elle "sait" qu'Alice a le bit 0 avec probabilité 1/2. Cela définit la distribution de probabilité de la chaine d'Alice. Nous ne nous attarderons pas sur ce type d'information dans le cadre de ce travail compte tenu du type d'entropie utilisée.

Pour calculer la min-entropie, nous allons utiliser la probabilité de succès d'Ève ou encore, la probabilité que sa chaine soit égale à la chaine d'Alice. La mesure de Breidbart retourne à Ève la même valeur qu'Alice avec probabilité $\cos^2 \frac{\pi}{8}$. Bien entendu, lorsqu'elle n'intercepte pas le qubit, elle ne sait rien sur la valeur chez Alice. Elle peut donc le deviner avec une probabilité de 1/2 seulement. Nous définissons la probabilité de succès d'Ève comme suit:

$$s = \left(\frac{1}{2}\right)^{(1-\delta)N} \cdot \left(\cos^2\left(\frac{\pi}{8}\right)\right)^{\delta N} .$$

La min-entropie d'Ève sur la chaine d'Alice est alors,

$$\begin{aligned} H_{min}(\mathsf{X} \mid \mathcal{E}) : &= -\lg(s) \\ &= (1-\delta)N \lg\left(\frac{1}{2}\right) + \delta N \lg\left(\cos^2\left(\frac{\pi}{8}\right)\right) . \end{aligned}$$

Soit e, l'erreur de Bob après la communication quantique. Lorsqu'un qubit est intercepté par l'ennemie et mesuré dans une base $\{M_i, M_j\}$ quelconque avant d'être transmis à Bob, la probabilité de Bob d'obtenir le même bit qu'Alice (sachant que les deux utilisent la même base de mesure) est de 3/4. Rappelons-nous que les qubits non interceptés par Ève sont transmis sans erreur à Bob. L'erreur e est donc définie comme suit:

$$e := \frac{\delta}{4} .$$

Pour pouvoir faire la correction d'erreurs ainsi que l'amplification de secret, nous voulons que la min-entropie de l'adversaire soit plus grande que la longueur du syndrome d'Alice et de Bob (3.2.2). Soit $\ell := N - k$, la longueur du syndrome. Pour établir k, nous utilisons la capacité du canal. Nous voulons donc que le taux de transmission satisfasse

$$\frac{k}{N} \leq 1 - h(e) .$$

En isolant k, nous avons la longueur maximale possible du syndrome.

$$\ell = N \cdot h(e) = -\frac{\delta N}{4} \lg\left(\frac{\delta}{4}\right) - N\left(1 - \frac{\delta}{4}\right) \lg\left(1 - \frac{\delta}{4}\right) .$$

Puisque nous pouvons mettre N en évidence dans la min-entropie d'Ève ainsi que dans la longueur du syndrome, nous pouvons simplifier l'analyse en deux dimensions (selon δ seulement).

$$\not{N}\left((1-\delta)\lg\left(\frac{1}{2}\right)+\delta\lg\left(\cos^2\left(\frac{\pi}{8}\right)\right)\right) > \not{N}\left(-\frac{\delta}{2}\lg\left(\frac{\delta}{2}\right)-\left(1-\frac{\delta}{2}\right)\lg\left(1-\frac{\delta}{2}\right)\right).$$

Dans la Figure 1, nous pouvons voir où la longueur du syndrome devient plus grande que la min-entropie d'Ève selon le nombre de qubits attaqués.

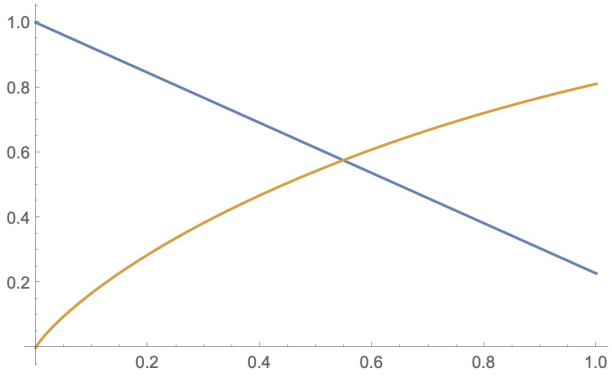


Fig. 5.2. Min-entropie d'Ève (bleu) et longueur du syndrome (jaune).

Le δ maximum où la min-entropie est encore plus grande que la longueur du syndrome est $\delta \approx 0.5486$ peu importe N (voir annexe A.1). Nous pouvons conclure que l'erreur maximum tolérée par Bob est $e = \frac{\delta}{4} \approx \frac{0.5486}{4} \approx 0.13715$. Nous avons une meilleure borne que BB84 non interactif (11%). Naturellement, garder les bases secrètes offre un clair avantage à Alice et Bob. Ce résultat n'est donc pas surprenant. Toutefois, notre analyse fixe une attaque pour Ève. Par conséquent, nos résultats sont sujets à amélioration.

Nous ne dépassons toujours pas le taux de 18.9% de Gottesman et Lo [**13**]. Suivant les travaux de Maurer, nous procédons à de la correction interactive pour pousser le taux d'erreurs. Nous commençons tout d'abord avec la tactique de filtrage.

5.4. Filtrage

5.4.1. Premier filtrage

Une fois la communication quantique complétée, Alice et Bob se retrouvent chacun avec une chaine de N bits classiques. Il est désormais important de visualiser les chaines par des chaines de paires de bits puisque la partie 2 manipule les bits par deux. Avant de commencer, ils permutent le tout. Par la suite, ils apparient chaque deux bits consécutifs. Alice et Bob ont donc $N/2$ paires de bits classiques. Comme le réarrangement des bits est annoncé sur le canal public classique, Ève sait comment apparier ses bits. Elle réorganise sa chaine comme

celle d'Alice et Bob. Par contre, grâce à la permutation, aux yeux d'Ève, toutes les paires formées de deux bits consécutifs sont équivalentes à des paires de bits choisies au hasard. Cela retire à Ève la possibilité d'avoir planifié une attaque qui prendrait en compte la position des bits et leur appartenance à une paire spécifique pendant la première partie. Elle est donc obligée de choisir au hasard les bits qu'elle veut intercepter lors de l'étape de communication quantique.

Nous commençons par définir les chaines avec une notation qui identifie clairement les paires après qu'Alice et Bob aient permuté et jeté les paires dont la parité ne concorde pas. Nous posons $2n$ comme étant la longueur des chaines après le tri des paires.

$$\begin{aligned} a &:= a_0^1\, a_1^1\, a_0^2\, a_1^2\, a_0^3\, a_1^3 ...\ a_0^n\, a_1^n \quad \text{(Alice)}\ , \\ b &:= b_0^1\, b_1^1\, b_0^2\, b_1^2\, b_0^3\, b_1^3 ...\ b_0^n\, b_1^n \quad \text{(Bob)}\ , \\ \hat{a} &:= \hat{a}_0^1\, \hat{a}_1^1\, \hat{a}_0^2\, \hat{a}_1^2\, \hat{a}_0^3\, \hat{a}_1^3 ...\ \hat{a}_0^n\, \hat{a}_1^n \quad \text{(Ève)}\ , \\ F_i^1 &:\ a_0^i \oplus a_1^i = b_0^i \oplus b_1^i\ . \end{aligned}$$

Les bits $a_0^i a_1^i$ forment donc la i-ème paire de bits dans la chaine d'Alice, de même pour Bob et Ève. F_i^1 représente l'évènement où Alice et Bob conservent un bit de la i-ème paire post-filtrage. Il y a n paires.

Notre adversaire a mesuré δN bits lors de la communication quantique. Après l'annonce des parités d'Alice et Bob, Ève regroupe les bits qu'elle a attaqué qui ont passé le filtrage dans un ensemble S. Plus précisément, l'ensemble contient les *positions* des bits attaqués qui font partis des paires conservées par Alice et Bob. Nous ne nous intéressons plus aux bits qui ne passent pas le tri puisqu'Alice et Bob les jettent. Du point de vue d'Ève, elle connait les bits de S avec probabilité de 85% et elle ne sait rien sur les autres.

Alice et Bob conservent un bit par paire i. Ève classifie les paires selon le type d'information qu'elle peut tirer sur le bit gardé par Alice et Bob. Elle forme l'ensemble $\mathcal{W}$ contenant toutes les paires où elle n'a mesuré aucun des deux qubits et l'ensemble $\mathcal{X}$ contenant les paires où elle a mesuré les deux qubits, mais n'a pas la même parité qu'Alice et Bob. Ces deux ensembles représentent donc les paires de bits où elle ne sait absolument rien. Elle crée aussi l'ensemble $\mathcal{Y}$ regroupant les paires où elle a mesuré un seul qubit et l'ensemble $\mathcal{Z}$ regroupant les paires où elle a mesuré les deux qubits et dont la parité concorde avec Alice et Bob. Les paires dans ces deux ensembles lui procurent de l'information pertinente. Les paires dans $\mathcal{Z}$ contiennent le plus d'information.

$$
\begin{aligned}
S &:= \left\{(i,j) \in \mathbb{N} \times \{0,1\} \mid \text{Ève a mesuré } \ \hat{a}_j^i\right\} \ , \\
\mathcal{W} &:= \left\{i \mid \Big((i,0),(i,1) \notin S\Big)\right\} \ , \\
\mathcal{X} &:= \left\{i \mid \Big((i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i \neq \hat{a}_0^i \oplus \hat{a}_1^i\Big)\right\} \ , \\
\mathcal{Y} &:= \left\{i \mid \Big((i,0) \in S \wedge (i,1) \notin S\Big) \vee \Big((i,0) \notin S \wedge (i,1) \in S\Big)\right\} \ , \\
\mathcal{Z} &:= \left\{i \mid (i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i = \hat{a}_0^i \oplus \hat{a}_1^i\right\} \ .
\end{aligned}
$$

À la fin du filtrage, Alice et Bob ont chacun une chaine contenant le deuxième bit de chaque paire dont la parité est la même. Soit $a_1a_2a_3...a_n$ où $a_i = a_1^i$ et $b_1b_2b_3...b_n$ où $b_i = b_1^i$. Ève construit une chaine $\hat{a}_1\hat{a}_2\hat{a}_3...\hat{a}_n$ en essayant de reproduire la chaine d'Alice.

Ève incorpore l'information qu'elle obtient de la communication classique à l'information venant de ses mesures à la communication quantique. Elle sait qu'Alice et Bob gardent toujours le deuxième bit de chaque paire conservée.

Lorsque:

- $i \in \mathcal{W}$, elle ne sait rien. Elle pose $\hat{a}_i =?$;
- $i \in \mathcal{X}$, elle ne sait rien. Elle pose $\hat{a}_i =?$;
- $i \in \mathcal{Y}$
  - $(i,0) \notin S, (i,1) \in S$, alors $\hat{a}_i = \hat{a}_1^i$;
  - $(i,0) \in S, (i,1) \notin S$, alors $\hat{a}_i = \hat{a}_0^i \oplus b_0^i \oplus b_1^i$;
- $i \in \mathcal{Z}$, alors $\hat{a}_i = \hat{a}_1^i$.

Lorsqu'une paire est dans $\mathcal{Y}$, même si le bit gardé par Alice et Bob n'est pas celui qu'Ève a mesuré auparavant, elle a néanmoins autant d'information sur celui-ci grâce à la parité annoncée. Une paire dans $\mathcal{Z}$ contient beaucoup d'information, puisqu'elle a mesuré les deux qubits et sa parité concorde avec celle d'Alice et Bob. Il n'est cependant pas plus avantageux pour elle de choisir un des deux bits au lieu de l'autre. Les deux sont indépendants et ont la même probabilité de succès 85%. Elle choisit simplement le même bit qu'Alice et Bob conserve.

Le but de cette section est de calculer la min-entropie d'Ève pour évaluer l'amélioration que procure la technique de filtrage. En supposant qu'elle limite ses attaques à des i/r avec la base de Breidbart, nous cherchons à savoir la quantité d'information qu'elle a obtenu selon son taux d'interception δ. Soient,

$$
\begin{aligned}
p_{\mathcal{W}} &:= P\left[i \in \mathcal{W}\right] = P\left[\Big((i,0),(i,1) \notin S\Big) \mid F_i^1\right] \ , \\
p_{\mathcal{X}} &:= P\left[i \in \mathcal{X}\right] = P\left[\Big((i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i \neq \hat{a}_0^i \oplus \hat{a}_1^i\Big) \mid F_i^1\right] \ , \\
p_{\mathcal{Y}} &:= P\left[i \in \mathcal{Y}\right] = P\left[\Big((i,0) \in S \wedge (i,1) \notin S\Big) \vee \Big((i,0) \notin S \wedge (i,1) \in S\Big) \mid F_i^1\right] \ , \\
p_{\mathcal{Z}} &:= P\left[i \in \mathcal{Z}\right] = P\left[(i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i = \hat{a}_0^i \oplus \hat{a}_1^i \mid F_i^1\right] \ ,
\end{aligned}
$$

qui représentent les probabilités d'être dans chaque groupe après le premier filtrage. Nous ne pourrons pas calculer ces probabilités directement. Il faudra d'abord les calculer avec un "ET" au lieu d'une condition ("$\ldots \wedge F_i^1$" au lieu de "$\ldots | F_i^1$"), pour ensuite utiliser la formule de la probabilité conditionnelle

$$P\left[A \mid B\right] = \frac{P\left[A \wedge B\right]}{P\left[B\right]} \ .$$

Une généralisation à plusieurs variables nous sera aussi utile:

$$P[A_1 \wedge A_2 \wedge ... \wedge A_k] = P[A_1] \cdot P[A_2 \mid A_1] \cdot ... \cdot P[A_k \mid A_1 \wedge A_2 \wedge ... \wedge A_{k-1}] \ .$$

Finalement, nous cherchons à calculer:

$$\begin{aligned} s_{\mathcal{W}} &:= P\left[\hat{a}_i = a_i \mid i \in \mathcal{W} \wedge F_i^1\right] \ , \\ s_{\mathcal{X}} &:= P\left[\hat{a}_i = a_i \mid i \in \mathcal{X} \wedge F_i^1\right] \ , \\ s_{\mathcal{Y}} &:= P\left[\hat{a}_i = a_i \mid i \in \mathcal{Y} \wedge F_i^1\right] \ , \\ s_{\mathcal{Z}} &:= P\left[\hat{a}_i = a_i \mid i \in \mathcal{Z} \wedge F_i^1\right] \ , \end{aligned}$$

qui représentent les probabilité de succès d'Ève de deviner le bit qu'Alice a choisi dans une paire (donc de deviner le deuxième bit de la paire d'Alice) selon le type de paire.

Pour alléger la notation dans les calculs, nous noterons

$$c := \cos^2\left(\frac{\pi}{8}\right) \text{ et } s := \sin^2\left(\frac{\pi}{8}\right) \ .$$

Puisque la majorité de nos ensembles sont conditionnés à l'évènement F_i^1, il est nécessaire de calculer sa probabilité. Puisqu'Alice et Bob ne conservent que les paires ayant la même parité, il ne peut y avoir 1 seule erreur dans une paire gardée, c'est-à-dire il y a 0 ou 2 erreurs qui se sont introduites dans la paire de Bob. Rappelons que $e = \delta/4$ est le taux d'erreurs introduit par Ève suite à la communication quantique.

$$P\left[F_i^1\right] = e^2 + (1-e)^2 = \left(\frac{\delta}{4}\right)^2 + \left(1 - \frac{\delta}{4}\right)^2 = \frac{\delta^2 - 4\delta + 8}{8} \ .$$

Soit e_1, l'erreur de Bob après le premier filtrage. Alice et lui gardent tous les deux le deuxième bit de leur paire. Bob a une erreur si ce bit est différent de celui d'Alice. Cela n'arrive que si les deux bits de la paire étaient erronés.

$$e_1 = \frac{P[\text{2 erreurs dans la paire}]}{P[F_i^1]} = \frac{e^2}{e^2 + (1-e)^2} = \frac{\left(\frac{\delta}{4}\right)^2}{\left(\frac{\delta}{4}\right)^2 \left(1 - \frac{\delta}{4}\right)^2} = \frac{\delta^2}{2(\delta^2 - 4\delta + 8)} \ .$$

Pour pouvoir calculer $p_{\mathcal{W}}$, $p_{\mathcal{X}}$, $p_{\mathcal{Y}}$ et $p_{\mathcal{Z}}$, il faut trouver:

$$P\left[\Big((i,0),(i,1) \notin S\Big) \wedge F_i^1\right] \ , \tag{1}$$

$$P\left[\Big((i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i \neq \hat{a}_0^i \oplus \hat{a}_1^i\Big) \wedge F_i^1\right] \ , \tag{2}$$

$$P\left[\Big(\Big((i,0) \in S \wedge (i,1) \notin S) \vee \Big((i,0) \notin S \wedge (i,1) \in S)\Big) \wedge F_i^1\right] \ , \tag{3}$$

$$P\left[(i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i = \hat{a}_0^i \oplus \hat{a}_1^i \wedge F_i^1\right] \ . \tag{4}$$

L'information que nous avons jusqu'à maintenant n'est pas suffisante pour calculer ces probabilités. En effet, les résultats d'Ève dépendent non seulement de ce qu'Alice envoie et ses mesures, mais aussi des résultats de Bob et ses parités annoncées. Il est donc important de tenir compte du côté de Bob. Cela rend les calculs plus complexes. Tout d'abord, nous analysons chacun des cas possibles. Puisqu'ils sont tous disjoints, nous additionnerons leur probabilité.

Les tableaux suivants représentent tous les cas possibles qui satisfont les conditions pour (1), (2), (3) et (4). Pour éviter les redondances, nous n'observons que les cas où $a_0^i a_1^i = 00$ (Alice a une paire 00). Nous pouvons voir qu'il y a une symétrie pour les autres cas $a_0^i a_1^i = 01, 10, 11$, puisque les bits chez Alice sont aléatoires. De plus, seulement le cas spécifique $(i,0) \notin S, (i,1) \in S$ (Ève a mesuré le *deuxième* bit de la paire) est représenté dans le tableau (3). Si $(i,0) \in S, (i,1) \notin S$ (Ève a mesuré le *premier* bit de la paire), les deux lignes du tableau seraient $00-0?-00$ et $00-1?-00$. Le résultat est le même, donc nous pouvons doubler notre réponse pour le tableau (3).

(1)		
$a_0^i a_1^i$	$\hat{a}_0^i \hat{a}_1^i$	$b_0^i b_1^i$
00	??	00

(2)		
$a_0^i a_1^i$	$\hat{a}_0^i \hat{a}_1^i$	$b_0^i b_1^i$
00	01	00
00	01	11
00	10	00
00	10	11

(3)		
$a_0^i a_1^i$	$\hat{a}_0^i \hat{a}_1^i$	$b_0^i b_1^i$
00	?0	00
00	?1	00

(4)		
$a_0^i a_1^i$	$\hat{a}_0^i \hat{a}_1^i$	$b_0^i b_1^i$
00	00	00
00	00	11
00	11	00
00	11	11

Tableau 5.1. Cas possibles chez Alice, Ève et Bob.

Calculons le seul cas possible pour (1).

$$
\begin{aligned}
&P\Big[(i,0),(i,1) \notin S \wedge a_0^i a_1^i = 00 \wedge b_0^i b_1^i = 00\Big] \\
&= P\left[(i,0),(i,1) \notin S\right] \cdot P\left[a_0^i a_1^i = 00 \mid (i,0),(i,1) \notin S\right] \\
&\qquad\qquad \cdot P\left[b_0^i b_1^i = 00 \mid a_0^i a_1^i = 00 \wedge (i,0),(i,1) \notin S\right] \\
&= (1-\delta)^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(1\cdot 1) \\
&= \frac{(1-\delta)^2}{4} \ .
\end{aligned}
$$

Nous avons alors,

$$
P\left[\Big((i,0),(i,1) \notin S) \wedge F_i^1\right] = 4 \cdot \Big(\frac{(1-\delta)^2}{4}\Big) = (1-\delta)^2 \ .
$$

Il ne faut pas oublier que le tableau ne montrait qu'un seul des quatre cas possible pour $a_0^i a_1^i$. Rappelons qu'ils sont tous équiprobables. Cela explique l'ajout de la constante multiplicative 4. Passons maintenant à (2), (3) et (4).

