Mise en œuvre et évaluation de dispositifs de cryptographie quantique à longueur d'onde télécom

Simon Fossier

Thales Research & Technology France

Laboratoire Charles Fabry de l'Institut d'Optique

Université Paris-Sud 11

THÈSE

présentée pour obtenir le grade de
Docteur en Sciences de l'Université Paris-Sud 11

Spécialité : Physique

par
Simon Fossier

Sujet :

Mise en œuvre et évaluation de dispositifs de cryptographie quantique à longueur d'onde télécom

Soutenue le 23 octobre 2009
devant la commission d'examen composée de :

Messieurs	Antonio Acín,	Rapporteur
	Thierry Debuisschert,	Examinateur
	Robert Frey,	Examinateur
	Philippe Grangier,	Membre invité
	Philippe Painchault,	Examinateur
	Jean-François Roch,	Rapporteur
	Emmanuel Rosencher,	Président
Madame	Rosa Tualle-Brouri,	Directrice de thèse

‘Curiouser and curiouser!’ cried Alice. ‘Now I’m opening out like the largest telescope that ever was!’

Lewis Carroll
Alice’s adventures in Wonderland

Remerciements

Si le travail de rédaction et de soutenance est un exercice principalement individuel, confinant parfois même à l'effort solitaire, la présence de collègues et d'amis lors des trois ans de thèse qui précèdent est quant à elle indispensable. Je souhaite ici les en remercier.

Parce qu'ils sont souvent omis ou trop peu mis en avant, je tiens à consacrer mes premiers remerciements aux personnels techniques et administratifs, qui sont souvent dans l'ombre des chercheurs mais constituent à mon sens une grande force pour un laboratoire quand on leur permet d'exprimer leurs compétences intelligemment et pleinement.

En particulier, les résultats de mon travail de thèse ne seraient pas ce qu'ils sont sans l'investissement et la qualité du travail d'André Villing, qui m'a beaucoup appris, tant sur le plan électronique qu'humain. Quitte à gêner un peu sa modestie, je lui exprime ici ma profonde admiration.

Cette thèse a été doublement encadrée par Thales R&T et le LCFIO. Mes remerciements vont d'une part à Thierry Debuisschert, avec qui j'ai eu l'occasion de beaucoup travailler sur le système expérimental, et avec qui j'ai pu découvrir (entre autres !) que prudence et pragmatisme ne sont pas incompatibles avec efficacité et résultats. Ils vont d'autre part à Rosa Tualle-Brouri et Philippe Grangier, qui ont su être présents lors de cette thèse tout en me laissant une certaine autonomie. Merci à tous trois de m'avoir fait confiance et apporté idées, connaissance et sérénité.

Ces remerciements ne sauraient oublier Jérôme Lodewyck et Eleni Diamanti, avec lesquels j'ai travaillé sur le système expérimental, ainsi qu'Anthony Leverrier, théoricien affirmé mais toujours à l'écoute des problématiques expérimentales. Nous avons partagé plus qu'un bureau et des problèmes scientifiques, et leur bonne humeur a beaucoup contribué au plaisir que j'ai pris pendant cette thèse.

Merci à toutes les personnes avec qui j'ai eu la chance de collaborer, et en particulier aux chercheurs et/ou thésards qui ont permis de mener à bien la démonstration SECOQC. Cette expérience restera sans aucun doute marquante dans mon parcours professionnel.

De même, j'ai beaucoup apprécié l'ambiance, studieuse et accueillante, des deux laboratoires entre lesquels j'ai joué l'électron libre. Cette ambiance est portée, parfois contre mauvaise fortune bon cœur, par les femmes et hommes qui y travaillent, et je tiens à saluer leur dévouement à leur recherche et à tous les remercier pour leur sourire et humour qui ont su me motiver.

Enfin, parce que la thèse a toujours tendance à dépasser la vie professionnelle, merci à tous ceux, amis, famille, palmipèdes ou félins en tout genre et toute matière, qui ont contribué à faire de ces trois ans une période agréable à vivre. Et merci, Koru.

Cette thèse clôt 8 ans d'études supérieures, et 22 ans d'études tout court. Toutes les rencontres de ces années ont contribué à me forger en tant que personne ; au delà de l'aspect scientifique, que ce texte soit donc un tribut à ces 24 premières années de vie, et un hommage à ceux qui y ont participé. Bon vent à tous !

Table des matières

Introduction

Le travail de thèse présenté dans ce manuscrit a été réalisé entre 2006 et 2009, dans le cadre d'une collaboration entre l'équipe de cryptographie quantique de Thales Research & Technology France et le groupe d'Optique quantique du laboratoire Charles Fabry de l'Institut d'Optique.

Il porte sur la réalisation expérimentale d'un démonstrateur fibré de cryptographie quantique à variables continues, ainsi que de la recherche de solutions originales pour optimiser son fonctionnement ou améliorer ses performances.

Etat de l'art en début de thèse

Le protocole que nous utilisons dans notre système a été proposé par Frédéric Grosshans et Philippe Grangier en 2002 [1, 2]. Il utilise des états cohérents et une mesure homodyne, particulièrement adaptés à une implémentation réelle.

Frédéric Grosshans, puis Jérôme Wenger, ont monté au cours de leurs thèses respectives [3, 4] une expérience de principe de cryptographie quantique en espace libre. En particulier, cette démonstration de principe a mené au développement d'un système de détection homodyne fonctionnant en régime pulsé, et limité par le bruit de photon.

Par la suite, Jérôme Lodewyck [5] a réalisé le transfert de l'expérience vers une technologie fibrée. Il a monté un système de cryptographie quantique uniquement composé de composants fibrés, et développé un logiciel assurant la modulation et la mesure des impulsions optiques, ainsi que les étapes classiques d'extraction de clé. Il a caractérisé le bruit du dispositif, qui détermine la sécurité de la transmission, puis réalisé une distribution quantique de clé à travers une fibre de 25 km.

Travail original

Le premier objectif de ce travail de thèse a été de sortir des conditions expérimentales optimales d'un laboratoire. Nous avons donc réalisé un système de démonstration de distribution quantique de clés entièrement composé de composants télécom fibrés standards, intégré dans des racks 19 pouces, de manière à le rendre transportable. Un travail important d'optimisation du montage optique et des composants a été mené, de manière à optimiser les pertes et assurer une stabilité maximale du système.

Par ailleurs, nous avons développé un logiciel complet de gestion du système. Il assure d'une part le bon fonctionnement de la transmission quantique : modulation et mesure, calibrations automatisées et rétrocontrôle en temps réel de tous les compo-

sants actifs du système. D'autre part, il permet d'effectuer la totalité des algorithmes nécessaires à l'extraction des clés secrètes. En particulier, un travail d'optimisation des performances du système a été mené, notamment concernant la réconciliation et l'amplification de confidentialité.

Le système a été transporté à Vienne (Autriche) pour une démonstration en vraie grandeur sur un réseau de cryptographie quantique. Il a fonctionné de façon continue, sans intervention humaine, pendant plus de 50 heures avec un taux de 8 à 10 kbit/s sur une fibre installée présentant 3 dB de pertes en ligne (équivalentes à 15 km de fibre standard).

Par ailleurs, nous avons étudié les possibilités d'amélioration du protocole de cryptographie quantique, en particulier :

- un nouveau protocole à modulation discrète, qui a été implémenté sur le système de démonstration et caractérisé de façon préliminaire.
- la possibilité théorique d'ajouter un pré-amplificateur paramétrique à la sortie du canal de transmission, qui permet de compenser les défauts du détecteur.
- les effets d'un amplificateur non-déterministe permettant d'améliorer le rapport signal à bruit d'un état cohérent, en particulier quand il est placé à la sortie du canal de transmission.

Collaborations

Outre la collaboration quotidienne entre Thales et l'Institut d'Optique, nous avons eu la chance de collaborer avec :

- Anthony Leverrier, de Télécom ParisTech, ainsi que Raúl García-Patrón, Nicolas Cerf et toute l'équipe QuIC de l'Université Libre de Bruxelles, qui nous ont fourni les preuves de sécurité du protocole.
- Les membres du projet SECOQC, et en particulier l'équipe Quantum Technologies de l'institut ARCS pour la mise au point du réseau de cryptographie quantique à Vienne.
- Philippe Painchault et Philippe Pache de Thales Communications France, ainsi que Romain Alléaume et l'équipe Information Quantique de Telecom ParisTech, pour l'implémentation d'une plateforme combinant cryptographies quantique et classique, dans le but de renouveler régulièrement avec la cryptographie quantique les clés de session de l'algorithme classique.

Organisation du manuscrit et guide de lecture

Ce manuscrit est divisé en quatre parties, présentant chacune un aspect du travail réalisé lors de cette thèse.

La première partie présente la théorie sous-jacente aux protocoles de cryptographie quantique : nous explorons dans un premier temps les techniques actuelles de cryptographie dans le chapitre 1, puis introduisons les notions de base de la cryptographie quantique. Le chapitre 2 est consacré aux états quantiques que nous utilisons dans le protocole, et aux outils théoriques dont nous avons besoin pour les manipuler. Le chapitre 3 introduit les notions d'information et d'entropie, nécessaires pour établir les preuves de sécurité de notre protocole, lesquelles sont présentées dans le chapitre 4.

La deuxième partie est consacrée à l'implémentation expérimentale du protocole de cryptographie quantique, et se divise en deux chapitres. Le chapitre 5 présente d'une part les caractéristiques des fibres optiques que nous utilisons, et d'autre part le fonctionnement des composants intervenant dans notre système. Le chapitre 6, quant à lui, expose les aspects d'intégration dans un démonstrateur transportable, la structure du programme de pilotage du système optique, et les résultats obtenus en terme de bruits et donc de taux théoriques de distribution de clé.

Après la transmission optique, notre protocole nécessite un traitement classique élaboré, afin d'extraire une clé secrète des données continues. La troisième partie du manuscrit est dédiée aux différents aspects de cette extraction : réconciliation des données (chapitre 7), amplification de confidentialité (chapitre 8). Le chapitre 9 décrit quant à lui la structure algorithmique de l'extraction, les enjeux de vitesse de traitement, puis les performances obtenues par le système lors d'un fonctionnement dans des conditions réelles.

Enfin, la quatrième partie est consacrée à des perspectives nouvelles d'amélioration du système. Le chapitre 10 présente principalement un nouveau protocole de cryptographie à variables continues permettant théoriquement d'augmenter la distance maximale de transmission. Le chapitre 11 (et dernier) est dédié à nos travaux théoriques sur la possibilité d'insérer des amplificateurs optiques dans notre système de cryptographie, de façon à compenser les imperfections des détecteurs, et donc d'améliorer les performances du système.

Première partie

Cryptographie quantique avec des variables continues

1 La cryptographie, des origines à la physique quantique

> Oui ! tout s'explique, tout s'enchaîne, tout est clair, et je comprends pourquoi Saknussemm, mis à l'index et forcé de cacher les découvertes de son génie, a dû enfouir dans un incompréhensible cryptogramme son secret...
>
> *Voyage au centre de la Terre*
> Jules Verne

Au cours de l'évolution des sociétés et des technologies, les techniques de sécurisation de l'information ont joué un rôle crucial de protection des données privées. Elles assurent au citoyen la confidentialité de l'échange de données personnelles, permettent aux entreprises de protéger leurs documents sensibles ou confidentiels, et sont utilisées jusqu'aux plus hauts niveaux de sécurité de l'État. Des techniques simples existent depuis longtemps, mais elles se sont perfectionnées plus récemment, avec le développement de la mécanique dans un premier temps, puis de l'électronique et de l'informatique. Ce chapitre apporte une vision globale des problématiques de cryptologie, en s'intéressant plus particulièrement aux méthodes de cryptographie algorithmique à clé privée ou secrète, utilisées couramment de nos jours, et aborde par la suite les techniques de cryptographie quantique, qui font l'objet de ce manuscrit.

1.1 κρυπτός / γράφειν : écrire le secret

kruptos : caché *graphein* : écrire

En le décomposant étymologiquement, le mot *cryptographie* est assez transparent et renvoie à la notion assez générale de l'écriture secrète. Dans ce cadre, la problématique la plus immédiate est de trouver une méthode pour transmettre un message entre deux points distants, de manière secrète, c'est-à-dire en en le rendant inaccessible à un espion (aussi appelé adversaire).

Deux solutions peuvent être employées. Tout d'abord, le canal de communication peut être rendu inaccessible à l'espionnage ; sécuriser physiquement le vecteur d'information est en effet une excellente option (la valise diplomatique accompagnée en est un bon exemple), mais nécessite des moyens particuliers, souvent indisponibles aux acteurs de la société civile.

Une autre approche consiste à considérer que le message transitant sur le canal peut être intercepté ou parfaitement copié, ce qui est par exemple le cas sur Internet. L'expéditeur va alors chercher à rendre ce message illisible à un espion, tout en s'assurant que le destinataire ait les moyens de le décrypter.

La cryptographie, dans son acception actuelle, renvoie à la deuxième approche. La cryptanalyse, science opposée et complémentaire, étudie quant à elle les failles de la cryptographie, et cherche à décrypter les messages codés transitant par le canal, sans que les interlocuteurs ne s'en aperçoivent. Ces deux domaines constituent les piliers de la cryptologie, la science du secret.

Considérations autour de la cryptographie

Notons tout d'abord que la cryptographie est à distinguer de la *stéganographie*, qui consiste à cacher le message écrit en clair dans un lieu rendu secret ; l'encre sympathique en est un exemple. Bien entendu, les différentes méthodes peuvent être couplées : ainsi, un expéditeur quelque peu paranoïaque pourra envoyer une lettre cryptée, écrite avec de l'encre sympathique, dans un coffre blindé. Nous ne nous intéresserons ici qu'au cryptage proprement dit.

De même, pour pouvoir prétendre à l'appellation « cryptographique », un protocole doit permettre au destinataire de retrouver le message d'origine sans erreur. Une technique consistant par exemple à supprimer les voyelles, pour rendre le message difficile à lire, tient plus de la stéganographie en ceci que le destinataire doit deviner les parties manquantes.

Finalement, l'espionnage, notion clé de la cryptologie, est à distinguer du *déni de service*, qui consiste à interrompre la transmission entre les interlocuteurs. Celui-ci, en effet, ne peut être évité que par une sécurisation physique du canal, ce qui relève d'une autre problématique que nous n'aborderons pas dans ce manuscrit.

Les acteurs

Comme tous les spécialistes, les cryptologues ont leurs traditions. Parmi celles-ci, nommer les différents personnages intervenant dans les transmissions est sans aucun doute la plus établie, et nous l'emploierons donc dans ce texte. Voici la distribution des rôles.

- Utilisateurs légitimes :
  - **Alice**, *expéditrice des messages, très bavarde.*
  - **Bob**, *destinataire des messages, entiché d'Alice, connecté à elle par un canal de communication.* Il est parfois appelé Bernard.
- Adversaire :
  - **Eve**, *adversaire d'Alice, espionnant les conversations.* Son nom vient de *eavesdropping*, écoute clandestine. Elle a accès à la totalité du canal de communication et de ce qui y transite, mais ne peut pas se faire passer pour Alice ou Bob (voir partie 1.2.4). En cryptographie quantique, on considère qu'elle possède une puissance de calcul infinie, et qu'elle n'est limitée que par les lois de la physique quantique.
- Autres personnages parfois rencontrés :
  - Carol(e), Dave, etc. Ils sont souvent utilisés si la communication fait intervenir plus de deux personnes.
  - On ne donne parfois à Eve que le pouvoir d'écouter les échanges, sans les modifier. Intervient alors Mallory (de *malicious*), attaquant actif.
  - On introduit parfois [6] beaucoup d'autres personnages ; citons Oscar (*opponent*), Trudy (*intruder*), Trent (*trusted arbitrator*, tiers d'arbitrage), etc.

1.2 Quelques techniques de cryptographie classique

Beaucoup de techniques de cryptographie ont existé au cours des siècles, déclinées en de nombreuses variantes. Cependant, les techniques cryptanalytiques ont évolué en parallèle, si bien que certains protocoles auparavant considérés comme sûrs sont devenus aujourd'hui triviaux à casser. Nous essayons ici de catégoriser les différents procédés cryptographiques, historiques et plus récents, dans l'objectif de montrer les avantages et failles de ceux-ci, et donc la difficulté de bâtir un algorithme suffisamment robuste.

1.2.1 La cryptographie par substitution

Plusieurs protocoles, dits de *chiffrement par substitution mono-alphabétique*, ont pour principe la substitution de chaque lettre ou groupe de lettres du message par une autre lettre ou groupe de lettres, comme illustré sur la figure 1.1. Les interlocuteurs possèdent une table de correspondance, qui leur permet d'établir l'équivalence entre le message en clair et le message crypté (ou *cryptogramme*). La notion de *lettres* est ici à prendre dans un sens générique, celui du *symbole*, l'unité de base d'écriture du message.

Parmi ces protocoles, citons le code de César, consistant à décaler les lettres du message d'un pas fixe, ou le carré de Polybe, dans lequel chaque lettre est codée par ses coordonnées 2D. Bien que rendant difficile la lecture, ces cryptosystèmes sont assez aisés à déchiffrer, du fait même de l'univocité de la transformation. Ainsi, la lettre la plus utilisée (comme *e* en français) sera toujours codée par le même bloc ; par conséquent, une simple analyse de la fréquence des différents blocs permet de remonter

à la fréquence des lettres du message, ce qui détermine avec une bonne approximation le message lui-même.

Alice

Message : BONJOUR BOB, COMMENT VAS-TU ?

Codage : + 5

Cryptogramme : GTSOTZW GTG, HTRRJSY AFX-YZ ?

Décodage : - 5

Message : BONJOUR BOB, COMMENT VAS-TU ?

Bob

Figure 1.1 : *Chiffre de César. Ici, la clé correspond à la valeur du décalage :* + 5.

Pour pallier cette faiblesse, plusieurs protocoles utilisent une substitution *poly-alphabétique*, une généralisation du code de César où le pas de substitution est lui-même défini grâce à une clé partagée *a priori* (voir Fig. 1.2). Le code de Vigenère et le cryptosystème Enigma utilisé par les Allemands pendant la Seconde Guerre mondiale font partie des plus célèbres codes poly-alphabétiques. Ces codes sont bien plus difficiles à déchiffrer, car le cryptanalyste doit retrouver la clé (ou, du moins, ses caractéristiques) pour décrypter le message. Cependant, tous ont été cassés au cours du XXe siècle — la victoire des Alliés n'y est pas étrangère.

Alice

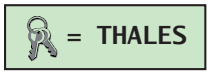
Message BONJOUR BOB, COMMENT VAS-TU ?

\+

Clé répétée THALEST HAL ESTHALE STH AL

=

Cryptogramme VWOVTML JPN, HHGUFZY NUA-UG ?



Bob

Cryptogramme VWOVTML JPN, HHGUFZY NUA-UG ?

\-

Clé répétée THALEST HAL ESTHALE STH AL

=

Message BONJOUR BOB, COMMENT VAS-TU ?

Figure 1.2 : *Codage par substitution polyalphabétique. Dans cet exemple, la clé partagée* a priori *est « Thales », répétée plusieurs fois pour coder le message. On voit cependant que si l'espion devine que le premier mot du message est « Bonjour », il est en mesure de décrypter entièrement le cryptogramme.*

1.2.2 Les principes de Kerckhoffs

En 1883, Auguste Kerckhoffs énonce [7] six principes à respecter pour assurer la confidentialité de transmissions cryptées :

- Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;
- Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;
- La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;
- Il faut qu'il soit applicable à la correspondance télégraphique ;
- Il faut qu'il soit portatif, et que son maniement n'exige pas le concours de plusieurs personnes ;
- Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

Ces principes sont un peu datés, et tiennent de la logique militaire de l'époque ; ils ont néanmoins permis de poser les bases d'une réflexion sur la place du secret dans la cryptographie. Essayons de les reformuler d'une manière plus adaptée à la cryptologie moderne.

Tout d'abord, le « système » qu'évoque Kerckhoffs recouvre à la fois les notions de protocole, d'algorithme, et d'implémentation. La remarque sur la simplicité et la portabilité est beaucoup moins justifiée, à l'ère de l'informatique, si l'on se place du point de vue de l'algorithme et du protocole — ils sont en effet gérés par l'ordinateur. En revanche, elle renvoie à une nécessité de simplicité de l'implémentation de l'interface utilisateur. De même, l'idée sous-jacente à la « correspondance télégraphique » est que le canal de communication, et donc le cryptogramme, sont supposés espionnables. Une interprétation plus actuelle des principes peut donc être :

- *Sécurité inconditionnelle :* Un message ne doit pas pouvoir être déchiffré sans la connaissance de la clé de chiffrement ;
- *Publicité du système :* Le système de chiffrement doit être entièrement public, et la sécurité du message ne doit donc pas dépendre d'un quelconque secret lié au mode de fonctionnement du système.

  Corollaire : La clé de chiffrement doit être modifiable à souhait, et donc être indépendante du système proprement dit.
- *Accessibilité du canal :* Le canal de transmission, et donc le cryptogramme qui y transite, sont supposés accessibles à l'espion.

Ces principes sont généralement respectés par les systèmes de cryptographie civils. En revanche, les militaires développent régulièrement des algorithmes secrets de chiffrement, la confidentialité du protocole étant supposée renforcer sa sécurité. Cette idée, en contradiction avec le point de vue de Kerckhoffs, est sujette à une vive controverse qui oppose régulièrement partisans du logiciel libre et défenseurs du code propriétaire fermé.

1.2.3 La cryptographie algorithmique moderne

Le chiffrement à clé symétrique

Le chiffrement à clé symétrique formalise le paradigme simple des systèmes à substitution : si les interlocuteurs partagent *a priori* une clé inconnue de l'adversaire, dite *clé secrète*, ils peuvent l'utiliser à la fois pour coder et décoder le message. Pour assurer

la sécurité, il faut néanmoins trouver un algorithme dont le déchiffrement soit simple avec la clé, mais particulièrement difficile, voire impossible, sans la connaissance de celle-ci.

Claude Shannon a exposé en 1948 [8] l'un des critères nécessaires à la sécurité inconditionnelle d'un protocole à clé symétrique : la longueur de la clé doit être au moins aussi longue que le message à chiffrer. Cette contrainte est particulièrement forte, mais a permis de montrer la sécurité d'un tel protocole, appelé *codage à masque jetable*, ou *one-time pad* en anglais. Ce protocole est d'une simplicité remarquable, car l'encodage consiste simplement à faire l'addition binaire[1], bit à bit, d'un message et d'une clé secrète de même longueur, le cryptogramme pouvant être ensuite décrypté par la même opération (voir figure 1.3). Il est aisé de montrer que ce protocole possède une sécurité inconditionnelle, sous réserve que chaque clé de chiffrement ne soit utilisée qu'une seule fois.