Pour (2):

$$
\begin{aligned}
P\Big[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 01 \wedge b_0^i b_1^i = 00\Big] &= \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(c\cdot s)(c\cdot s) = \frac{\delta^2 c^2 s^2}{4} \ , \\
P\Big[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 01 \wedge b_0^i b_1^i = 11\Big] &= \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(c\cdot s)(s\cdot c) = \frac{\delta^2 c^2 s^2}{4} \ , \\
P\Big[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 10 \wedge b_0^i b_1^i = 00\Big] &= \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(s\cdot c)(s\cdot c) = \frac{\delta^2 s^2 c^2}{4} \ , \\
P\Big[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 10 \wedge b_0^i b_1^i = 11\Big] &= \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(s\cdot c)(c\cdot s) = \frac{\delta^2 s^2 c^2}{4} \ .
\end{aligned}
$$

Donc,

$$
P\left[\Big((i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i \neq \hat{a}_0^i \oplus \hat{a}_1^i \wedge F_i^1\right] = 4\cdot\delta^2\Big(\frac{1}{4}\Big)(c^2s^2 + c^2s^2 + s^2c^2 + s^2c^2) = 4\delta^2c^2s^2 \ .
$$

Pour (3):

$$
\begin{aligned}
P\Big[(i,0) \notin S \wedge (i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_1^i = 0 \wedge b_0^i b_1^i = 00\Big] &= (1-\delta)\delta\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)\cdot c\cdot(1\cdot c) \\
&= \frac{(1-\delta)\delta c^2}{4} \ , \\
P\Big[(i,0) \notin S \wedge (i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_1^i = 1 \wedge b_0^i b_1^i = 00\Big] &= (1-\delta)\delta\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)\cdot s\cdot(1\cdot s) \\
&= \frac{(1-\delta)\delta s^2}{4} \ .
\end{aligned}
$$

Il ne faut pas oublier que les calculs ci-dessus sont pour le cas $(i,0) \notin S$ et $(i,1) \in S$. Le cas $(i,0) \in S$ et $(i,1) \notin S$ donne le même résultat. Il faut donc doubler le tout. Alors,

$$\begin{aligned} P\left[\Big(\big((i,0) \in S \wedge (i,1) \notin S\big) \vee \big((i,0) \notin S \wedge (i,1) \in S\big)\Big) \wedge F_i^1\right] &= 2 \cdot 4 \cdot (1-\delta)\delta\Big(\frac{1}{4}\Big)(c^2+s^2) \\ &= 2\delta(1-\delta)(c^2+s^2) \ . \end{aligned}$$

Pour (4):

$$P\left[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 00 \wedge b_0^i b_1^i = 00\right] = \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(c\cdot c)(c\cdot c) = \frac{\delta^2 c^4}{4} \ ,$$

$$P\left[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 00 \wedge b_0^i b_1^i = 11\right] = \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(c\cdot c)(s\cdot s) = \frac{\delta^2 c^2 s^2}{4} \ ,$$

$$P\left[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 11 \wedge b_0^i b_1^i = 00\right] = \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(s\cdot s)(s\cdot s) = \frac{\delta^2 s^4}{4} \ ,$$

$$P\left[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 11 \wedge b_0^i b_1^i = 11\right] = \delta^2\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)(s\cdot s)(c\cdot c) = \frac{\delta^2 s^2 c^2}{4} \ .$$

Donc,

$$\begin{aligned} P\left[(i,0),(i,1) \in S \wedge a_0^i \oplus a_1^i = \hat{a}_0^i \oplus \hat{a}_1^i \wedge F_i^1\right] &= 4 \cdot \delta^2\Big(\frac{1}{4}\Big)(c^4 + c^2 s^2 + s^4 + s^2 c^2) \\ &= \delta^2(c^4 + 2c^2 s^2 + s^4) \ . \end{aligned}$$

Nous pouvons enfin calculer les probabilités d'être dans chaque ensemble $\mathcal{W}$, $\mathcal{X}$, $\mathcal{Y}$ et $\mathcal{Z}$. N'oublions pas qu'il faut diviser par la probabilité de l'évènement conditionneur F_i^1.

$$\begin{aligned} p_{\mathcal{W}} &= (1-\delta)^2 \div \frac{\delta^2 - 4\delta + 8}{8} = \frac{8(1-\delta)^2}{\delta^2 - 4\delta + 8} \ , \\ p_{\mathcal{X}} &= 4\delta^2 c^2 s^2 \div \frac{\delta^2 - 4\delta + 8}{8} = \frac{32\delta^2 c^2 s^2}{\delta^2 - 4\delta + 8} \ , \\ p_{\mathcal{Y}} &= 2\delta(1-\delta)(c^2+s^2) \div \frac{\delta^2 - 4\delta + 8}{8} = \frac{16\delta(1-\delta)(c^2+s^2)}{\delta^2 - 4\delta + 8} \ , \\ p_{\mathcal{Z}} &= \delta^2(c^4 + 2c^2 s^2 + s^4) \div \frac{\delta^2 - 4\delta + 8}{8} = \frac{8\delta^2(c^4 + 2c^2 s^2 + s^4)}{\delta^2 - 4\delta + 8} \ . \end{aligned}$$

Nous pouvons voir sur la figure 5.3 que la quantité de bits qui sont inconnus d'Ève (dans $p_{\mathcal{W}}$ et $p_{\mathcal{X}}$) est vite surpassée par la quantité de bits dans $p_{\mathcal{Y}}$ et $p_{\mathcal{Z}}$ lorsque celle-ci attaque plus que la moitié de la chaine ($\delta > 0.5$). Notre intuition nous laisse croire qu'Ève commence a gagner un peu trop d'information pour qu'Alice et Bob puissent établir une clé sure. Nous verrons à la fin de cette sous-section le taux maximal δ exact pour garantir la sécurité.

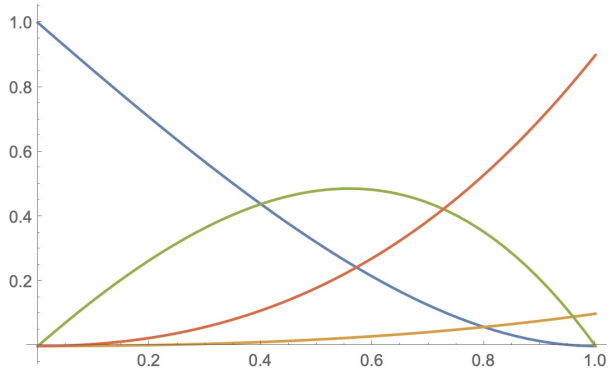


Fig. 5.3. $p_{\mathcal{W}}$ (bleu), $p_{\mathcal{X}}$ (jaune), $p_{\mathcal{Y}}$ (vert) et $p_{\mathcal{Z}}$ (rouge) en fonction de δ.

Nous savons que $s_{\mathcal{W}} = s_{\mathcal{X}} = 1/2$. Il reste à calculer $s_{\mathcal{Y}}$ et $s_{\mathcal{Z}}$. Nous avons besoin de

$$P\left[\hat{a}_i = a_i \wedge (i \in \mathcal{Y}) \wedge F_i^1\right] \quad \text{et} \quad P\left[\hat{a}_i = a_i \wedge (i \in \mathcal{Z}) \wedge F_i^1\right] \ .$$

Nous pourrons ensuite diviser ces deux probabilités par les probabilités des évènements conditionneurs $(i \in \mathcal{Y}) \wedge F_i^1$ et $(i \in \mathcal{Z}) \wedge F_i^1$ pour obtenir les probabilités conditionnelles $s_{\mathcal{Y}}$ et $s_{\mathcal{Z}}$. Les tableaux suivant sont simplement ceux du tableau 5.1 avec la condition additionnelle que le $\hat{a}_i$ d'Ève corresponde au a_i d'Alice, c'est-à-dire qu'Ève réussit à deviner le bit d'Alice. Elle ne fait que reconnaître son manque d'information.

(5)		
$a_0^i a_1^i$	$\hat{a}_0^i \hat{a}_1^i$	$b_0^i b_1^i$
00	?0	00

(6)		
$a_0^i a_1^i$	$\hat{a}_0^i \hat{a}_1^i$	$b_0^i b_1^i$
00	00	00
00	00	11

Tableau 5.2. Cas où Ève devine correctement.

Pour (5):

$$P\left[(i,0) \notin S \wedge (i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_1^i = 0 \wedge b_0^i b_1^i = 00\right] = \frac{(1-\delta)\delta c^2}{4} \ .$$

Il ne faut pas oublier que les calculs ci-dessus sont pour le cas $(i,0) \notin S, (i,1) \in S$. Le cas $(i,0) \in S, (i,1) \notin S$ donne le même résultat. Il faut donc doubler le tout. De plus, il faut multiplier par 4 pour tenir compte des trois autres possibilités chez Alice $(01, 10, 11)$. Alors,

$$\begin{aligned} P\left[\hat{a}_i = a_i \wedge \Big((i,0),(i,1) \in \mathcal{Y}\Big) \wedge F_i^1\right] &= 2 \cdot 4 \cdot (1-\delta)\delta\Big(\frac{1}{4}\Big)c^2 \\ &= 2\delta(1-\delta)c^2 \ . \end{aligned}$$

Pour (6):

$$P\Big[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 00 \wedge b_0^i b_1^i = 00\Big] = \frac{\delta^2 c^4}{4} \ ,$$
$$P\Big[(i,0),(i,1) \in S \wedge a_0^i a_1^i = 00 \wedge \hat{a}_0^i \hat{a}_1^i = 00 \wedge b_0^i b_1^i = 11\Big] = \frac{\delta^2 c^2 s^2}{4} \ .$$

Donc,

$$\begin{aligned} P\Big[\hat{B}_{(i,0),(i,1)} = B_{(i,0),(i,1)} \wedge \Big((i,0),(i,1)\Big) \in \mathcal{Z}\Big) \wedge F_i^1\Big] &= 4 \cdot \delta^2 \Big(\frac{1}{4}\Big)(c^4 + c^2 s^2) \\ &= \delta^2 (c^4 + c^2 s^2) \ . \end{aligned}$$

Finalement, on peut diviser ces probabilités pour obtenir les probabilités d'Ève de deviner le bit d'Alice conditionnées sur l'évènement F_i^1.

$$\begin{aligned} s_{\mathcal{W}} &= \frac{1}{2} \ , \\ s_{\mathcal{X}} &= \frac{1}{2} \ , \\ s_{\mathcal{Y}} &= \frac{2\delta(1-\delta)c^2}{\delta(1-\delta)(2c^2+2s^2)} = \frac{c^2}{c^2+s^2} \approx 0.971405 \ , \\ s_{\mathcal{Z}} &= \frac{\delta^2(c^4+c^2s^2)}{\delta^2(c^4+2c^2s^2+s^4)} = \frac{c^4+c^2s^2}{c^4+2c^2s^2+s^4} = \frac{c^2}{c^2+s^2} \approx 0.971405 \ . \end{aligned}$$

Contrairement à l'intuition initiale que la probabilité de succès d'Ève en étant dans $\mathcal{Z}$ soit plus grande que celle en étant dans $\mathcal{Y}$, nous pouvons voir qu'il n'y a pas de différence entre les deux. Pourtant, lorsque Ève est dans $\mathcal{Z}$, elle a mesuré les deux bits de la paire et elle est certaine qu'elle a la même parité qu'Alice et Bob. Cela veut dire qu'elle se tromperait seulement si ses deux mesures ont donné le mauvais résultat, ce qui est peu probable. Lorsqu'elle est dans $\mathcal{Y}$, elle n'a mesuré qu'un seul bit de la paire, ce qui lui donne moins de garanties sur son succès. Pourtant, les deux probabilités sont identiques! Rappelons que si Ève ne mesure qu'un seul bit dans une paire, elle laisse nécessairement l'autre bit intact. Elle transmet donc une paire à Bob qui ne peut contenir plus qu'une erreur. Puisqu'Alice et Bob comparent leur parité, une telle paire n'est conservée que si elle est sans erreur (puisque 2 erreurs seraient impossible). Il se peut qu'Ève obtienne le mauvais résultat suite à sa mesure. Par contre, Bob ne peut avoir une seule erreur dans sa paire. Si Ève obtient la mauvaise valeur, alors Bob doit tout de même retrouver la valeur originale qui serait, à ce moment, beaucoup moins probable. La méthode de filtrage offre donc à Ève une bonne probabilité de deviner le bit d'Alice lorsque sa paire est dans $\mathcal{Y}$.

Calculons maintenant la probabilité d'Ève de deviner la chaine d'Alice.

$$\begin{aligned} s_1 &:= (1/2)^{n\cdot(p_{\mathcal{W}}+p_{\mathcal{X}})}(s_{\mathcal{Y}})^{n\cdot p_{\mathcal{Y}}}(s_{\mathcal{Z}})^{n\cdot p_{\mathcal{Z}}} \\ &= \left(\frac{1}{2}\right)^{n\left(\frac{8((1-\delta)^2+4\delta^2c^2s^2)}{\delta^2-4\delta+8}\right)}\left(\frac{c^2}{c^2+s^2}\right)^{n\left(\frac{8\delta(1-\delta)(2c^2+2s^2)}{\delta^2-4\delta+8}\right)}\left(\frac{c^2}{c^2+s^2}\right)^{n\left(\frac{8\delta^2(c^4+2c^2s^2+s^4)}{\delta^2-4\delta+8}\right)} . \end{aligned}$$

La min-entropie d'Ève après le filtrage est alors,

$$H^1_{min} := -\lg(s_1) .$$

Ensuite, la longueur du syndrome est

$$\begin{aligned} \ell &\approx n\cdot h(e_1) \\ &= -\frac{\delta^2 n}{2(\delta^2-4\delta+8)}\lg\left(\frac{\delta^2}{2(\delta^2-4\delta+8)}\right) - \left(1-\frac{\delta^2 n}{2(\delta^2-4\delta+8)}\right)\lg\left(1-\frac{\delta^2}{2(\delta^2-4\delta+8)}\right) . \end{aligned}$$

Dans la Figure 3, nous pouvons voir où la longueur du syndrome devient plus grande que la min-entropie d'Ève lorsque δ augmente.

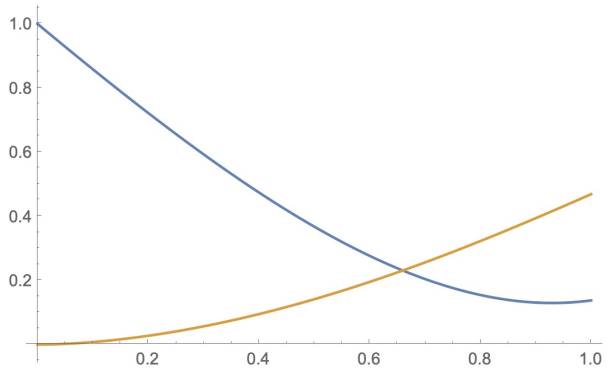


Fig. 5.4. Min-entropie d'Ève (bleu) et longueur du syndrome (jaune).

Le δ maximum où la min-entropie est encore plus grande que la longueur du syndrome est $\delta \approx 0.659748$ peu importe n (voir annexe A.2). L'erreur maximum tolérée par Bob est donc d'environ 16.49%. Le premier filtrage offre une amélioration du taux que nous obtenons avec la correction d'erreurs non-interactive traditionnelle (13.72%). Maintenant, nous appliquons une seconde fois la méthode de filtrage pour essayer de dépasser le taux obtenu grâce à la correction interactive de Gottesman et Lo [**13**] (18.9%).

5.4.2. Deuxième filtrage

L'amélioration précédente laisse croire que réappliquer le filtrage pourrait encore améliorer le taux maximal pour lequel Alice et Bob obtiennent une clé secrète. Cependant, l'analyse se complique. En employant la même tactique qu'à l'analyse du premier filtrage, les différents cas à analyser augmentent beaucoup. Il y a quatre types de bits pouvant former des paires résultant en plus d'une douzaine de catégories. On se retrouve donc avec une tâche beaucoup plus ardue.

Similairement au premier filtrage, Ève devine le bit conservé par Alice avec une probabilité de 1/2 si les deux bits de la paire proviennent de $\mathcal{Y}$ (même chose si les deux proviennent de $\mathcal{Z}$) et si la parité de la paire n'est pas la même que celle d'Alice et Bob. Rappelons que si elle n'a pas la même parité qu'Alice et Bob, elle choisit un des deux bits au hasard puisqu'ils ont la même probabilité de succès. Par contre, puisqu'elle sait qu'il y a une erreur sans savoir de quel bit provient cette erreur, son résultat est aussi bon qu'un choix purement aléatoire avec un succès de 1/2. Puisque les bits de $\mathcal{Y}$ et de $\mathcal{Z}$ ont la même probabilité de succès, nous pourrions essayer de les rassembler en une seule catégorie pour simplifier l'analyse de toutes les possibilités de paires. Supposons qu'elle ne différencie pas les bits provenant de $\mathcal{Y}$ des bits provenant de $\mathcal{Z}$. Si elle a une paire avec un bit de $\mathcal{Y}$ et un bit de $\mathcal{Z}$ avec une parité différente de celle d'Alice et Bob, elle va choisir un des deux bits au hasard puisqu'ils ont la même probabilité de succès. Comme elle ne sait pas de quel bit provient l'erreur, elle devine le bit conservé par Alice avec probabilité 1/2. En revanche, si elle décide qu'elle garde toujours le bit provenant de $\mathcal{Z}$ spécifiquement, elle aura une meilleure probabilité de deviner le bit d'Alice. Encore plus surprenant, sa probabilité de succès en choisissant le bit de $\mathcal{Y}$ est moins que 1/2, voulant dire qu'elle devrait prendre le complément de ce bit. Il est dont crucial de conserver les étiquettes des types de bits du filtrage précédant, puisque cela change comment Ève doit modeler sa chaine pour être le plus proche de celle d'Alice. Il est possible qu'il existe une meilleure solution pour déterminer l'information qu'Ève obtient suite au méthode de filtrage qui serait plus simple à analyser. Nous laissons cette question ouverte.

Nous obtenons un taux d'erreurs maximal toléré de 19.07% (voir annexe A.3). Nous dépassons la plus haute tolérance d'erreurs de 18.9% [**13**] (à notre connaissance). Cependant, en plus d'avoir une analyse de sécurité plus complexe, nous nous retrouvons dans une situation peu pratique. Rappelons que cette méthode diminue à la fois le taux d'erreurs de Bob et celui d'Ève. La min-entropie d'Ève est moins élevée qu'au premier filtrage, et le taux d'erreurs à corriger chez Bob aussi. L'intersection de la min-entropie d'Ève et de la longueur du syndrome est plus loin qu'au premier filtrage, mais très près de l'axe des x (voir fig. 5.5). En théorie, un deuxième filtrage améliore la borne, mais diminue presque totalement la min-entropie d'Ève. Pour résoudre ce problème, nous introduisons l'étape de

concentration. Nous verrons plus tard comment la concentration met Alice et Bob dans une situation plus favorable à la réapplication de la technique de filtrage. Nous ne montrons pas les détails de l'analyse d'un deuxième filtrage qui suit immédiatement un premier filtrage. Toutefois, l'analyse du deuxième filtrage après concentration est similaire. Elle est cependant plus complexe puiqu'elle tient compte de l'étape de concentration en sus.

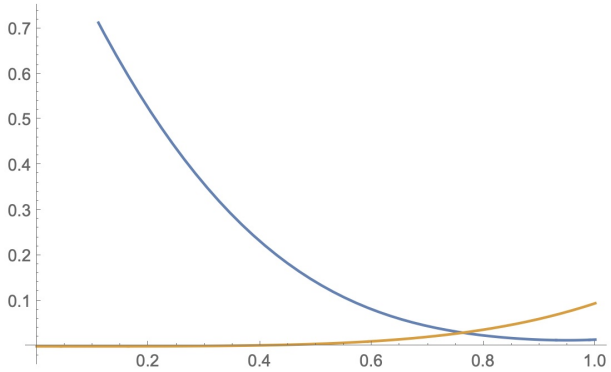


Fig. 5.5. Min-entropie d'Ève (bleu) et longueur du syndrome (jaune) après deux filtrages.

5.5. Concentration

Comme au filtrage, Alice et Bob permutent leur chaine pour ensuite apparier les bits consécutifs. Encore une fois, Ève ne peut planifier préalablement une attaque quantique basée sur les paires formées à cause de la permutation. Elle sait tout de même quelle est la permutation. Nous utilisons une notation similaire à l'étape du filtrage pour les chaines des trois partis post-permutation. Pour simplifier la notation, nous posons $m = n/2$.

$$
\begin{aligned}
A :&= A^1_{i_1}\, A^1_{i_2}\, A^2_{i_3}\, A^2_{i_4}\, A^3_{i_5}\, A^3_{i_6} \ldots A^m_{i_{2m-1}}\, A^m_{i_{2m}} \quad \text{(Alice)}\ , \\
B :&= B^1_{i_1}\, B^1_{i_2}\, B^2_{i_3}\, B^2_{i_4}\, B^3_{i_5}\, B^3_{i_6} \ldots B^m_{i_{2m-1}}\, B^m_{i_{2m}} \quad \text{(Bob)}\ , \\
\hat{A} :&= \hat{A}^1_{i_1}\, \hat{A}^1_{i_2}\, \hat{A}^2_{i_3}\, \hat{A}^2_{i_4}\, \hat{A}^3_{i_5}\, \hat{A}^3_{i_6} \ldots \hat{A}^m_{i_{2m-1}}\, \hat{A}^m_{i_{2m}} \quad \text{(Ève)}\ .
\end{aligned}
$$

L'indice j du haut identifie la j-ème paire de la chaine. L'indice du bas i_k représente la position du k-ième bit pré-permutation. Contrairement au filtrage, nous n'avons pas à définir un évènement pour conditionner nos probabilités. En effet, il n'y a pas d'annonce sur le canal classique qui pourrait influencer les résultats d'Ève comme à l'étape du filtrage. Les probabilités sont directement calculables sans devoir conditionner à un autre évènement. La concentration sert à diminuer considérablement la corrélation entre Alice et Ève même si celle entre Alice et Bob diminue un peu aussi. Rappelons que le filtrage annonçait les parités ainsi que les paires conservées. L'erreur chez Bob diminuait donc au détriment du gain d'information chez Ève.

Ève sépare les paires en différent type, dépendant de la nature des deux bits. Par "nature", nous voulons dire de quel ensemble provient le bit ($\mathcal{W}$, $\mathcal{X}$, $\mathcal{Y}$ ou $\mathcal{Z}$). Ces quatre ensembles contiennent les indices qui identifient les paires suite au premier filtrage. Seulement un bit est conservé par paires. Ces indices deviennent alors la position des bits dans la chaine résultante, donc avant la permutation de la concentration. Ève veut se rappeler de la provenance de chaque bit, puisqu'ils ont des probabilités de succès différentes. Après la permutation de la concentration, les indices i_k indiquent alors leur position pré-permutation. Ève utilise donc i_k pour savoir de quel ensemble ($\mathcal{W}, \mathcal{X}, \mathcal{Y}$ ou $\mathcal{Z}$) provient chaque bit. Naturellement, la j-ième paire chez Ève est définie comme suit: $\hat{A}^j_{i_{2j-1}}\hat{A}^j_{i_{2j}}$.

$$
\begin{aligned}
\mathcal{A} &:= \{j \mid i_{2j-1}, i_{2j} \in \mathcal{W}\} \ , \\
\mathcal{B} &:= \{j \mid i_{2j-1}, i_{2j} \in \mathcal{X}\} \ , \\
\mathcal{C} &:= \{j \mid i_{2j-1}, i_{2j} \in \mathcal{Y}\} \ , \\
\mathcal{D} &:= \{j \mid i_{2j-1}, i_{2j} \in \mathcal{Z}\} \ , \ , \\
\mathcal{E} &:= \{j \mid (i_{2j-1} \in \mathcal{W} \wedge i_{2j} \in \mathcal{X}) \vee (i_{2j-1} \in \mathcal{X} \wedge i_{2j} \in \mathcal{W})\} \ , \\
\mathcal{F} &:= \{j \mid (i_{2j-1} \in \mathcal{W} \wedge i_{2j} \in \mathcal{Y}) \vee (i_{2j-1} \in \mathcal{Y} \wedge i_{2j} \in \mathcal{W})\} \ , \\
\mathcal{G} &:= \{j \mid (i_{2j-1} \in \mathcal{W} \wedge i_{2j} \in \mathcal{Z}) \vee (i_{2j-1} \in \mathcal{Z} \wedge i_{2j} \in \mathcal{W})\} \ , \\
\mathcal{H} &:= \{j \mid (i_{2j-1} \in \mathcal{X} \wedge i_{2j} \in \mathcal{Y}) \vee (i_{2j-1} \in \mathcal{Y} \wedge i_{2j} \in \mathcal{X})\} \ , \\
\mathcal{I} &:= \{j \mid (i_{2j-1} \in \mathcal{X} \wedge i_{2j} \in \mathcal{Z}) \vee (i_{2j-1} \in \mathcal{Z} \wedge i_{2j} \in \mathcal{X})\} \ , \\
\mathcal{J} &:= \{j \mid (i_{2j-1} \in \mathcal{Y} \wedge i_{2j} \in \mathcal{Z}) \vee (i_{2j-1} \in \mathcal{Z} \wedge i_{2j} \in \mathcal{Y})\} \ .
\end{aligned}
$$

Selon la nature des bits dans chaque paire, ces dernières ont différentes probabilités d'avoir la même parité qu'Alice et Bob. Ève a précisément dix types de paires possibles. Puisqu'Alice et Bob font un XOR dans chaque paire, Ève fait de même. Puisqu'aucune information sur les paires n'est annoncée (comme la parité à l'étape de filtration), Alice et Bob peuvent conserver sans inquiétude la parité de chaque paire pour former une nouvelle chaine 2 fois plus courte. Cela a pour but d'augmenter l'incertitude chez leur adversaire. En effet, le moment qu'un des deux bits est inconnu d'Ève, la parité de la paire lui devient inconnue aussi. Puisque l'erreur de Bob est relativement basse grâce au filtrage, la concentration par XOR devrait, jusqu'à un certain point, moins l'affecter qu'Ève (voir fig. 5.9). Alice et Bob ne font aucun échange public pour cette partie, donc Ève ne peut faire mieux que conserver les parités de toutes ses paires. La chaine résultante chez Ève sera

$$(\hat{A}^1_{i_1} \oplus \hat{A}^1_{i_2})(\hat{A}^2_{i_3} \oplus \hat{A}^2_{i_4})(\hat{A}^3_{i_5} \oplus \hat{A}^3_{i_6})...(\hat{A}^m_{i_{2m-1}} \oplus \hat{A}^m_{i_{2m}}) \ .$$

Voyons comment cette méthode affecte le taux d'erreurs de nos partenaires et celui de l'adversaire. Similairement à l'analyse du filtrage, nous calculons la probabilité de succès de chaque type de paire ainsi que la proportion de chaque ensemble pour déterminer la nouvelle

min-entropie d'Ève. Les probabilités suivantes représentent les proportions des ensembles.

$$
\begin{aligned}
p_{\mathcal{A}} &:= P[j \in \mathcal{A}] = P[i_{2j-1}, i_{2j} \in \mathcal{W}] \ , \\
p_{\mathcal{B}} &:= P[j \in \mathcal{B}] = P[i_{2j-1}, i_{2j} \in \mathcal{X}] \ , \\
p_{\mathcal{C}} &:= P[j \in \mathcal{C}] = P[i_{2j-1}, i_{2j} \in \mathcal{Y}] \ , \\
p_{\mathcal{D}} &:= P[j \in \mathcal{D}] = P[i_{2j-1}, i_{2j} \in \mathcal{Z}] \ , \\
p_{\mathcal{E}} &:= P[j \in \mathcal{E}] = P[(i_{2j-1} \in \mathcal{W} \wedge i_{2j} \in \mathcal{X}) \vee (i_{2j-1} \in \mathcal{X} \wedge i_{2j} \in \mathcal{W})] \ , \\
p_{\mathcal{F}} &:= P[j \in \mathcal{F}] = P[(i_{2j-1} \in \mathcal{W} \wedge i_{2j} \in \mathcal{Y}) \vee (i_{2j-1} \in \mathcal{Y} \wedge i_{2j} \in \mathcal{W})] \ , \\
p_{\mathcal{G}} &:= P[j \in \mathcal{G}] = P[(i_{2j-1} \in \mathcal{W} \wedge i_{2j} \in \mathcal{Z}) \vee (i_{2j-1} \in \mathcal{Z} \wedge i_{2j} \in \mathcal{W})] \ , \\
p_{\mathcal{H}} &:= P[j \in \mathcal{H}] = P[(i_{2j-1} \in \mathcal{X} \wedge i_{2j} \in \mathcal{Y}) \vee (i_{2j-1} \in \mathcal{Y} \wedge i_{2j} \in \mathcal{X})] \ , \\
p_{\mathcal{I}} &:= P[j \in \mathcal{I}] = P[(i_{2j-1} \in \mathcal{X} \wedge i_{2j} \in \mathcal{Z}) \vee (i_{2j-1} \in \mathcal{Z} \wedge i_{2j} \in \mathcal{X})] \ , \\
p_{\mathcal{J}} &:= P[j \in \mathcal{J}] = P[(i_{2j-1} \in \mathcal{Y} \wedge i_{2j} \in \mathcal{Z}) \vee (i_{2j-1} \in \mathcal{Z} \wedge i_{2j} \in \mathcal{Y})] \ .
\end{aligned}
$$

Ensuite, nous avons les probabilités de succès de chaque type de paire.

$$
\begin{aligned}
s_{\mathcal{A}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{A}] \ , \\
s_{\mathcal{B}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{B}] \ , \\
s_{\mathcal{C}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{C}] \ , \\
s_{\mathcal{D}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{D}] \ , \\
s_{\mathcal{E}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{E}] \ , \\
s_{\mathcal{F}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{F}] \ , \\
s_{\mathcal{G}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{G}] \ , \\
s_{\mathcal{H}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{H}] \ , \\
s_{\mathcal{I}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{I}] \ , \\
s_{\mathcal{J}} &:= P[\hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \mid j \in \mathcal{J}] \ .
\end{aligned}
$$

Les probabilités d'appartenance aux différents types de paires se calculent assez facilement. Il n'y a pas d'évènement conditionnel et les évènements décrivant la nature de chaque bit sont indépendants grâce à la permutation. Par exemple, l'évènement où le premier bit d'une paire provient de $\mathcal{Y}$, ne nous dit rien sur le deuxième bit de la paire. La probabilité d'appartenance à l'un des ensembles $\mathcal{W}, \mathcal{X}, \mathcal{Y}$ ou $\mathcal{Z}$ est uniquement déterminée par la proportion des ensembles en question.

$$
\begin{aligned}
p_{\mathcal{A}} &= (p_{\mathcal{W}})^2 \ , \quad p_{\mathcal{E}} = 2 \cdot p_{\mathcal{W}} \cdot p_{\mathcal{X}} \ , \quad p_{\mathcal{I}} = 2 \cdot p_{\mathcal{X}} \cdot p_{\mathcal{Z}} \ , \\
p_{\mathcal{B}} &= (p_{\mathcal{X}})^2 \ , \quad p_{\mathcal{F}} = 2 \cdot p_{\mathcal{W}} \cdot p_{\mathcal{Y}} \ , \quad p_{\mathcal{J}} = 2 \cdot p_{\mathcal{Y}} \cdot p_{\mathcal{Z}} \ . \\
p_{\mathcal{C}} &= (p_{\mathcal{Y}})^2 \ , \quad p_{\mathcal{G}} = 2 \cdot p_{\mathcal{W}} \cdot p_{\mathcal{Z}} \ , \\
p_{\mathcal{D}} &= (p_{\mathcal{Z}})^2 \ , \quad p_{\mathcal{H}} = 2 \cdot p_{\mathcal{X}} \cdot p_{\mathcal{Y}} \ ,
\end{aligned}
$$

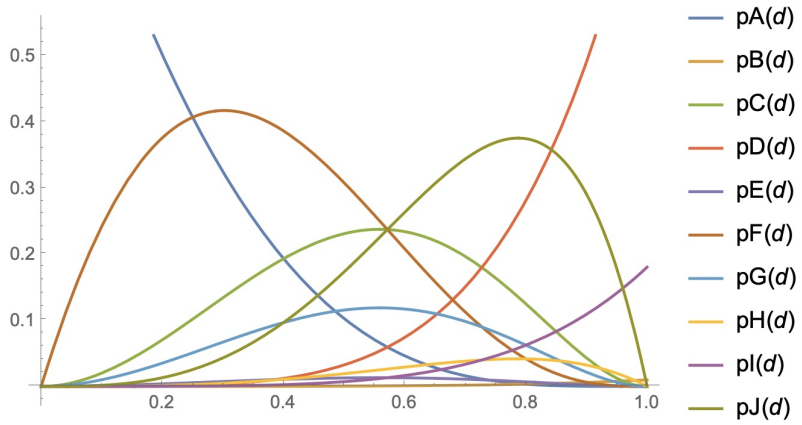


Fig. 5.6. Distribution de probabilité des ensembles $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, $\mathcal{I}$ et $\mathcal{J}$.

La figure 5.6 nous montre la répartition des différents ensembles selon le taux d'attaque δ. Nous savons qu'Ève ne sait rien sur la parité d'une paire si au moins un des deux bits lui est inconnu. Puisqu'Ève n'a aucune information sur un bit provenant de $\mathcal{W}$ ou $\mathcal{X}$, nous pouvons dire que $s_{\mathcal{A}} = s_{\mathcal{B}} = s_{\mathcal{E}} = s_{\mathcal{F}} = s_{\mathcal{G}} = s_{\mathcal{H}} = s_{\mathcal{I}} = 1/2$. Il est difficile de discerner ces groupes dans la figure précédente. La figure 5.7 regroupe $\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}$ et $\mathcal{I}$ pour mieux visualiser la probabilité d'être dans une paire totalement inconnue d'Ève. Nous voulons naturellement que cette probabilité soit grande pour maximiser son incertitude. Cette portion domine jusqu'à ce qu'on approche un taux d'attaque $\delta = 0.6$.

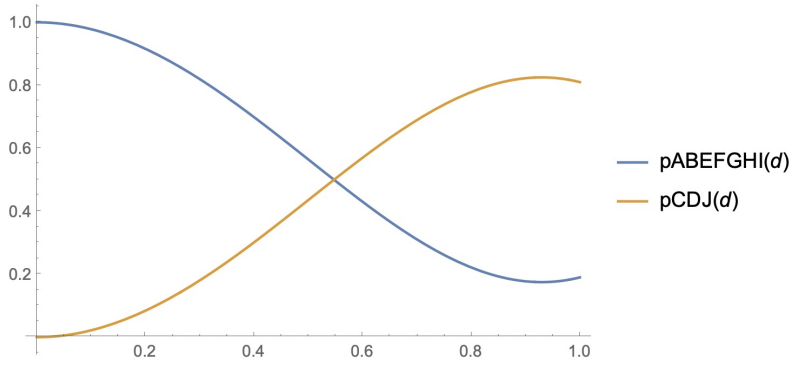


Fig. 5.7. Probabilité des groupes où Ève ne sait rien (bleu) et où Ève sait beaucoup (jaune).

Pour les autres groupes, elle ne réussit à deviner la parité d'Alice que si la paire contient 0 ou 2 erreurs. Nous pouvons utiliser les probabilités de succès des groupes $\mathcal{Y}$ et $\mathcal{Z}$.

$$\begin{aligned} s_{\mathcal{C}} &= (s_{\mathcal{Y}})^2 + (1 - s_{\mathcal{Y}})^2 \approx 0.944 \ , \\ s_{\mathcal{D}} &= (s_{\mathcal{Z}})^2 + (1 - s_{\mathcal{Z}})^2 \approx 0.944 \ , \\ s_{\mathcal{J}} &= s_{\mathcal{Y}} \cdot s_{\mathcal{Z}} + (1 - s_{\mathcal{Y}})(1 - s_{\mathcal{Z}}) \approx 0.944 \ . \end{aligned}$$

Rappelons que $s_{\mathcal{Y}} = s_{\mathcal{Z}}$, ce qui explique les résultats identiques. La probabilité de succès d'Ève est donc,

$$s_2 := (1/2)^{m\cdot(p_{\mathcal{A}}+p_{\mathcal{B}}+p_{\mathcal{E}}+p_{\mathcal{F}}+p_{\mathcal{G}}+p_{\mathcal{H}}+p_{\mathcal{I}})} \cdot (s_{\mathcal{C}})^{m\cdot p_{\mathcal{C}}} \cdot (s_{\mathcal{D}})^{m\cdot p_{\mathcal{D}}} \cdot (s_{\mathcal{J}})^{m\cdot p_{\mathcal{J}}} \ .$$

La min-entropie d'Ève devient

$$H_{min}(\mathsf{X} \mid \mathcal{E}) := -\lg(s_2) \ .$$

L'erreur de Bob dépend de l'intrusion ou non d'une seule erreur dans sa paire. Bien entendu, cette erreur peut être dans le premier ou le deuxième bit.

$$e_2 = 2 \cdot e_1(1 - e_1) \ .$$

Finalement, la longueur du syndrome est

$$\ell \approx m \cdot h(e_2) = m(-e_2 \cdot \lg(e_2) - (1 - e_2) \cdot \lg(e_2)) \ .$$

Encore une fois, nous n'avons pas à considérer la variable m ($n/2$ plus précisément) lors de la comparaison de l'entropie avec la longueur du syndrome. Nous ne faisons que varier le taux d'attaque δ.

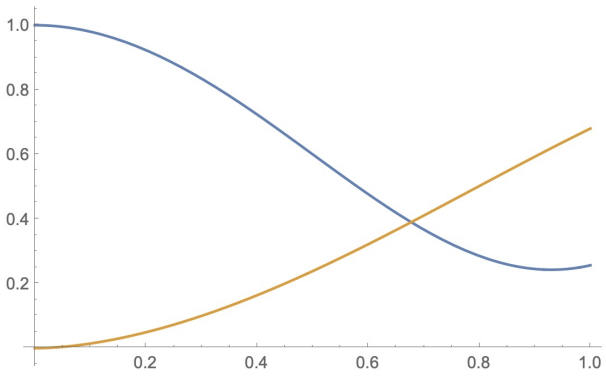
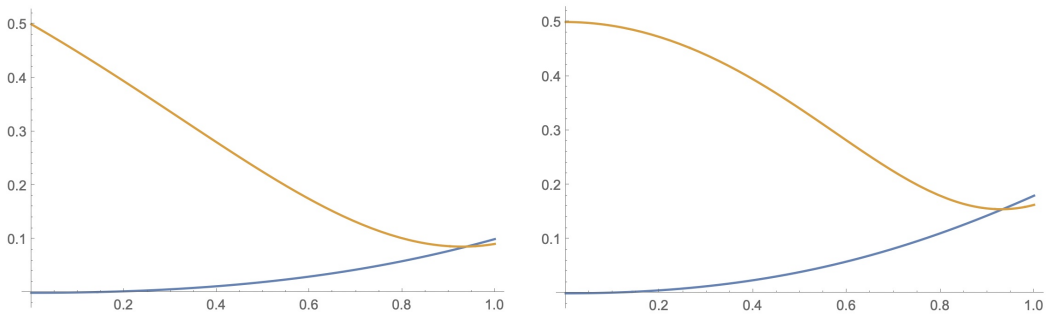


Fig. 5.8. Min-entropie d'Ève (bleu) et longueur du syndrome (jaune) après concentration.

Le delta maximum toléré par Alice et Bob est maintenant 0.67737. Ceci donne un taux d'erreurs de 16.93% (voir annexe A.4). Ce n'est pas une augmentation importante par rapport au premier filtrage. Cependant, ce n'était pas le but premier de cette étape. La figure 5.8 montre que la min-entropie d'Ève s'est considérablement éloignée de l'axe des

abscisses. Encore plus important, nous avons rétabli une distance entre l'erreur de Bob et celle de notre adversaire (voir fig. 5.9). Nous nous retrouvons donc avec un scénario propice à une reapplication de filtrage.



(a) Filtrage (b) Concentration

Fig. 5.9. Erreur de Bob (bleu) et erreur d'Ève (jaune).