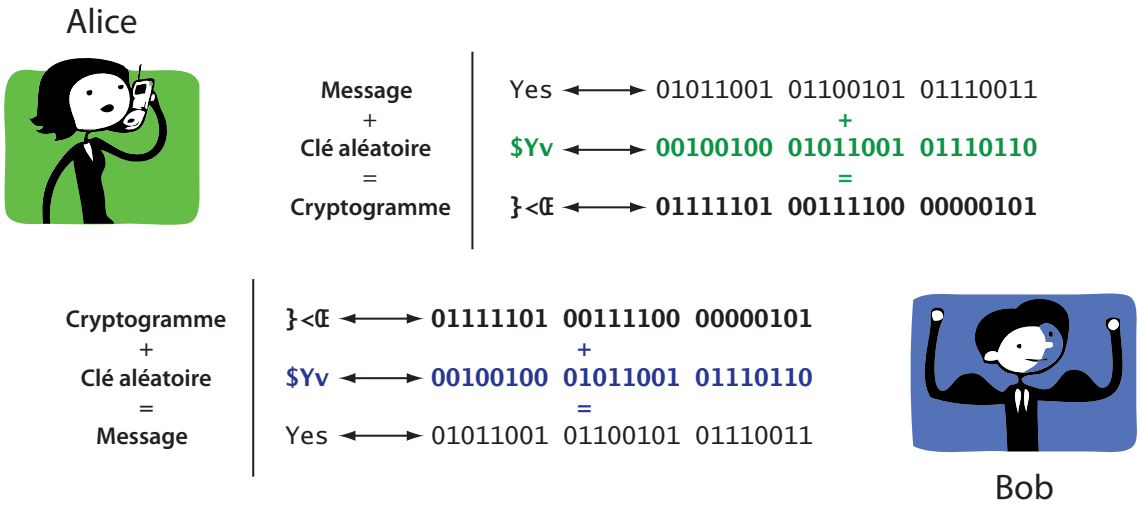


Figure 1.3 : *Principe du codage à masque jetable. La clé doit être aléatoire, et utilisée une seule fois : dans ce cas, la sécurité du codage est démontrée.*

Si le critère de Shannon n'est pas respecté, la sécurité ne peut être démontrée formellement. En revanche, les protocoles peuvent être testés contre un grand nombre d'attaques spécifiques, de manière intense. En particulier, les protocoles doivent s'assurer que la clé soit suffisamment longue pour empêcher une attaque dite *exhaustive*, consistant à tester toutes les clés possibles. Par conséquent, la sécurité est dite *calculatoire* : le concepteur du protocole essaie de trouver la meilleure attaque contre le système, et estime la durée minimale que prendrait cette attaque pour déterminer la clé. Si cette durée est déraisonnable (et ceci est subjectif : par exemple, 100 ans sont-ils suffisants ?), le protocole est considéré comme sûr.

De très nombreux protocoles à chiffrement symétrique existent. Citons l'un des plus célèbres, DES (Data Encryption Standard [9]), qui n'est plus utilisé de nos jours car sa clé de chiffrement de 56 bits est trop courte, et autorise une attaque exhaustive très rapide. On considère de nos jours qu'une clé de 80 bits est un minimum, mais

1. L'addition binaire se comprend comme l'opération *ou exclusif*, de sorte que $1 \oplus 1 = 0$.

une clé de 128, voire 256 bits, est recommandée. Parmi les algorithmes les plus utilisés aujourd'hui, on trouve 3DES (Triple Data Encryption Standard [10]), Blowfish [11] et AES (Advanced Encryption Standard [12]).

Le chiffrement à clés asymétriques

La nécessité du partage d'une clé secrète est l'une des limites du chiffrement symétrique ; en effet, il suppose une transmission physique de la clé entre les interlocuteurs, avec les contraintes que ceci comporte. Cette transmission peut en outre être impossible, comme par exemple pour le cas d'un consommateur qui veut acheter sur Internet : il est difficile d'imaginer que le site marchand envoie par transporteur sécurisé une clé secrète, que le consommateur utiliserait ensuite pour envoyer ses données bancaires !

Le chiffrement asymétrique offre une solution à ce problème. Le destinataire génère tout d'abord une paire de clés complémentaires : la clé de décodage, appelée *clé privée*, est conservée par le destinataire, et la *clé publique* d'encodage est mise à disposition du public sur un serveur de clés authentifié. Si Alice veut envoyer un message codé à Bob, il lui suffit de récupérer la clé publique de Bob sur le serveur, d'encrypter le message avec celle-ci, et d'envoyer le cryptogramme à Bob. Celui-ci peut ensuite le décrypter aisément avec sa clé privée.

On ne connaît actuellement pas de moyen pour démontrer formellement la sécurité des protocoles à clés asymétriques : il s'agit à nouveau d'une sécurité calculatoire, reposant sur le fait que retrouver la clé privée à partir du message et de la clé publique est extrèmement difficile.

Ce type de protocole est utilisé couramment sur tous les sites sécurisés du Web, car il ne requiert pas de contact *a priori* pour transmettre un message. En revanche, il nécessite des clés beaucoup plus longues que pour l'encryptage symétrique. On considère aujourd'hui que pour une sécurité à assez long terme (typiquement 40 ans), une clé d'au moins 4096 bits devrait être utilisée, ce qui n'est que très rarement le cas en pratique. Parmi les algorithmes les plus connus, citons RSA (pour Rivest, Shamir et Adleman [13]) et DSA (Digital Signature Algorithm [14]).

1.2.4 Autour de l'encryptage...

Authentification : certification et signature

Pour être certain de la sécurité de la transmission, il faut que Bob puisse s'assurer que le message qu'il a reçu provient bien d'Alice et non d'Eve, et qu'il n'a pas été modifié. Dans le cas contraire, Eve pourrait utiliser des attaques du type *man in the middle*, c'est-à-dire qu'elle pourrait se faire passer alternativement pour Alice et Bob, non pas pour décrypter leurs messages mais pour les manipuler en leur faisant croire qu'ils discutent effectivement.

La certification des clés publiques et la signature électronique permettent d'éviter ce type d'attaques, à travers une infrastructure de gestion de clés (PKI, pour *Public-Key Infrastructure*). Dans un premier temps, quand Bob reçoit la clé d'Alice, il contacte un organisme de certification (de confiance), qui lui permet de vérifier le lien entre la clé et l'identité d'Alice. Ensuite, pour éviter que l'espion ne modifie le message qu'elle a envoyé, Alice le signe avec sa clé privée, grâce à un algorithme de signature

électronique. Bob peut alors vérifier l'intégrité du message en utilisant la clé publique d'Alice.

Cette procédure d'authentification, comprenant la vérification d'identité et d'intégrité, est cruciale en cryptographie dès lors qu'Alice et Bob utilisent un canal de communication classique. En cryptographie quantique, Alice et Bob utilisent un canal classique pour échanger toutes les informations classiques nécessaires pour l'extraction de leur clé, et ils doivent ainsi s'assurer que celui-ci est authentifié.

Aspects légaux

La cryptographie, sa légalité, et sa liberté d'emploi ont été le sujet de nombreuses controverses. Les États ont longtemps considéré les techniques cryptographiques comme des armes de guerre, et donc relevant d'une prérogative de l'armée, son emploi dans la société civile — ou pire, à des fins économiques — étant souvent illégal.

En France, l'usage de la cryptographie devient vraiment possible à partir de 1996, mais sous de nombreuses contraintes : une demande d'autorisation du procédé est requise, et la taille des clés de cryptage est limitée à 40 bits.
En 1999, la loi française s'assouplit encore, à la surprise des défenseurs de la cryptographie libre, en autorisant l'usage de tout procédé cryptographique mettant en œuvre des clés de 128 bits maximum, et en transformant la demande d'autorisation en une simple déclaration.
Depuis 2004 et jusqu'à aujourd'hui, il n'y a plus de taille limite des clés de cryptage, et l'usage de la cryptographie est libre, mais la création d'un nouveau procédé est toujours soumis à déclaration.

Dans le reste du monde, les lois relatives à l'usage de la cryptographie varient beaucoup en fonction des pays, et il n'existe qu'assez peu d'accords internationaux, mais la dynamique actuelle reste axée sur la libéralisation.

1.3 La cryptographie quantique

1.3.1 Idée

A l'heure actuelle, on ne connaît qu'une seule technique pour crypter des données tout en en démontrant formellement la sécurité par la théorie de l'information : c'est le codage à masque jetable [15] (et ses variantes), décrit dans la section 1.2.3. Comme nous l'avons expliqué, l'inconditionalité de la sécurité repose sur deux hypothèses : la clé est aussi longue que le message, et elle n'est utilisée qu'une seule fois. L'implication est directe : pour crypter, mettons, un film de 800 mégaoctets, il faudra 800 mégaoctets de clé, partagée *a priori* par Alice et Bob. Cette taille est très importante, comparée aux quelques centaines de bits nécessaires pour coder des gigaoctets de message avec les autres protocoles. Comment, donc, transmettre cette clé entre Alice et Bob, qui plus est de manière sûre, pour pouvoir établir une chaîne complète de secret ?

La technique employée pour les données militaires très sensibles est le transport d'une grande quantité de clé par valise diplomatique, la clé étant ensuite employée dans les communications et renouvelée une fois le stock épuisé. C'est la technique utilisée pour le fameux *téléphone rouge*, implémenté par « Alice » Krouchtchev et « Bob » F.

Kennedy en 1963. Néanmoins, ce type de méthode, outre sa non-disponibilité à la société civile, nécessite un contact direct ou un intermédiaire.

Se pose donc la problématique suivante : est-il possible de distribuer des clés secrètes entre deux interlocuteurs (1) à distance, (2) à la demande, et (3) avec une sécurité démontrable ?

La cryptographie quantique propose une solution à ce problème. Formellement, elle peut être définie comme l'association de deux techniques : la distribution quantique de clés secrètes, et l'utilisation de ces clés dans le codage à masque jetable. Néanmoins, on appelle usuellement « cryptographie quantique » les techniques de distribution à distance de clés secrètes qui utilisent la physique quantique pour les démonstrations de sécurité. Elle est d'ailleurs dénommée *quantum key distribution*, ou QKD, en anglais.

1.3.2 Principe général

Dans les protocoles de télécommunications classiques, l'information est codée par Alice sur l'amplitude d'impulsions lumineuses intenses, qui se propagent dans des fibres optiques. En mesurant cette amplitude, Bob est à même de récupérer l'information initiale. Bien entendu, l'information qui transite sur le canal est parfaitement accessible à Eve, qui peut prélever une partie du signal, et éventuellement amplifier le signal de manière à cacher son prélèvement ; Bob est alors incapable de se rendre compte de l'espionnage.

Protocole général de cryptographie quantique

La cryptographie quantique s'inspire des protocoles classiques, à la différence que le support de l'information est maintenant une particule se comportant de manière quantique :

- Alice génère un état quantique déterminé, et code une information aléatoire sur deux observables qui ne commutent pas ;
- Alice envoie cet état à Bob par un canal adapté, dit *canal quantique* ;
- Bob reçoit l'état, et mesure *de manière aléatoire* l'une des deux observables pour en récupérer l'information.

Raisonnement qualitatif sur la sécurité

Un objet élémentaire, tel qu'un photon ou un électron, est soumis aux principes suivants, posés par la physique quantique :

- **Théorème de non-clonage :** Il est impossible de réaliser des copies parfaites (des clones) d'états quantiques inconnus.
- **Relations d'incertitude de Heisenberg :** Si deux observables d'un état quantique ne commutent pas, il est impossible de réaliser simultanément, et avec une précision arbitraire, les mesures correspondantes.

Si, maintenant, Eve cherche à mesurer l'information codée sur l'état quantique, ces deux principes nous assurent qu'elle va (1) faire une erreur sur sa mesure, et (2) perturber l'état qui transite entre Alice et Bob. Elle va donc introduire du bruit — ou

des erreurs — sur la mesure de Bob, et ce dernier va donc remarquer la présence de l'espion.

Ainsi, dans un système de cryptographie quantique, *tout espionnage se traduira par du bruit ajouté sur la mesure de Bob*, ce bruit devenant donc la *signature de l'espion*.

Preuves de sécurité

Le raisonnement qualitatif ci-dessus ne suffit bien entendu pas à démontrer la sécurité d'un protocole, et il est nécessaire de formaliser les notions d'information, de mesure, etc., dans un calcul dit « preuve de sécurité inconditionnelle ». Ces preuves utilisent des outils de théorie de l'information et de physique quantique pour déterminer la quantité d'information accessible à l'espion, et donc la quantité d'information secrète ayant été transmise par le canal quantique. Le chapitre 4 expose les preuves de sécurité appliquées au protocole étudié dans ce manuscrit.

1.3.3 Quelques protocoles de cryptographie quantique

En pratique, tous les protocoles de cryptographie quantique utilisent les photons comme vecteurs d'information, car ils sont relativement faciles à produire, faciles à manipuler et voyagent (très) rapidement dans des fibres optiques, tout en subissant peu d'atténuation. Une fois ce choix fait, il existe à peu près autant de protocoles que de propriétés sur lesquelles coder l'information : polarisation, amplitude, phase, fréquence et temps.

Historiquement, le premier protocole à avoir été imaginé et implémenté est appelé BB84 [16] — méthode standard pour nommer un protocole, qui vient des inventeurs éponymes Gilles **B**ennett et Claude **B**rassard et de l'année de publication, 19**84**. L'information est codée sur la polarisation de photons uniques, en choisissant deux bases de polarisations non indépendantes (par exemple H/V et $\circlearrowleft/\circlearrowright$) pour assurer la sécurité. Ce protocole a connu plusieurs variantes [17, 18], et a été implémenté de nombreuses fois en 25 ans.

E91, un autre protocole [19], imaginé par Artur Ekert, utilise des états intriqués EPR codés en polarisation et a été développé indépendament de BB84. Ces deux protocoles sont en général considérés comme les protocoles fondateurs de la cryptographie quantique.

D'autres protocoles utilisent des états laser très atténués, tout en effectuant une mesure discrète (détecteurs ou compteurs de photons). Citons par exemple les protocoles DPS (Differential Phase Shift) [20], où l'information est codée sur les phases successives des impulsions, mais aussi les protocoles à codage fréquentiel [21, 22, 23], et les protocoles à codage temporel [24, 25]. Un protocole dans lequel l'information est codée sur le temps de détection des photons est d'ailleurs développé au sein de Thales Research & Technology France [26].

État de l'art

Outre les développements effectués dans les centres de recherche de grands groupes (Thales, Toshiba, NEC, ...), quelques start-ups ont déjà vu le jour, et vendent des systèmes de cryptographie quantique, telles que MagiQ (États-Unis), idQuantique

(Suisse) ou SmartQuantum (Lannion, France). Il est donc clair qu'étant donné l'état de l'art, concurrencer les acteurs présents dans la communauté ne se résume pas à imaginer des protocoles, mais aussi à développer des démonstrateurs complets, incluant protocole, preuves de sécurité, intégration, pilotage et extraction de la clé.

1.3.4 Avantages et limitations de la cryptographie quantique

L'avantage principal de la cryptographie quantique est sa capacité à établir une chaîne complète de sécurité pour la transmission d'un message confidentiel. Cet atout en fait une option intéressante pour l'échange de données particulièrement sensibles. Du fait même de sa nature, néanmoins, un certain nombre de contraintes existe :

- Puisque toute mesure détruit l'état quantique, la transmission doit se faire à travers une fibre optique « noire », c'est-à-dire sans système d'amplification classique tel que les répéteurs utilisés dans l'industrie des télécommunications.
- Les pertes de la fibre de transmission ne peuvent donc pas être compensées, et ceci induit une distance limite de transmission, au delà de laquelle l'espion possède plus d'information que ce qu'Alice et Bob partagent. Il ne reste alors plus de « matériau » pour extraire une clé secrète. Actuellement, cette distance maximale est typiquement comprise entre 20 et 200 kilomètres, en fonction des protocoles.
- Il existe d'autres contraintes ou limitations, moins intrinsèques, mais plus liées à la relative jeunesse du domaine, telles qu'une limitation du débit maximal de transmission, dû par exemple à la fréquence maximale de détection des photons. Cet aspect « composants » évolue assez vite actuellement, ce qui permet d'envisager une accélération notable des vitesses de génération de clé secrète.

Combinaison cryptographie quantique / cryptographie classique

Les systèmes actuels qui implémentent la totalité des étapes d'extraction de clé sont assez peu nombreux, et présentent tous des taux de génération de clé de l'ordre de 1 à 10 kbit/s à une distance de fibre d'environ 20 km. Ces systèmes, employés tels quels avec un encryptage de type masque jetable, ne permettent donc pas de coder des messages aux vitesses requises par les applications usuelles, c'est-à-dire de l'ordre du Mbit/s ou Gbit/s.

Les algorithmes de cryptographie symétrique, en revanche, ont été implémentés dans des systèmes dédiés, encryptant les données à des vitesses atteignant le Gbit/s. Néanmoins, le fait qu'une seule clé soit utilisée pour crypter de grandes quantités de message pose le problème de « l'usure » de cette clé. Citons l'exemple de quelques attaques très médiatisées [27, 28] contre AES, dans laquelle les imperfections de l'infrastructure des processeurs sont utilisées pour retrouver une clé AES complète. Ces attaques nécessitent souvent la présence de collisions, c'est-à-dire la répétition de plusieurs blocs de données cryptés avec la même clé dans le cache du processeur.

Pour réduire le risque de ce type d'attaques, il est possible de combiner les forces des deux cryptographies, en utilisant la cryptographie quantique pour rafraîchir de manière régulière les clés de cryptage de l'algorithme classique. Compte tenu des débits accessibles actuellement, une clé AES de 256 bits peut être régénérée plusieurs fois par

seconde avec la cryptographie quantique. Bien que n'atteignant pas une sécurité inconditionnelle, cet assemblage permet de rendre beaucoup plus difficile le « cassage » du code classique, et conserve l'avantage de vitesse de la cryptographie algorithmique. C'est donc un bon exemple de synergie entre deux technologies *a priori* concurrentes.

Thales (à travers les unités Research & Technology et Communications), le LCFIO et Telecom ParisTech ont mis en place une collaboration, dans le cadre d'un projet ANR, SEQURE [29]. Ce projet a pour but d'implémenter une telle structure mixte, en couplant la technologie quantique présentée dans ce manuscrit avec les encrypteurs rapides Mistral Gigabit commercialisés par Thales, et d'effectuer une démonstration d'encryptage sur une fibre installée sur le terrain.

2 États gaussiens et variables continues

> God runs electromagnetics by wave theory on Monday, Wednesday and Friday, and the Devil runs them by quantum theory on Tuesday, Thursday and Saturday.
>
> Sir William Bragg

De nombreux protocoles de cryptographie quantique sont fondés sur un codage discret de l'information, en utilisant des photons uniques. Au contraire, dans notre système, l'information est codée sur des variables continues, à savoir les deux quadratures d'un état cohérent du champ électromagnétique. Nous présentons dans ce chapitre les propriétés principales des quadratures du champ et des états gaussiens, et décrivons le protocole à variables continues utilisé dans la suite de ce manuscrit.

2.1 États gaussiens, états cohérents

2.1.1 Quadratures du champ électromagnétique

Quadratures classiques

Le champ électrique classique d'un mode de la lumière peut s'exprimer comme

$$\vec{E}\,(\vec{r},t) = \left(Ae^{i\left(\vec{k}.\vec{r}-\omega t\right)} + A^* e^{-i\left(\vec{k}.\vec{r}-\omega t\right)}\right)\vec{\varepsilon} \tag{2.1}$$

et peut être caractérisé de manière scalaire par une amplitude complexe $\underline{E} = Ae^{i\varphi}$. Les quadratures classiques de la lumière sont définies comme les projections de cette amplitude sur les axes du plan complexe :

$$X = \frac{\underline{E} + \underline{E}^*}{2} = A\cos\varphi \qquad P = \frac{\underline{E} - \underline{E}^*}{2i} = A\sin\varphi \tag{2.2}$$

et de façon générale par

$$X_\theta = X\cos\theta + P\sin\theta = A\cos\left(\varphi - \theta\right) \tag{2.3}$$

Quantification du champ

Dans le cadre de la quantification du champ, on remplace l'amplitude complexe du champ par l'opérateur annihilation $\hat{a}$ et son conjugué l'opérateur création $\hat{a}^\dagger$, et on peut donc définir par extension les observables quadrature quantique :

$$\hat{X} = \sqrt{N_0}(\hat{a} + \hat{a}^\dagger) \qquad \hat{P} = \sqrt{N_0}\frac{(\hat{a} - \hat{a}^\dagger)}{i} \tag{2.4}$$

Les valeurs propres de ces observables peuvent prendre une infinité de valeurs possibles, réparties dans un continuum, et on parle donc de *variables continues*.

Les opérateurs $\hat{a}$ et $\hat{a}^\dagger$ vérifiant, par construction, la relation de commutation $\left[\hat{a},\hat{a}^\dagger\right] = 1$, les opérateurs quadrature vérifient

$$\left[\hat{X},\hat{P}\right] = 2iN_0 \tag{2.5}$$

Les deux observables $\hat{X}$ et $\hat{P}$ ne commutent donc pas. Il en découle la relation de Heisenberg suivante :

$$\Delta X\, \Delta P \geqslant N_0 \tag{2.6}$$

où Δ correspond à l'écart-type du bruit sur la mesure associée à la quadrature considérée. Cette relation implique l'impossibilité de réaliser une mesure simultanée, avec une précision arbitraire, de la valeur des deux quadratures.

Bruit de photon

Dans les expressions précédentes, le terme N_0 est un terme de dimensionnement, qui provient de la quantification du champ et dépend du choix des unités. Dans le cadre de la quantification du champ, il s'exprime comme $N_0 = \hbar\omega/2\varepsilon_0 V$, ce qui correspond au carré du champ électromagnétique associé à un photon dont le volume de quantification est V.

Dans l'optique de ce manuscrit, il apparaît plutôt, à travers l'expression (2.6), comme une valeur de référence du bruit sur la mesure des quadratures. N_0 est appelé *niveau de bruit quantique standard*, ou plus couramment *bruit de photon* en optique quantique. En anglais, le terme consacré est *shot noise*, ce qui correspond à la notion de bruit de grenaille [1].

Hamiltonien et nombre de photons

Le hamiltonien de l'oscillateur harmonique peut se définir à partir des quadratures :

$$\hat{H} = \frac{\hbar\omega}{4N_0}\left(\hat{X}^2 + \hat{P}^2\right) = \hbar\omega\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) \tag{2.7}$$

1. Ce nom provient d'une interprétation corpusculaire classique du comportement de la lumière, où l'on considère que les photons arrivent séparément, comme de la grenaille, sur le détecteur. Pour évaluer le bruit d'intensité δI d'un laser, on procède de manière statistique : si le détecteur détecte N photons, le bruit sur ce nombre sera $\delta N \propto \sqrt{N}$. On peut donc montrer que $\delta I \propto A$, où A est l'amplitude du signal. Or $\delta I = \delta(A^2) \propto A\,\delta A$, et donc $\frac{A\,\delta A}{A} = \delta A =$ constante. Le bruit sur l'amplitude est donc constant, ce qui renvoie à l'équation 2.6.