5.6. Deuxième filtrage après concentration

Le filtrage nous aide à augmenter considérablement le taux d'erreurs toléré, mais diminue énormément le taux d'erreurs d'Ève. Il n'est donc pas pertinent de réappliquer immédiatement un autre filtrage pour rendre le taux d'erreurs d'Ève négligeable. C'est pourquoi nous ajoutons la méthode de concentration après le premier filtrage. Nous distançons l'erreur d'Ève de 0, même si cela n'améliore pas le taux d'erreurs toléré de beaucoup. Maintenant, nous procédons à une deuxième étape de filtrage. Alice et Bob permutent leur chaine. Ève fait de même. Nous supposons que les chaines sont de longueur $2r$ après qu'Alice et Bob aient filtré les paires dont la parité n'est pas la même. Ils ont donc r paires. Nous définissons leur nouvelle chaine comme suit:

$$\begin{aligned}
\mathbf{A} &:= \mathbf{A}^1_{j_1}\mathbf{A}^1_{j_2}\mathbf{A}^2_{j_3}\mathbf{A}^2_{j_4}\mathbf{A}^3_{j_5}\mathbf{A}^3_{j_6}...\mathbf{A}^r_{j_{2r-1}}\mathbf{A}^r_{j_{2r}} \quad \text{(Alice)} \ , \\
\mathbf{B} &:= \mathbf{B}^1_{j_1}\mathbf{B}^1_{j_2}\mathbf{B}^2_{j_3}\mathbf{B}^2_{j_4}\mathbf{B}^3_{j_5}\mathbf{B}^3_{j_6}...\mathbf{B}^r_{j_{2r-1}}\mathbf{B}^r_{j_{2r}} \quad \text{(Bob)} \ , \\
\hat{\mathbf{A}} &:= \hat{\mathbf{A}}^1_{j_1}\hat{\mathbf{A}}^1_{j_2}\hat{\mathbf{A}}^2_{j_3}\hat{\mathbf{A}}^2_{j_4}\hat{\mathbf{A}}^3_{j_5}\hat{\mathbf{A}}^3_{j_6}...\hat{\mathbf{A}}^r_{j_{2r-1}}\hat{\mathbf{A}}^r_{j_{2r}} \quad \text{(Ève)} \ , \\
F^2_k &: \mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \mathbf{B}^k_{j_{2k-1}} \oplus \mathbf{B}^k_{j_{2k}} \ .
\end{aligned}$$

L'indice k du haut identifie la k-ième paire de la chaine et l'indice j_t du bas représente la position du t-ième bit pré-permutation. L'évènement F^2_k représente l'évènement où Alice et Bob conservent la k-ième paire post-filtrage. Il est important de noter que nous sommes rendus au deuxième filtrage. L'évènement F^1_j sera toujours présent implicitement dans cette section. Pour alléger la notation, nous omettrons de l'écrire dans chaque probabilité. Toutes

les probabilités $P[E]$ seront en fait $P[E \mid F_j^1]$, c'est-à-dire toutes probabilités seront conditionnées sur le fait qu'un premier filtrage a déjà eu lieu.

Nous tenons à prévenir le lecteur que les calculs qui suivent sont laborieux. Le filtrage en soit nécessite une analyse en profondeur pour inclure la condition provenant de l'échange public des parités entre Alice et Bob. Le premier filtrage a une analyse de plusieurs pages avec seulement un groupe S de base (techniquement deux groupes, S et $\overline{S}$). Pour ce deuxième filtrage, nous avons dix groupes de base. Heureusement, une fois certaines définitions expliquées et certains calculs préparatoires, les étapes deviennent similaires au premier filtrage.

Notons qu'il est important de garder les groupes des méthodes précédentes séparés même s'ils ont le même succès ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}$ et $\mathcal{J}$). En effet, l'influence qu'ils ont sur les résultats de Bob varie. En conséquence, si nous fusionnons des types différents ensemble, les calculs ne seront pas exacts.

Comme au premier filtrage, Ève sépare les paires en groupe selon la provenance des bits. Nous avons 58 cas. Nous omettons donc d'énoncer tous les groupes explicitement.

$$
\begin{aligned}
\mathcal{AA} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{A}\} \ , \\
\mathcal{BB} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{B}\} \ , \\
\mathcal{CC}n &:= \{k \mid (j_{2k-1}, j_{2k} \in \mathcal{C}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} \neq \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}})\} \ , \\
\mathcal{CC} &:= \{k \mid (j_{2k-1}, j_{2k} \in \mathcal{C}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}})\} \ , \\
\mathcal{DD}n &:= \{k \mid (j_{2k-1}, j_{2k} \in \mathcal{D}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} \neq \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}})\} \ , \\
\mathcal{DD} &:= \{k \mid (j_{2k-1}, j_{2k} \in \mathcal{D}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}})\} \ , \\
\mathcal{EE} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{E}\} \ , \\
\mathcal{FF} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{F}\} \ , \\
\mathcal{GG} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{G}\} \ , \\
\mathcal{HH} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{H}\} \ , \\
\mathcal{II} &:= \{k \mid j_{2k-1}, j_{2k} \in \mathcal{I}\} \ , \\
\mathcal{JJ}n &:= \{k \mid (j_{2k-1}, j_{2k} \in \mathcal{J}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} \neq \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}})\} \ , \\
\mathcal{JJ} &:= \{k \mid (j_{2k-1}, j_{2k} \in \mathcal{J}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}})\} \ .
\end{aligned}
$$

Ces 13 groupes englobent les types de paires dont les deux bits ont la même nature, c'est-à-dire les paires dont les deux bits proviennent du même groupe de l'étape de concentration. Nous séparons certains en deux puisque la parité d'Ève joue sur le succès du groupe (comme $\mathcal{W}$ et $\mathcal{X}$ au premier filtrage). Si la paire d'Ève n'a pas la même parité que celles d'Alice et Bob, alors elle ne pourra pas extraire la même information. Les 45 ensembles restants sont

tous de la forme

$$E_1E_2 := \{k \mid (j_{2k-1} \in E_1 \wedge j_{2k} \in E_2) \vee (j_{2k-1} \in E_2 \wedge j_{2k} \in E_1)\} \ ,$$

où E_1 et $E_2 \in \{\mathcal{A}.\mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}, \mathcal{J}\}$ et $E_1 \neq E_2$. Nous considérons $E_1E_2 = E_2E_1$. Cela fait donc $C(2, 10)$ groupes ($C(k, n)$ dénote la combinaison de k éléments parmi n).

Lorsque Alice et Bob ont fini de trier les paires dont la parité est la même, ils conservent chacun le deuxième bit de chaque paire valide. Soit $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3...\mathbf{A}_r$ où $\mathbf{A}_k = \mathbf{A}^k_{j_{2k}}$ et $\mathbf{B}_1\mathbf{B}_2\mathbf{B}_3...\mathbf{B}_r$ où $\mathbf{B}_k = \mathbf{B}^k_{j_{2k}}$, les deux chaines résultantes d'Alice et Bob. Ève obtient une chaine $\hat{\mathbf{A}}_1\hat{\mathbf{A}}_2\hat{\mathbf{A}}_3...\hat{\mathbf{A}}_r$ qui tente de copier la chaine d'Alice.

Ève modèle sa chaine selon l'information qu'elle a accumulée jusque-là. Elle procède de la même façon qu'au premier filtrage. Il y a plusieurs ensembles qui ne lui sont pas utiles puisqu'ils forment des paires dans lesquelles les deux bits lui sont inconnus. Dans ces cas, elle ne peut rien faire. Par contre, si un seul des bits de la paire est inconnu, elle peut se servir de l'autre pour essayer de retrouver le bit qui sera gardé par Alice et Bob. Finalement, lorsque les deux bits proviennent de "bons" ensembles de la concentration, elle peut conserver l'un des deux si sa parité concorde avec celle annoncée. Dans le cas contraire, elle ne sait rien.

Lorsque:

(1) $k \in \mathcal{AA}$, $\mathcal{BB}$, $\mathcal{CC}n$, $\mathcal{DD}n$, $\mathcal{EE}$, $\mathcal{FF}$, $\mathcal{GG}$, $\mathcal{HH}$, $\mathcal{II}$, $\mathcal{JJ}$, $\mathcal{AB}$, $\mathcal{AE}$, $\mathcal{AF}$, $\mathcal{AG}$, $\mathcal{AH}$, $\mathcal{AI}$, $\mathcal{BE}$, $\mathcal{BF}$, $\mathcal{BG}$, $\mathcal{BH}$, $\mathcal{BI}$, $\mathcal{EF}$, $\mathcal{EG}$, $\mathcal{EH}$, $\mathcal{EI}$, $\mathcal{FG}$, $\mathcal{FH}$, $\mathcal{FI}$, $\mathcal{GH}$, $\mathcal{GI}$ ou $\mathcal{HI}$,
elle ne sait rien. Elle pose $\hat{\mathbf{A}}_k =?$;

(2) $k \in \mathcal{AC}$, $\mathcal{AD}$, $\mathcal{AJ}$, $\mathcal{BC}$, $\mathcal{BD}$, $\mathcal{BJ}$, $\mathcal{CE}$, $\mathcal{CF}$, $\mathcal{CG}$, $\mathcal{CH}$, $\mathcal{CI}$, $\mathcal{DE}$, $\mathcal{DF}$, $\mathcal{DG}$, $\mathcal{DH}$, $\mathcal{DI}$, $\mathcal{EJ}$, $\mathcal{FJ}$, $\mathcal{GJ}$, $\mathcal{HJ}$, $\mathcal{IJ}$
- $j_{2k} \notin \mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}$ ou $\mathcal{I}$ et $j_{2k-1} \in \mathcal{C}$, $\mathcal{D}$ ou $\mathcal{J}$, alors $\hat{\mathbf{A}}_k = \hat{\mathbf{A}}^k_{j_{2k}}$;
- $j_{2k-1} \in \mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}$ ou $\mathcal{I}$ et $j_{2k} \notin \mathcal{C}$, $\mathcal{D}$ ou $\mathcal{J}$, alors $\hat{\mathbf{A}}_k = \hat{\mathbf{A}}^k_{j_{2k-1}} \oplus \mathbf{B}^k_{j_{2k-1}} \oplus \mathbf{B}^k_{j_{2k}}$;

(3) $k \in \mathcal{CD}$, alors Ève choisit le bit $\in \mathcal{D}$;

(4) $k \in \mathcal{CJ}$, alors Ève choisit le bit $\in \mathcal{J}$;

(5) $k \in \mathcal{DJ}$, alors $\hat{\mathbf{A}}_k = \hat{\mathbf{A}}^k_{j_{2k}}$;

(6) $k \in \mathcal{CC}$, $\mathcal{DD}$, $\mathcal{JJ}$, alors $\hat{\mathbf{A}}_k = \hat{\mathbf{A}}^k_{j_{2k}}$.

Certains cas peuvent sembler non intuitifs. Comme nous l'avons expliqué à la section 5.4.2, il faut se rappeler de la provenance de chaque bit même si différents groupes à l'étape précédente ont la même probabilité de succès. C'est pour cette raison que les cas (3), (4) et (5) sont séparés. Pour chacun, nous avons testé toutes les possibilités pour voir laquelle est la plus avantageuse pour Ève. C'est un simple calcul lorsque nous avons un répertoire de données préparé. Comme (5) n'a pas d'avantage a choisir un bit spécifique, nous aurions pu le jumeler au cas (6).

Avant de procéder comme au premier filtrage, nous devons préparer le terrain. La technique brute avec laquelle nous analysons tous les cas possibles fonctionne toujours. Au premier filtrage, nous avions besoin des probabilités des divers résultats de mesure chez

Ève et chez Bob. Au deuxième filtrage (après concentration), nous devons plutôt utiliser les probabilités des divers résultats de la concentration. Nous retournons à l'analyse de la concentration pour calculer des probabilités qui seront utiles pour calculer tous les cas possibles.

Nous définissons,

$$\begin{aligned}
p_{bbb}(E) :&= P[\text{Après concentration: bit d'Alice = bit d'Ève = bit de Bob}] \\
&= P[A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} = \hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = B^j_{i_{2j-1}} \oplus B^j_{i_{2j}}] \ , \\
p_{bnbb}(E) :&= P[\text{Après concentration: bit d'Alice } \neq \text{ bit d'Ève = bit de Bob}] \\
&= P[A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} \neq \hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} = B^j_{i_{2j-1}} \oplus B^j_{i_{2j}}] \ , \\
p_{bbnb}(E) :&= P[\text{Après concentration: bit d'Alice = bit d'Ève } \neq \text{ bit de Bob}] \\
&= P[A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} = \hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}} \neq B^j_{i_{2j-1}} \oplus B^j_{i_{2j}}] \ , \\
p_{bnbnb}(E) :&= P[\text{Après concentration: bit d'Alice } = \text{ bit de Bob} \neq \text{ bit d'Ève }] \\
&= P[A^j_{i_{2j-1}} \oplus A^j_{i_{2j}} = B^j_{i_{2j}} \oplus B^j_{i_{2j+1}} \neq \hat{A}^j_{i_{2j-1}} \oplus \hat{A}^j_{i_{2j}}] \ ,
\end{aligned}$$

où $E \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$. Cette définition n'inclut pas les autres groupes de la concentration puisqu'au moins un des deux bits d'Ève n'est pas une valeur binaire, mais un "?". L'addition bit par bit n'a donc aucun sens. Ce sont les probabilités de différentes possibilités de relations entre Alice, Ève et Bob après la concentration. Nous utilisons ce type d'approche puisque les résultats sont plus généraux comparativement à l'utilisation de valeurs explicites. Nous pourrons donc les utiliser dans nos prochains calculs.

	Alice	Ève	Bob	Probabilités	
$\mathcal{Y}$	00	?0	00	$\frac{c^2}{c^2+s^2}$	①
	00	?1	00	$\frac{s^2}{c^2+s^2}$	②
$\mathcal{Z}$	00	00	00	$\frac{c^4}{c^4+2c^2s^2+s^4}$	③
	00	00	11	$\frac{c^2s^2}{c^4+2c^2s^2+s^4}$	④
	00	11	00	$\frac{s^4}{c^4+2c^2s^2+s^4}$	⑤
	00	11	11	$\frac{c^2s^2}{c^4+2c^2s^2+s^4}$	⑥

Tableau 5.3. Résultats possibles chez Alice, Ève et Bob des bits provenant de $\mathcal{Y}$ ou $\mathcal{Z}$.

Nous avons besoin des résultats possibles à la première filtration. Le tableau ci-dessus montre les résultats possibles et leurs probabilités selon leur type $\mathcal{Y}$ ou $\mathcal{Z}$. Rappelons que Bob conserve toujours le deuxième bit de sa paire.

La colonne d'Alice est fixée à la possibilité 00 afin de bien visualiser ce qu'il se passe. Rappelons que les quatre possibilités (00, 01, 10 et 11) sont toutes équiprobables pour Alice. Puisque nous tenons compte de la relation de la paire d'Alice avec celle de Bob et celle d'Ève et non de leur valeur en tant que telle, regarder le cas 00 est suffisant. Rappelons aussi que $\mathcal{C}$ contient les paires donc les deux bits proviennent de $\mathcal{Y}$, $\mathcal{D}$ représente les paires de deux bits de type $\mathcal{Z}$ et $\mathcal{J}$ englobe les paires ayant un bit de $\mathcal{Y}$ et un de $\mathcal{Z}$.

$$
\begin{aligned}
p_{bbb}(\mathcal{C}) &= ① \cdot ① + ② \cdot ② = \frac{c^4 + s^4}{(c^2 + s^2)^2} \ , \\
p_{bnbb}(\mathcal{C}) &= 0 \ , \\
p_{bbnb}(\mathcal{C}) &= 0 \ , \\
p_{bnbnb}(\mathcal{C}) &= ① \cdot ② + ② \cdot ① = \frac{2c^2 s^2}{(c^2 + s^2)^2} \ .
\end{aligned}
$$

Comme une paire de type $\mathcal{C}$ contient deux bits de types $\mathcal{Y}$, il est impossible que Bob ait un résultat différent d'Alice, d'où les probabilités nulles pour $p_{bnbb}(\mathcal{C})$ et $p_{bbnb}(\mathcal{C})$.

$$
\begin{aligned}
p_{bbb}(\mathcal{D}) &= ③ \cdot ③ + ④ \cdot ④ + ⑤ \cdot ⑤ + ⑥ \cdot ⑥ = \frac{c^8 + 2c^4 s^4 + s^8}{(c^4 + 2c^2 s^2 + s^4)^2} \ , \\
p_{bnbb}(\mathcal{D}) &= ③ \cdot ⑥ + ⑥ \cdot ③ + ④ \cdot ⑤ + ⑤ \cdot ④ = \frac{2c^6 s^2 + 2c^2 s^6}{(c^4 + 2c^2 s^2 + s^4)^2} \ , \\
p_{bbnb}(\mathcal{D}) &= ③ \cdot ④ + ④ \cdot ③ + ⑤ \cdot ⑥ + ⑥ \cdot ⑤ = \frac{2c^6 s^2 + 2c^2 s^6}{(c^4 + 2c^2 s^2 + s^4)^2} \ , \\
p_{bnbnb}(\mathcal{D}) &= ③ \cdot ⑤ + ⑤ \cdot ③ + ④ \cdot ⑥ + ⑥ \cdot ④ = \frac{4c^4 s^4}{(c^4 + 2c^2 s^2 + s^4)^2} \ .
\end{aligned}
$$

$$
\begin{aligned}
p_{bbb}(\mathcal{J}) &= \frac{① \cdot ③ + ③ \cdot ① + ② \cdot ⑤ + ⑤ \cdot ②}{2} = \frac{c^6 + s^6}{(c^2 + s^2)(c^4 + 2c^2 s^2 + s^4)} \ , \\
p_{bnbb}(\mathcal{J}) &= \frac{① \cdot ⑥ + ⑥ \cdot ① + ② \cdot ④ + ④ \cdot ②}{2} = \frac{c^4 s^2 + c^2 s^4}{(c^2 + s^2)(c^4 + 2c^2 s^2 + s^4)} \ , \\
p_{bbb}(\mathcal{J}) &= \frac{① \cdot ④ + ④ \cdot ① + ② \cdot ⑥ + ⑥ \cdot ②}{2} = \frac{c^4 s^2 + c^2 s^4}{(c^2 + s^2)(c^4 + 2c^2 s^2 + s^4)} \ , \\
p_{bbb}(\mathcal{J}) &= \frac{① \cdot ⑤ + ⑤ \cdot ① + ② \cdot ③ + ③ \cdot ②}{2} = \frac{c^2 s^4 + c^4 s^2}{(c^2 + s^2)(c^4 + 2c^2 s^2 + s^4)} \ .
\end{aligned}
$$

Nous divisons les cas de l'ensemble $\mathcal{J}$ par deux puisque nous devons enlever les permutations des deux bits de $\mathcal{Y}$ et $\mathcal{Z}$.

Nous définissons un type de probabilité semblable à ce que nous venons de faire, mais pour les ensembles dans lesquels Ève ne sait rien.

$$p_{bb}(E) := P[\text{Après concentration: bit d'Alice = bit Bob}] = P[A^j_{i_{2j}} \oplus A^j_{i_{2j+1}} = B^j_{i_{2j}} \oplus B^j_{i_{2j+1}}] \ ,$$
$$p_{bnb}(E) := P[\text{Après concentration: bit d'Alice } \neq \text{ bit Bob}] = P[A^j_{i_{2j}} \oplus A^j_{i_{2j+1}} \neq B^j_{i_{2j}} \oplus B^j_{i_{2j+1}}] \ ,$$

où $E \in \{\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}\}$. Comme les probabilités définies plus haut, celles-ci seront utiles pour les calculs qui s'en viennent. Nous les calculons de la même façon, mais sans tenir compte des résultats chez Ève.

	Alice	Ève	Bob	Probabilités	
$\mathcal{W}$	00	??	00	1	⑦
$\mathcal{X}$	00	??	00	$\frac{1}{2}$	⑧
	00	??	11	$\frac{1}{2}$	⑨

Tableau 5.4. Résultats possibles chez Alice et Bob des bits provenant de $\mathcal{W}$ ou $\mathcal{X}$.

$$p_{bb}(\mathcal{A}) = ⑦ \cdot ⑦ = 1 \ , \qquad p_{bnb}(\mathcal{A}) = 0 \ ,$$
$$p_{bb}(\mathcal{B}) = ⑧ \cdot ⑧ + ⑨ \cdot ⑨ = \frac{1}{2} \ , \qquad p_{bnb}(\mathcal{B}) = ⑧ \cdot ⑨ + ⑨ \cdot ⑧ = \frac{1}{2} \ ,$$
$$p_{bb}(\mathcal{E}) = \frac{⑦ \cdot ⑧ + ⑧ \cdot ⑦}{2} = \frac{1}{2} \ , \qquad p_{bnb}(\mathcal{E}) = \frac{⑦ \cdot ⑨ + ⑨ \cdot ⑦}{2} = \frac{1}{2} \ ,$$
$$p_{bb}(\mathcal{F}) = \frac{① \cdot ⑦ + ⑦ \cdot ① + ② \cdot ⑦ + ⑦ \cdot ②}{2} = 1 \ , \qquad p_{bnb}(\mathcal{F}) = 0 \ ,$$
$$p_{bb}(\mathcal{G}) = \frac{③ \cdot ⑦ + ⑦ \cdot ③ + ⑤ \cdot ⑦ + ⑦ \cdot ⑤}{2} = \frac{c^4 + s^4}{c^4 + 2c^2 s^2 + s^4} \ ,$$
$$p_{bnb}(\mathcal{G}) = \frac{④ \cdot ⑦ + ⑦ \cdot ④ + ⑥ \cdot ⑦ + ⑦ \cdot ⑤}{2} = \frac{2c^2 s^2}{c^4 + 2c^2 s^2 + s^4} \ ,$$
$$p_{bb}(\mathcal{H}) = \frac{① \cdot ⑧ + ⑧ \cdot ① + ② \cdot ⑧ + ⑧ \cdot ②}{2} = \frac{1}{2} \ ,$$
$$p_{bnb}(\mathcal{H}) = \frac{① \cdot ⑨ + ⑨ \cdot ① + ② \cdot ⑨ + ⑨ \cdot ②}{2} = \frac{1}{2} \ ,$$
$$p_{bb}(\mathcal{I}) = \frac{③ \cdot ⑧ + ⑧ \cdot ③ + ⑤ \cdot ⑧ + ⑧ \cdot ⑤ + ④ \cdot ⑨ + ⑨ \cdot ④ + ⑥ \cdot ⑨ + ⑨ \cdot ⑥}{2} = \frac{1}{2} \ ,$$
$$p_{bnb}(\mathcal{I}) = \frac{③ \cdot ⑨ + ⑨ \cdot ③ + ⑤ \cdot ⑨ + ⑨ \cdot ⑤ + ④ \cdot ⑧ + ⑧ \cdot ④ + ⑥ \cdot ⑧ + ⑧ \cdot ⑥}{2} = \frac{1}{2} \ .$$

Passons maintenant aux calculs principaux. Nous commençons par la probabilité de l'évènement qui servira a conditionner toutes nos probabilités. Rappelons que l'erreur de Bob après la concentration est e_2 et qu'Alice et Bob ne conservent une paire que s'il y a 0 ou 2 erreurs chez Bob.

$$P\left[F_k^2\right] = (e_2)^2 + (1-e_2)^2 \ .$$

Nous pouvons tout de suite calculer l'erreur de Bob après le second filtrage:

$$e_3 = \frac{(e_2)^2}{(e_2)^2 + (1-e_2)^2} \ .$$

Pour calculer la probabilité de succès d'Ève, nous devons tout d'abord calculer la probabilité d'être dans chaque ensemble. Cela fait donc 58 probabilités à calculer. Heureusement pour nous, la majorité suit le même patron. Les probabilités que nous venons de calculer nous aideront à généraliser les formules suivantes.

Soit,

$$\begin{aligned}
p_{\mathcal{AA}} &:= P[k \in \mathcal{AA}] = P[(j_{2k-1}, j_{2k} \in \mathcal{A}) \mid F_k^2] \ , \\
p_{\mathcal{BB}} &:= P[k \in \mathcal{BB}] = P[(j_{2k-1}, j_{2k} \in \mathcal{B}) \mid F_k^2] \ , \\
p_{\mathcal{CC}n} &:= P[k \in \mathcal{CC}n] = P[(j_{2k-1}, j_{2k} \in \mathcal{C}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} \neq \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}}) \mid F_k^2] \ , \\
p_{\mathcal{CC}} &:= P[k \in \mathcal{CC}] = P[(j_{2k-1}, j_{2k} \in \mathcal{C}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}}) \mid F_k^2] \ , \\
p_{\mathcal{DD}n} &:= P[k \in \mathcal{DD}n] = P[(j_{2k-1}, j_{2k} \in \mathcal{D}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} \neq \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}}) \mid F_k^2] \ , \\
p_{\mathcal{DD}} &:= P[k \in \mathcal{DD}] = P[(j_{2k-1}, j_{2k} \in \mathcal{D}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}}) \mid F_k^2] \ , \\
p_{\mathcal{EE}} &:= P[k \in \mathcal{EE}] = P[(j_{2k-1}, j_{2k} \in \mathcal{E}) \mid F_k^2] \ , \\
p_{\mathcal{FF}} &:= P[k \in \mathcal{FF}] = P[(j_{2k-1}, j_{2k} \in \mathcal{F}) \mid F_k^2] \ , \\
p_{\mathcal{GG}} &:= P[k \in \mathcal{GG}] = P[(j_{2k-1}, j_{2k} \in \mathcal{G}) \mid F_k^2] \ , \\
p_{\mathcal{HH}} &:= P[k \in \mathcal{HH}] = P[(j_{2k-1}, j_{2k} \in \mathcal{H} \mid F_k^2] \ , \\
p_{\mathcal{II}} &:= P[k \in \mathcal{II}] = P[(j_{2k-1}, j_{2k} \in \mathcal{I}) \mid F_k^2] \ , \\
p_{\mathcal{JJ}n} &:= P[k \in \mathcal{JJ}n] = P[(j_{2k-1}, j_{2k} \in \mathcal{J}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} \neq \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}}) \mid F_k^2] \ , \\
p_{\mathcal{JJ}} &:= P[k \in \mathcal{JJ}] = P[(j_{2k-1}, j_{2k} \in \mathcal{J}) \wedge (\mathbf{A}^k_{j_{2k-1}} \oplus \mathbf{A}^k_{j_{2k}} = \hat{\mathbf{A}}^{\,i}_{j_{2k-1}} \oplus \hat{\mathbf{A}}^{\,i}_{j_{2k}}) \mid F_k^2] \ .
\end{aligned}$$

Nous épargnons au lecteur de la lecture de la liste exhaustive restante.

$$p_{E_1E_2} := P[k \in E_1E_2] = P[(j_{2k-1} \in E_1 \wedge j_{2k} \in E_2) \vee (j_{2k-1} \in E_2 \wedge j_{2k} \in E_1) \mid F_k^2] \ ,$$

où $E_1 \neq E_2$. De plus, $p_{E_1E_2} = p_{E_2E_1}$.

Comme au premier filtrage, nous devons, en premier lieu, trouver les probabilités non conditionnées (énoncée comme $P[... \wedge F_k^2]$) pour ensuite diviser par la probabilité de l'évènement conditionnel. Au lieu d'énoncer tous les résultats possibles relatifs à chaque cas, nous présentons trois tableaux généraux qui tiennent compte de toutes possibilités dépendant de combien de bit sont inconnus d'Ève. Le lecteur peut s'y référer pour comprendre les calculs qui suivront.

Deux bits inconnus d'Ève		
$\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}$	$\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}$	$\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}$
00	??	00
00	??	11

Un bit inconnu d'Ève		
$\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}$	$\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}$	$\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}$
00	?0	00
00	?1	00
00	?0	11
00	?1	11

Aucun bit inconnu d'Ève		
$\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}$	$\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}$	$\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}$
00	00	00
00	00	11
00	01	00
00	01	11
00	10	00
00	10	11
00	11	00
00	11	11

Tableau 5.5. Résultats possibles chez Alice, Ève et Bob après filtration, concentration et seconde filtration.

Nous offrons des patrons pour les calculs. Tous les ensembles associés à un même type de tableau (ex: les deux bits proviennent d'ensembles pour lesquels Ève ne sait rien) suivent le même patron.

Commençons par calculer $p_{E_1E_2}$ où $(E_1, E_2) \in \{\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}\}^2$. Rappelons que selon leur définition, $p_{E_1E_2} = p_{E_2E_1}$. Le patron suivant fonctionne pour les deux ordres, mais nous conserverons qu'un seul des deux lorsque nous calculerons la probabilité de succès d'Ève. Bien entendu, nous devons tout d'abord calculer $P[(j_{2k-1}, j_{2k} \in E \wedge F_k^2]$ avant de calculer la probabilité conditionnelle. Le premier tableau (Deux bits inconnus d'Ève) montre les deux seuls cas qui sont possibles pour ces ensembles.

Les tableaux de 5.5 ne représentent que les cas où Alice envoie des 0 dès le début de la communication quantique. Heureusement, les autres possibilités sont toutes équiprobables. Il suffit donc de multiplier nos résultats par 4. Même si nous visualisons des valeurs fixes, nous utilisons nos probabilités de relation calculées plus haut. En connaissant la valeur

d'Alice, il n'est pas important de savoir la valeur spécifique des bits de Bob, mais de leur *relation* par rapport au bits d'Alice. Ceci nous permet de simplement multiplier le tout par 4 pour tenir compte des autres valeurs possibles chez Alice. Il sera de même pour les autres probabilités qui tiennent aussi compte de la relation avec les bits d'Ève.

$$P\left[\left(j_{2k-1}, j_{2k} \in EE\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\ = (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot ((p_{bb}(E) \cdot (p_{bb}(E)) \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EE\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\ = (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot ((p_{bnb}(E) \cdot (p_{bnb}(E)) \ .$$

Maintenant, le total en additionnant tous les cas et en multipliant par 4 est

$$P[(j_{2k}, j_{2k+1} \in EE \wedge F_k^2] = 4 \cdot \left((p_E)^2 \cdot \frac{1}{4} \cdot \left(p_{bb}(E)^2 + p_{bnb}(E)^2\right)\right) = (p_E)^2\left(p_{bb}(E)^2 + p_{bnb}(E)^2\right) \ .$$

Il est bien de noter que le tout est symétrique. L'ordre de E_1 et E_2 dans le nom $p_{E_1E_2}$ n'a donc pas d'importance, ce qui confirme que nous pouvons conserver $p_{E_1E_2}$ ou $p_{E_2E_1}$.

Ensuite, calculons les probabilités des groupes où les deux ensembles sont bien connus d'Ève et identiques, mais Ève n'a pas la même parité qu'Alice et Bob. Soit les ensembles p_{EEn} où $E \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$.

$$P\left[\left(j_{2k-1}, j_{2k} \in EEn\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 01\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\ = (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbb}(E) \cdot p_{bnbnb}(E) \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EEn\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 01\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\ = (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbnb}(E) \cdot p_{bnbb}(E) \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EEn\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 10\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\ = (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbnb}(E) \cdot p_{bbb}(E) \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EEn\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 10\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\ = (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbb}(E) \cdot p_{bbnb}(E) \ .$$

Le total de ces probabilités est

$$P[(j_{2k-1}, j_{2k} \in EEn \wedge F_k^2] = 4 \cdot \left((p_E)^2 \cdot \frac{1}{4} \cdot \left(2 \cdot p_{bbb}(E) \cdot p_{bnbnb}(E) + 2 \cdot p_{bbnb}(E) \cdot p_{bnbb}(E)\right)\right) \\ = 2 \cdot (p_E)^2 \cdot \left(p_{bbb}(E) \cdot p_{bnbnb}(E) + p_{bbnb}(E) \cdot p_{bnbb}(E)\right) \ .$$

Calculons les cas contraires, donc lorsque les deux ensembles sont identiques et bien connus d'Ève et sa parité est la même qu'Alice et Bob. Nous cherchons les probabilités p_{EE} où $E \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$.

$$P\left[\left(j_{2k-1}, j_{2k} \in EE\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^{\,k}_{j_{2k-1}}\hat{\mathbf{A}}^{\,k}_{j_{2k}} = 00\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right]$$
$$= (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbb}(E)^2 \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EE\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^{\,k}_{j_{2k-1}}\hat{\mathbf{A}}^{\,k}_{j_{2k}} = 00\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right]$$
$$= (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbnb}(E)^2 \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EE\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^{\,k}_{j_{2k-1}}\hat{\mathbf{A}}^{\,k}_{j_{2k}} = 11\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right]$$
$$= (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbnb}(E)^2 \ ,$$

$$P\left[\left(j_{2k-1}, j_{2k} \in EE\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^{\,k}_{j_{2k-1}}\hat{\mathbf{A}}^{\,k}_{j_{2k}} = 11\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right]$$
$$= (p_E)^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbb}(E)^2 \ .$$

Le total de ces cas est

$$P[(j_{2k-1}, j_{2k} \in EE \wedge F_k^2] = 4 \cdot \left((p_E)^2 \cdot \frac{1}{4} \cdot \left(p_{bbb}(E)^2 + p_{bbnb}(E)^2 + p_{bnbnb}(E)^2 + p_{bnbb}(E)^2\right)\right)$$
$$= (p_E)^2 \cdot \left(p_{bbb}(E)^2 \cdot p_{bnbb}(E)^2 + p_{bbnb}(E)^2 \cdot p_{bnbnb}(E)^2\right) \ .$$

Maintenant, calculons les probabilités des groupes tels que l'un des deux ensembles est inconnu d'Ève. Donc $E_1 \in \{\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}\}$ et $E_2 \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$. Il est important de remarquer que le tableau "un bit inconnu d'Ève" (voir 5.5) ne prend pas en compte l'ordre des bits. Puisque les deux proviennent de groupes différents, l'ordre dans la paire est important. Il faut alors multiplier le tout par 2 pour tenir compte des paires du type E_2E_1.

Pour $p_{E_1E_2}$,

$$P\left[\Big(j_{2k-1}\in E_1\wedge j_{2k}\in E_2\Big)\wedge\Big(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}=00\Big)\wedge\Big(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}=?0\Big)\wedge\Big(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}=00\Big)\right]$$
$$=(p_{E_1}\cdot p_{E_2})\cdot\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)\cdot p_{bb}(E_1)\cdot p_{bbb}(E_2)\ ,$$
$$P\left[\Big(j_{2k-1}\in E_1\wedge j_{2k}\in E_2\Big)\wedge\Big(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}=00\Big)\wedge\Big(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}=?0\Big)\wedge\Big(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}=11\Big)\right]$$
$$=(p_{E_1}\cdot p_{E_2})\cdot\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)\cdot p_{bnb}(E_1)\cdot p_{bbnb}(E_2)\ ,$$
$$P\left[\Big(j_{2k-1}\in E_1\wedge j_{2k}\in E_2\Big)\wedge\Big(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}=00\Big)\wedge\Big(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}=?1\Big)\wedge\Big(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}=00\Big)\right]$$
$$=(p_{E_1}\cdot p_{E_2})\cdot\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)\cdot p_{bb}(E_1)\cdot p_{bnbnb}(E_2)\ ,$$
$$P\left[\Big(j_{2k-1}\in E_1\wedge j_{2k}\in E_2\Big)\wedge\Big(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}}=00\Big)\wedge\Big(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}}=?1\Big)\wedge\Big(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}}=11\Big)\right]$$
$$=(p_{E_1}\cdot p_{E_2})\cdot\Big(\frac{1}{2}\cdot\frac{1}{2}\Big)\cdot p_{bnb}(E_1)\cdot p_{bnbb}(E_2)\ .$$

Le total est

$$P[(j_{2k-1},j_{2k}\in E_1E_2\wedge F^2_k]$$
$$=2\cdot4\cdot(p_{E_1}\cdot p_{E_2})\Big(p_{bb}(E_1)p_{bbb}(E_2)+p_{bnb}(E_1)p_{bbnb}(E_2)$$
$$+p_{bb}(E_1)p_{bnbnb}(E_2)+p_{bnb}(E_1)p_{bnbb}(E_2)\Big)\Big)\ .$$

Pour finir, nous calculons les probabilités pour les groupes ayant deux ensembles distincts qu'Ève connait bien. Donc $E_1,E_2\in\{\mathcal{C},\mathcal{D},\mathcal{J}\}$ et $E_1\neq E_2$.

$$
\begin{aligned}
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 00\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbb}(E_1) \cdot p_{bbb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 00\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbnb}(E_1) \cdot p_{bbnb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 01\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbb}(E_1) \cdot p_{bnbnb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 01\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bbnb}(E_1) \cdot p_{bnbb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 10\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbnb}(E_1) \cdot p_{bbb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 10\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbb}(E_1) \cdot p_{bbnb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 11\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 00\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbnb}(E_1) \cdot p_{bnbnb}(E_2) \ , \\
P\left[\left(j_{2k-1}, j_{2k} \in E_1E_2\right) \wedge \left(\mathbf{A}^k_{j_{2k-1}}\mathbf{A}^k_{j_{2k}} = 00\right) \wedge \left(\hat{\mathbf{A}}^k_{j_{2k-1}}\hat{\mathbf{A}}^k_{j_{2k}} = 11\right) \wedge \left(\mathbf{B}^k_{j_{2k-1}}\mathbf{B}^k_{j_{2k}} = 11\right)\right] \\
= (p_{E_1} \cdot p_{E_2}) \cdot \left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot p_{bnbb}(E_1) \cdot p_{bnbb}(E_2) \ .
\end{aligned}
$$

Nous épargnons au lecteur la lecture peu esthétique de la somme finale, mais rappelons que le tout doit être multiplié par 2 pour l'ordre contraire des bits dans la paire et par 4 pour les quatre cas possibles chez Alice.

Il reste seulement à diviser chaque total obtenu par $P\left[F_k^2\right]$ pour avoir les probabilités conditionnelles. Nous passons tout de suite aux calculs des probabilités de succès. Comme à l'analyse du premier filtrage (5.4.1), nous calculons d'abord les probabilités non conditionnées et divisons ensuite par les probabilités des conditions même (qui sont les totaux que nous venons de calculer).

Nous allons accélérer le pas. Pour calculer les probabilités de succès de chaque ensemble, nous prenons les mêmes cas ci-dessus, mais nous n'additionnons que ceux qui satisfont un succès chez Ève selon sa méthode d'attaque à la page 64. Ensuite, nous divisons chaque probabilité par le total calculé plus haut pour obtenir la probabilité de succès d'Ève de chaque ensemble. Pour voir les calculs détaillés de cette section, veuillez voir l'annexe A.5.

Enfin, nous calculons s_3, la probabilité de succès d'Ève.

$$s_3 := \prod_{A_1A_2} \left(\frac{1}{2}\right)^{n(p_{A_1A_2})} \cdot \prod_{B} \left(\frac{1}{2}\right)^{n(p_{BBn})} \cdot \prod_{C} (s_{CC})^{n(p_{CC})} \cdot \prod_{D_1D_2} (s_{D_1D_2})^{n(p_{D_1D_2})} \cdot \prod_{E_1E_2} (s_{E_1E_2})^{n(p_{E_1E_2})} ,$$

où $A_1, A_2 \in \{\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}\}$ et A_1 et A_2 sont en ordre alphabétique (pour éviter les doublures A_1A_2 et A_2A_1), $B \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$, $C \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$, $D_1 \in \{\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}\}$, $D_2 \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$ et $E_1, E_2 \in \{\mathcal{C}, \mathcal{D}, \mathcal{J}\}$ et E_1 et E_2 sont en ordre alphabétique.

La min-entropie d'Ève est

$$H_{min}(\mathsf{X} \mid \mathcal{E}) := -\lg(s_3) .$$

La longueur du syndrome selon l'erreur e_3 chez Bob est

$$\ell \approx n \cdot h(e_3) .$$

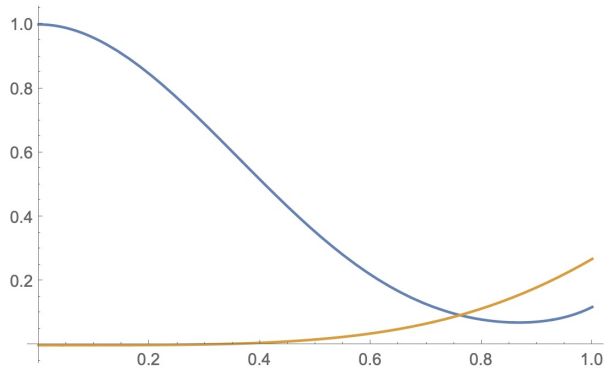
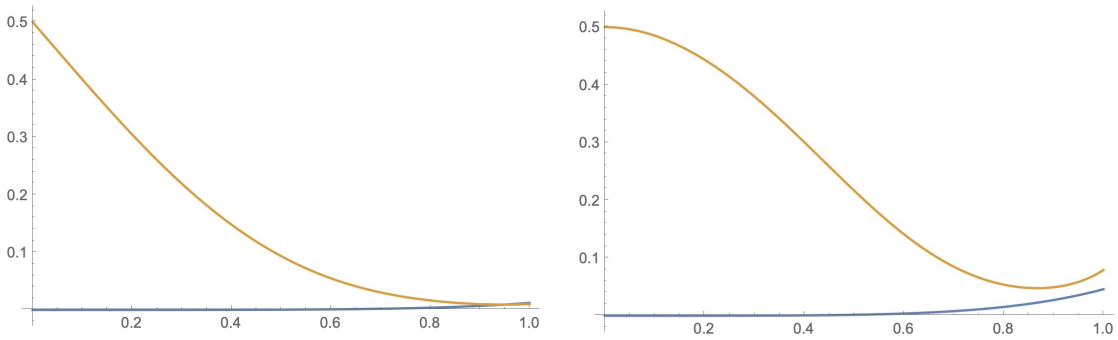


Fig. 5.10. Min-entropie d'Ève (bleu) et longueur du syndrome (jaune) après un filtrage, une concentration et un autre filtrage.