L'opérateur nombre de photons $\hat{n} = \hat{a}^\dagger \hat{a}$ apparaît dans l'expression du Hamiltonien, et l'on retrouve l'expression de l'énergie $\hat{E}$ de l'oscillateur harmonique :

$$\langle E \rangle = \hbar\omega \left(\langle \hat{n} \rangle + \frac{1}{2} \right) \tag{2.8}$$

2.1.2 États gaussiens

Fonction de Wigner

Puisqu'il est impossible d'avoir accès en même temps aux deux quadratures, il n'est pas possible de déterminer exactement la distribution du champ électrique dans l'espace des phases, et donc de définir la probabilité conjointe de mesure d'un couple de quadratures $\Pr(x,p)$. En revanche, pour toute quadrature $\hat{X}_\theta$, la distribution de probabilité marginale $\Pr(x_\theta)$ peut être déterminée aussi précisément que souhaité.

Pour représenter malgré tout la distribution du champ, il est possible de définir des *distributions de quasi-probabilité*, telles que la fonction de Wigner W, qui se comporte de manière analogue aux distributions de probabilité sans toutefois en posséder toutes les propriétés [30]. La fonction de Wigner d'un état monomode, $W(x,p)$, se définit formellement comme

$$W(x,p) = \frac{1}{2\pi\hbar} \int_{-\infty}^{\infty} e^{-ipx'/\hbar} \langle x + \frac{x'}{2} | \, \hat{\rho} \, | x - \frac{x'}{2} \rangle \, \mathrm{d}x' \tag{2.9}$$

où $\hat{\rho}$ est la matrice densité de l'état. Cette fonction est normée, mais à la différence d'une probabilité elle peut prendre des valeurs négatives. L'une des propriétés fondamentales de la fonction de Wigner est son lien avec les distributions de probabilité marginales de toutes les quadratures, qui peuvent s'exprimer en fonction de W :

$$\Pr(x_\theta) = \int_{-\infty}^{\infty} W(x,p) \mathrm{d}p_\theta \tag{2.10}$$

où $p_\theta = x_{\theta+\pi/2}$. La distribution de probabilité selon une quadrature est donc « l'ombre » de la fonction de Wigner selon la quadrature orthogonale, comme illustré sur la figure 2.1 et suivantes.

État gaussien

Un état gaussien est défini comme un état dont la fonction de Wigner est gaussienne, en conséquence de quoi les probabilités marginales de toutes les quadratures suivent elles aussi une distribution gaussienne. On peut en particulier montrer qu'un état pur est gaussien si et seulement si sa fonction de Wigner est positive pour tout x et p (théorème de Hudson-Piquet [31, 32]). Nous verrons plus loin que les états gaussiens ont des propriétés très intéressantes pour la théorie de l'information et les preuves de sécurité.

Citons quelques états gaussiens importants pour la suite :

- **L'état vide** $|0\rangle$: c'est l'état gaussien le plus simple, défini comme état fondamental de l'oscillateur harmonique, c'est-à-dire tel que $\hat{a}\,|0\rangle = 0$. Le nombre moyen de photons dans cet état est nul, d'où son nom.
  Il vérifie $\langle\hat{X}\rangle = \langle\hat{P}\rangle = 0$ et $\Delta X = \Delta P = \sqrt{N_0}$.

- **L'état cohérent** $|\alpha\rangle$: défini formellement comme vecteur propre de l'opérateur annihilateur, c'est-à-dire vérifiant $\hat{a}\,|\alpha\rangle = \alpha\,|\alpha\rangle$.
  Il vérifie $\langle\hat{X}\rangle = 2\sqrt{N_0}\,\mathrm{Re}(\alpha)$, $\langle\hat{P}\rangle = 2\sqrt{N_0}\,\mathrm{Im}(\alpha)$ et $\Delta X = \Delta P = \sqrt{N_0}$.

- **Les états comprimés monomodes** : puisque la relation d'incertitude 2.6 ne fait intervenir que le produit des bruits sur les deux quadratures, il est possible d'imaginer des états ayant un bruit inférieur au bruit de photon sur l'une des quadratures, au prix d'un bruit amplifié sur l'autre. Par exemple, pour un état comprimé selon $\hat{X}$ avec un facteur s :
  $\langle\hat{X}\rangle = 2\sqrt{N_0}\,\mathrm{Re}(\alpha)$, $\langle\hat{P}\rangle = 2\sqrt{N_0}\,\mathrm{Im}(\alpha)$, $\Delta X = \sqrt{\frac{N_0}{s}}$ et $\Delta P = \sqrt{sN_0}$.

- **L'état thermique** : il s'agit d'une superposition statistique d'états cohérents, qui correspond à l'état obtenu quand le champ radiatif est à l'équilibre avec un réservoir à une température T. Il peut s'exprimer sous la forme

$$\rho = \frac{1}{\pi\overline{n}} \int e^{-|\alpha|^2/\overline{n}}\,|\alpha\rangle\langle\alpha|\,\mathrm{d}\alpha$$

  où $\overline{n}$ est le nombre moyen de photons thermiques. Sa fonction de Wigner est similaire à celle de l'état vide, mais élargie puisque le nombre de photons dans l'état n'est pas nul.

- **Les états bimodes intriqués en quadrature, ou états EPR** : de manière analogue à l'expérience de pensée réalisée par Einstein, Podolsky et Rosen en 1935 [33], on peut remarquer que les quadratures d'un état bimode ρ_{AB} vérifient $[\hat{X}_A - \hat{X}_B, \hat{P}_A + \hat{P}_B] = 0$. La mécanique quantique autorise alors la connaissance parfaite des quantités $(\hat{X}_A - \hat{X}_B)$ et $(\hat{P}_A + \hat{P}_B)$ simultanément.
  Il est donc possible d'imaginer des états parfaitement intriqués, c'est-à-dire dont les opérateurs $\hat{X}_A$ et $\hat{X}_B$ sont parfaitement corrélés, et $\hat{P}_A$ et $\hat{P}_B$ parfaitement anti-corrélés. Un état EPR de variance VN_0 vérifie :
  $\left\langle\hat{X}_A^2\right\rangle = \left\langle\hat{X}_B^2\right\rangle = \left\langle\hat{P}_A^2\right\rangle = \left\langle\hat{P}_B^2\right\rangle = VN_0$,
  $\left\langle\hat{X}_A\hat{X}_B\right\rangle = \left\langle\hat{P}_A\hat{P}_B\right\rangle = \pm\sqrt{V^2-1}\,N_0$.

Propriétés des états cohérents

Les états cohérents sont faciles à produire : il s'agit par exemple des états de sortie d'un laser standard très au dessus du seuil, et sans diffusion de phase.

Les états cohérents peuvent être exprimés dans la base de Fock par :

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}}\,|n\rangle \tag{2.11}$$

où $|n\rangle$ est l'état nombre contenant exactement n photons, tel que $\hat{a}^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle$.

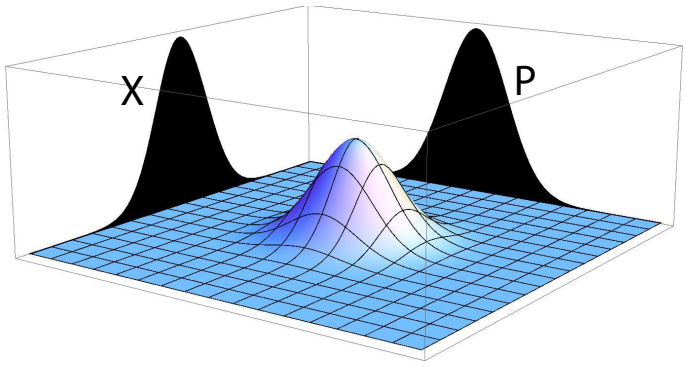


Figure 2.1 : *Fonction de Wigner et distributions en $\hat{X}$ et $\hat{P}$ de l'état vide $|0\rangle$.*

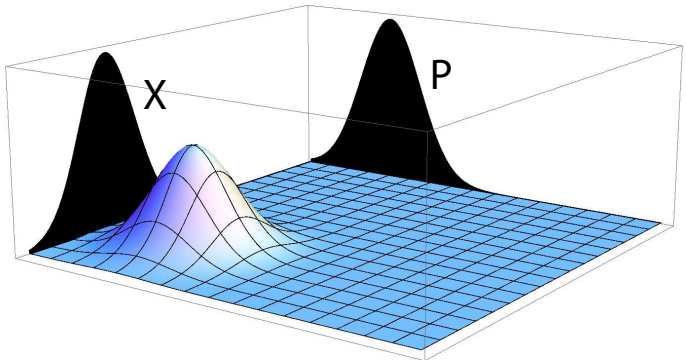


Figure 2.2 : *Fonction de Wigner et distributions en $\hat{X}$ et $\hat{P}$ d'un état cohérent $|\alpha\rangle$.*

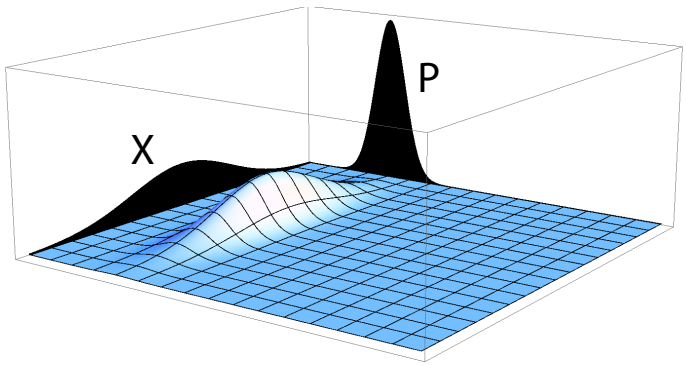


Figure 2.3 : *Fonction de Wigner et distributions en $\hat{X}$ et $\hat{P}$ d'un état comprimé selon $\hat{X}$.*

Les états cohérents sont des états pointeurs (*pointer states* en anglais), c'est-à-dire des états privilégiés de la lumière ne s'intriquant pas avec leur environnement, et n'étant donc pas soumis à décohérence [34]. En particulier, un état cohérent $|\alpha\rangle$ entrant dans une lame séparatrice de transmission $\sqrt{\eta}$ ne s'intrique pas avec le vide : la transformation s'écrit

$$|\alpha\rangle_{\mathrm{a}}\,|0\rangle_{\mathrm{b}} \qquad \rightarrow \qquad |\sqrt{\eta}\,\alpha\rangle_{\mathrm{c}}\,|\sqrt{1-\eta}\,\alpha\rangle_{\mathrm{d}} \tag{2.12}$$

Par conséquent, un état cohérent $|\alpha\rangle$ subissant des pertes en amplitude g deviendra l'état $|g\alpha\rangle$.

2.1.3 Transformations gaussiennes

Les opérations unitaires préservant le caractère gaussien des états sur lesquels elles sont appliquées sont appelées transformations unitaires gaussiennes.

Matrices de covariance

Les états gaussiens à n modes ont la propriété d'être entièrement définis par les valeurs moyennes et les covariances des $2n$ quadratures formant une base ($\hat{X}$ et $\hat{P}$ pour un état monomode).

Outre l'expression standard $\vec{r} = (x_1,p_1,x_2,\cdots,p_n)$ dans la base des $2n$ quadratures, il est donc utile de définir la *matrice de covariance* d'un état gaussien, qui s'écrit

$$\gamma_{2n} = \begin{pmatrix} \langle \hat{X}_1^2 \rangle & \langle \hat{X}_1\hat{P}_1 \rangle & \langle \hat{X}_1\hat{X}_2 \rangle & \cdots & \langle \hat{X}_1\hat{P}_{n-1} \rangle & \langle \hat{X}_1\hat{X}_n \rangle & \langle \hat{X}_1\hat{P}_n \rangle \\ \langle \hat{P}_1\hat{X}_1 \rangle & \langle \hat{P}_1^2 \rangle & \langle \hat{P}_1\hat{X}_2 \rangle & \cdots & \langle \hat{P}_1\hat{P}_{n-1} \rangle & \langle \hat{P}_1\hat{X}_n \rangle & \langle \hat{P}_1\hat{P}_n \rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle \hat{P}_n\hat{X}_1 \rangle & \langle \hat{P}_n\hat{P}_1 \rangle & \langle \hat{P}_n\hat{X}_2 \rangle & \cdots & \langle \hat{P}_n\hat{P}_{n-1} \rangle & \langle \hat{P}_n\hat{X}_n \rangle & \langle \hat{P}_n^2 \rangle \end{pmatrix} \tag{2.13}$$

Transformations symplectiques

Le groupe symplectique est un ensemble important, qui correspond aux transformations qui sont linéaires en les opérateurs annihilation et création, et qui conservent les relations de commutation. En particulier, leur caractère linéaire en $\hat{a}$ et $\hat{a}^\dagger$ assure qu'elles conservent le caractère gaussien d'un état ; le groupe symplectique est donc un sous-ensemble des transformations gaussiennes.

Les transformations symplectiques peuvent être décrites par une matrice symplectique S_y vérifiant

$$S_y \Omega S_y^T = \Omega, \text{ où } \Omega = \bigoplus_{i=1}^{n} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{2.14}$$

telle que

$$\vec{r}_{\mathrm{out}} = S_y \vec{r}_{\mathrm{in}} \tag{2.15}$$

Dans le formalisme des matrices de covariance, on obtient

$$\gamma_{\mathrm{out}} = \left\langle \vec{r}_{\mathrm{out}}\,\vec{r}_{\mathrm{out}}^{\,T} \right\rangle = \left\langle S_y\,\vec{r}_{\mathrm{in}}\,\vec{r}_{\mathrm{in}}^{\,T}\,S_y^T \right\rangle = S_y \gamma_{\mathrm{in}} S_y^T$$

Théorème de Williamson [35] Pour tout état gaussien de moyenne nulle, et donc totalement défini par sa matrice de covariance γ, il existe une transformation symplectique S qui le transforme en un produit tensoriel d'états thermiques, sous la forme :

$$\lambda = S_y \gamma S_y^T = \bigoplus_{k=1}^{N} \begin{pmatrix} \lambda_k & 0 \\ 0 & \lambda_k \end{pmatrix} \tag{2.16}$$

Qui plus est, dans cette expression, les n valeurs propres λ_k correspondent aux valeurs propres de la matrice $-\Omega\gamma\Omega\gamma$, et sont appelées *valeurs propres symplectiques* de γ.

Quelques transformations

Les transformations symplectiques sur les états gaussiens peuvent être avantageusement réalisées en utilisant le formalisme des matrices de covariances. Nous décrivons ici les transformations utilisées au cours de ce manuscrit, le formalisme symplectique étant décrit plus en détail dans [36].

Rotation de phase : Une rotation de phase θ d'un état monomode est décrite par la transformation :

$$\begin{pmatrix} x \\ p \end{pmatrix}_{out} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix}_{in} \tag{2.17}$$

Lame séparatrice : Une lame séparatrice de transmission η couplant les deux modes d'un état bimode est décrite par la transformation :

$$\begin{pmatrix} x_1 \\ p_1 \\ x_2 \\ p_2 \end{pmatrix}_{out} = \begin{pmatrix} \sqrt{\eta} & 0 & \sqrt{1-\eta} & 0 \\ 0 & \sqrt{\eta} & 0 & \sqrt{1-\eta} \\ -\sqrt{1-\eta} & 0 & \sqrt{\eta} & 0 \\ 0 & -\sqrt{1-\eta} & 0 & \sqrt{\eta} \end{pmatrix} \begin{pmatrix} x_1 \\ p_1 \\ x_2 \\ p_2 \end{pmatrix}_{in} \tag{2.18}$$

Amplification paramétrique optique : Un amplificateur paramétrique optique dépendant de la phase, qui comprime le bruit sur l'une des quadratures d'un état monomode tout en amplifiant celui de la quadrature orthogonale, est décrit par la transformation :

$$\begin{pmatrix} x \\ p \end{pmatrix}_{out} = \begin{pmatrix} e^{-s} & 0 \\ 0 & e^{s} \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix}_{in} \tag{2.19}$$

Un amplificateur paramétrique optique indépendant de la phase, qui amplifie les deux quadratures du mode 1 tout en les couplant avec un mode de bruit 2, est décrit par la transformation :

$$\begin{pmatrix} x_1 \\ p_1 \\ x_2 \\ p_2 \end{pmatrix}_{out} = \begin{pmatrix} \sqrt{g} & 0 & \sqrt{g-1} & 0 \\ 0 & \sqrt{g} & 0 & -\sqrt{g-1} \\ \sqrt{g-1} & 0 & \sqrt{g} & 0 \\ 0 & -\sqrt{g-1} & 0 & \sqrt{g} \end{pmatrix} \begin{pmatrix} x_1 \\ p_1 \\ x_2 \\ p_2 \end{pmatrix}_{in} \tag{2.20}$$

Mesures projectives

Le résultat d'une mesure projective M sur un état ρ est donné de façon générale par $\mathrm{Tr}[\rho M]$. Dans le cas d'une mesure projective de la valeur des quadratures d'un état gaussien, le formalisme symplectique peut être encore utilisé pour simplifier les calculs.

Détection homodyne Considérons une détection homodyne, dont le principe est détaillé dans la section 2.2.2, qui effectue par exemple une mesure sur la quadrature $\hat{X}_B$ d'un état gaussien bimode, de matrice de covariance

$$\gamma_{AB} = \begin{pmatrix} \gamma_A & \sigma_{AB} \\ \sigma_{AB}^T & \gamma_B \end{pmatrix} \tag{2.21}$$

Une mesure homodyne correspond à une projection sur un état gaussien infiniment comprimé [37], et l'expression de la matrice de covariance après la projection s'écrit alors [38] :

$$\gamma_A^{\text{out}} = \gamma_A - \sigma_{AB}(X\gamma_B X)^{\text{MP}}\sigma_{AB}^T \tag{2.22}$$

avec $X = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ et où MP représente le pseudo-inverse de Moore-Penrose [2].

Détection hétérodyne De même, la détection hétérodyne, au sens que nous lui donnons dans ce manuscrit, est une double mesure homodyne, qui détermine de manière simultanée la valeur des quadratures $\hat{X}_B$ et $\hat{P}_B$. L'état projeté s'écrit alors :

$$\gamma_A^{\text{out}} = \gamma_A - \sigma_{AB}(\gamma_B + \mathbb{I}_2)^{-1}\sigma_{AB}^T \tag{2.23}$$

2.2 Cryptographie quantique avec des variables continues

2.2.1 Protocoles de cryptographie quantique à variables continues

A partir de 1999, des articles proposent de nouveaux protocoles de cryptographie quantique utilisant un codage de l'information sur les quadratures d'états gaussiens de la lumière [39, 40, 41]. Par opposition aux protocoles à variables discrètes, leur originalité est l'utilisation d'une détection cohérente pour effectuer la mesure chez Bob. Les premiers protocoles de cryptographie quantique à variables continues proposent l'utilisation d'états comprimés en quadrature comme support de l'information. Néanmoins, ces états, délicats à produire et à transmettre, sont assez peu adaptés à des

2. Le pseudo-inverse de Moore-Penrose est une généralisation de l'inverse, valide pour des matrices non-inversibles. Formellement, le pseudo-inverse de M est la seule matrice M^{MP} telle que $MM^{\text{MP}}M = M$, $M^{\text{MP}}MM^{\text{MP}} = M^{\text{MP}}$, $(MM^{\text{MP}})^* = MM^{\text{MP}}$ et $(M^{\text{MP}}M)^* = M^{\text{MP}}M$.
La plupart des logiciels de calcul formel possèdent une implémentation de cette fonction ; pour le cas simple où la matrice est diagonale, calculer le pseudo-inverse revient à inverser tous les termes non-nuls de la diagonale.

applications dans lesquelles les distances mises en jeu sont de l'ordre de 10 à 100 km. En effet, les états comprimés s'intriquent avec l'environnement, ce qui fait disparaître leur caractère comprimé au cours de la propagation, à la différence des états cohérents qui sont des états pointeurs de la lumière (et dont le bruit est donc indépendant des pertes).

Protocole GG02

Le protocole que nous étudions ici a été choisi à la fois pour des raisons historiques et pratiques. Il s'agit du protocole proposé par Grosshans et Grangier dans [1, 2] en 2002, qui met en jeu des états cohérents et une détection homodyne. Il est plus facile de travailler avec des états cohérents qu'avec des états comprimés, car ils peuvent être produits aisément avec un laser standard, et ne sont pas soumis à la décohérence ; en outre, les performances des deux types de protocoles sont assez similaires. Le choix d'une détection homodyne, lui aussi, tient d'un certain pragmatisme : les performances d'un système à détection hétérodyne (par exemple dans [42]) sont presque les mêmes que celles d'un système à détection homodyne, mais deux détecteurs sont nécessaires (au lieu d'un) pour le cas hétérodyne. Nous décrivons ci-dessous le détail du protocole GG02.

- Alice génère des impulsions cohérentes limitées par le bruit de photon, et choisit, pour chaque impulsion et de manière aléatoire, deux valeurs de quadratures x_A et p_A parmi une distribution gaussienne centrée sur 0 et de variance $V_A N_0$.
- Pour coder l'information, elle déplace ensuite chaque impulsion dans l'espace des phases, de manière à ce que l'état cohérent soit centré sur $\langle \hat{X} \rangle = x_A$ et $\langle \hat{P} \rangle = p_A$.
- Elle envoie les impulsions à Bob à travers un canal quantique : fibre standard ou espace libre.
- Bob choisit, pour chaque impulsion et de manière aléatoire, une quadrature à mesurer : $\hat{X}_B$ ou $\hat{P}_B$.
- Enfin, Bob mesure pour chaque impulsion, grâce à une détection homodyne, la valeur de la quadrature choisie, x_B ou p_B.
- Une fois la mesure effectuée, Bob annonce à Alice, par un canal authentifié, son choix de quadrature de mesure. Alice peut alors oublier l'autre quadrature.

Après ces étapes, Alice et Bob partagent des données corrélées, puisque la mesure de Bob est affectée (au minimum) par le bruit de photon. En outre, leurs données ont été *a priori* espionnées par Eve.

Il reste donc tout d'abord à évaluer la qualité de la transmission, pour déterminer la quantité de secret restant dans les données : c'est l'objet de l'évaluation des paramètres et des calculs de sécurité.

S'il reste un secret, Alice et Bob doivent ensuite l'extraire, le plus efficacement possible, pour pouvoir enfin générer une clé binaire, parfaitement secrète. Ces étapes de traitement classique des données incluent en particulier un algorithme de réconciliation, consistant à extraire une chaîne binaire identique (mais seulement partiellement secrète) à partir des données corrélées d'Alice et Bob, puis un algorithme d'amplification de confidentialité, qui transforme cette chaîne en une clé parfaitement secrète.

Pour l'étape de réconciliation, deux choix sont possibles. Si les données d'Alice servent de base à l'élaboration de la clé, c'est-à-dire que Bob corrige ses erreurs pour retrouver les valeurs d'Alice, la réconciliation est dite *directe* ; si ce sont les données

de Bob qui servent de référence, elle est dite *inverse*.