Nous avons haussé δ à 0.761099. Nous pouvons maintenant tolérer un taux d'erreurs maximum d'environ 19.03%. C'est une nette amélioration du 16.93% après concentration et nous dépassons la plus haute borne à notre connaissance (18.9% [**13**]). En regardant les deux graphiques à la figure 5.11, nous pouvons voir que l'ajout de la concentration entre les deux filtrages hausse le taux d'erreurs d'Ève. Remarquons surtout que le taux d'erreurs de Bob a aussi monté, mais moins rapidement. Les deux restent distancés jusqu'à la plus grosse valeur que peut prendre δ, soit 1. Cela veut dire qu'avec la communication interactive présentée plus haut, il serait possible de maintenir le taux d'erreurs chez Bob plus bas que celui d'Ève



(a) Deux filtrages sans concentration (b) Deux filtrages avec concentration au milieu

Fig. 5.11. Erreur de Bob (bleu) et erreur d'Ève (jaune).

même si celle-ci attaque l'entièreté des qubits à la communication quantique. Par Csiszár et Körner [**11**], il serait possible de générer une clé secrète sous ces conditions, c'est-à-dire avec un taux d'erreurs initial de 25%. Ceci est énorme puisque nous atteindrions la borne maximale pour un protocole BB84. Nous n'avons présentement par les moyens ni le temps de montrer exactement comment cela serait possible, mais nous proposons l'idée de répéter l'application alternée du filtrage et de la concentration. Gardons tout de même en tête que notre analyse repose sur l'hypothèse qu'Ève attaque seulement avec des i/r dans la base de Breidbart. Ainsi, nos résultats ne sont pas généralisés à toutes les attaques possibles.

Chapitre 6

Variantes intéressantes

6.1. Préparation

Nous présentons des variations à notre protocole. L'étape que nous voulons préserver est le non-dévoilement des bases. Les autres étapes sont sujettes à amélioration. Premièrement, nous proposons une variante pour la préparation de la chaine d'Alice. Cette dernière pourrait préparer ses bits dans plus de deux bases différentes. Par exemple, le protocole aux six états [**6**] [**2**] utilise trois bases au lieu de deux pour augmenter le taux d'erreurs toléré sur le canal.

$$\{|0\rangle, |1\rangle\} \qquad \{|+\rangle, |-\rangle\} \qquad \{|i\rangle, |-i\rangle\},$$

où $|i\rangle := \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$ et $|-i\rangle := \frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle$. Nous pourrions incorporer ceci dans notre travail. Alice préparerait chacun de ses bits dans l'une des trois bases ci-dessus. Les bases sont toutes équiprobables et Ève ne les connaitra jamais. Son incertitude sera alors plus grande que dans notre proposition originale. Par contre, l'analyse de sécurité contre des attaques i/r avec la base de Breidbart ne serait plus adéquate avec l'ajout de la troisième base. En effet, la base de Breidbart n'est pas optimale du point de vue d'Ève pour retirer le plus d'information possible d'un qubit préparé aléatoirement dans une base parmi ces trois. Il faudrait plutôt trouver une base équidistante des trois.

Lorsque Bob doit choisir les bases au hasard, avoir trois bases peut être un désavantage, puisque Bob est plus à risque de ne pas coïncider avec Alice dans son choix de mesure. Lorsque la séquence de bases est préalablement partagée par nos deux partis, ceci ne serait pas un problème.

Par contre, la tolérance d'erreurs maximale pour un protocole aux six états est de 26.4% [**13**]. Il faudrait donc pouvoir dépasser ce taux pour que ce soit pertinent.

Une autre modification intéressante serait de passer à une dimension plus grande que 2. Alice envoie actuellement des registres de dimension 2. Les trois bases énoncées plus haut sont des bases mutuellement non biaisées. Cela veut dire que si un bit est préparé dans une de ces trois bases, un adversaire mesurant dans une des deux autres bases aura un résultat

complètement aléatoire. C'est un des principes fondateurs de BB84 et du protocole à 6 états. Ève est sure d'obtenir le bon résultat que si elle prend la bonne base. Soit k qubits formant un registre de dimension $d := 2^k$. Lorsqu'on augmente la dimension, nous augmentons aussi la quantité de bases qui peuvent être mutuellement non biaisées. Pour un espace de Hilbert de dimension d, il existe $d+1$ bases mutuellement non biaisées. Cerf, Bourennane, Karlsson et Gisin ont publié un travail portant sur deux différents protocoles inspirés de BB84 [**10**]. Dans le premier, Alice choisit seulement deux bases mutuellement non biaisées dans l'espace de Hilbert de dimension $d > 2$. Elle prépare chaque k bits dans l'une de ces deux bases. Dans le deuxième, elle choisit entre une des $d + 1$ bases pour préparer le registre. Leurs résultats montrent que les deux méthodes tolèrent graduellement un taux d'erreurs plus gros à chaque incrémentation de d. Même si la deuxième variante en avantageuse, ils concluent que la première est préférable dans la pratique puisqu'elle ne force pas Alice et Bob à jeter environ $(d-1)/d$ bits. Si Bob possède déjà la séquence de bases, ces résultats deviennent encourageants. Cela laisse croire que nous pourrions significativement augmenter le taux d'erreurs toléré en envoyant des qubits dans plusieurs bases sans le désavantage de jeter une si grosse portion de la chaine (puisque Bob n'a pas à deviner les bases). Nous ne connaissons pas de travaux qui traitent du taux d'erreurs maximal toléré lorsque nous utilisons $d + 1$ bases (où $d > 2$) et un protocole interactif. Il est fort probable que ce soit une borne plus haute que 26.4%. Ce serait donc un plus grand défi à relever.

Un désavantage d'utiliser plus que deux bases est la quantité de bits nécessaires pour décrire une base. Lorsque Alice et Bob peuvent seulement choisir parmi deux bases, un bit d'information est suffisant pour définir la base choisie (0 pour $\{M_0, M_1\}$ et 1 pour $\{M_+, M_-\}$). Ainsi, une chaine θ de longueur n est suffisante pour préparer une chaine x de longueur n. Si nous avons plus de choix de bases, alors nous avons besoin de plus qu'un bit de θ pour préparer un bit de x. Nous réduisons alors la longueur de la chaine x envoyée à la communication quantique.

6.2. Filtrage

Une longueur de bloc autre que 2 a été considérée lors de l'étude de cette technique. Voici les différentes variations qui ont été analysées:

1. Blocs de longueur **3**: Alice et Bob comparent la parité du bloc. Si leur trio ont la même parité, ils conservent **un bit**.
2. Blocs de longueur **4**: Alice et Bob comparent la parité du bloc. Si leur quatuor ont la même parité, ils conservent **un bit**.
3. Blocs de longueur **3**: Alice et Bob comparent la parité du bloc. Si leur trio ont la même parité, ils conservent **deux bits**.

4. Bloc de longueur **3**: Alice et Bob comparent la parité du bloc. Si leur trio ont la même parité, ils conservent le **XOR de deux bits**.

6.2.1. Bloc 3 : 1 bit

Comme le filtrage présenté au chapitre 3, Alice et Bob permutent leur chaine. Cependant, ils remplacent les paires par des trios. Ils conservent tout de même un seul bit dont la position est préalablement choisie (le troisième du trio par exemple) après avoir comparé la parité de chaque trio. Cette technique augmente le taux d'erreurs maximum initial de 13.7% à 16.1% (voir annexe A.6). Il s'agit d'une bonne amélioration, mais ce n'est pas aussi avantageux que le filtrage original avec des blocs de longueur 2 (16.5%). De plus, la complexité d'analyse de sécurité pour un filtrage dépend de la longueur des blocs. Plus ils sont gros, plus il existe de groupes à calculer individuellement. Il n'y a donc aucun avantage à utiliser cette variation.

6.2.2. Bloc 4 : 1 bit

Cette variation est similaire à la précédente. Alice et Bob permutent leur chaine, les séparent en quatuors, annoncent les parités et conservent un seul bit parmi chaque bloc satisfaisant le filtrage de parité. Le taux d'erreurs maximum toléré en utilisant cette méthode est 16% (voir annexe A.7). Encore une fois, nous remarquons que la technique par paires est plus avantageuse. Nous voyons même qu'augmenter la longueur des blocs semble baisser graduellement l'efficacité du protocole. En effet, augmenter la longueur augmente aussi la probabilité qu'Alice et Bob conserve un bloc contenant des erreurs. Nous en concluons que les blocs de longueur 2 sont optimaux pour la technique de filtrage lorsque nous conservons un seul bit.

6.2.3. Bloc 3 : 2 bits

De toute évidence, augmenter la longueur des blocs tout en continuant de garder un seul bit semble un peu excessif et peu productif. Le filtrage coupe déjà la chaine d'Alice et Bob d'au moins la moitié dû à un seul bit gardé dans une paire. En réalité, ils en perdent encore plus puisqu'ils jettent les paires dont la parité diffère. Cela dépend par contre de la quantité d'erreurs introduite par Ève. Les deux variations présentées ci-dessus étaient donc déjà un pas en arrière en termes de productivité. Qu'arrive t'il si nous augmentons les blocs tout en augmentant la quantité de bits conservés? Nous avons testé cette proposition pour des blocs de longueur 3. Donc Alice et Bob permutent leur chaine, ils forment des trios, ils comparent leur parité et ils conservent deux bits par bloc ayant la même parité. C'est possible puisque l'annonce de parité ne dévoile qu'un seul bit d'informations. Il serait alors contre-productif de garder qu'un seul bit du trio. Surprenamment, le taux d'erreurs maximum toléré est le même que lorsque nous gardons qu'un seul bit dans un trio (voir annexe A.8). Malgré le

taux plus bas que le filtrage original, ce n'est pas pour autant inutile. Le taux d'erreurs est autant affecté que la variation 1, mais Alice et Bob peuvent conserver une plus grande partie de la chaine.

6.2.4. Bloc 3 : XOR 2 bits

Il est intéressant de se rappeler l'avantage de l'étape de concentration. Cette méthode est excellente pour augmenter l'incertitude chez Ève après le filtrage qui diminue son taux erreur. Comme la variation 3 nous dit qu'il est possible de conserver deux bits dans un trio lors d'un filtrage à bloc de longueur 3, il semble naturel d'essayer de conserver le XOR de ces deux bits à la place. En effet, cette variation procure un semblant de concentration dans l'étape de filtration. La figure 6.1 montre que l'erreur d'Ève n'augmente pas autant qu'au filtrage original. La méthode de concentration est légèrement meilleure pour augmenter le taux d'erreurs chez Ève, mais la variation 4 reste néanmoins efficace. De ce fait, ne serait-il pas plus pertinent d'utiliser cette variation au lieu d'une étape de filtration (version originale) suivie d'une étape de concentration? En fait, en ne faisant que la variation 4 du filtrage, nous remarquons que le taux d'erreurs maximum toléré ne dépasse pas 16.1% (voir annexe A.9). Ce résultat est faible lorsqu'on le compare au 16.9% pour un filtrage original suivi d'une concentration. Rappelons que le but premier de ce travail est d'augmenter le taux d'erreurs maximum toléré sur le canal par Alice et Bob. Concernant le côté pratique de la chose, cette variation ne rend pas l'analyse de sécurité moins complexe puisque la longueur des blocs est plus grande, résultant en plus de cas à analyser. Notre procédure originale reste donc la plus avantageuse.

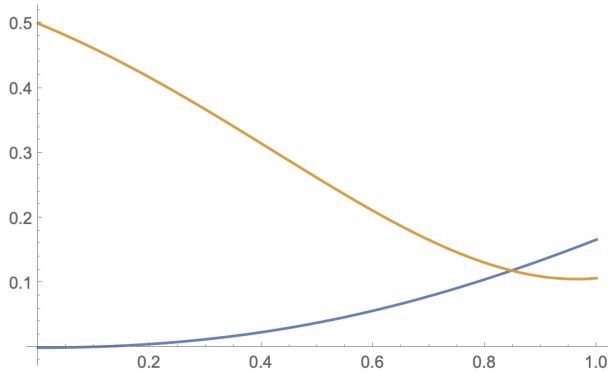


Fig. 6.1. Erreur de Bob (bleu) et erreur d'Ève (jaune) après filtrage variation 4.

6.3. Concentration

Maintenant, qu'en est-il de la concentration? Pourquoi avons-nous utilisés des blocs de longueur 2? Y a-t-il un avantage? Il se trouve que prendre des blocs de longueur 3 offre un

meilleur résultat que celui que nous avons obtenu. Faire une addition bit par bit de trios sans rien dévoiler à Ève augmente le delta maximum à 0.6687605, donnant ainsi un taux d'erreurs tolérable de 17.19% (16.93% pour longueur 2). Nous nous sommes limités aux paires pour simplifier l'analyse de sécurité. L'étape de concentration telle quelle n'est pas particulièrement plus difficile pour une longueur de 3, mais elle rend l'analyse du deuxième filtrage beaucoup trop fastidieuse. Il n'est donc pas irréaliste de penser qu'utiliser une concentration par trio entre les filtrages aurait augmenté encore plus notre taux d'erreurs maximum toléré.

6.4. Répétition: filtrage et concentration

Nous avons étudié l'impact d'un filtrage, de deux filtrages, d'un filtrage suivi d'une concentration et d'un filtrage suivi d'une concentration, suivi d'un second filtrage. Nous avons vu les avantages du filtrage ainsi que les bénéfices d'ajouter une concentration entre deux filtrages. Nous encourageons donc l'analyse de l'application successive de ces deux techniques en prenant soin de les alterner. Avec le résultat final de la section 5.6, nous espérons que des futurs travaux puissent repousser le taux maximal d'erreurs toléré par un protocole QKD avec des qubits BB84 à 25%.

Chapitre 7

Conclusion

Nous avons proposé une version du protocole BB84 dont les bases de mesure ne sont jamais dévoilées. En ajoutant un troisième parti, nous avons établi comment Alice et Bob peuvent commencer leur communication quantique avec un θ secret en commun. L'analyse de ce protocole est motivée par le désir de hausser le taux d'erreurs maximal toléré entre Alice et Bob sur Terre. En augmentant la tolérance aux erreurs, nous augmentons la distance maximale sur laquelle un QKD est possible.

Au chapitre 2, nous avons défini des concepts fondamentaux et des définitions importantes de la théorie quantique et de la cryptographie. Le chapitre 3 présente en détail le fameux protocole BB84 (avec correction non interactive) qui sert de base à notre travail. Nous avons décrit les étapes du protocole pour établir une clé finale sure. Les attaques de type i/r avec les mesures projectives $\{M_0, M_1\}$ et $\{M_+, M_-\}$ étaient considérées pour l'analyse de sécurité. Nous avons vu comment pouvoir calculer le plus gros taux d'erreurs tolérée (11%).

Au chapitre 4, nous avons présenté notre version modifiée de BB84. Les étapes définies sont similaires au protocole présenté au chapitre 3. La principale différence est le secret de la séquence de bases θ et la communication classique interactive. Avec l'aide d'un satellite qui se déplace entre Alice et Bob, ceux-ci peuvent s'entendre sur un θ avant l'envoi de qubits BB84 vers Bob. Puisqu'ils connaissent la séquence de bases, il n'est pas nécessaire de sacrifier la moitié de leur chaine comme dans le protocole original. En ajoutant au secret des bases la correction d'erreurs interactive, nous voulons pouvoir dépasser la meilleure tolérance connue (18.9% [**13**]).

Ensuite, nous avons présenté une analyse restreinte de quelques variations de notre modèle dans le chapitre 5. Nous avons choisi de limiter notre adversaire en lui imposant un type d'attaque. Nous supposons qu'Ève fait des attaques i/r dans la base de Breidbart. Nous avons expliqué pourquoi cette base est bénéfique pour Ève dans notre contexte. Toutefois, nous ne savons pas s'il s'agit de la meilleure mesure possible à long terme. Sous cette hypothèse, nous avons vu comment Ève peut astucieusement manipuler l'information qu'elle

obtient grâce à cette attaque pour minimiser son incertitude sur la chaine d'Alice. En premier lieu, nous avons vu que le secret des bases permet une tolérance d'erreurs allant jusqu'à 13.72%. Nous dépassons ainsi le taux de 11% pour un protocole BB84 avec correction d'erreurs non interactive. Ensuite, nous avons analysé deux techniques interactives classiques qui haussent le taux davantage. La méthode de filtrage augmente le taux à 16.49%. Par contre, le taux d'erreurs de notre adversaire devient trop petit lorsque δ augmente. Cela nous empêche de réappliquer la méthode une deuxième fois. Nous avons alors introduit la concentration qui monte le taux d'erreurs chez l'adversaire et qui hausse notre tolérance d'erreurs à 16.93%. Cette technique a pour but principal d'ajouter des erreurs chez Ève pour pouvoir réappliquer le filtrage et non de hausser le taux maximal toléré par Alice et Bob. Une fois cela fait, nous avons réappliqué la méthode de filtrage pour augmenter le taux d'erreurs à 19.03% sans trop diminuer le taux d'erreurs chez Ève. Ainsi, sous une hypothèse d'attaque, nous avons pu dépasser la plus haute tolérance connue pour un protocole BB84 avec correction interactive (18.9%).

Finalement, nous avons suggéré des variations à notre propre protocole au chapitre 6. Nous ne sommes pas obligés de préparer la chaine initiale d'Alice dans les bases rectilinéaire et diagonale. Nous pouvons choisir parmi 3 (ou plus) bases mutuellement non biaisées pour augmenter l'incertitude chez Ève. Dans un tel contexte, notre analyse limitée aux attaques de Breidbart n'est plus pertinente. De plus, cela impliquerait que nous aurions un taux maximal plus gros à dépasser (26.4% ou plus). Nous avons aussi présenté des variations du filtrage. Nous remarquons qu'augmenter la longueur des blocs et modifier notre choix de bits n'est pas plus avantageux que le filtrage original présenté au chapitre 4. Cependant, la concentration peut être améliorée en augmentant la longueur des blocs.

En résumé, nous avons réussi à dépasser la plus haute borne connue pour des protocoles QKD avec qubits BB84 dans l'hypothèse où Ève doit attaquer dans la base de Breidbart. Nous avons aussi donné quelques pistes pour améliorer notre maximum atteint (19.03%) et peut-être même atteindre la borne maximale (25%).

De toute évidence, il n'est pas garanti que l'attaque de Breidbart est la meilleure attaque contre notre protocole. Nous savons seulement que cette mesure offre la meilleure probabilité de succès pour un bit BB84 si l'adversaire n'apprend jamais la base exacte. Nous sommes conscients que des attaques collectives ou cohérentes pourraient lui fournir plus d'information. Nous espérons que cette hypothèse puisse être levée dans des travaux futurs. Lorsque Ève mesure les qubits dans la base de Breidbart, elle ne peut pas gagner d'information sur la séquence de base, c'est-à-dire son incertitude sur θ est et demeure maximale. C'est pourquoi nous pouvons réutiliser les bases pour des QKD futures. Si elle attaque différemment, nous espérons qu'elle gagne de l'information sur θ au détriment de la chaine d'Alice. En d'autres mots, nous espérons que la min-entropie de θ diminue seulement si la min-entropie de la chaine d'Alice augmente en conséquence. Ainsi, en mettant θ et la clé brute $\hat{x}$ dans la

fonction de hachage, à l'étape d'amplification de secret, nous pourrions générer la clé finale ainsi qu'une nouvelle séquence de bases. Puisque l'incertitude perdue sur θ est transférée à celle de $\hat{x}$, nous pourrions générer une chaine assez longue pour n bases et au moins 1 bit de clé. Avec de légères modifications, nous croyons que le protocole proposé dans ce mémoire serait encore pertinent avec une analyse complète des attaques d'Ève. Nous laissons la porte ouverte pour les personnes intéressées.

Références bibliographiques

[1] E. Arikan : Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, juill. 2009.

[2] H. Bechmann-Pasquinucci et N. Gisin : Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography. *Physical Review A*, 59(6):4238–4248, juin 1999.

[3] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail et J. Smolin : Experimental quantum cryptography. *Journal of Cryptology*, 5:3–28, janv. 1992.

[4] C. H. Bennett et G. Brassard : Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, déc. 2014.

[5] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres et W. K. Wootters : Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70:1895–1899, mars 1993.

[6] D. Bruss : Optimal eavesdropping in quantum cryptography with six states. *Physical Review Letters*, 81(14):3018–3021, oct. 1998.

[7] D. Bruss, A. Ekert et C. Macchiavello : Optimal universal quantum cloning and state estimation. *Physical Review Letters*, 81:2598–2601, sept. 1998.

[8] V. Bužek et M. Hillery : Quantum copying: Beyond the no-cloning theorem. *Physical Review A*, 54:1844–1852, sept. 1996.

[9] J. L. Carter et M. N. Wegman : Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, avr. 1979.

[10] N.J. Cerf, M. Bourennane, A. Karlsson et N. Gisin : Security of quantum key distribution using d-level systems. *Physical Review Letters*, 88(12), mars 2002.

[11] I. Csiszár et J. Körner : Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 24(3):339–348, 1978.

[12] E. N. Gilbert : A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.

[13] D. Gottesman et H.-K. Lo : Proof of security of quantum key distribution with two-way classical communications. *IEEE Transactions on Information Theory*, 49(2):457, 2001.

[14] A. Kerckhoffs : La cryptographie militaire. *Journal des sciences militaires*, 4:5–38, 161–191, janv. 1883.

[15] S.-K. Liao, W.-Q. Cai, J. Handsteiner, B. Liu, J. Yin, L. Zhang, D. Rauch, M. Fink, J.-G. Ren, W.-Y. Liu, Y. Li, Q. Shen, Y. Cao, F.-Z. Li, J.-F. Wang, Y.-M. Huang, L. Deng, T. Xi, L. Ma, T. Hu, L. Li, N.-L. Liu, F. Koidl, P. Wang, Y.-A. Chen, X.-B. Wang, M. Steindorfer, G. Kirchner, C.-Y. Lu, R. Shu, R. Ursin, T. Scheidl, C.-Z. Peng, J.-Y. Wang, A. Zeilinger et

J.-W. Pan : Satellite-relayed intercontinental quantum network. *Physical Review Letters*, 120:030501, janv. 2018.

[16] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li, F.-Z. Li, X.-W. Chen, L.-H. Sun, J.-J. Jia, J.-C. Wu, X.-J. Jiang, J.-F. Wang, Y.-M. Huang, Q. Wang, Y-L. Zhou, L. Deng, T. Xi, L. Ma, T. Hu, Q. Zhang, Y.-A Chen, N.-L. Liu, X.-B. Wang, Z.-C. Zhu, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y Wang et J.-W. Pan : Satellite-to-ground quantum key distribution. *Nature*, 549(7670):43–47, août 2017.

[17] U. M. Maurer : Protocols for secret key agreement by public discussion based on common information. *In* Ernest F. Brickell, éditeur : *Advances in Cryptology — CRYPTO' 92*, pages 461–470, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[18] J. L. Park : The concept of transition in quantum mechanics. *Foundations of Physics*, 1:23–33, mars 1970.

[19] R. Renner : Security of quantum key distribution. *International Journal of Quantum Information*, 6(1):1–127, 2008.

[20] R. Renner, N. Gisin et B. Kraus : Information-theoretic security proof for quantum-key-distribution protocols. *Physical Review A*, 72(1), juill. 2005.

[21] R. Renner et R. König : Universally composable privacy amplification against quantum adversaries. *TCC 2005*, 3378, 2004.

[22] K. Saeedi, S. Simmons, J. Z. Salvail, P. Dluhy, H. Riemann, N. V. Abrosimov, P. Becker, H.-J. Pohl, J. J. L. Morton et M. L. W. Thewalt : La cryptographie militaire. *Journal des sciences militaires*, 342:830–833, nov. 2013.

[23] L. Salvail : IFT6195: Sujets en informatique quantique.

[24] C. E. Shannon : A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423, 1948.

[25] C. E. Shannon : Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4): 656–715, 1949.

[26] R. Varshamov : Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741, janv. 1957.

[27] W. K. Wootters et W. H. Zurek : A single quantum cannot be cloned. *Nature*, 299:802–803, oct. 1982.

[28] A. D. Wyner : The wire-tap channel. *The Bell System Technical Journal*, 54(8):1355–1387, 1975.

[29] J. Yin, Y. Cao, Y.H. li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai, G.-B. Li, Q.-M. Lu, Y.-H. Gong, Y. Xu, S.-L. Li, F.-Z. Li, Y.-Y. Yin, Z.-Q. Jiang, M. Li, J.-J. Jia, G. Ren, D. He, Y.-L. Zhou, X.-X. Zhang, N. Wang, X. Chang, Z.-C. Zhu, N.-L. Liu, Y.A. Chen, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang et J.-W. Pan : Satellite-based entanglement distribution over 1200 kilometers. *Science*, 356, juin. 2017.

[30] J. Yin, Y.-H. Li, S.-K. Liao, M. Yang, Y. Cao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, S.-L. Li, R. Shu, Y.-M. Huang, L. Deng, L. Li, Q. Zhang, N.-Le Liu, Y.-A. Chen, C.-Y. Lu, X.-B. Wang, J.-Y. Wang F. Xu, C.-Z. Pengand A. K. Ekert et J.-W. Pan : Entanglement-based secure quantum cryptography over 1,120 kilometres. *Nature*, 582:501–505, juin 2020.

Annexe A

Calculs sur Mathematica

A.1. Préparation

```
(* Constantes *)
s := (Sin[Pi/8])^2
c := (Cos[Pi/8])^2

(* min-entropie d'Eve et longueur du syndrome sans n *)
H[d_] := -Log2[((1/2)^(1 - d))*(c^d)]
l[d_] := -(d/4)*Log2[d/4] - (1 - (d/4))*Log2[1 - (d/4)]
Plot[{H[d], l[d]}, {d, 0, 1}]
FindRoot[H[d] - l[d], {d, 0.1}]
```

A.2. Filtrage

```
(* Probabilité qu'Alice et Bob aient la même parité *)
F1[d_] := (d^2 - 4 d + 8)/8

(* Erreur de Bob *)
e1[d_] := (d/4)^2/((d/4)^2 + (1 - (d/4))^2)

(* Filtrage 1: probabilité d'être dans W X, Y et Z *)
pw[d_] := (1 - d)^2/F1[d]
px[d_] := d^2*(4*c^2*s^2)/F1[d]
py[d_] := (d*(1 - d)*(2 c^2 + 2 s^2))/F1[d]
pz[d_] := (d^2*(c^4 + s^4 + 2*c^2*s^2))/F1[d]
Plot[{pw[d], px[d], py[d], pz[d]}, {d, 0, 1}]
(* probabilités de succès de chaque ensemble *)
```

```
sw := 0.5
sx := 0.5
sy := c^2/(c^2 + s^2)
sz := (c^4 + c^2*s^2)/(c^4 + s^4 + 2 c^2*s^2)

s1[d_] := (0.5^pw[d])*(0.5^px[d])*(sy^py[d])*(sz^pz[d])
H1[d_] := -Log2[s1[d]]
l1[d_] := -e1[d]*Log2[e1[d]] - (1 - e1[d])*Log2[1 - e1[d]]
Plot[{H1[d], l1[d]}, {d, 0, 1}]
FindRoot[H1[d] - l1[d], {d, 0.1}]
Plot[{e1[d], 1 - s1[d]}, {d, 0, 1}]
```

A.3. Second filtrage

```
(* Après filtrage 1: probabilité de chaque resultats d'Eve et Bob *)
pxbbb[d_] := ((1 - d)^2/2 +
    d^2 (c^2*s^2))/(d^2 (4 c^2*s^2) + (1 - d)^2)
pxbbnb[d_] := (d^2 (c^2*s^2))/(d^2 (4 c^2*s^2) + (1 - d)^2)
pxbnbb[d_] := (d^2 (c^2*s^2))/(d^2 (4 c^2*s^2) + (1 - d)^2)
pxbnbnb[d_] := ((1 - d)^2/2 +
    d^2 (c^2*s^2))/(d^2 (4 c^2*s^2) + (1 - d)^2)
pybbb[d_] := c^2/(c^2 + s^2)
pybnbnb[d_] := s^2/(c^2 + s^2)
pzbbb[d_] := c^4/(c^4 + s^4 + 2 c^2*s^2)
pzbbnb[d_] := c^2*s^2/(c^4 + s^4 + 2 c^2*s^2)
pzbnbb[d_] := c^2*s^2/(c^4 + s^4 + 2 c^2*s^2)
pzbnbnb[d_] := s^4/(c^4 + s^4 + 2 c^2*s^2)

(* Probabilite qu'Alice et Bob aient la meme parite *)
f2[d_] := e1[d]^2 + (1 - e1[d])^2

(* Erreur de Bob *)
E2[d_] := e1[d]^2/f2[d]

(* Après filtrage 2: probabilité d'etre dans A, B, C, D, E, F, G, H et I *)
pXa[d_] := (8 (1 - d)^2)/(d^2 - 4 d + 8)
pXb[d_] := (32 d^2 c^2 s^2)/(d^2 - 4 d + 8)
PA[d_] := (pXa[d]^2 + (pXa[d]*pXb[d]) + (pXb[d]^2/2))/f2[d]
```

```
PAa[d_] := pXa[d]^2/f2[d]
PAab[d_] := (pXa[d]*pXb[d])/f2[d]
PAb[d_] := (pXb[d]^2/2)/f2[d]
PB[d_] := (2*py[d]^2*pybbb[d]*pybnbnb[d])/f2[d]
PC[d_] := (py[d]^2 (pybbb[d]^2 + pybnbnb[d]^2))/f2[d]
PD[d_] := (2*pz[d]^2*(pzbbb[d]*pzbnbnb[d] + pzbbnb[d]*pzbnbb[d]))/f2[d]
PE[d_] := (pz[d]^2*(pzbbb[d]^2 + pzbnbb[d]^2 + pzbbnb[d]^2 +
      pzbnbnb[d]^2))/f2[d]
PF[d_] := ((2*pXa[d]*py[d]*(pybbb[d] + pybnbnb[d])) + (pXb[d]*
      py[d]*(pybbb[d] + pybnbnb[d])))/f2[d]
PFa[d_] := (2*pXa[d]*py[d]*(pybbb[d] + pybnbnb[d]))/f2[d]
PFb[d_] := (pXb[d]*py[d]*(pybbb[d] + pybnbnb[d]))/f2[d]
PGa[d_] := (2*pXa[d]*pz[d]*(pzbbb[d] + pzbnbnb[d]))/f2[d]
PGb[d_] := (pXb[d]*
    pz[d]*(pzbbb[d] + pzbbnb[d] + pzbnbb[d] + pzbnbnb[d]))/f2[d]
PH[d_] := (2*py[d]*pz[d]*(pybbb[d]*pzbnbnb[d] + pybnbnb[d]*pzbbb[d]))/
  f2[d]
PI[d_] := (2*py[d]*pz[d]*(pybbb[d]*pzbbb[d] + pybnbnb[d]*pzbnbnb[d]))/
  f2[d]

S2[d_] := ((1/2)^PA[d])*((1/2)^PB[d])*((c^4/(c^4 + s^4))^
    PC[d])*((1/2)^PD[d])*(((c^8 + c^4 s^4)/(c^8 + 2 c^4 s^4 + s^8))^
    PE[d])*((c^2/(c^2 + s^2))^
    PF[d])*(((2*pXa[d]*pzbbb[d] + pXb[d]*pzbbb[d] +pXb[d]*pzbbnb[d])/(2*pXa[d]*(pzbbb[d]
        pXb[d]*(pzbbb[d] + pzbbnb[d] pzbnbb[d] + pzbnbnb[d])))^
    PG[d])*((c^2/(c^2 + s^2))^PH[d])*((c^6/(c^6 + s^6))^PI[d])

h2[d_] := -Log2[S2[d]]
L2[d_] := -E2[d]*Log2[E2[d]] - (1 - E2[d])*Log2[1 - E2[d]]
Plot[{PA[d] + PB[d] + PC[d] + PD[d] + PE[d] + PF[d] + PGa[d] +
   PGb[d] + PH[d] + PI[d]}, {d, 0, 1}]
Plot[{h2[d], L2[d]}, {d, 0, 1}]
FindRoot[h2[d] - L2[d], {d, 0.1}]
Plot[{E2[d], 1 - S2[d]}, {d, 0, 1}]
```

A.4. Concentration

```
(* Concentration bloc 2 post filtrage 1 *)
```

```
pa[d_] := pw[d]^2
pb[d_] := px[d]^2
pc[d_] := py[d]^2
pd[d_] := pz[d]^2
pe[d_] := 2*pw[d]*px[d]
pf[d_] := 2*pw[d]*py[d]
pg[d_] := 2*pw[d]*pz[d]
ph[d_] := 2*px[d]*py[d]
pi[d_] := 2*px[d]*pz[d]
pj[d_] := 2*py[d]*pz[d]
Plot[pa[d]+pb[d]+pc[d]+pd[d]+pe[d]+pf[d]+pg[d]+ph[d]+pi[d]+pj[d], {d, 0, 1}]

(* Probabilité de succès de chaque ensemble *)
sc := sy^2 + (1 - sy)^2
sd := sz^2 + (1 - sz)^2
sj := sy*sz + (1 - sy)*(1 - sz)

(* Erreur de Bob *)
e2[d_] := 2*(1 - e1[d])*e1[d]

s2[d_] := ((1/2)^(pa[d] + pb[d] + pe[d] + pf[d] + pg[d] + ph[d] +
      pi[d]))*(sc^(pc[d] + pd[d]))*(sj^pj[d])

H2[d_] := -Log2[s2[d]]
l2[d_] := -e2[d]*Log2[e2[d]] - (1 - e2[d])*Log2[1 - e2[d]]
FindRoot[H2[d] - l2[d], {d, 0.5}]
Plot[{H2[d], l2[d]}, {d, 0, 1}]
Plot[{e2[d], 1 - s2[d]}, {d, 0, 1}]
```