2.2.2 Pourquoi travailler avec des variables continues ?

QKD à variables discrètes : des détecteurs contraignants

Dans tous les protocoles de cryptographie quantique proposés jusqu'à la fin des années 1990, la détection des photons s'effectue grâce à un détecteur (ou un compteur) de photons uniques. Ces détecteurs sont généralement des photodiodes à avalanche très sensibles, avec un gain très élevé. Les technologies silicium permettent d'atteindre, dans le cas de la détection de photons dans le visible, des efficacités de l'ordre de 50 % avec peu de bruit, tout en conservant des prix assez réduits.

Dans le cas de la cryptographie quantique, ces détecteurs présentent néanmoins un certain nombre d'inconvénients pratiques. Tout d'abord, la plupart des systèmes de cryptographie quantique fonctionnent à longueur d'onde télécom (autour de 1 550 nm), ce qui correspond à la bande C d'absorption minimale des fibres optiques. Or, les détecteurs silicium ne fonctionnement efficacement que jusqu'environ $\lambda = 1\,000$ nm ; il a donc fallu développer des détecteurs spécifiques pour les applications télécom.

Les technologies à base d'arséniure de gallium (InGaAs) ont été massivement utilisées, mais elles présentent des performances nettement inférieures aux photodiodes silicium : typiquement 15 % d'efficacité, mais beaucoup de bruit d'obscurité (typ. 3 000 coups/s) et un temps mort après chaque détection dû aux *afterpulses*, pendant lequel aucune détection n'est possible, ce qui limite la fréquence de détection du détecteur à typiquement 100 kHz. En outre, le prix d'un détecteur de photons unique InGaAs utilisable pour la cryptographie quantique est de l'ordre de 10 000 euros.

Néanmoins, une technologie à base de NbN supraconducteur refroidi à l'hélium se développe actuellement : elle permet d'obtenir un bruit d'obscurité très faible (typiquement 10 coups par seconde), ce qui est essentiel pour la cryptographie quantique. Par ailleurs, ce type de détecteur présente jusque 10 % d'efficacité, et un temps de montée de l'ordre de 100 ps.

La détection homodyne : efficace et économique

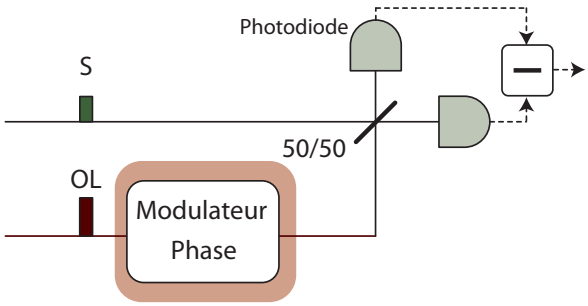


Figure 2.4 : *Schéma de principe d'une détection homodyne*

La détection homodyne est un système interférentiel, dans lequel le signal à mesurer S se couple avec un oscillateur local intense OL dans une lame séparatrice 50/50 (figure

2.4). La valeur de l'intensité optique des deux sorties + et − de la lame est alors mesurée avec deux photodiodes standard. La différence $\Delta\hat{I}$ des deux photocourants peut donc s'écrire

$$\begin{aligned}\Delta\hat{I} &= \hat{n}_+ - \hat{n}_- = \hat{a}_+^\dagger \hat{a}_+ - \hat{a}_-^\dagger \hat{a}_- && (2.24)\\ &= (\hat{a}_S^\dagger + \hat{a}_{OL}^\dagger)(\hat{a}_S + \hat{a}_{OL}) - (\hat{a}_S^\dagger - \hat{a}_{OL}^\dagger)(\hat{a}_S - \hat{a}_{OL}) && (2.25)\\ &= 2\,\hat{a}_{OL}^\dagger \hat{a}_S + 2\,\hat{a}_S^\dagger \hat{a}_{OL} && (2.26)\end{aligned}$$

Si l'on suppose que l'oscillateur local est suffisamment intense pour que ses fluctuations soient négligeables (hypothèse de champ classique), on peut écrire $\hat{a}_{OL} = \sqrt{I_{OL}}e^{i\varphi}$:

$$\Delta\hat{I} = 2\sqrt{I_{OL}}(\hat{a}_S e^{-i\varphi} + \hat{a}_S^\dagger e^{i\varphi}) = \frac{2\sqrt{I_{OL}}}{\sqrt{N_0}}\hat{X}_{S/OL} \qquad (2.27)$$

où $\hat{X}_{S/OL}$ correspond à la quadrature du signal en phase avec l'oscillateur local. En contrôlant la phase de l'oscillateur local, la détection homodyne permet donc de mesurer n'importe quelle quadrature du signal, le signal étant amplifié par un facteur égal à l'amplitude de l'oscillateur local.

L'intérêt de ce système est de ne pas nécessiter de composants spécifiques (voir section 5.5 pour l'implémentation). On peut donc mettre en œuvre une détection homodyne limitée au bruit de photon avec des composants télécom commerciaux standards.

3 Notions de théorie de l'information

> It is a very sad thing that nowadays there is so little useless information.
>
> *A Few Maxims for the Instruction of the Over-Educated*, Oscar Wilde

3.1 L'information : de nombreuses acceptions, parfois contradictoires

Dans le langage courant, la notion d'*information* est omniprésente, et assez difficile à définir. Comme nous le verrons dans ce chapitre, elle a été formalisée dans le cadre de la théorie de l'information et utilisée pour les preuves de sécurité de cryptographie quantique ; or, le sens formel qu'elle y prend n'est pas exactement celui qu'on lui prête couramment. Nous essayons dans ce chapitre de classifier les différents sens de « l'information », des significations courantes aux notions formelles.

3.1.1 L'information comme enregistrement

Nous recevons en permanence de l'information, en tant qu'êtres vivants, par les sensations que nous éprouvons, qu'elles viennent de nous-même (faim, soif, fatigue) ou de l'extérieur (stimulation visuelle, auditive, olfactive). Les technologies que nous avons conçues, elles aussi, reçoivent de l'information, par les différents capteurs qui les composent, et elles les transforment en général en données.

Cette information est brute, non interprétée. Elle n'existe qu'en tant qu'*enregistrement*, et est générique, indépendante de toute signification, bien que demandant un substrat physique pour nous parvenir.

3.1.2 L'information comme augmentation de savoir

Considérons deux livres de 300 pages : le premier contient, sur chaque page, la même recette de mousse au chocolat ; le deuxième, en revanche, contient 300 recettes différentes. Bien que ces livres contiennent le même nombre de mots, donc la même quantité d'information-enregistrement, il est clair qu'on dira du deuxième qu'il contient « plus d'information ».

Cette « information utile » est donc à distinguer de la première, et se conçoit comme *vecteur d'augmentation des savoirs, des connaissances*. Dans ce cadre, toute redondance d'un message n'apporte pas plus d'information.

Par ailleurs, pour acquérir de l'information-savoir, il est nécessaire d'interpréter l'information-enregistrement. L'information utile d'un message est dépendante du contexte dans lequel nous recevons le message, mais aussi de l'ensemble de nos connaissances *a priori* : un message en japonais, même passionnant, reste inutile pour qui ne le comprend pas, et ne lui apporte donc pas d'information.

3.1.3 Lien entre ces deux informations

Une source d'information qui ne transmet que peu de savoir tout en émettant beaucoup de données est dite redondante. Exemple : une phrase du type « Le code est 36 ; en binaire 100100 ; le produit de 3 et 12 ; le carré de 6 » est lourde, mais très robuste aux pertes : si je n'entends qu'une partie sur les quatre, je reçois toute l'information qui y est contenue.

Inversement, une source d'information qui transmet beaucoup de savoir avec très peu de données est économe en bande passante, mais très fragile. Transmettre la phrase « Code binaire : 100100 » est dangereux : si je rate l'un des bits du code, ou même si je rate le mot *binaire*, j'interprète mal l'information.

Que dire de la phrase « Nous sommes en juillet, c'est l'été » ? Elle contient deux informations-enregistrements, mais la deuxième n'apporte pas vraiment d'information-savoir : *a priori*, une fois que je sais que nous sommes en juillet, je me doute que c'est l'été, et je n'apprends donc rien quand je lis la deuxième partie. Mais un Australien qui lit cette phrase (en anglais) comprendra que le rédacteur vient de l'hémisphère nord, et obtient donc deux informations. On voit donc que la notion d'information peut être toute relative au contexte dans lequel elle est obtenue.

3.2 Quantifier l'information

3.2.1 Le bit, *binary digit*

L'information, dans sa première acception de quantité de données enregistrées ou transmises, est aisée à définir, puisqu'elle est liée à la notion de codage. On considère un message M, composé de m lettres choisies dans un alphabet de N lettres. La quantité d'information nécessaire pour coder le message correspond au nombre de lettres binaires qui sont utilisées à cet effet. La *lettre binaire*, ou *binary digit*, a donné le mot-valise *bit*.

Puisqu'il existe N lettres dans l'alphabet, le nombre de bits nécessaires pour définir de manière non ambiguë une lettre est $\log_2 N$, et donc la taille binaire du message s'écrit

$$I(M) = m \log_2 N \tag{3.1}$$

Ce *bit* correspond à l'unité couramment utilisée en informatique : on dira qu'un fichier fait 50 kilobit (kb), ou qu'une clé USB peut contenir 1 gigabit (Gb), c'est à dire que l'on peut enregistrer sur celle-ci un message binaire de 10^9 lettres au maximum.

3.2.2 Le bit, *binary unit*

La *théorie de l'information* est le domaine qui formalise le deuxième type d'information, l'information utile, celle qui fournit une nouvelle connaissance. Elle définit une nouvelle unité, l'unité binaire, ou *binary unit*, qui s'abbrévie en... *bit*. Cette conjonction n'est pas vraiment fortuite, puisque les deux notions de bits sont intimement liées, mais quelque peu malheureuse : elle entraîne souvent des confusions, par exemple entre le nombre de bits (digits) de données qui transitent sur un canal, et le nombre de bits (units) d'information apportés par ces données. Mais l'usage est plus qu'ancré, et peu d'auteurs distinguent explicitement les « bigits » des « binits » dans leurs textes.

Nous décrivons maintenant les principales notions intervenant dans les calculs d'information.

3.2.3 Information d'une variable classique

L'information propre

Une question est posée : à cette question, plus la probabilité d'une réponse donnée est faible, plus l'information que cette réponse nous apporte est importante. Inversement, si une réponse est très probable, on s'attend à ce qu'elle ait lieu, et donc le fait qu'elle ait en effet lieu n'est pas très informatif.

Considérons donc une variable aléatoire X, pouvant donner N résultats x_i avec une probabilité p_i. L'*information propre* de x_i est définie comme :

$$I(x_i) = \log_2 \frac{1}{p_i} = -\log_2 p_i \tag{3.2}$$

L'information propre est donc d'autant plus importante que le résultat est inattendu ; pour cette raison, elle a parfois été appelée *quantité de surprise*.

Exemple : On considère une pièce « pipée » : elle tombe sur pile 7 fois sur 8. Les informations propres sont donc

$$I(\text{pile}) = -\log_2 \frac{7}{8} \approx \underbrace{0{,}19 \text{ bit}}_{\text{peu surprenant}} \qquad I(\text{face}) = -\log_2 \frac{1}{8} = \underbrace{3 \text{ bit}}_{\text{surprenant}}$$

L'entropie

L'entropie est une notion centrale de la théorie de l'information. Elle correspond à l'incertitude moyenne sur le résultat du tirage d'une variable aléatoire, ou de manière

équivalente à l'information moyenne fournie par un tirage d'une variable aléatoire. Elle est donc définie comme :

$$H(X) = \sum_i p_i\, I(x_i) = -\sum_i p_i \log_2 p_i \tag{3.3}$$

On montre facilement que l'entropie est maximale quand les N probabilités p_i sont toutes égales à $\frac{1}{N}$, auquel cas $H(X) = \log_2 N$. Ceci est assez intuitif, puisqu'il devient alors impossible de prédire « sur quel résultat le tirage a le plus de chances de tomber » : l'incertitude est maximale.

Exemple : Reprenons notre pièce pipée ; l'entropie du tirage de la pièce vaut :

$$H(X_{\text{pipée}}) = \frac{7}{8} \times 0{,}19 + \frac{1}{8} \times 3 = 0{,}54 \text{ bit}$$

Considérons maintenant une pièce équilibrée, donc des probabilités 0,5 et un résultat imprédictible :

$$H(X_{\text{équilibrée}}) \;=\; -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1 \text{ bit}$$

A la limite, si la pièce ne tombe jamais sur face, il n'y a aucune incertitude sur le résultat, et l'entropie est nulle. Ce résultat donne une autre définition du bit : *un bit est la quantité d'information fournie par la réalisation d'une option dans une alternative équiprobable.*

Entropie d'une distribution gaussienne Dans son article fondateur [8], Claude Shannon a montré deux résultats importants :

1. À variance fixée, la distribution de probabilité qui maximise l'entropie est la distribution gaussienne.
2. L'entropie d'une variable aléatoire gaussienne X_G s'écrit

$$H(X_G) = \frac{1}{2}\log_2 \left\langle X_G^2 \right\rangle + C \tag{3.4}$$

où C est une constante, dépendant du choix des unités.

Pourquoi ce logarithme ?

Le choix du logarithme dans les formules d'information n'est pas arbitraire : les quantités d'informations sont construites de façon que la mesure soit additive. Or, puisque la probabilité de deux événements indépendants x et y s'écrit $\Pr(x,y) = \Pr(x)\Pr(y)$, il faut choisir $I(x) \propto \log \Pr(x)$ pour avoir $I(x,y) = I(x) + I(y)$.

3.2.4 Informations de deux variables classiques

Information mutuelle propre

On considère maintenant deux variables aléatoires X et Y, *a priori* dépendantes l'une de l'autre, pouvant prendre des valeurs x_i et y_j avec des probabilités $\Pr(x_i)$ et $\Pr(y_j)$. On note $\Pr(x_i,y_j)$ la probabilité d'occurrence conjointe de x_i et y_i, et

$\Pr(x_i|y_j) = \Pr(x_i,y_j)/\Pr(y_j)$ la probabilité conditionnelle de x_i sachant y_j, c'est-à-dire la probabilité que x_i se produise sachant que y_j s'est produit.

L'information mutuelle propre est alors définie comme

$$I(x_i : y_j) = \log_2 \frac{\Pr(x_i,y_j)}{\Pr(x_i)\Pr(y_j)} = \log_2 \frac{\Pr(x_i|y_j)}{\Pr(x_i)} \tag{3.5}$$

$I(x_i : y_j)$ est une mesure de corrélation entre les deux résultats :

- $I(x_i : y_j) = 0$ implique $\Pr(x_i,y_j) = \Pr(x_i)\Pr(y_j)$, c'est-à-dire que les événements sont statistiquement indépendants,
- $I(x_i : y_j) < 0$: si x_i se produit, y_j devient moins probable
- $I(x_i : y_j) > 0$: si x_i se produit, y_j devient plus probable

À noter également, la présence des deux-points dans $I(x_i : y_j)$, et non d'une virgule. En effet, la virgule, en théorie des probabilités, dénote une intersection : $I(x_i,y_j)$ correspond à l'information propre de l'événement « x_i et y_j », et non à leur information *mutuelle* propre.

Information mutuelle moyenne

Par extension, l'information mutuelle moyenne correspond à la corrélation moyenne entre les variables aléatoires X et Y. Elle est notée $I(X : Y)$, ou plus simplement I_{XY}, et définie comme

$$I(X : Y) = \sum_{i,j} \Pr(x_i,y_j) I(x_i : y_j) \tag{3.6}$$

L'information mutuelle est donc symétrique : $I(X : Y) = I(Y : X) = I_{XY}$. Intuitivement, elle correspond à ce qu'on apprend sur X en connaissant Y. Ceci est plus clair en la réécrivant en fonction des entropies mettant en jeu X et Y :

$$\underbrace{I(X : Y)}_{\text{ce que } Y \text{ dit sur } X} = H(X) + H(Y) - H(X,Y) \tag{3.7}$$

$$= \underbrace{H(X)}_{\text{ce qu'il y a à dire sur } X} - \underbrace{H(X|Y)}_{\text{ce que } Y \text{ ne dit pas sur } X} \tag{3.8}$$

Entropie conditionnelle : $H(X|Y)$ se lit « entropie de X *sachant* Y ». Elle est définie comme en (3.3), à partir de la probabilité conditionnelle $\Pr(x_i|y_j)$, qui est la probabilité d'occurrence de x_i sachant que y_j s'est produit. Il est à noter que l'entropie conditionnelle $H(X|Y)$ est toujours positive ou nulle quand X et Y sont des variables aléatoires classiques, le cas limite $H(X|Y) = 0$ étant atteint quant X et Y sont parfaitement corrélées classiquement ($X = Y$).

3.2.5 Information de variables quantiques

On se place maintenant dans le cas d'un état quantique, défini par sa matrice densité ρ. Les résultats des mesures sur un état quantique sont par nature incertains, mais la probabilité d'obtenir un résultat donné, elle, est parfaitement déterminée. Il est donc possible de définir des entropies et des informations associées à cette incertitude. La notion d'entropie d'un état quantique a été introduite dans [43], et la thèse de Raúl

García-Patrón [36] offre une synthèse de l'état actuel des travaux sur celle-ci. La suite de ce chapitre est largement inspirée de ces travaux.

Entropie de Von Neumann

L'entropie de Von Neumann [44] d'un état quelconque ρ est une généralisation de l'entropie classique H, définie par analogie avec l'information classique. En effet, on peut toujours décomposer ρ dans sa base d'états propres $\{|i\rangle\langle i|\}$:

$$\rho = \sum_i \lambda_i \, |i\rangle\langle i| \tag{3.9}$$

et on peut ainsi définir la variable aléatoire λ, qui correspond à la probabilité de mesure de ρ dans l'état $|i\rangle\langle i|$. L'entropie classique $H(\lambda)$ s'écrit alors

$$H(\lambda) = -\sum_i \lambda_i \log_2 \lambda_i = -\mathrm{Tr}(\rho \log_2 \rho) \tag{3.10}$$

On définit donc l'entropie de Von Neumann de manière générale comme

$$\boxed{S(\rho) = -\mathrm{Tr}(\rho \log_2 \rho)} \tag{3.11}$$

Cas gaussien Pour un état gaussien à n modes ρ, caractérisé par sa matrice de covariance γ, on peut calculer l'entropie de Von Neumann à partir des valeurs propres symplectiques définies en 2.1.3 :

- Puisque (1) l'entropie de Von Neumann est définie grâce à une trace, que (2) la trace est invariante par des transformations unitaires, et que (3) l'opérateur déplacement est une transformation unitaire, l'entropie $S(\rho)$ est égale à l'entropie $S(\rho_0)$, où ρ_0 est l'état ρ déplacé de façon à le centrer sur zéro.
- De la même façon, les transformations symplectiques sont un sous-ensemble des transformations unitaires, et l'expression de $S(\rho_0)$ doit être égale à $S(\rho_\lambda)$, où ρ_λ est l'état de matrice de covariance $\lambda = S_y \gamma_0 S_y^T$ apparaissant dans le théorème de Williamson (2.16), c'est-à-dire le produit de n états thermiques. On peut donc écrire :

  $$S(\rho) = \sum_{k=1}^{n} S(\rho_{\lambda_k}) \tag{3.12}$$

  où ρ_{λ_k} est un état thermique de matrice de covariance $\lambda_k \mathbb{I}$, dont l'entropie de von Neumann s'écrit $S(\rho_{\lambda_k}) = G(\frac{\lambda_k - 1}{2})$, avec $G(x) = (x+1)\log_2(x+1) - x\log_2(x)$. Cette expression est assez non-triviale à démontrer, le lecteur se référera à [36] pour plus de détails.

Finalement, l'entropie d'un état gaussien s'écrit

$$S(\rho) = \sum_i G\left(\frac{\lambda_i - 1}{2}\right) \tag{3.13}$$

où les λ_i sont les n valeurs propres symplectiques de γ, et $G(x) = (x+1)\log_2(x+1) - x\log_2 x$.

Information mutuelle quantique

Toujours par analogie, on définit l'information mutuelle entre deux parties A et B d'un état quantique bipartite ρ_{AB} comme

$$S(A:B) = S(A) + S(B) - S(A,B) \tag{3.14}$$

Cas pur Dans le cas d'un état pur bimode ψ_{AB}, on peut montrer que $S(A,B) = 0$, ainsi que $S(A) = S(B)$ [1].

Information conditionnelle. Contrairement à sa version classique, l'information conditionnelle $S(A|B) = S(A,B) - S(B)$ d'un état quantique ρ_{AB} peut être négative. Par exemple, dans le cas d'un état EPR $|\psi\rangle_{\mathrm{AB}} = \frac{1}{2}(|0\rangle_{\mathrm{A}}|0\rangle_{\mathrm{B}} + |1\rangle_{\mathrm{A}}|1\rangle_{\mathrm{B}})$, on trouve $S(A|B) = -1$. Des variables quantiques A et B peuvent donc être mieux corrélées que ce qui est autorisé par l'information classique, ce qui est une particularité fondamentale de la physique quantique appelée *intrication quantique*.

Examples

	État quantique	$S(A,B)$	$S(A:B)$	$S(A\|B)$
(1)	$\rho \otimes \sigma$	$S(A) + S(B)$	0	$S(A)$
(2)	$\sum_i^N \|i\rangle\langle i\| \otimes \|i\rangle\langle i\|$	$\log_2 N$	$\log_2 N$	0
(3)	$\sum_i^N \|i\rangle \otimes \|i\rangle$	0	$2\log_2 N$	$\boxed{-\log_2 N}$

(1) : état parfaitement décorrélé, (2) : état parfaitement corrélé classiquement
(3) : état parfaitement intriqué

3.3 Information et variables continues

Dans le cadre de la cryptographie quantique à variables continues, nous avons naturellement accès à plusieurs variables aléatoires : X_A, P_A, $\hat{X}_B$, $\hat{P}_B$, qui correspondent aux distributions de quadrature accessibles à Alice et Bob, et $\hat{X}_E$ et $\hat{P}_E$, les quadratures accessibles à Eve.

3.3.1 Modèle du canal gaussien

Le canal gaussien est une modélisation des perturbations subies par l'état cohérent au cours de sa transmission entre Alice et Bob. Il a pour but d'exprimer X_B et P_B en fonction de X_A et P_A [2], et est l'élément de base des calculs de sécurité du protocole.

On modélise les perturbations subies par l'état par une atténuation en intensité $G = g^2$, et un bruit ajouté X_{ch} défini à l'entrée du canal, supposé indépendant de X_A. En cryptographie quantique, nous cherchons systématiquement à majorer l'information

1. Ceci provient directement de l'inégalité d'Araki-Lieb, selon laquelle $S(A,B) \geqslant |S(A) - S(B)|$. Puisque $S(A,B) = 0$, les deux termes $S(A)$ et $S(B)$ doivent être égaux.