A.5. Second filtrage après concentration

```
b2P3bbb := (c^4 + s^4)/(c^2 + s^2)^2
b2P3bnbb := 0
b2P3bbnb := 0
b2P3bnbnb := 2 (c^2 s^2)/(c^2 + s^2)^2
b2P4bbb := (c^8 + 2 c^4 s^4 + s^8)/(c^4 + 2 c^2 s^2 + s^4)^2
b2P4bnbb := (2 c^6 s^2 + 2 c^2 s^6)/(c^4 + 2 c^2 s^2 + s^4)^2
b2P4bbnb := (2 c^6 s^2 + 2 c^2 s^6)/(c^4 + 2 c^2 s^2 + s^4)^2
```

```
b2P4bnbnb := (4 c^4 s^4)/(c^4 + 2 c^2 s^2 + s^4)^2
b2P10bbb := (c^6 + s^6)/((c^2 + s^2) (c^4 + 2 c^2 s^2 + s^4))
b2P10bnbb := (c^4 s^2 +
    s^4 c^2)/((c^2 + s^2) (c^4 + 2 c^2 s^2 + s^4))
b2P10bbnb := (c^4 s^2 +
    s^4 c^2)/((c^2 + s^2) (c^4 + 2 c^2 s^2 + s^4))
b2P10bnbnb := (c^2 s^4 + s^2 c^4)/((c^2 + s^2) (c^4 + 2 c^2 s^2 + s^4))
p0 := (c^4 + s^4)/(c^4 + 2 s^2 c^2 + s^4)

F2[d_] := e2[d]^2 + (1 - e2[d])^2

P11[d_] := (b2p1[d]^2)/F2[d]
P22[d_] := (b2p2[d]^2*(1/2))/F2[d]
P33[d_] := ((b2p3[d]^2)*(b2P3bbb^2 + b2P3bnbnb^2))/F2[d]
P33n[d_] := ((b2p3[d]^2)*(b2P3bbb*b2P3bnbnb + b2P3bnbnb*b2P3bbb))/F2[d]
P44[d_] := ((b2p4[d]^2)*(b2P4bbb^2 + b2P4bbnb^2 + b2P4bnbnb^2 +
      b2P4bnbb^2))/F2[d]
P44n[d_] := ((b2p4[d]^2)*(b2P4bbb*b2P4bnbnb + b2P4bbnb*b2P4bnbb +
      b2P4bnbnb*b2P4bbb + b2P4bnbb*b2P4bbnb))/F2[d]
P55[d_] := (b2p5[d]^2*(1/2))/F2[d]
P66[d_] := (b2p6[d]^2)/F2[d]
P77[d_] := (b2p7[d]^2*(p0^2 + (1 - p0)^2))/F2[d]
P88[d_] := (b2p8[d]^2*(1/2))/F2[d]
P99[d_] := (b2p9[d]^2*(1/2))/F2[d]
P1010[d_] := ((b2p10[d]^2)*(b2P10bbb^2 + b2P10bbnb^2 + b2P10bnbnb^2 +
      b2P10bnbb^2))/F2[d]
P1010n[d_] := ((b2p10[d]^2)*(b2P10bbb*b2P10bnbnb +
      b2P10bbnb*b2P10bnbb + b2P10bnbnb*b2P10bbb +
      b2P10bnbb*b2P10bbnb))/F2[d]
P12[d_] := (b2p1[d]*b2p2[d])/F2[d]
P13[d_] := (2*b2p1[d]*b2p3[d]*(b2P3bbb + b2P3bnbnb))/F2[d]
P14[d_] := (2*b2p1[d]*b2p4[d]*(b2P4bbb + b2P4bnbnb))/F2[d]
P15[d_] := (b2p1[d]*b2p5[d])/F2[d]
P16[d_] := (2*b2p1[d]*b2p6[d])/F2[d]
P17[d_] := (2*b2p1[d]*b2p7[d]*p0)/F2[d]
P18[d_] := (b2p1[d]*b2p8[d])/F2[d]
P19[d_] := (b2p1[d]*b2p9[d])/F2[d]
P110[d_] := (2*b2p1[d]*b2p10[d]*(b2P10bbb + b2P10bnbnb))/F2[d]
```

```
P23[d_] := (2*b2p2[d]*
    b2p3[d]*(0.5*b2P3bbb + 0.5*b2P3bbnb + 0.5*b2P3bnbnb +
      0.5*b2P3bnbb))/F2[d]
P24[d_] := (2*b2p2[d]*
    b2p4[d]*(0.5*b2P4bbb + 0.5*b2P4bbnb + 0.5*b2P4bnbnb +
      0.5*b2P4bnbb))/F2[d]
P25[d_] := (b2p2[d]*b2p5[d])/F2[d]
P26[d_] := (b2p2[d]*b2p6[d])/F2[d]
P27[d_] := (2*b2p2[d]*b2p7[d]*(0.5*p0 + 0.5*(1 - p0)))/F2[d]
P28[d_] := (b2p2[d]*b2p8[d])/F2[d]
P29[d_] := (b2p2[d]*b2p9[d])/F2[d]
P210[d_] := (2*b2p2[d]*
    b2p10[d]*(0.5*b2P10bbb + 0.5*b2P10bbnb + 0.5*b2P10bnbnb +
      0.5*b2P10bnbb))/F2[d]
P34[d_] :=
 2*(b2p3[d]*
     b2p4[d]*(b2P3bbb*b2P4bbb + b2P3bbnb*b2P4bbnb +
       b2P3bbb*b2P4bnbnb + b2P3bbnb*b2P4bnbb + b2P3bnbnb*b2P4bbb +
       b2P3bnbb*b2P4bbnb + b2P3bnbnb*b2P4bnbnb + b2P3bnbb*b2P4bnbb))/
   F2[d]
P35[d_] := (2*b2p3[d]*
    b2p5[d]*(0.5*b2P3bbb + 0.5*b2P3bbnb + 0.5*b2P3bnbnb +
      0.5*b2P3bnbb))/F2[d]
P36[d_] := (2*b2p3[d]*b2p6[d]*(b2P3bbb + b2P3bnbnb))/F2[d]
P37[d_] := (2*b2p3[d]*
    b2p7[d]*(p0*b2P3bbb + (1 - p0)*b2P3bbnb +
      p0*b2P3bnbnb + (1 - p0)*b2P3bnbb))/F2[d]
P38[d_] := (2*b2p3[d]*
    b2p8[d]*(0.5*b2P3bbb + 0.5*b2P3bbnb + 0.5*b2P3bnbnb +
      0.5*b2P3bnbb))/F2[d]
P39[d_] := (2*b2p3[d]*
    b2p9[d]*(0.5*b2P3bbb + 0.5*b2P3bbnb + 0.5*b2P3bnbnb +
      0.5*b2P3bnbb))/F2[d]
P310[d_] :=
 2*(b2p3[d]*
     b2p10[d]*(b2P3bbb*b2P10bbb + b2P3bbnb*b2P10bbnb +
       b2P3bbb*b2P10bnbnb + b2P3bbnb*b2P10bnbb + b2P3bnbnb*b2P10bbb +
       b2P3bnbb*b2P10bbnb + b2P3bnbnb*b2P10bnbnb +
```

```
       b2P3bnbb*b2P10bnbb))/F2[d]
P45[d_] := (2*b2p4[d]*
    b2p5[d]*(0.5*b2P4bbb + 0.5*b2P4bbnb + 0.5*b2P4bnbnb +
      0.5*b2P4bnbb))/F2[d]
P46[d_] := (2*b2p4[d]*b2p6[d]*(b2P4bbb + b2P4bnbnb))/F2[d]
P47[d_] := (2*b2p4[d]*
    b2p7[d]*(p0*b2P4bbb + (1 - p0)*b2P4bbnb +
      p0*b2P4bnbnb + (1 - p0)*b2P4bnbb))/F2[d]
P48[d_] := (2*b2p4[d]*
    b2p8[d]*(0.5*b2P4bbb + 0.5*b2P4bbnb + 0.5*b2P4bnbnb +
      0.5*b2P4bnbb))/F2[d]
P49[d_] := (2*b2p4[d]*
    b2p9[d]*(0.5*b2P4bbb + 0.5*b2P4bbnb + 0.5*b2P4bnbnb +
      0.5*b2P4bnbb))/F2[d]
P410[d_] :=
 2*(b2p4[d]*
     b2p10[d]*(b2P4bbb*b2P10bbb + b2P4bbnb*b2P10bbnb +
       b2P4bbb*b2P10bnbnb + b2P4bbnb*b2P10bnbb + b2P4bnbnb*b2P10bbb +
       b2P4bnbb*b2P10bbnb + b2P4bnbnb*b2P10bnbnb +
       b2P4bnbb*b2P10bnbb))/F2[d]
P56[d_] := (b2p5[d]*b2p6[d])/F2[d]
P57[d_] := (2*b2p5[d]*b2p7[d] (0.5*p0 + 0.5*(1 - p0)))/F2[d]
P58[d_] := (b2p5[d]*b2p8[d])/F2[d]
P59[d_] := (b2p5[d]*b2p9[d])/F2[d]
P510[d_] := (2*b2p5[d]*
    b2p10[d]*(0.5*b2P10bbb + 0.5*b2P10bbnb + 0.5*b2P10bnbnb +
      0.5*b2P10bnbb))/F2[d]
P67[d_] := (2*b2p6[d]*b2p7[d]*p0)/F2[d]
P68[d_] := (b2p6[d]*b2p8[d])/F2[d]
P69[d_] := (b2p6[d]*b2p9[d])/F2[d]
P610[d_] := (2*b2p6[d]*b2p10[d]*(b2P10bbb + b2P10bnbnb))/F2[d]
P78[d_] := (2*b2p7[d]*b2p8[d]*(0.5*p0 + 0.5*(1 - p0)))/F2[d]
P79[d_] := (2*b2p7[d]*b2p9[d]*(0.5*p0 + 0.5*(1 - p0)))/F2[d]
P710[d_] := (2*b2p7[d]*
    b2p10[d]*(p0*b2P10bbb + (1 - p0)*b2P10bbnb +
      p0*b2P10bnbnb + (1 - p0)*b2P10bnbb))/F2[d]
P89[d_] := (b2p8[d]*b2p9[d])/F2[d]
P810[d_] := (2*b2p8[d]*
```

```
    b2p10[d]*(0.5*b2P10bbb + 0.5*b2P10bbnb + 0.5*b2P10bnbnb +
      0.5*b2P10bnbb))/F2[d]
P910[d_] := (2*b2p9[d]*
    b2p10[d]*(0.5*b2P10bbb + 0.5*b2P10bbnb + 0.5*b2P10bnbnb +
      0.5*b2P10bnbb))/F2[d]

succ33 := (b2P3bbb^2)/(b2P3bbb^2 + b2P3bnbnb^2)
succ44 := (b2P4bbb^2 + b2P4bbnb^2)/(b2P4bbb^2 + b2P4bbnb^2 +
    b2P4bnbnb^2 + b2P4bnbb^2)
succ1010 := (b2P10bbb^2 + b2P10bbnb^2)/(b2P10bbb^2 + b2P10bbnb^2 +
    b2P10bnbnb^2 + b2P10bnbb^2)
succ13 := b2P3bbb/(b2P3bbb + b2P3bnbnb)
succ14 := b2P4bbb/(b2P4bbb + b2P4bnbnb)
succ110 := b2P10bbb/(b2P10bbb + b2P10bnbnb)
succ23 := (0.5*b2P3bbb + 0.5*b2P3bbnb)/(0.5*b2P3bbb + 0.5*b2P3bbnb +
    0.5*b2P3bnbnb + 0.5*b2P3bnbb)
succ24 := (0.5*b2P4bbb + 0.5*b2P4bbnb)/(0.5*b2P4bbb + 0.5*b2P4bbnb +
    0.5*b2P4bnbnb + 0.5*b2P4bnbb)
succ210 := (0.5*b2P10bbb + 0.5*b2P10bbnb)/(0.5*b2P10bbb +
    0.5*b2P10bbnb + 0.5*b2P10bnbnb + 0.5*b2P10bnbb)
succ34 := (b2P4bbb*b2P3bbb + b2P4bbnb*b2P3bbnb + b2P4bbb*b2P3bnbnb +
    b2P4bbnb*b2P3bnbb)/(b2P3bbb*b2P4bbb + b2P3bbnb*b2P4bbnb +
    b2P3bbb*b2P4bnbnb + b2P3bbnb*b2P4bnbb + b2P3bnbnb*b2P4bbb +
    b2P3bnbb*b2P4bbnb + b2P3bnbnb*b2P4bnbnb + b2P3bnbb*b2P4bnbb)
succ35 := (0.5*b2P3bbb + 0.5*b2P3bbnb)/(0.5*b2P3bbb + 0.5*b2P3bbnb +
    0.5*b2P3bnbnb + 0.5*b2P3bnbb)
succ36 := b2P3bbb/(b2P3bbb + b2P3bnbnb)
succ37 := (p0*b2P3bbb + (1 - p0)*b2P3bbnb)/(p0*b2P3bbb + (1 - p0)*
     b2P3bbnb + p0*b2P3bnbnb + (1 - p0)*b2P3bnbb)
succ38 := (0.5*b2P3bbb + 0.5*b2P3bbnb)/(0.5*b2P3bbb + 0.5*b2P3bbnb +
    0.5*b2P3bnbnb + 0.5*b2P3bnbb)
succ39 := (0.5*b2P3bbb + 0.5*b2P3bbnb)/(0.5*b2P3bbb + 0.5*b2P3bbnb +
    0.5*b2P3bnbnb + 0.5*b2P3bnbb)
succ310 := (b2P10bbb*b2P3bbb + b2P10bbnb*b2P3bbnb +
    b2P10bbb*b2P3bnbnb + b2P10bbnb*b2P3bnbb)/(b2P3bbb*b2P10bbb +
    b2P3bbnb*b2P10bbnb + b2P3bbb*b2P10bnbnb + b2P3bbnb*b2P10bnbb +
    b2P3bnbnb*b2P10bbb + b2P3bnbb*b2P10bbnb + b2P3bnbnb*b2P10bnbnb +
    b2P3bnbb*b2P10bnbb)
```

```
succ45 := (0.5*b2P4bbb + 0.5*b2P4bbnb)/(0.5*b2P4bbb + 0.5*b2P4bbnb +
    0.5*b2P4bnbnb + 0.5*b2P4bnbb)
succ46 := b2P4bbb/(b2P4bbb + b2P4bnbnb)
succ47 := (p0*b2P4bbb + (1 - p0)*b2P4bbnb)/(p0*b2P4bbb + (1 - p0)*
     b2P4bbnb + p0*b2P4bnbnb + (1 - p0)*b2P4bnbb)
succ48 := (0.5*b2P4bbb + 0.5*b2P4bbnb)/(0.5*b2P4bbb + 0.5*b2P4bbnb +
    0.5*b2P4bnbnb + 0.5*b2P4bnbb)
succ49 := (0.5*b2P4bbb + 0.5*b2P4bbnb)/(0.5*b2P4bbb + 0.5*b2P4bbnb +
    0.5*b2P4bnbnb + 0.5*b2P4bnbb)
succ410 := (b2P4bbb*b2P10bbb + b2P4bbnb*b2P10bbnb +
    b2P4bbb*b2P10bnbnb + b2P4bbnb*b2P10bnbb)/(b2P4bbb*b2P10bbb +
    b2P4bbnb*b2P10bbnb + b2P4bbb*b2P10bnbnb + b2P4bbnb*b2P10bnbb +
    b2P4bnbnb*b2P10bbb + b2P4bnbb*b2P10bbnb + b2P4bnbnb*b2P10bnbnb +
    b2P4bnbb*b2P10bnbb)
succ510 := (0.5*b2P10bbb + 0.5*b2P10bbnb)/(0.5*b2P10bbb +
    0.5*b2P10bbnb + 0.5*b2P10bnbnb + 0.5*b2P10bnbb)
succ610 := b2P10bbb/(b2P10bbb + b2P10bnbnb)
succ710 := (p0*b2P10bbb + (1 - p0)*b2P10bbnb)/(p0*b2P10bbb + (1 - p0)*
     b2P10bbnb + p0*b2P10bnbnb + (1 - p0)*b2P10bnbb)
succ810 := (0.5*b2P10bbb + 0.5*b2P10bbnb)/(0.5*b2P10bbb +
    0.5*b2P10bbnb + 0.5*b2P10bnbnb + 0.5*b2P10bnbb)
succ910 := (0.5*b2P10bbb + 0.5*b2P10bbnb)/(0.5*b2P10bbb +
    0.5*b2P10bbnb + 0.5*b2P10bnbnb + 0.5*b2P10bnbb)

(* Erreur de Bob *)
e3[d_] := e2[d]^2/F2[d]
S3[d_] := ((1/2)^(P11[d] + P22[d] + P33n[d] + P44n[d] + P55[d] +
      P66[d] + P77[d] + P88[d] + P99[d] + P1010n[d] + P12[d] +
      P15[d] + P16[d] + P17[d] + P18[d] + P19[d] + P25[d] + P26[d] +
      P27[d] + P28[d] + P29[d] + P56[d] + P57[d] + P58[d] + P59[d] +
      P67[d] + P68[d] + P69[d] + P78[d] + P79[d] + P89[d]))*(succ33^
    P33[d])*(succ44^P44[d])*(succ1010^P1010[d])*(succ13^
    P44[d])*(succ14^P14[d])*(succ110^P110[d])*(succ23^P23[d])*(succ24^
    P24[d])*(succ210^P210[d])*(succ34^P34[d])*(succ35^P35[d])*(succ36^
    P36[d])*(succ37^P37[d])*(succ38^P38[d])*(succ39^P39[d])*(succ310^
    P310[d])*(succ45^P45[d])*(succ46^P46[d])*(succ47^P47[d])*(succ48^
    P48[d])*(succ49^P49[d])*(succ410^P410[d])*(succ510^
    P510[d])*(succ610^P610[d])*(succ710^P710[d])*(succ810^
```

```
    P810[d])*(succ910^P910[d])
FindRoot[-Log2[
    S3[d]] - (-e3[d]*Log2[e3[d]] - (1 - e3[d]) (Log2[1 - e3[d]])), {d,
   0.5}]
Plot[{-Log2[S3[d]], -e3[d]*
    Log2[e3[d]] - (1 - e3[d]) (Log2[1 - e3[d]])}, {d, 0, 1}]
Plot[{e3[d], 1 - S3[d]}, {d, 0, 1}]
```

A.6. Bloc 3 : 1 bit

Nouvelle page de code

```
c := (Cos[Pi/8])^2
s := (Sin[Pi/8])^2

mp[d_] := (1 - d/4)^3 + 3*(d/4)^2*(1 - d/4)
X1[d_] := (1 - d)^3/mp[d]
X2[d_] := (2*d*(1 - d)^2*(c^2 + s^2))/mp[d]
Y[d_] := (d*(1 - d)^2*(c^2 + s^2))/mp[d]
Z[d_] := (2*d^2*(1 - d)*(c^4 + 6*c^2 s^2 + s^4))/mp[d]
U[d_] := (d^2*(1 - d)*(c^4 + 6*c^2 s^2 + s^4))/mp[d]
V[d_] := (d^3*(c^6 + 6*c^4 s^2 + 9*c^2 s^4))/mp[d]
W[d_] := (d^3*(9*c^4 s^2 + 6*c^2 s^4 + s^6))/mp[d]
succX := 1/2
succY := (c^2)/(c^2 + s^2)
succZ := (2*c^4 + 6*c^2 s^2)/(2*(c^4 + 6*c^2 s^2 + s^4))
succU := (c^4 + 2*c^2 s^2 + s^4)/(c^4 + 6*c^2 s^2 + s^4)
succV := (c^6 + 4*c^4 s^2 + 3*c^2 s^4)/(c^6 + 6*c^4 s^2 + 9*c^2 s^4)
succW := (6*c^4 s^2 + 2*c^2 s^4)/(9*c^4 s^2 + 6*c^2 s^4 + s^6)

err[d_] := (2*(d/4)^2*(1 - d/4))/((1 - d/4)^3 +
    3*(d/4)^2*(1 - d/4))(* fois 2? *)

Succ[d_] := (succX^(X1[d] + X2[d]))*(succY^(Y[d]))*(succZ^(Z[
      d]))*(succU^(U[d]))*(succV^(V[d]))*(succW^(W[d]))
MinEntEve[d_] := -Log2[Succ[d]]
longSyn[d_] := -err[d]*Log2[err[d]] - (1 - err[d]) (Log2[1 - err[d]])
N[FindRoot[MinEntEve[d] - longSyn[d], {d, 0.1}]]
```

```
Plot[{MinEntEve[d], longSyn[d]}, {d, 0, 1}]
```

A.7. Bloc 4 : 1 bit

```
mp4[d_] := (1 - d/4)^4 + 6*(d/4)^2*(1 - d/4)^2 + (d/4)^4;
X14[d_] := (1 - d)^4/mp4[d];
X24[d_] := (3*d*(1 - d)^3*(c^2 + s^2))/mp4[d];
Y4[d_] := (d*(1 - d)^3*(c^2 + s^2))/mp4[d];
X34[d_] := (3*d^2*(1 - d)^2*(c^4 + 6*c^2 s^2 + s^4))/mp4[d];
Z4[d_] := (3*d^2*(1 - d)^2*(c^4 + 6*c^2 s^2 + s^4))/mp4[d];
U4[d_] := (3*d^3*(1 - d)*(c^6 + 15*c^4 s^2 + 15*c^2 s^4 + s^6))/mp4[d];
V4[d_] := (d^3*(1 - d)*(c^6 + 15*c^4 s^2 + 15*c^2 s^4 + s^6))/mp4[d];
W14[d_] := (d^4*(c^8 + 12*c^6 s^2 + 38*c^4 s^4 + 12*c^2 s^6 + s^8))/
   mp4[d];
W24[d_] := (d^4*(16*c^6 s^2 + 32*c^4 s^4 + 16*c^2 s^6))/mp4[d];
succX4 = succX;
succY4 = succY;
succZ4 = (3*c^4 + 9*c^2 s^2)/(3*(c^4 + 6*c^2 s^2 + s^4));
succU4 = (3*(c^6 + 10*c^4 s^2 + 5*c^2 s^4))/(3*(c^6 + 15*c^4 s^2 +
       15*c^2 s^4 + s^6));
succV4 = (c^6 + 7*c^4 s^2 + 7*c^2 s^4 + s^6)/(c^6 + 15*c^4 s^2 +
     15*c^2 s^4 + s^6);
succW14 = 0.95;
succW24 = 0.5;

err4[d_] := (3*(d/4)^2*(1 - d/4)^2 + (d/4)^4)/((1 - d/4)^4 +
    6*(d/4)^2*(1 - d/4)^2 + (d/4)^4)
Succ4[d_] := (succX4^(X14[d] + X24[d] + X34[d]))*(succY4^(Y4[
      d]))*(succZ4^(Z4[d]))*(succU4^(U4[d]))*(succV4^(V4[
      d]))*(succW14^(W14[d]))*(succW24^(W24[d]))
MinEntEve4[d_] := -Log2[Succ4[d]]
longSyn4[d_] := -err4[d]*
   Log2[err4[d]] - (1 - err4[d]) (Log2[1 - err4[d]])
N[FindRoot[MinEntEve4[d] - longSyn4[d], {d, 0.1}]]
Plot[{MinEntEve4[d], longSyn4[d]}, {d, 0, 1}]
```

A.8. Bloc 3 : 2 bits

```
PA[d_] := (1 - d)^3/mp[d]
```

```
PB[d_] := (d*(1 - d)^2*(c^2 + s^2))/mp[d]
PC[d_] := (2*d*(1 - d)^2*(c^2 + s^2))/mp[d]
PD[d_] := (2*d^2*(1 - d)*(c^4 + 6*c^2 s^2 + s^4))/mp[d]
PE[d_] := (d^2*(1 - d)*(c^4 + 6*c^2 s^2 + s^4))/mp[d]
PF[d_] := (d^3*(9*c^4 s^2 + 6*c^2 s^4 + s^6))/mp[d]
PG[d_] := (d^3*(c^6 + 6*c^4 s^2 + 9*c^2 s^4))/mp[d]
succA1 := 1/2
succA2 := 1/2
succB1 := 1/2
succB2 := 1/2
succC1 := c^2/(c^2 + s^2)
succC2 := 1/2
succD1 := (c^4 + 3*c^2 s^2)/(c^4 + 6*c^2 s^2 + s^4)
succD2 := (c^4 + 2*c^2 s^2 + s^4)/(c^4 + 6*c^2 s^2 + s^4)
succE1 := (c^4 + 3*c^2 s^2)/(c^4 + 6*c^2 s^2 + s^4)
succE2 := (c^4 + 3*c^2 s^2)/(c^4 + 6*c^2 s^2 + s^4)
succF1 := (6*c^4 s^2 + 2*c^2 s^4)/(9*c^4 s^2 + 6*c^2 s^4 + s^6)
succF2 := (6*c^4 s^2 + 2*c^2 s^4)/(9*c^4 s^2 + 6*c^2 s^4 + s^6)
succG1 := (c^6 + 4*c^4 s^2 + 3*c^2 s^4)/(c^6 + 6*c^4 s^2 + 9*c^2 s^4)
succG2 := (c^6 + 4*c^4 s^2 + 3*c^2 s^4)/(c^6 + 6*c^4 s^2 + 9*c^2 s^4)

err3[d_] := (2*(d/4)^2*(1 - d/4))/((1 - d/4)^3 + 3*(d/4)^2*(1 - d/4))
Succ3[d_] := (succA1^(PA[d]/2))*(succA2^(PA[d]/2))*(succB1^(PB[d]/
      2))*(succB2^(PB[d]/2))*(succC1^(PC[d]/2))*(succC2^(PC[d]/
      2))*(succD1^(PD[d]/2))*(succD2^(PD[d]/2))*(succE1^(PE[d]/
      2)) (succE2^(PE[d]/2))*(succF1^(PF[d]/2))*(succF2^(PF[d]/
      2))*(succG1^(PG[d]/2))*(succG2^(PG[d]/2))
MinEntEve3[d_] := -Log2[Succ3[d]]
longSyn3[d_] := -err3[d]*
   Log2[err3[d]] - (1 - err3[d]) (Log2[1 - err3[d]])
N[FindRoot[MinEntEve3[d] - longSyn3[d], {d, 0.1}]]
Plot[{MinEntEve3[d], longSyn3[d]}, {d, 0, 1}]
```

A.9. Bloc 3 : XOR 2 bits

```
succA := 1/2
succB := c^2/(c^2 + s^2)
succC := 1/2
```

```
succD := (c^4 + 3*c^2 s^2)/(c^4 + 6*c^2 s^2 + s^4)
succE := (c^4 + 2*c^2 s^2 + s^4)/(c^4 + 6*c^2 s^2 + s^4)
succF := (3*c^4 s^2 + 4*c^2 s^4 + s^6)/(9*c^4 s^2 + 6*c^2 s^4 + s^6)
succG := (c^6 + 4*c^4 s^2 + 3*c^2 s^4)/(c^6 + 6*c^4 s^2 + 9*c^2 s^4)

err32[d_] := (2*(d/4)^2*(1 - (d/4)))/((1 - d/4)^3 +
    3*(d/4)^2*(1 - d/4))
succ32[d_] := (succA^PA[d])*(succB^PB[d])*(succC^PC[d])*(succD^
    PD[d])*(succE^PE[d])*((1 - succF)^PF[d])*(succG^PG[d])
MinEntEve32[d_] := -Log2[succ32[d]]
longSyn32[
  d_] := (-err32[d]*
    Log2[err32[d]]) - ((1 - err32[d])*(Log2[1 - err32[d]]))
N[FindRoot[MinEntEve32[d] - longSyn32[d], {d, 0.1}]]
Plot[{MinEntEve32[d], longSyn32[d]}, {d, 0, 1}]
```