2. Jusqu'ici, nous avons noté $\hat{\cdot}$ les variables quantiques, pour les distinguer des variables classiques. Dans la suite du manuscrit, nous omettrons cette notation pour alléger les formules, en gardant à l'esprit que seules X_A et P_A sont des variables classiques.

disponible à l'espion. Puisque la distribution gaussienne est la distribution qui a la plus grande entropie à variance donnée, c'est celle qui apportera le plus d'information à l'espion : l'optimalité des états gaussiens a été démontrée dans [45, 46]. Nous supposons donc que X_{ch} est un bruit gaussien, ce qui avantage systématiquement Eve.

En écrivant l'état envoyé par Alice comme $X_{\text{out}} = X_A + X_0$, où X_0 correspond à la quadrature du bruit de photon (gaussien), on déduit

$$X_B = g(X_A + X_0 + X_{\text{ch}}) = g(X_A + X_N) \tag{3.15}$$

où gX_N est le bruit total (modélisé comme gaussien) sur la mesure de Bob. Rappelons que dans ce modèle, le bruit X_N est indépendant de X_A, ce qui permettra dans les calculs qui suivent d'annuler les termes $\langle X_A X_N \rangle$.

3.3.2 Information mutuelle classique

Pour calculer l'information mutuelle I_{AB} entre Alice et Bob, on décompose X_B suivant (3.15). Puisque le bruit et le message sont supposés indépendants, on utilise l'additivité de l'entropie entre X_A et X_N :

$$\begin{aligned}
I_{AB} &= H(X_A) + H(X_B) - H(X_A,X_B) && (3.16)\\
&= H(X_A) + H(X_B) - \cancel{H(X_A,gX_A)} - H(X_A,gX_N) && (3.17)\\
&= \cancel{H(X_A)} + H(X_B) - \cancel{H(X_A)} - H(gX_N) && (3.18)\\
&= \frac{1}{2}\log_2 \frac{\langle X_B^2 \rangle}{G\langle X_N^2 \rangle} && (3.19)
\end{aligned}$$

puisque les distributions de X_B et X_N sont supposées gaussiennes dans notre modèle. On voit en particulier que, contrairement à l'entropie, l'information mutuelle n'est pas définie à une constante près.

3.3.3 Formule de Shannon

Dans le cas de distributions gaussiennes, la formule de Shannon permet d'établir le lien entre information mutuelle classique et rapport signal à bruit. Si l'on définit le rapport signal à bruit comme

$$\text{SNR} = \frac{\langle X_A^2 \rangle}{\langle X_N^2 \rangle} \tag{3.20}$$

on peut réécrire (3.19) sous la forme

$$\begin{aligned}
I_{AB} &= \frac{1}{2}\log_2 \frac{G\langle (X_A + X_N)^2 \rangle}{G\langle X_N^2 \rangle} && (3.21)\\
&= \frac{1}{2}\log_2 \frac{\langle X_A^2 \rangle + \cancel{\langle X_A X_N \rangle} + \langle X_N^2 \rangle}{\langle X_N^2 \rangle} && (3.22)\\
&= \frac{1}{2}\log_2 \left(1 + \frac{\langle X_A^2 \rangle}{\langle X_N^2 \rangle}\right) && (3.23)
\end{aligned}$$

ce qui donne

$$\boxed{I_{AB} = \frac{1}{2}\log_2 (1 + \text{SNR})} \tag{3.24}$$

Formule de Shannon

En mettant en relation information et rapport signal à bruit, la formule de Shannon permet donc d'établir un pont entre théorie de l'information et résultats physiques, c'est-à-dire entre les caractéristiques de la mesure de Bob et la quantité d'information que ces mesures lui permettent de partager avec Alice.

Variance conditionnelle, information conditionnelle

La variance conditionnelle de B connaissant A est définie comme :

$$\begin{aligned} V_{B|A} &\hat{=} \min_{\alpha} \left\langle (X_B - \alpha X_A)^2 \right\rangle && (3.25)\\ &= \min_{\alpha} \left\langle ((g-\alpha)X_A + gX_N)^2 \right\rangle && (3.26)\\ &= \min_{\alpha} \left[(g-\alpha)^2 \left\langle X_A^2 \right\rangle + g(g-\alpha)\cancel{\langle X_A X_N \rangle} + g^2 \left\langle X_N^2 \right\rangle \right] && (3.27)\\ V_{B|A} &= g^2 \left\langle X_N^2 \right\rangle \qquad \text{(le minimum étant atteint pour } \alpha = g) && (3.28) \end{aligned}$$

A partir de cette expression, nous pouvons réécrire la formule de Shannon sous la forme

$$\boxed{I_{AB} = \frac{1}{2} \log_2 \frac{\langle X_B^2 \rangle}{V_{B|A}}} \tag{3.29}$$

Puisque l'information mutuelle s'écrit aussi

$$I_{AB} = H(X_B) - H(X_B|X_A) \tag{3.30}$$

nous pouvons identifier tous les termes deux à deux, ce qui nous permet de mettre en évidence une propriété des variables aléatoires gaussiennes, pour lesquelles $H(X_B|X_A) = V_{B|A}$.

Enfin, notons que nous pouvons également retrouver une expression classique de la variance conditionnelle en écrivant $\alpha_{\min} = g = \frac{\langle X_A X_B \rangle}{\langle X_A^2 \rangle}$:

$$\begin{aligned} V_{B|A} &= \min_{\alpha} \left\langle (X_B - \alpha X_A)^2 \right\rangle && (3.31)\\ &= \left\langle \left(X_B - \frac{\langle X_A X_B \rangle}{\langle X_A^2 \rangle} X_A \right)^2 \right\rangle && (3.32)\\ \text{c'est-à-dire} \quad V_{B|A} &= \left\langle X_B^2 \right\rangle - \frac{\langle X_A X_B \rangle^2}{\langle X_A^2 \rangle} && (3.33) \end{aligned}$$

3.3.4 Théorème de Holevo

La formule (ou borne) de Holevo [47] permet de déterminer une borne supérieure à l'entropie mutuelle quantique d'un état bipartite ρ_{AB}. Si l'on considère une mesure de Bob, pouvant donner un ensemble de résultats $\{x_i\}$ avec des probabilités p_i, l'information mutuelle est bornée par

$$\boxed{S(A:B) \leqslant S(\rho_A) - \sum_i p_i \, S\left(\rho_A^{x_i}\right)} \tag{3.34}$$

Borne de Holevo

où $\rho_A^{x_i}$ correspond à l'état d'Alice conditionné au résultat x_i sur la mesure de Bob.

Il a été démontré dans [48] que cette borne de Holevo pouvait être utilisée pour borner l'information acquise par l'espion dans un protocole de cryptographie quantique, et nous l'utiliserons donc dans le preuves de sécurité présentées dans le chapitre suivant.

4 Preuves de sécurité

No one can build his security upon the nobleness of another person.

Willa Cather

Dans ce chapitre, nous formalisons les preuves de sécurité relatives au protocole de cryptographie quantique que nous avons implémenté. Nous modélisons tout d'abord les pertes et bruits intervenant dans le système, et exposons les problématiques et hypothèses nécessaires pour les calculs de sécurité. Nous calculons ensuite l'information secrète disponible à Alice et Bob pour l'extraction d'une clé secrète, dans le cas d'une réconciliation inverse et dans le mode réaliste.

4.1 Modélisation des bruits et des pertes

4.1.1 Bruit de photon

Les états générés par Alice sont des états cohérents, dont les quadratures peuvent s'écrire

$$X_{\text{in}} = X_A + X_0 \qquad P_{\text{in}} = P_A + P_0 \tag{4.1}$$

où X_A/P_A sont les deux quadratures classiques sur lesquelles Alice a centré l'état, et X_0/P_0 sont les quadratures du bruit de photon. Exprimons donc d'abord la variance de la distribution en quadrature des états envoyés par Alice [1] :

$$\begin{aligned} V_{\text{in}} &= \left\langle X_{\text{in}}^2 \right\rangle && (4.2)\\ &= \underbrace{\left\langle X_A^2 \right\rangle}_{V_A N_0} + \underbrace{\cancel{\left\langle X_A X_0 + X_0 X_A \right\rangle}}_{\substack{\text{car le bruit de photon est} \\ \text{indépendant de la modulation}}} + \underbrace{\left\langle X_0^2 \right\rangle}_{N_0} && (4.3)\\ &= (V_A + 1)\, N_0 \quad \hat{=} \quad V\, N_0 && (4.4) \end{aligned}$$

1. La plupart des valeurs que nous définissons dans ce chapitre conservent la symétrie entre X et P. Sauf cas particulier, nous n'expliciterons donc que les expressions selon X.

4.1.2 Canal de transmission

Le canal de transmission est modélisé comme un canal gaussien ; les états y sont soumis à des pertes en intensité $T = t^2$ et à un bruit ajouté gaussien de variance εN_0, défini à l'entrée du canal. On définit alors le bruit total $\chi_{\text{line}} N_0$ ajouté par le canal [2] :

$$\chi_{\text{line}} = \underbrace{\frac{1-T}{T}}_{\substack{\text{bruit dû aux pertes} \\ \textit{ramenés à l'entrée du canal}}} + \underbrace{\varepsilon}_{\text{excès de bruit}} \tag{4.5}$$

4.1.3 Système de détection

Le système de détection peut présenter des imperfections, dues à l'efficacité quantique limitée des photodiodes, aux pertes dans les différents composants qui le composent, ou au bruit ajouté par l'électronique intégrée. De manière similaire au canal de transmission, nous modélisons le système de détection par un canal gaussien qui présente des pertes η et ajoute un bruit électronique $v_{\text{el}} N_0$ défini à la sortie du canal, c'est-à-dire au niveau des détecteurs.

Le bruit total $\chi_{\text{hom}} N_0$ ajouté par la détection homodyne s'écrit alors :

$$\chi_{\text{hom}} = \frac{1-\eta}{\eta} + \frac{v_{\text{el}}}{\eta} \tag{4.6}$$

4.1.4 Variances et bruits totaux

Avec cette modélisation, on exprime la variance des données de Bob, $V_B N_0$:

$$\begin{aligned} V_B &= \eta T (V_{\text{in}} + \chi_{\text{line}}) + \eta \chi_{\text{hom}} && (4.7) \\ &= \eta T \left[V_A + \varepsilon + \frac{1 + v_{\text{el}}}{\eta T} \right] && (4.8) \end{aligned}$$

Si l'on définit $G = \eta T$, on peut exprimer le bruit total $\chi_{\text{tot}} N_0$ ajouté dans le système :

$$\begin{aligned} \chi_{\text{tot}} &= \chi_{\text{line}} + \frac{\chi_{\text{hom}}}{T} && (4.9) \\ &= \underbrace{\frac{1-G}{G}}_{\substack{\text{bruit total dû aux pertes} \\ \textit{ramenés à l'entrée du canal}}} + \underbrace{\varepsilon + \frac{v_{\text{el}}}{G}}_{\text{bruit en excès total}} && (4.10) \end{aligned}$$

et donc

$$V_B = G(V + \chi_{\text{tot}}) \tag{4.11}$$

2. Formellement, $\chi_{\text{line}} N_0$ est *la variance* du bruit total ajouté par le canal. Cependant, dans les analyses de sécurité de ce chapitre, les bruits considérés sont gaussiens. Pour ne pas trop alourdir les phrases, nous parlerons donc souvent par métonymie de *la valeur d'un bruit* plutôt que de *la valeur de la variance de ce bruit.*

4.2 Sécurité du protocole

4.2.1 Problématique

Une fois la transmission quantique effectuée, Alice et Bob partagent des données corrélées, et ils cherchent à en extraire une clé secrète. On peut montrer que l'information accessible à l'espion doit être inférieure à l'information I_{AB}, partagée par Alice et Bob, pour que cette extraction soit possible [49].

Pour évaluer I_{AB}, Alice et Bob doivent évaluer les *paramètres caractérisant la transmission*, à l'aide d'un échantillon aléatoire des données. Une fois ces paramètres connus, il est possible de calculer l'information mutuelle I_{AB} entre Alice et Bob. Dans notre cas, il s'agit d'évaluer les différents bruits intervenant dans le système, ainsi que les niveaux de signal et de pertes. Ceci nous permet d'évaluer le rapport signal à bruit, qui est directement lié à I_{AB} d'après la formule de Shannon (3.24).

Reste donc à évaluer la quantité d'information accessible à l'espion : c'est le rôle des calculs de sécurité. Dans le reste de ce chapitre, nous nous placerons dans le cas d'une réconciliation inverse, telle que définie dans la section 2.2.1, puisque nos calculs montrent qu'elle fournit à Alice et Bob un avantage toujours supérieur à la réconciliation directe[3]. Dans ce cadre, l'information accessible à l'espion correspond à la connaissance qu'Eve a sur les données de Bob, c'est-à-dire à l'information mutuelle entre Bob et Eve, I_{BE}.

4.2.2 Différentes classes d'attaques

Pour calculer I_{BE}, il est nécessaire de modéliser le système de cryptographie quantique correctement, mais aussi de bien délimiter l'ensemble des opérations qu'Eve est autorisée à effectuer : on parle de *classes d'attaques*, dont nous donnons quelques exemples ci-dessous.

- *Attaques individuelles :* Eve est autorisée à interagir de manière individuelle avec chaque état cohérent envoyé par Alice, et à stocker son état-sonde dans une mémoire quantique. Elle peut ensuite attendre que Bob révèle quel choix de quadrature il a effectué (c'est l'étape dite de *sifting*) pour réaliser la mesure la plus adaptée en fonction de cette quadrature. En revanche, on suppose qu'elle effectue cette mesure avant la réconciliation (cf. chapitre 7). L'information maximale qu'Eve obtient sur la clé de Bob est alors limitée par l'information mutuelle classique (dite de Shannon), I_{BE}, définie dans la partie 3.2.4.
- *Attaques collectives :* Eve interagit toujours de manière individuelle avec les états cohérents, mais est autorisée à attendre que la totalité des étapes classiques d'extraction de clé soit terminée pour effecter la mesure la plus adaptée. Devetak et Winter ont montré [48] que l'information mutuelle à utiliser dans ce cas est l'information mutuelle quantique (dite de Holevo), χ_{BE}, définie dans la partie 3.2.5.

3. Les calculs pour la réconciliation directe peuvent être dérivés directement des méthodes utilisées dans ce chapitre, voir [1, 5] pour plus de détails.

- Les *attaques cohérentes* sont les attaques les plus puissantes autorisées par la mécanique quantique. Eve est alors autorisée à interagir collectivement avec tous les états envoyés par Alice, et à réaliser des mesures conjointes sur toutes ses sondes après la totalité des étapes d'extraction. La sécurité contre ce type d'attaques est dite *inconditionnelle*, puisqu'aucune hypothèse n'est faite sur les capacités de l'espion.

Dans le cas des systèmes de cryptographie quantique utilisant les quadratures d'états cohérents — tels que celui présenté dans ce manuscrit — il a été montré dans [50] que la borne χ_{BE} définie pour les attaques collectives reste valable asymptotiquement pour des attaques cohérentes arbitraires. Par conséquent, les résultats que nous calculons dans la suite de ce chapitre pour les attaques collectives restent valables pour des attaques cohérentes générales, ce qui garantit la sécurité inconditionnelle du protocole.

Mode réaliste

Dans tous les protocoles de cryptographie quantique actuellement mis en pratique, on considère que les sites physiques où sont placés les dispositifs d'Alice et de Bob sont sûrs. Ce mode de fonctionnement est généralement appelé *mode réaliste*, et repose sur l'idée simple que si l'espion a accès aux sites physiques, il peut avoir accès à l'ordinateur d'Alice ou Bob, et donc directement espionner le message.

En particulier, le mode réaliste implique que l'espion n'a pas accès au détecteur de Bob. Dans le cas des variables continues, nous modélisons l'hypothèse en supposant que l'espion ne peut avoir d'influence sur les pertes du détecteur χ_{hom} et sur le bruit électronique v_{el} ; nous nous placerons dans ce cadre pour la suite [4].

4.2.3 Modélisation à intrication virtuelle

Le protocole à variables continues présenté en 2.2.1 correspond à un schéma asymétrique, dit *prepare and measure* (P&M) : Alice prépare un état selon une certaine distribution, puis l'envoie à Bob qui le mesure. On peut définir un autre protocole, parfaitement équivalent, fondé sur l'intrication :

- Alice génère une paire EPR (voir partie 2.1.2) parfaitement intriquée en quadratures, de variance VN_0.
- Elle conserve un mode de l'état pour elle, et envoie l'autre moitié à Bob, par le canal de transmission.
- Elle effectue ensuite une mesure hétérodyne des deux quadratures de son mode, et obtient les valeurs x_A et p_A. Puisque les deux modes de l'état sont parfaitement intriqués, cette mesure projette le mode de Bob dans l'état cohérent $|x_A - ip_A\rangle$. Par ailleurs, puisque la distribution des quadratures de l'état EPR est une gaussienne de variance $VN_0 = (V_A + 1)N_0$, la distribution envoyée à Bob est exactement la même que dans le protocole P&M.

4. Les premières preuves de sécurité ont été établies hors du mode réaliste [1], mais il est aisé de retrouver les formules non-réalistes (qualifiées de « paranoïaques » dans [51]) à partir des preuves que nous présentons ci-après, en remplaçant les bruits par χ'_{tot}, χ'_{line} et χ'_{hom}, tels que $\chi'_{\text{tot}} = \chi'_{\text{line}} = \chi_{\text{tot}}$ et $\chi'_{\text{hom}} = 0$. Les formules non-réalistes sont explicitées dans l'annexe A.2

- Le mode de Bob subit ensuite les mêmes pertes et bruits que dans le protocole P&M, et s'intrique éventuellement avec des états-sondes d'Eve. De même, Bob récupère l'information en réalisant une mesure homodyne de l'état qu'il reçoit.

L'intérêt de ce schéma, résumé sur la figure 4.1, est de faire intervenir un seul état pur à trois modes intriqués entre Alice, Bob et Eve, qui modélise la distribution d'états préparés. Il est donc possible d'exprimer la fonction d'onde ou la matrice de covariance de cet état, et donc de réaliser les calculs d'information.

Dans le mode réaliste, il est nécessaire de modéliser le bruit électronique v_{el} et les pertes de la détection η. Pour ce faire, nous introduisons un état EPR de variance vN_0 dans le système de Bob. La moitié de cet état (correspondant à un état thermique de variance vN_0) se mélange avec l'état sortant du canal de transmission dans une lame séparatrice de transmission η. Enfin, la variance vN_0 est choisie de façon que la variance de Bob au niveau des détecteurs soit augmentée de v_{el}, ce qui revient à poser $v = 1 + \frac{v_{\text{el}}}{1-\eta}$.

4.3 Expression de l'information secrète

4.3.1 Attaques individuelles

La quantité d'information secrète disponible après la transmission peut s'écrire $\Delta I_{\text{Shannon}} = I_{AB} - I_{BE}$ [49].

Information entre Alice et Bob : $\mathbf{I_{AB}}$

L'information partagée par Alice et Bob est calculée à partir de la formule de Shannon 3.24 :

$$I_{AB} = \frac{1}{2}\log_2(1+\text{SNR}) \tag{4.12}$$

$$= \frac{1}{2}\log_2\left(1+\frac{V_A}{1+\chi_{\text{tot}}}\right) \tag{4.13}$$

$$\boxed{I_{AB} = \frac{1}{2}\log_2\left(\frac{V+\chi_{\text{tot}}}{1+\chi_{\text{tot}}}\right)} \tag{4.14}$$

L'information mutuelle I_{AB} que partagent Alice et Bob après la transmission croît donc de façon logarithmique avec la puissance en sortie d'Alice, et diminue avec le bruit total ajouté dans la ligne.

Information accessible à l'espion : $\mathbf{I_{BE}}$

Pour calculer l'information accessible à l'espion, on part de la formule 3.29 :

$$I_{BE} = \frac{1}{2}\log_2\frac{V_B}{V_{B|E}} \tag{4.15}$$

On connaît $V_B = G(V+\chi_{\text{tot}})N_0$, reste donc à déterminer $V_{B|E}$. Dans le cas de la réconciliation inverse, les données de Bob servent de base à la construction de la clé.

Le but pour Alice et pour Eve est donc de retrouver la valeur des données bruitées reçues par Bob.

Par ailleurs, on se place toujours dans le mode réaliste, dans lequel Eve n'a pas d'information sur χ_{hom}. On peut donc écrire $V_{B|E} = \eta V_{B_1|E} + \eta\chi_{\text{hom}} N_0$, où B_1 correspond au mode de Bob à la sortie du canal de transmission.

- Alice a une information : la valeur classique de la quadrature qu'elle a envoyée, x_A. Elle doit donc estimer la valeur de la quadrature X_{B_1}, qui peut s'écrire $x_{B_1} = t\,x_A + x_{N,A}$. Dans cette expression, $X_{N,A}$ est l'estimateur d'Alice sur les données de Bob, qui correspond à l'erreur commise par Alice quand elle estime X_{B_1}. Sa définition nous permet de l'exprimer en fonction de la variance conditionnelle $V_{B_1|A} = \langle X_{N,A}^2 \rangle$.
- De la même manière, Eve possède une information p_E sur l'état B_1, et peut donc écrire la quadrature $p_{B_1} = \lambda\, p_E + p_{N,E}$. L'erreur résiduelle d'Eve est $P_{N,E}$, telle que $\langle P_{N,E}^2 \rangle = V_{B_1|E}$.

On peut donc calculer le commutateur

$$\begin{aligned}
[X_{N,A}\,,\,P_{N,E}] &= [X_{B_1} - X_A\,,\,P_{B_1} - \lambda\,P_E] && (4.16)\\
&= [X_{B_1}\,,\,P_{B_1}] \underbrace{-\lambda\cancel{[X_{B_1}\,,\,P_E]} - \cancel{[X_A\,,\,P_{B_1}]} + \lambda\cancel{[X_A\,,\,P_E]}}_{\text{les quadratures de deux modes distincts commutent}} && (4.17)\\
[X_{N,A}\,,\,P_{N,E}] &= 2iN_0 && (4.18)
\end{aligned}$$

Cette relation de commutation nous permet d'écrire l'inégalité de Heisenberg

$$V_{B_1|A} V_{B_1|E} \geqslant N_0^2 \tag{4.19}$$

Reste à exprimer $V_{B_1|A}$. Pour ce faire, on utilise la modélisation à intrication virtuelle : la matrice de covariance γ_{AB1} de l'état EPR s'écrit

$$\gamma_{AB_1} = \begin{pmatrix} V & \sqrt{T(V^2-1)} \\ \sqrt{T(V^2-1)} & T(V+\chi_{\text{line}}) \end{pmatrix} N_0 \tag{4.20}$$

On peut donc exprimer (voir eq. 3.33)

$$\begin{aligned}
V_{B_1|A} &= \langle X_{B_1}^2 \rangle - \frac{\langle X_A X_{B_1} \rangle^2}{\langle X_A^2 \rangle} && (4.21)\\
&= T(V+\chi_{\text{line}})N_0 - \frac{T(V^2-1)}{V} N_0 && (4.22)\\
&= T\left(\frac{1}{V} + \chi_{\text{line}}\right) N_0 && (4.23)
\end{aligned}$$

La borne sur $V_{B|E}$ s'écrit donc

$$V_{B|E} = \eta V_{B_1|E} + \eta\chi_{\text{hom}} N_0 \geqslant \frac{\eta + \eta T \chi_{\text{hom}}(\frac{1}{V} + \chi_{\text{line}})}{T(\frac{1}{V} + \chi_{\text{line}})} N_0 \tag{4.24}$$

On exprime enfin la borne sur l'information accessible à Eve :

$$\boxed{I_{BE} \leqslant \frac{1}{2} \log_2 \frac{T^2(V+\chi_{\text{tot}})(\frac{1}{V} + \chi_{\text{line}})}{1 + T\chi_{\text{hom}}(\frac{1}{V} + \chi_{\text{line}})}} \tag{4.25}$$

4.3.2 Attaques collectives

Dans le cas d'attaques collectives, l'information secrète extractible s'écrit $\Delta I_{\text{Holevo}} = I_{AB} - \chi_{BE}$, où l'expression (4.14) de I_{AB} reste valable. Pour calculer l'information mutuelle quantique $\chi_{BE} = S(B:E)$, nous devons d'abord modéliser le protocole selon le mode réaliste, avec un schéma de type intrication virtuelle (voir figure 4.1 pour les notations des modes). Nous résumons ici le raisonnement [51] menant à l'expression du taux secret.

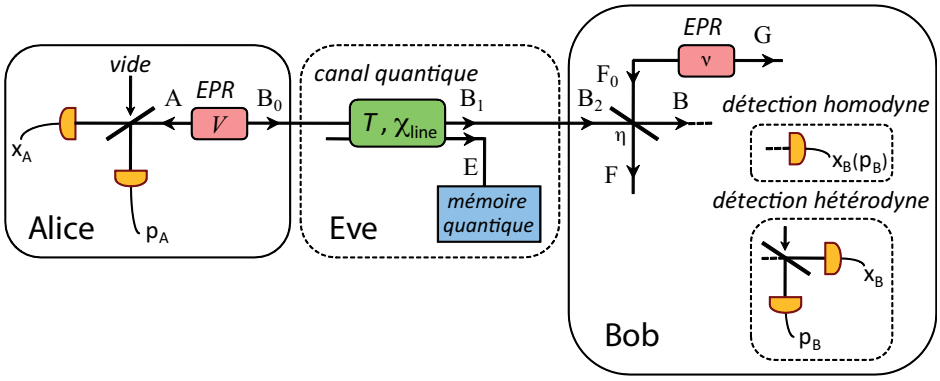


Figure 4.1 : *Modélisation à intrication virtuelle du protocole dans le mode réaliste.*

Calcul de l'information secrète

Nous bornons χ_{BE} à partir de la formule de Holevo (3.34) :

$$\chi_{BE} \leqslant S(\rho_E) - \int \Pr(x_B)\, S\left(\rho_E^{x_B}\right) \mathrm{d}x_B \tag{4.26}$$

a. Réécriture de l'inégalité

- L'action de l'espion étant optimale, Eve purifie l'état d'Alice et Bob : l'état ρ_{AB_1E} est un état pur. En tant que tel, on sait (voir section 3.2.5) que $S(\rho_E) = S(\rho_{AB_1})$.
- De même, après la mesure projective de Bob sur X_B, l'état $\rho_{AEFG}^{x_B}$ est pur, et donc $S\left(\rho_E^{x_B}\right) = S\left(\rho_{AFG}^{x_B}\right)$. Par ailleurs, l'état ρ_{ABFG} est un état gaussien (il s'agit d'un mélange d'états gaussiens), et en tant que tel la projection de l'état par une mesure sur X_B ne dépend pas de la mesure x_B réalisée. Le terme $S(\rho_{AFG}^{x_B})$ est donc indépendant de x_B, et peut être sorti de l'intégrale dans (4.26).
  On peut donc réécrire :

$$\chi_{BE} \leqslant S(\rho_{AB_1}) - S\left(\rho_{AFG}^{x_B}\right) \tag{4.27}$$

b. Expression de $\gamma_{\mathbf{AB_1}}$ et $\gamma_{\mathbf{AFG}}^{\mathbf{x_B}}$

- La matrice γ_{AB_1} est exprimée en (4.20).
- Pour calculer $\gamma_{AFG}^{x_B}$, on exprime d'abord $\gamma_{AB_2F_0G} = \gamma_{AB_2} \oplus \gamma_{F_0G}^{EPR}$.
- Les modes B_2 et F_0 sont ensuite couplés par une lame séparatrice de transmission η :

$$\gamma_{ABFG} = Y^T\, \gamma_{AB_2F_0G}\, Y \tag{4.28}$$

avec $Y = \left(\mathbb{I}_A \oplus S_{B_2F_0}^{\text{beamsplitter}} \oplus \mathbb{I}_G\right)$.

- On réorganise ensuite γ_{ABFG} en $\gamma_{AFGB} = \begin{bmatrix} \gamma_{AFG} & \sigma_{AFG;B} \\ \sigma_{AFG;B}^T & \gamma_B \end{bmatrix}$.
- On utilise enfin (2.21) pour calculer $\gamma_{AFG}^{x_B}$:

$$\gamma_{AFG}^{x_B} = \gamma_{AFG} - \sigma_{AFG;B}^T (X\gamma_B X)^{\mathrm{MP}} \sigma_{AFG;B} \tag{4.29}$$

c. Expression de $\chi_{\mathbf{BE}}$

Après ces calculs, nous disposons de l'expression des matrices de covariance. Nous devons alors en extraire les valeurs propres symplectiques, c'est-à-dire les valeurs propres de $-\Omega\gamma\Omega\gamma$ (voir équation 2.16). Ce calcul passe par la diagonalisation de la matrice, ce qui est particulièrement fastidieux car les éléments des matrices ont une expression compliquée ; il est alors possible de s'aider d'un logiciel de calcul formel.

Une fois les valeurs propres symplectiques déterminées, nous pouvons enfin utiliser l'expression 3.13 pour déterminer χ_{BE} :

$$\boxed{\chi_{\mathrm{BE}} = G\left(\frac{\lambda_1 - 1}{2}\right) + G\left(\frac{\lambda_2 - 1}{2}\right) - G\left(\frac{\lambda_3 - 1}{2}\right) - G\left(\frac{\lambda_4 - 1}{2}\right)} \tag{4.30}$$

où $\lambda_{1,2}$ sont les valeurs propres symplectiques de γ_{AB_1} et $\lambda_{3,4,5}$ les valeurs propres symplectiques de $\gamma_{AFG}^{x_B}$. Elles s'expriment [5] :

$$\lambda_{1,2}^2 = \frac{1}{2}(A \pm \sqrt{A^2 - 4B}) \qquad \lambda_{3,4}^2 = \tfrac{1}{2}(C \pm \sqrt{C^2 - 4D}) \tag{4.31}$$

$$A = V^2(1-2T) + 2T + T^2(V + \chi_{\mathrm{line}})^2 \qquad B = T^2(V\chi_{\mathrm{line}} + 1)^2$$

$$C = \frac{V\sqrt{B} + T(V + \chi_{\mathrm{line}}) + A\chi_{\mathrm{hom}}}{T(V + \chi_{\mathrm{tot}})} \qquad D = \sqrt{B}\frac{V + \sqrt{B}\chi_{\mathrm{hom}}}{T(V + \chi_{\mathrm{tot}})}$$

Evolution de l'information secrète avec les paramètres

Grâce à ces formules, nous pouvons déterminer l'information secrète disponible à Alice et Bob en fonction des différents paramètres du système. Pour toutes les courbes qui suivent, nous partons des paramètres suivants : $V_A = 20$; $T = 0{,}5$; $\varepsilon = 0{,}01$; $\eta = 0{,}5$; $v_{\mathrm{el}} = 0{,}01$.

Dans la figure 4.2, nous faisons varier la variance de modulation. Sans surprise, I_{AB} varie logarithmiquement ; l'information secrète, quant à elle, tend vers une valeur limite aux grandes variances.

Dans la figure 4.3, nous jouons sur les bruits en excès dans le système : ε et v_{el}. La différence de comportement de l'information secrète est frappante : l'excès de bruit du canal quantique ε fait très rapidement chuter l'information secrète, pour atteindre zéro autour de 0,3 ; au contraire, le bruit électronique n'a quasiment pas d'effet sur l'information secrète. C'est tout l'intérêt du mode réaliste : quand le bruit électronique augmente, χ_{BE} diminue, ce qui revient à dire que l'espion ne peut pas utiliser le bruit électronique pour acquérir de l'information.

Enfin, nous observons un phénomène similaire dans la figure 4.4, dans laquelle les transmissions du système sont le paramètre fluctuant. Quand T fluctue, l'information de l'espion se rapproche très vite de l'information I_{AB} (mais ne la croise pas). Au contraire, quand η diminue, l'information secrète disponible diminue de façon presque linéaire : ceci n'est dû qu'à la diminution progressive du SNR avec η, et non à une intervention de l'espion.

5. Le calcul de λ_5 donne 1 dans tous les cas, c'est-à-dire $G(\frac{\lambda_5 - 1}{2}) = 0$.

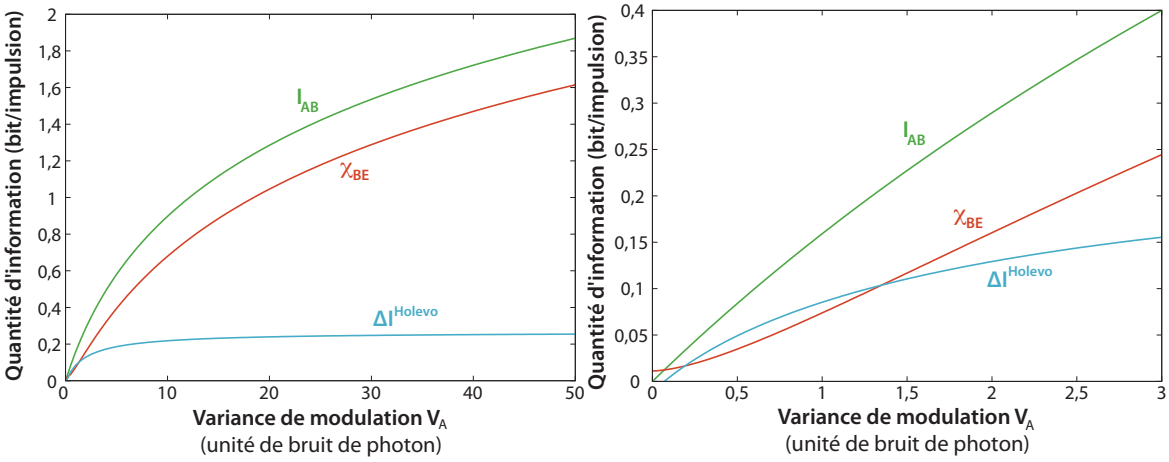


Figure 4.2 : *Information en fonction de la variance de modulation.*

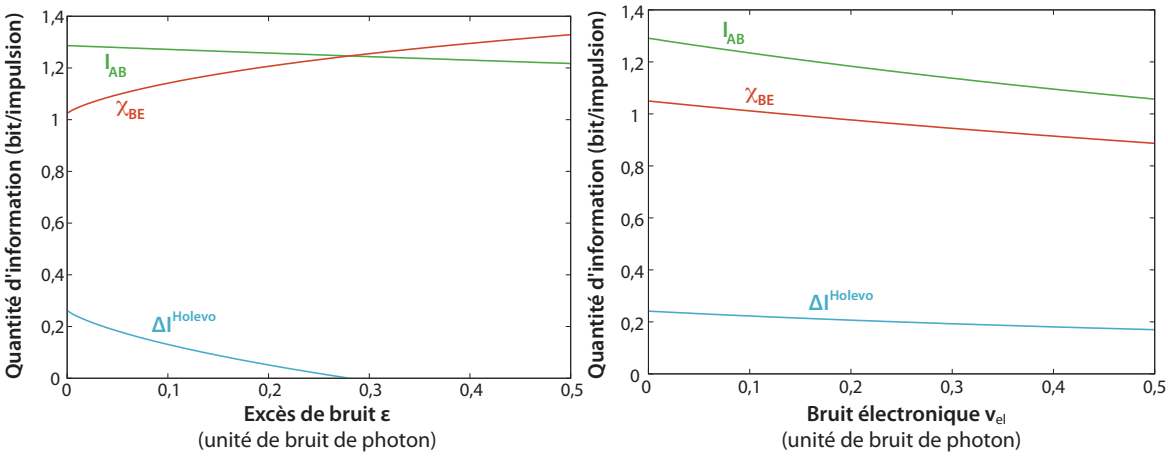


Figure 4.3 : *Information en fonction, respectivement, de l'excès de bruit du canal quantique et du bruit électronique.*

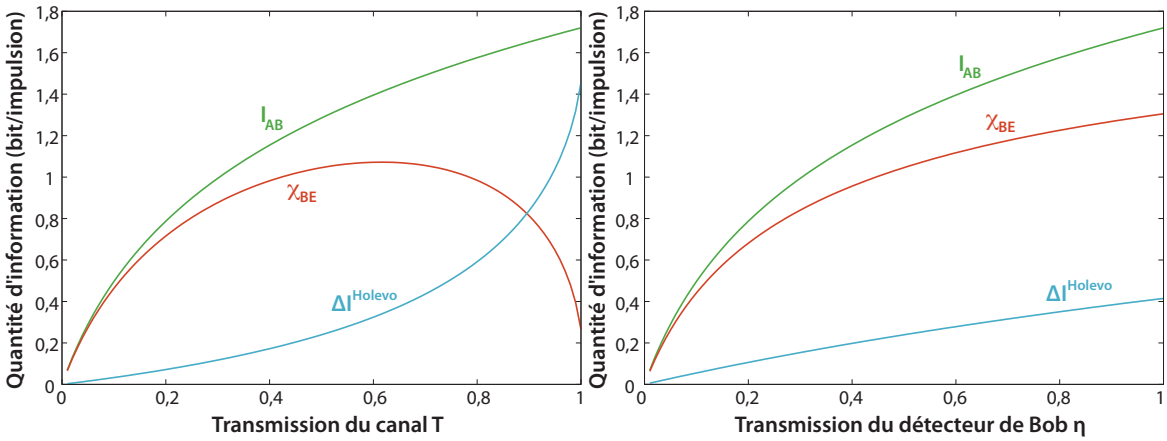


Figure 4.4 : *Information en fonction, respectivement, de la transmission du canal quantique et de la transmission du détecteur de Bob.*

Deuxième partie

Mise en œuvre expérimentale

5 Implémentation optique du protocole

> Le capitaine Nemo, muni d'une lunette à réticules, qui, au moyen d'un miroir, corrigeait la réfraction, observa l'astre qui s'enfonçait peu à peu au-dessous de l'horizon en suivant une diagonale très allongée. (...) Nous étions au pôle même.
>
> *Vingt mille lieues sous les mers*
> Jules Verne

Le système de cryptographie quantique présenté dans ce document a été entièrement conçu et réalisé avec de l'optique fibrée. La figure 5.1 présente le schéma optique global du démonstrateur, simplifié des polariseurs et isolateurs assurant une bonne qualité de faisceau tout au long de la propagation.

Dans ce chapitre, nous exposons dans un premier temps les spécificités du travail avec des fibres optiques, ainsi que les caractéristiques des fibres utilisées. Nous détaillons ensuite l'implémentation des composants essentiels de notre système, à savoir la source des états cohérents utilisés, le système de modulation en amplitude et phase utilisé par Alice pour l'encodage, le système de multiplexage permettant de transmettre signal et OL dans une seule fibre de transmission, ainsi que le système de mesure de Bob.

5.1 Optique fibrée

Les expériences d'optique mettant en jeu des phénomènes quantiques sont généralement réalisées en espace libre. En effet, la majorité des états quantiques intéressants (états intriqués, comprimés, etc.) sont sensibles à la décohérence, qui les ramène vers des états cohérents ; il est alors nécessaire de limiter au maximum les pertes et de contrôler au mieux les propriétés de l'état utilisé. De ce point de vue, les performances des composants libres sont particulièrement adaptées (haute réflectivité des miroirs, micro-contrôles du positionnement des composants, faible atténuation du milieu de propagation, etc.).

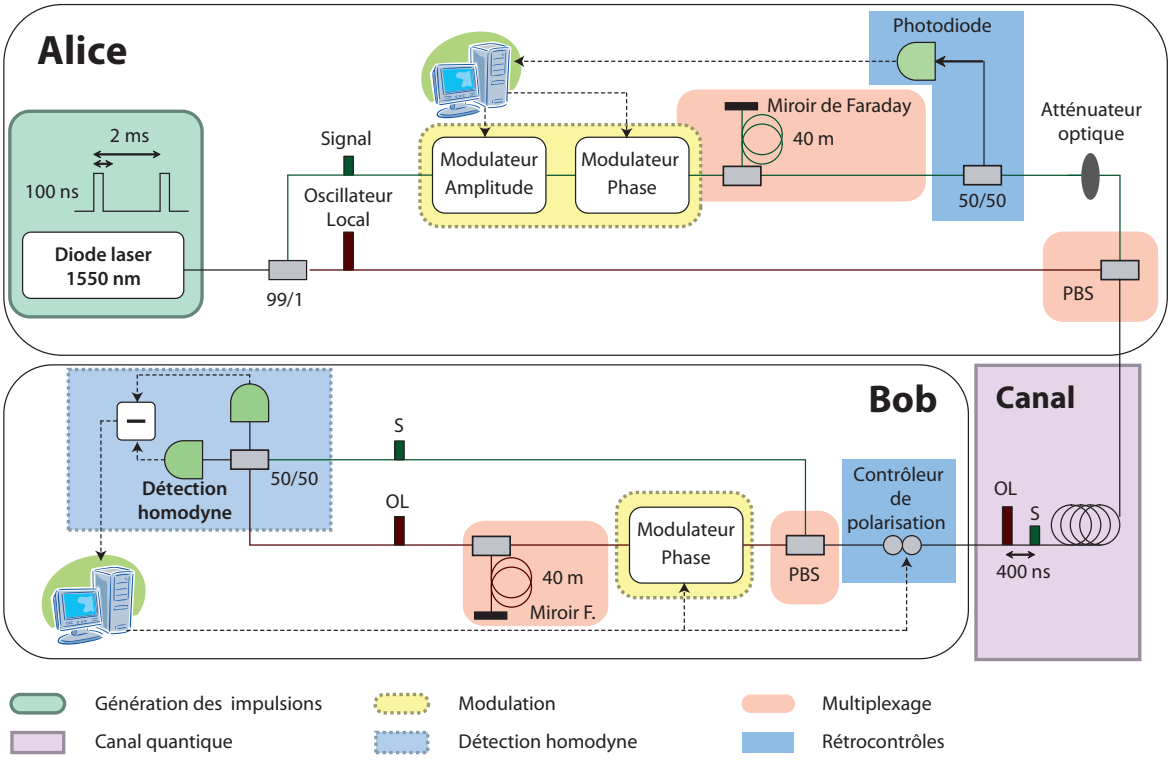


Figure 5.1 : *Schéma du système optique.*

Néanmoins, la difficulté majeure du travail en espace libre vient des difficultés d'alignement et de pointage, ainsi que des problématiques de couplage de modes. La stabilité des réglages est cruciale, et la qualité du support des composants est donc importante : ainsi, des tables optiques sur coussin d'air sont le plus souvent utilisées. Dans le cadre d'un prototype de cryptographie quantique, qui doit pouvoir être déplacé et intégré dans des boîtiers rackables, ce type de contraintes rend l'implémentation assez délicate. Par ailleurs, la transmission en espace libre pose, à longue distance, des problèmes de pointage et de fluctuations, liés aux perturbations atmosphériques. Notons toutefois que certains systèmes utilisant une propagation libre en ont démontré la faisabilité [52, 53].

Dans ce contexte, l'avantage évident de l'optique fibrée est la résolution des problèmes d'alignement et de guidage de la lumière. Par ailleurs, la faible dimension des composants et la flexibilité des fibres permet d'intégrer les montages fibrés dans des volumes assez petits. Il y a néanmoins un prix à payer : les connexions entre fibres introduisent des pertes, les fibres présentent une biréfringence importante, les chemins optiques et les phases dans les fibres sont difficiles à contrôler, etc. Nous présentons dans cette section les caractéristiques des fibres optiques que nous utilisons.

5.1.1 Différents types de fibres optiques

Il existe plusieurs types de fibres optiques, principalement à base de plastique ou de silice. Dans le système que nous avons développé, nous utilisons des fibres optiques mo-

nomodes en silice, conçues pour un fonctionnement optimal à longueur d'onde télécom, autour de 1 550 nm (pic de transmission de la fibre).

Structure d'une fibre multimode

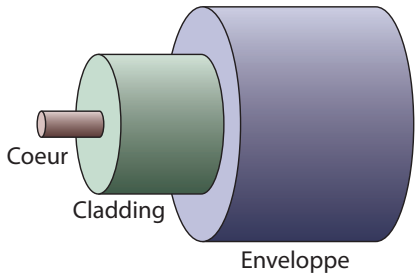


Figure 5.2 : *Structure d'une fibre optique typique*

Une fibre optique en verre est composée d'un cœur et d'un *cladding* (ou gaine optique), tous deux en silice, et éventuellement d'une gaine de protection en plastique. La figure 5.2 présente la structure optique d'une fibre typique.

La propagation de la lumière dans une fibre multimode peut être décrite assez précisément en utilisant l'optique géométrique. Les indices optiques du cœur et du cladding sont choisis de façon qu'une onde électromagnétique qui rentre dans la fibre, selon un angle suffisamment petit, ne subisse que des réflexions totales au cours de sa propagation, ce qui lui permet d'être transmise avec des pertes théoriquement nulles.

Citons quelques types de fibre multimode :

- Fibre à saut d'indice : c'est la fibre la plus utilisée. Puisque l'indice optique du cœur est constant, la vitesse de phase des ondes est constante lors de la propagation. Néanmoins, par des considérations géométriques, on voit que le vecteur d'onde effectif (selon l'axe) de deux impulsions qui entrent selon un angle différent n'est pas le même : il y a donc dispersion modale, en fonction de l'angle d'entrée (voir figure 5.3).
- Fibre à gradient d'indice constant. Dans cette fibre, l'indice optique est une fonction linéaire de la distance à l'axe, ce qui permet de réduire la dispersion modale.
- Fibre à gradient d'indice linéaire. L'indice optique est parabolique, centré sur l'axe de la fibre ; un calcul d'optique géométrique montre que le chemin optique dans une telle fibre est indépendant de l'angle d'entrée, ce qui revient à dire que le vecteur d'onde effectif de toutes les impulsions est la même [54]. Par conséquent, c'est le type de fibre qui réduit le mieux la dispersion modale.

Fibre monomode

Comme leurs noms l'indiquent, les fibres monomodes sont conçues pour ne laisser qu'un seul mode de l'onde électromagnétique se propager, alors que les fibres multimodes en autorisent plusieurs. La fibre optique joue le rôle d'un guide d'onde : plus son diamètre est élevé, plus le nombre de modes pouvant se propager est important.

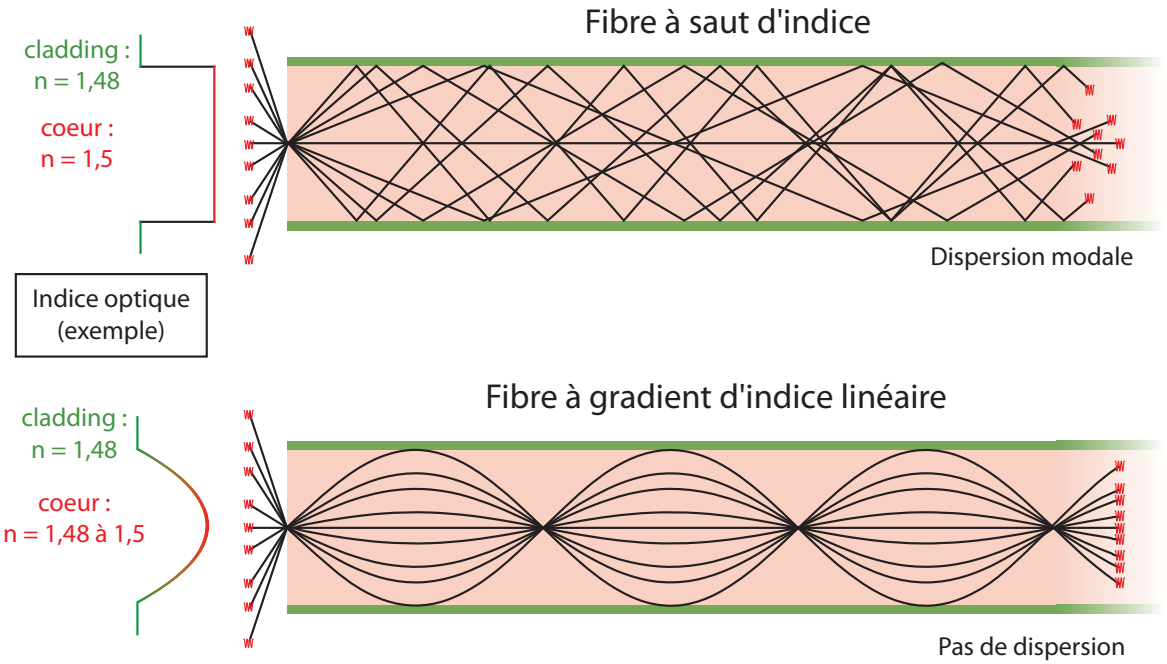


Figure 5.3 : *Propagation de l'onde en fonction de l'angle d'incidence, dans une fibre à saut d'indice et dans une fibre à gradient linéaire d'indice. Les angles d'incidence du schéma sont volontairement exagérés, de façon à amplifier les effets de dispersion modale. En pratique, pour des fibres multimodes ayant un cœur d'environ 80 µm, l'angle maximal d'acceptance est de l'ordre de 30°.*

Typiquement, une fibre optique monomode à 1 550 nm a un cœur de 8 à 10 micromètres, alors que les fibres multimodes ont des cœurs de plusieurs dizaines de micromètres, voire plus pour les fibres conçues pour les lasers de puissance. Bien entendu, le caractère monomode d'une fibre dépend de la longueur d'onde de la lumière qui s'y propage : une fibre monomode à 1 550 nm est multimode à 650 nm.

Les fibres monomodes ont plusieurs atouts : tout d'abord, il n'y a plus de dispersion des modes spatiaux avec la distance, ce qui rend les fibres monomodes particulièrement adaptées à des transmissions à haut débit et à longue distance. De plus, la taille du mode optique transmis est défini par la géométrie de la fibre (la taille du cœur), et il est donc facile de coupler deux fibres monomodes avec une très bonne efficacité, en les plaçant en vis-à-vis. On atteint une efficacité de 95 % (−0,3 dB) avec un coupleur de fibres standard, et 98% ou plus avec des coupleurs spécifiquement conçus pour limiter les pertes. Par conséquent, la plupart des systèmes de cryptographie quantique utilisent une fibre monomode standard comme canal quantique de transmission.

Fibre à maintien de polarisation

Une fibre standard, en tant que milieu matériel de propagation, présente de la biréfringence dès qu'elle est soumise à une contrainte extérieure, ce qui est le cas quasi-systématiquement, ne serait-ce que du fait des contraintes internes de la structure. La polarisation de l'impulsion lumineuse qui s'y propage n'est donc pas constante, que

ce soit dans l'espace ou dans le temps, puisque les contraintes sur la fibre évoluent en permanence (par exemple du fait des variations de température).

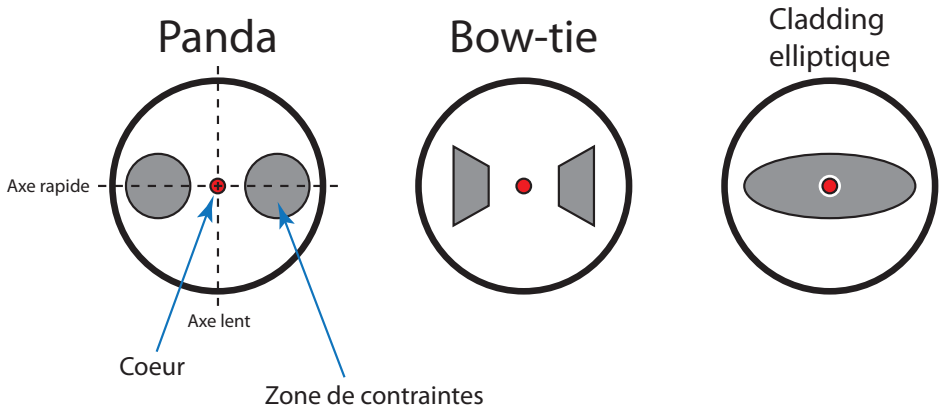


Figure 5.4 : *Coupe orthogonale à l'axe de différents types de fibres à maintien de polarisation*

Pour les applications où un contrôle de la polarisation est nécessaire, un type de fibre, dit *à maintien de polarisation (PM)* et présenté sur la figure 5.4, a été développé à partir des années 1980. Il s'agit d'une fibre monomode spéciale dans laquelle des zones de contrainte internes ont été créées de manière à créer deux axes privilégiés orthogonaux de polarisation. Ces axes sont appelés *rapide* et *lent* du fait de leurs indices optiques respectifs n_1 et n_2, qui vérifient $n_1 - n_2 \approx 10^{-3}$. Ainsi, si une onde lumineuse entre dans la fibre PM avec une polarisation alignée avec l'un des deux axes, et que les contraintes extérieures restent petites devant les contraintes internes, la polarisation de l'onde est conservée. En revanche, si la polarisation n'est pas parfaitement alignée en entrée, elle devient légèrement elliptique en sortie. Par conséquent, la connexion de deux fibres PM est assez délicate, car les axes privilégiés doivent être positionnés en vis-à-vis avec une très bonne précision.

Les fibres PM que nous avons choisi d'utiliser sont de type Panda, dans lesquelles deux zones cylindriques de contraintes sont placées de manière symétrique par rapport au cœur. Les zones de contraintes sont dopées au bore, et ont un coefficient d'expansion thermique plus fort que le reste du cladding ($3.10^{-6}/\,^{\circ}\mathrm{C}$ contre $5.10^{-7}/\,^{\circ}\mathrm{C}$) ; en refroidissant, la zone dopée se contracte plus que le cladding, ce qui crée une tension interne importante et donc une forte biréfringence. Il existe d'autres types de fibres à maintien de polarisation, qui utilisent des techniques similaires : citons les fibres *bow-tie* et les fibres à cladding elliptique.

Tous les composants que nous utilisons dans le système sont sertis à des fibres PM, à l'exception des lignes à retard (voir dans la suite du chapitre), du contrôleur de polarisation, et des photodiodes, qui utilisent des fibres monomode standard. Par ailleurs, nous avons choisi de suivre la convention utilisée dans les télécommunications : dans la quasi-totalité des fibres PM de notre système, la lumière est polarisée selon l'axe lent des fibres.

5.1.2 Pertes des fibres optiques

Pour optimiser les performances de notre système de cryptographie, il nous faut limiter les pertes de certaines parties du montage optique. Les pertes dans le système d'Alice n'interviennent pas dans l'expression du taux secret, qui ne fait apparaître que le niveau de sortie V_A. En revanche, les pertes η de la voie signal de Bob dégradent l'information secrète disponible, et doivent donc être éliminées autant que possible. Les pertes sur la voie de l'oscillateur local n'ont pas d'effet en tant que telles, mais un trop faible niveau d'OL au niveau du détecteur diminue le niveau de bruit de photon, et augmente donc le niveau relatif de bruit électronique. Enfin, les pertes de la fibre de transmission ne sont pas sous notre contrôle, mais ont un rôle crucial dans la distance limite de transmission et le taux secret.

Pertes en ligne

Chaque type de fibre optique est optimisé pour un usage spécifique, et les pertes linéiques acceptables dépendent de l'utilisation recherchée. Dans le cas des fibres à cœur de silice, un pic principal de transmission existe autour de 1 550 nm. C'est donc la longueur d'onde utilisée pour les télécommunications à longue distance.

Les pertes en ligne des fibres monomodes standard que nous utilisons est de 0,19 dB/km, ce qui correspond à la valeur standard pour les fibres optiques utilisées de manière commerciale. Quelques constructeurs proposent des fibres optimisées en transmission, mais elles sont bien plus chères et n'atteignent que des atténuations d'environ 0,15 dB/km. Les fibres PM présentent une atténuation plus importante, spécifiée à 0,5 dB/km.

Pertes dues à la courbure

Un rayon de courbure minimal admissible est généralement spécifié lors de l'achat de fibres optiques : dans le cas de nos fibres PM, celui-ci est de 40 mm. En effet, au delà d'une certaine courbure, la fibre ne guide plus parfaitement la lumière, ce qui se traduit par des pertes dépendant de la courbure. Nous avons mesuré les pertes en courbure de l'une des fibres PM utilisées sur le système (figure 5.5). On voit que le seuil réel au delà duquel la fibre présente des pertes (14 mm) est bien plus bas que celui spécifié par les constructeurs. En dessous de 6 mm, plus aucune lumière n'est guidée, et la fibre casse généralement vers 3 mm de rayon de courbure.

En pratique, il est difficile d'imposer une telle courbure involontairement, d'autant que les fibres sont gainées : les fibres destinées à être régulièrement manipulées ont d'ailleurs une gaine de 3 mm de diamètre, ce qui rend une rupture involontaire presque impossible. Dans notre cas, les fibres ont une gaine de 900 µm, et nous les fixons dans des roues de lovage de 8 cm, de façon à respecter les 40 mm de rayon de courbure.

Au final, l'effet des pertes des fibres de nos boîtiers reste faible, d'autant que nous avons optimisé le montage de façon à ce que la longueur de la voie signal chez Bob soit aussi petite que possible (environ 5 m, c'est-à-dire un contrôleur de polarisation et deux coupleurs). Les pertes principales de Bob viennent des photodiodes de la détection homodyne et des quelques coupleurs de fibre restants.

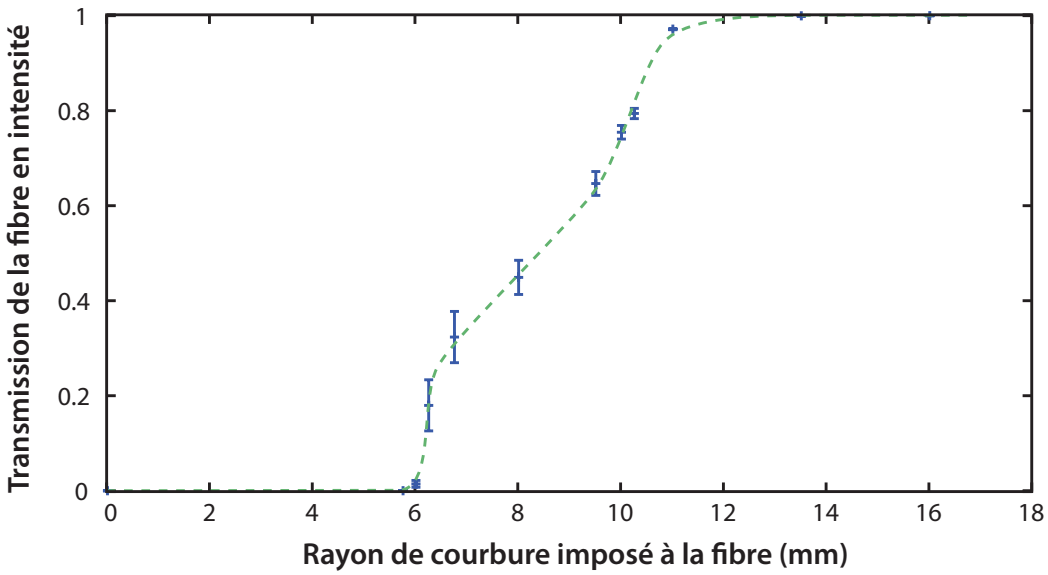


Figure 5.5 : *Transmission totale d'une fibre monomode standard, enroulée sur 3 tours autour d'un cylindre, en fonction du rayon de courbure du cœur de la fibre. Au dessus d'un seuil de 14 mm, la fibre transmet normalement la lumière. En dessous, elle ne la guide plus parfaitement, et devient parfaitement opaque en dessous d'un rayon de courbure de 6 mm.*

5.1.3 Polarisation dans une fibre

Fibres monomodes standard

Dans les fibres monomodes standard, la polarisation n'est pas maintenue, et la polarisation du signal sortant de la fibre peut donc varier au cours du temps. Pour évaluer les vitesses de variation de la polarisation dans de telles fibres, nous avons placé un rouleau de 25 km de fibre monomode en extérieur, et polarisé la lumière en sortie de la fibre : la puissance de sortie est donc proportionnelle à la projection de la polarisation sur l'axe du polariseur. La figure 5.6 présente la dérive de puissance dans ces conditions.

D'un côté, on voit apparaître une dérive importante, due aux contraintes thermiques imposées à la fibre. Ces contraintes évoluent lentement, mais ont un effet cumulé important. On voit ici que la polarisation fluctue sur des échelles de temps de l'ordre de l'heure, et qu'elle ne trouve pas de point d'équilibre au cours des 40 heures du test.

Par ailleurs, on voit sur la partie gauche du graphe, entre minuit et 8 heures, que les contraintes mécaniques (comme des vibrations ou des courants d'air ; ici, le vent) peuvent aussi faire fluctuer la polarisation. Il s'agit principalement de petites fluctuations autour d'un point d'équilibre, mais elles sont rapides.

Fibres monomodes à maintien de polarisation

En théorie, les fibres PM maintiennent efficacement la polarisation, avec un *crosstalk* spécifié à moins de −30 dB pour 100 m. En pratique, pour atteindre cette précision, il faut s'assurer d'un couplage parfait à l'entrée de la fibre, et d'une absence de contraintes sur celle-ci. En particulier, les axes de propagation de deux fibres PM connectées doivent être ajustés à mieux qu'un demi-degré pour que la fuite de puissance sur le mauvais axe soit plus petit que le crosstalk.

Le problème principal vient donc de l'efficacité des connecteurs de fibre : ils permettent d'aligner les axes, mais ont souvent un peu de jeu pour faciliter leur emploi (de l'ordre de 2 à 3 °), et l'angle relatif des deux fibres est donc difficile à contrôler. La figure 5.7 présente la dérive de polarisation dans une fibre PM de 20 m, le long de laquelle 10 connecteurs ont été placés. En entrée, un polariseur est placé de façon à ce que la polarisation soit linéaire (à −30 dB près), selon l'axe lent. A chaque connexion, un peu de puissance « fuit » dans l'axe rapide de la fibre, et la polarisation devient légèrement plus elliptique. En sortie de la fibre, la polarisation est donc assez elliptique, et on voit que la perte de puissance selon l'axe lent peut atteindre presque 50 % — la polarisation n'est alors plus réellement contrôlée.

Pour éviter ce type d'effets, nous avons inséré régulièrement des polariseurs dans la chaîne optique, de façon à « redresser » la polarisation du signal. Par ailleurs, nous avons soudé les composants entre eux grâce à une soudeuse optique prenant en charge les fibres PM, et qui permet d'obtenir des rapports d'extinction d'environ −25 dB à chaque connexion.

5.1.4 Effets non-linéaires dans les fibres optiques

Dans le cas où une puissance importante est envoyée dans une fibre optique, la polarisation diélectrique ne peut plus être considérée comme proportionnelle au champ électrique, et s'écrit alors $\mathbf{P} = \varepsilon_0(\chi^{(1)}\mathbf{E} + \chi^{(2)}\mathbf{EE} + \chi^{(3)}\mathbf{EEE} + \dots)$. Des effets non-linéaires peuvent alors apparaître [55].

Dans la mesure où le milieu est centro-symétrique, le terme en $\chi^{(2)}$, qui correspond à la susceptibilité du second ordre, responsable entre autres de l'effet Pockels, disparaît. Les termes du troisième ordre, en revanche, peuvent apparaître, en particulier auto-modulation de phase, diffusion Raman ou Brillouin ; cependant, ils ne deviennent vraiment significatifs que pour des puissances de l'ordre du watt. Dans notre système, nous n'utilisons que de faibles puissances optiques, typiquement moins de 10 mW-crête, de sorte que ces effets sont négligeables.

5.1.5 Composants fibrés disponibles

Depuis le développement des télécommunications fibrées, de nombreux composants conçus pour l'espace libre ont été transposés pour la technologie fibrée. Nous présentons dans le tableau 5.8 une liste non-exhaustive des composants disponibles en optique fibrée à 1 550 nm.

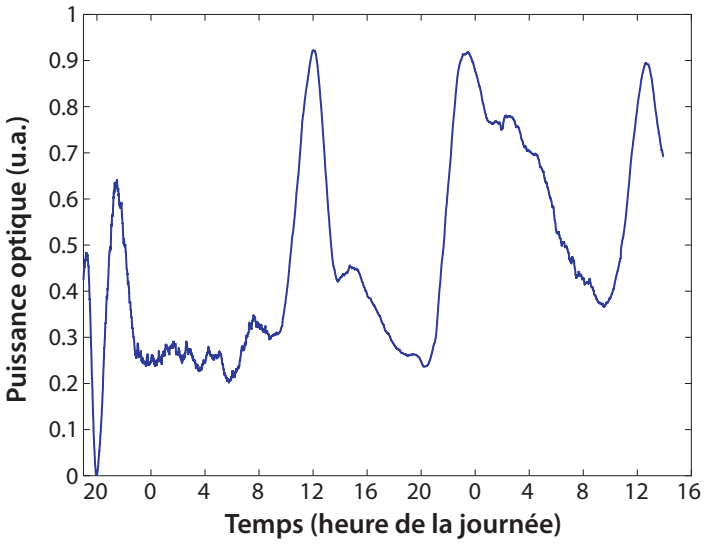


Figure 5.6 : *Puissance optique selon un axe de polarisation donné, en sortie d'un rouleau de 25 km de fibre monomode, placé en extérieur à l'abri du soleil et du vent. La mesure est faite pendant 40 heures, de 19h jusqu'à 14h le surlendemain. Les contraintes thermiques font évoluer la biréfringence du rouleau de fibre, et la polarisation en sortie fluctue en conséquence.*

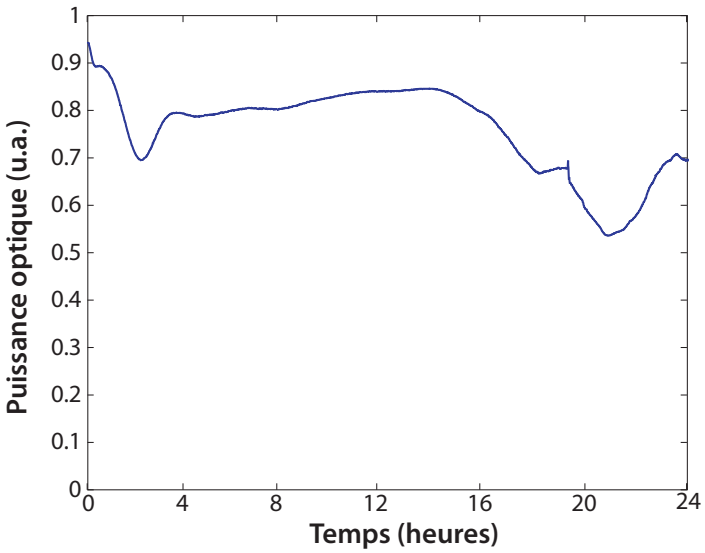


Figure 5.7 : *Puissance optique sortant d'une chaîne composée d'une diode laser polarisée, suivie de dix jarretières PM d'environ 1 m connectorisées entre elles, et polarisée en sortie selon l'axe privilégié des fibres PM. La fluctuation de puissance de sortie est due à l'imprécision cumulée des connexions entre fibres PM, du fait du jeu existant dans les connecteurs.*

Composant	Pertes	SM	PM	Commentaire
Diodes laser	—	×	✓	
Coupleurs	0,05 dB	✓	✓	Version fibrée des séparateurs de faisceau.
Polariseurs	0,3 dB	✓	✓	
Atténuateurs	0-80 dB	✓	✓	Souvent fondés sur un montage en espace libre, intégré avec des fibres.
Modul. de phase	2 dB	×	✓	
Modul. d'amplitude	3 dB	×	✓	
Photodiodes	1 dB	✓	×	À base d'InGaAs.
PBS	0,3 dB	✓	✓	Séparateur de polarisation.
WDM	0,5 dB	✓	✓	Wavelength Division Multiplexer. Permet de faire voyager deux longueurs d'ondes dans la même fibre.
Amplificateurs	—	✓	✓	Généralement, EDFA (Erbium-Doped Fiber Amplifier).
Circulateurs	0,8 dB	✓	✓	
Isolateurs	0,5 dB	✓	✓	
Rotateur de Faraday	0,3 dB	✓	✓	Non-réciproque, fait tourner la polarisation de 45 °
Miroir de Faraday	0,4 dB	✓	✓	Rotateur suivi d'un miroir ; réfléchit la lumière en lui imposant une rotation de polarisation de 90 °

Figure 5.8 : *Quelques composants disponibles en optique fibrée à 1550 nm. Les pertes correspondent aux pertes d'insertion, excluant les effets des connecteurs. Les colonnes SM et PM correspondent à la disponibilité de ces produits en version monomode et/ou maintien de polarisation chez les revendeurs de composants télécom habituels.*

Note sur les coupleurs fibrés

De nombreux coupleurs interviennent dans notre implémentation, et il convient d'être attentif aux performances de ceux-ci. En effet, certaines entreprises proposent des coupleurs à maintien de polarisation dont le rapport de couplage varie beaucoup avec les conditions de température. De même, certains coupleurs sont connectorisés avec des fibres PM, mais le composant lui-même n'est pas conçu spécifiquement, ce qui conduit à des dérives en polarisation à la sortie du coupleur.

Nous avons choisi des coupleurs de la société Novawave Technologies, pour lesquels nous avons mesuré une stabilité de l'ordre de 0,1 % en puissance de sortie simple, et de 0,4 % en puissance de sortie polarisée.

5.2 Source laser

Pour séparer clairement les différents états encodés, et ainsi faciliter l'analyse de sécurité, nous utilisons le régime impulsionnel. Les impulsions que nous utilisons doivent répondre à une contrainte principale : le « fond » lumineux résiduel de l'oscillateur local doit être petit par rapport au signal, en particulier pour ne pas qu'il interfère avec celui-ci lors du multiplexage. Dans la mesure où l'oscillateur local (OL) contient environ 10^8 photons par impulsion et le signal seulement quelques uns, le rapport d'extinction de l'OL doit être d'au moins 90 dB.

Dans les premières versions de notre système [56], les impulsions lumineuses étaient découpées dans un faisceau continu, à l'aide de modulateurs d'amplitude. Chaque modulateur n'éteignant le signal qu'à 30 dB au mieux, nous avons alors dû coupler plusieurs modulateurs en série. Outre le surcoût et la difficulté du contrôle de la stabilité de l'extinction, cette chaîne de modulateurs introduisait des pertes totales d'environ 10 dB, ce qui rendait le niveau d'oscillateur trop faible au niveau de la détection.

Nous avons donc choisi d'utiliser une diode laser télécom Alcatel LMI1905 en tant que source d'états cohérents, diode qui a la particularité de supporter une modulation impulsionnelle directe. Nous la modulons directement, en lui imposant des impulsions de courant très au dessus du seuil, d'une durée de 100 ns et émises toutes les 2 µs. Aucun courant ne parcourt la diode hors l'impulsion, l'intensité émise par la diode est alors nulle, hormis éventuellement une faible fluorescence (de toute façon incohérente avec le signal).

Cette diode fournit une puissance maximale de 15 mW-crête, et sa largeur spectrale est spécifiée à environ 2 MHz, soit 16 femtomètres à mi-hauteur, ce qui est très bon en théorie. Néanmoins, cette largeur correspond au régime continu et stabilisé en température. En régime impulsionnel, un « chirp » de longueur d'onde apparaît au cours de l'impulsion, comme on le voit sur les figures 5.9 et 5.10. La longueur d'onde dérive d'environ 45 pm (6 THz) au cours de l'impulsion, mais cette dérive est stable d'une impulsion sur l'autre. Nous verrons dans la section 5.4.3 que cette largeur spectrale introduit des contraintes sur l'équilibrage de l'interférence signal/OL.

5.3 Modulation

Pour ajuster l'amplitude et la phase de chaque impulsion, nous utilisons des modulateurs électro-optiques fibrés de la société Eospace. Ces modulateurs présentent des pertes d'insertion de 2 à 3 dB, ont une bande passante d'environ 30 GHz, et sont sensibles à la polarisation ; ils sont donc connectorisés avec des fibres PM Panda.

5.3.1 Principe de fonctionnement

Les modulateurs de phase sont composés d'un cristal de niobate de lithium ($LiNbO_3$), qui ne présente pas de symétrie axiale et est donc sensible à l'effet Pockels : l'indice optique du matériau dépend linéairement de la tension appliquée aux bornes du cristal.

Les modulateurs d'amplitude sont composés de deux voies montées selon une configuration Mach-Zehnder (fig. 5.11). Un cristal est placé dans l'une des voies, et la tension appliquée sur celui-ci fait varier le déphasage entre les deux ondes de sortie, qui inter-

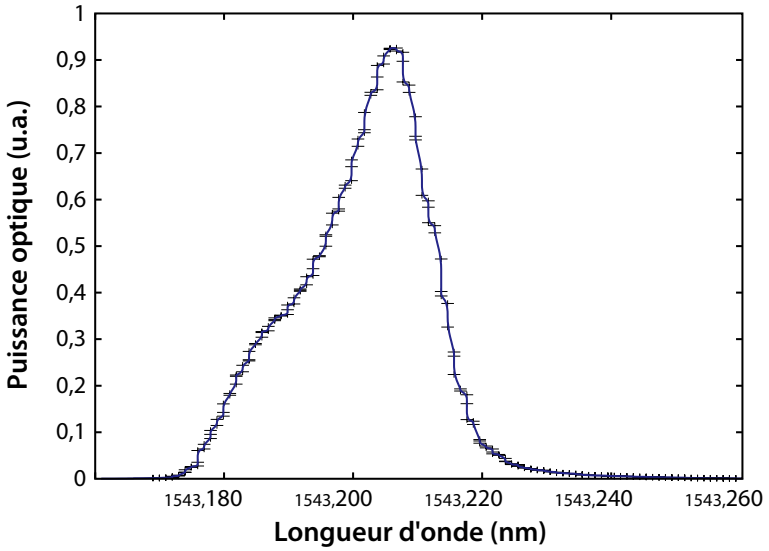


Figure 5.9 : *Spectre d'émission de la diode, en régime impulsionnel.*

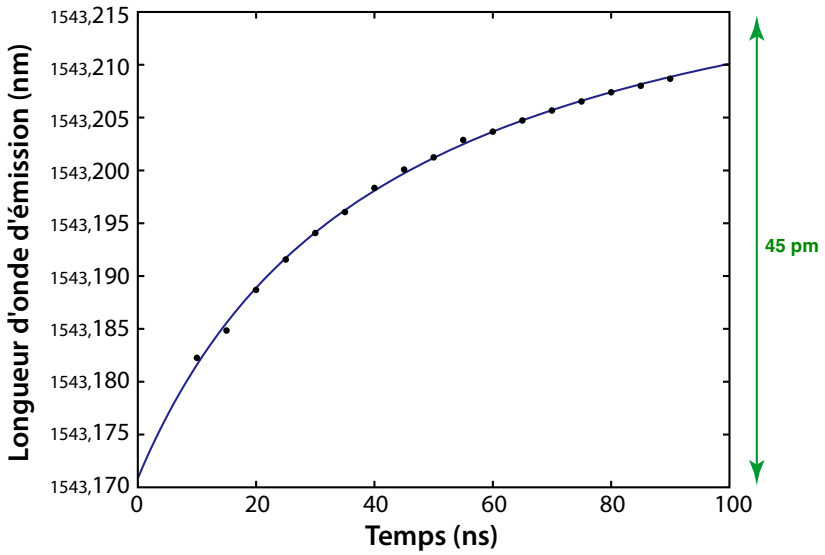


Figure 5.10 : *Dérive de la longueur d'onde d'émission de la diode en fonction du temps depuis le début de l'impulsion. Nous avons obtenu ces points en découpant des fenêtres de 5 ns dans l'impulsion de 100 ns grâce à un modulateur électro-optique, puis avec un analyseur de spectre optique.*

fèrent ensuite. L'extinction entre l'interférence constructive (transmission maximale) et destructive (extinction) est d'environ −30 dB pour nos modulateurs, et peut monter jusque −50 dB pour certains modulateurs spécifiquement conçus, au prix de pertes en ligne plus importantes.

Deux technologies de modulateurs d'amplitude existent actuellement, dénommées *Z-cut* et *X-cut*. Dans la structure *Z-cut*, l'un des guides d'onde est placé sous l'électrode

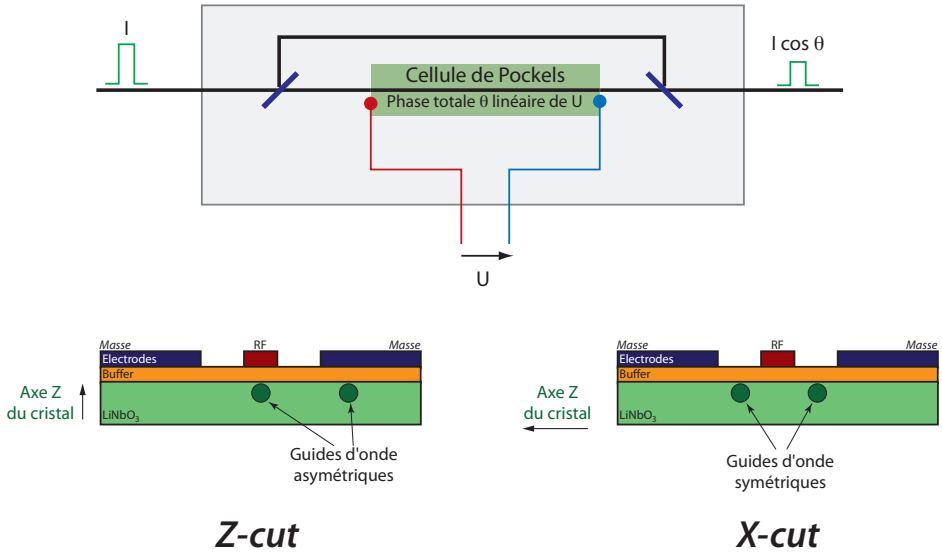


Figure 5.11 : *Schéma de principe d'un modulateur d'amplitude*

RF, ce qui améliore l'efficacité de l'effet non-linéaire, mais introduit une asymétrie entre électrodes. Au contraire, les électrodes sont symétriques dans le schéma *X-cut*, mais plus éloignées de l'électrode RF.

La technologie *Z-cut* permet ainsi d'obtenir des rapports d'extinction plus importants, des V_π plus petits et une meilleure bande passante. En revanche, dans les modulateurs *Z-cut*, la modulation d'amplitude se traduit par une modulation de phase concommitante. A l'opposé, la phase n'est presque pas affectée par le modulateur dans la technologie *X-cut*.

Puisque nous n'utilisons de toute façon pas les modulateurs à leurs limites, nous avons préféré utiliser une technologie *X-cut*, ce qui évite de devoir contrôler la modulation conjointe de phase.

5.3.2 Comportement

Courbes caractéristiques

Les caractéristiques des modulateurs (figure 5.12) correspondent bien au modèle théorique. Il est à noter que l'extinction des modulateurs d'amplitude n'est pas la même pour toutes les arches. En particulier, l'extinction de l'arche qui correspond à une tension de commande de -10 V n'a qu'un rapport d'extinction d'environ -15 dB, à comparer à environ -30 dB pour les deux pieds de l'arche centrée sur zéro. Nous avons choisi de nous placer sur la pente descendante de cette arche : transmission maximale pour $U \approx 0$ V, et transmission minimale pour $U \approx 3{,}5$ V.

Sensibilité à la température

Le niobate de lithium est un matériau particulièrement sensible aux variations de température, et le déphasage global (à tension nulle) imposé par une cellule de Pockels

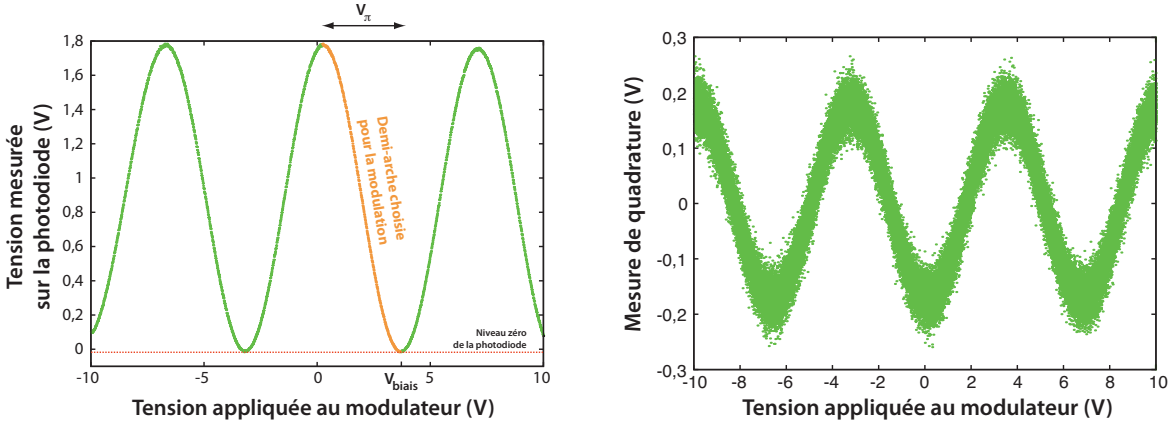


Figure 5.12 : *Caractéristiques de nos modulateurs d'amplitude et de phase.*

varie donc au cours du temps si le modulateur n'est pas parfaitement thermalisé. Pour piloter un modulateur, deux paramètres sont pertinents : le V_π, c'est-à-dire la variation de tension nécessaire pour déphaser de π la cellule, et la tension de biais, correspondant à la tension de commande pour atteindre un réglage de référence.

Pour les modulateurs d'amplitude, le V_π correspond à la tension nécessaire pour passer d'une transmission totale à une extinction totale, et on définit généralement la tension de biais comme la tension la plus proche de 0 V générant une extinction totale. En pratique, le V_π reste quasi-constant dans des conditions de température habituelles (10 °C - 30 °C) ; en revanche, la tension de biais fluctue beaucoup, typiquement d'un volt sur une échelle de 30 minutes en fonctionnement normal.

Pour les modulateurs de phase, contrôler la tension de biais (et donc la phase absolue du signal en sortie de chaque modulateur) n'a pas vraiment d'intérêt ; en effet, la phase qui nous intéresse est la phase relative signal/OL au niveau de l'interférence, et celle-ci est contrôlée par ailleurs (voir partie 6.2.3). En revanche, le V_π doit être très bien contrôlé, pour ne pas rajouter de bruit technique dans le système. La section 6.2.3 détaille les rétrocontrôles assurant le bon fonctionnement des modulateurs.

5.3.3 Modulation gaussienne

Le protocole de cryptographie quantique nécessite une modulation gaussienne des quadratures du signal. Exprimée en fonction de la phase et de l'amplitude, cette distribution correspond à une modulation uniforme de la phase sur $[0; 2\pi]$, et à une modulation selon une distribution de Maxwell pour l'amplitude :

$$\Pr(\text{Amplitude} = a) = \frac{a}{V_A N_0} e^{-\frac{a^2}{2V_A N_0}}$$

Puisque cette distribution en amplitude n'est pas bornée et autorise donc des amplitudes très grandes, il n'est pas possible de la mettre en œuvre parfaitement. Nous la tronquons donc à une amplitude $a_{\max} = 4\sqrt{V_A N_0}$, que nous associons à un réglage à transmission maximale du modulateur d'amplitude. La fraction des impulsions que

nous perdons ainsi peut être calculée par

$$1 - \int_0^{4\sqrt{V_A N_0}} \Pr(\text{Amplitude} = a)\,\mathrm{d}a = 0{,}04\%.$$

Qui plus est, la distribution n'est pas réellement continue, car les cartes d'acquisition sont par nature discrètes. Si le pas de discrétisation est très petit devant le bruit de photon, l'effet sur l'excès de bruit reste néanmoins limité.

Frédéric Grosshans a étudié dans sa thèse [3] l'effet de l'approximation des valeurs continues par une modulation discrétisée. Il conclut que, pour une information mutuelle I_{AB} de l'ordre de 1 bit, les cartes d'acquisition devraient garantir une profondeur d'acquisition de plus de 9 bits, ce qui est le cas pour nos cartes (12 bits).

Jérôme Lodewyck a quant à lui étudié [5] l'évolution de la variance réelle des données en fonction de l'amplitude de coupure. En coupant à $4\sqrt{V_A N_0}$, la variance est diminuée d'environ 0,3 %, alors que seules 0,04 % des impulsions sont perdues. En effet, les impulsions à grande amplitude contribuent davantage à la variance que les amplitudes faibles.

Avec ces considérations, nous pouvons choisir un régime « raisonnable » pour lequel l'effet de ces imperfections sur la sécurité est *a priori* limité. Nous avons choisi $a_{\max} = 4\sqrt{V_A N_0}$, de façon à ce que la variation de variance soit faible devant la variance d'excès de bruit typique du système.

5.4 Multiplexage et contrôle de la polarisation

5.4.1 Multiplexage : deux signaux, une fibre

En sortie du système d'Alice, nous disposons de deux trains d'impulsions, signal et OL, que nous devons transmettre à Bob. En pratique, il n'est pas envisageable d'utiliser deux fibres pour transmettre ces signaux. En effet, deux fibres différentes impliqueraient deux chemins optiques différents ; cumulée sur plusieurs dizaines de kilomètres, la différence de phase pourrait alors devenir difficile à contrôler.

Nous devons donc multiplexer nos deux signaux dans le même canal. Plusieurs techniques existent, directement inspirées des télécommunications classiques : multiplexage en fréquence, en temps, en polarisation... Dans notre cas, utiliser un simple multiplexage temporel pose un problème de démultiplexage : un switch optique induit 2 décibels de pertes, ce qui dégrade significativement les performances de notre système. Au contraire, un simple multiplexage en polarisation pose le problème des fuites de l'oscillateur local sur la polarisation du signal. Nous avons donc choisi un multiplexage à la fois en temps et en polarisation, qui résout les deux problèmes, en permettant un démultiplexage aisé avec un séparateur de polarisation [57].

L'oscillateur local est donc envoyé tel quel dans la fibre de transmission, alors que le signal est retardé de 400 ns et transmis selon une polarisation orthogonale au signal. Les deux signaux voyagent ainsi dans la même fibre, et subissent les mêmes contraintes : puisque les dérives de longueur et de biréfringence sont principalement thermiques, leur variation sur 400 ns peut être considérée comme nulle.

Pour réaliser le multiplexage temporel, nous avons ajouté chez Alice un système de ligne à retard sur le signal (voir figures 5.1 et 5.13). Le signal, polarisé selon l'axe

lent de la fibre, traverse un séparateur de polarisation (PBS) vers une fibre optique monomode standard de 40 mètres (soit 200 ns de délai). À l'extrémité de la fibre est installé un miroir de Faraday, composé d'un rotateur de Faraday et d'un miroir standard. Le signal subit donc une rotation de polarisation de 45 ° à l'aller, est réfléchi, et subit de nouveau une rotation de 45 ° au retour, dans le même sens qu'à l'aller. Il parcourt donc la fibre de 40 m une nouvelle fois, mais selon l'axe rapide, et sort de fait du PBS par la troisième voie [1].

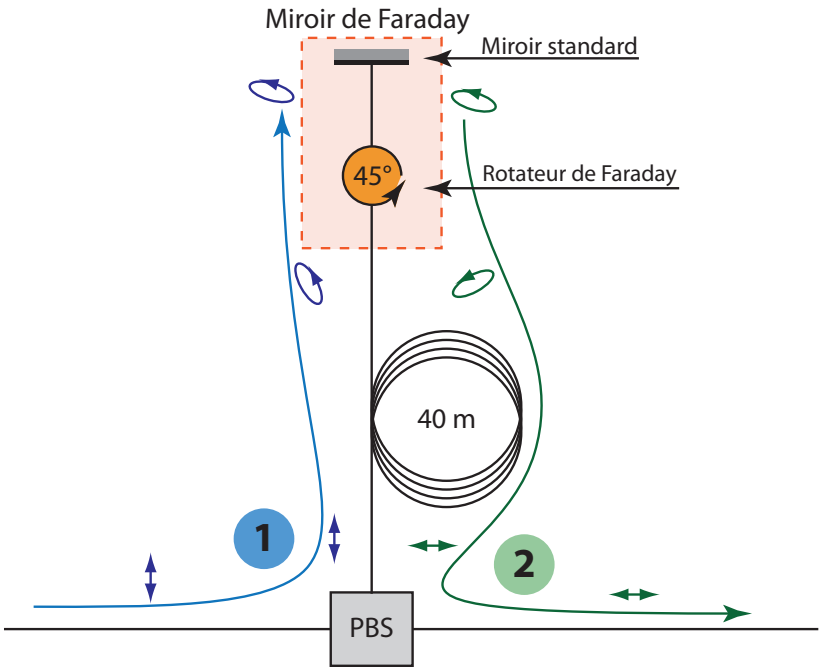


Figure 5.13 : *Schéma de montage de la ligne à délai de 400 ns. Le rotateur de Faraday est un composant non-réciproque qui impose une rotation de polarisation de $\pi/4$ indépendamment du sens de parcours. Le signal de retour (flèche verte 2) est donc orthogonal au signal d'aller (flèche bleue 1), et revient au PBS avec une polarisation orthogonale à l'axe privilégié (lent) de la fibre PM. Il sort donc par la troisième voie du PBS.*

L'intérêt de ce type de montage, par rapport à un simple circulateur, vient du fait que les dérives de polarisation subies par le signal à l'aller sont exactement compensées au retour. Il est ainsi possible de se passer d'une fibre à maintien de polarisation pour la ligne à retard, ce qui réduit d'autant le coût du système.

5.4.2 Contrôle de la polarisation et démultiplexage

Dans le cas où les signaux n'auraient pas subi de biréfringence entre Alice et Bob, il serait possible de les démultiplexer aisément avec un PBS placé à l'entrée de Bob. Néanmoins, les fibres de transmission installées sur le terrain sont généralement des

1. En pratique, l'axe rapide de la troisième voie du polariseur est soudée à l'axe lent du composant suivant. On peut ainsi faire astucieusement repasser le signal sur l'axe habituel de propagation avec une simple soudure optique.

fibres monomodes standard, et elles ne maintiennent donc pas la polarisation. Il est donc nécessaire de corriger la dérive de polarisation en entrée de Bob.

Pour ce faire, nous utilisons un contrôleur dynamique de polarisation General Photonics *Polarite III*. Le contrôleur est composé d'une fibre monomode standard sur laquelle sont fixés quatre composants piézoélectriques pilotables par quatre entrées de tension 0-12 V. Il est ainsi possible d'imposer des contraintes contrôlées sur la fibre et donc de changer l'état de polarisation de la lumière qui la traverse.

Pour trouver l'état de polarisation visé (linéaire et selon l'axe lent de la fibre PM), nous utilisons les deux mesures du système de détection présentées dans la figure 5.17 (mesure homodyne et mesure de l'oscillateur local) comme signaux de contrôle. Dans la mesure où la polarisation peut être quelconque en sortie de la fibre de transmission, nous devons d'abord trouver un réglage grossier du contrôleur, pour lequel la polarisation est proche de l'optimum. Pour ce faire, nous explorons de manière aléatoire la sphère de Poincaré, en imposant quatre critères de réussite :

- le niveau d'oscillateur local est dans une plage acceptable (entre 0,3 et 1,5 V).
- la détection homodyne ne sature pas, ce qui correspond à un niveau moyen entre -1 V et 1 V.
- la variance du signal homodyne correspond à une valeur acceptable du bruit de photon (entre 300 et 2 000 mV^2).
- les trois points précédents restent vrais dans un petit disque autour du point considéré.

L'expérience a montré que les quatre critères ne peuvent être vérifiés que dans le cas où la polarisation est presque parfaitement redressée. Une fois le réglage grossier effectué, une recherche de proche en proche, en optimisant le niveau d'oscillateur local, nous permet de trouver le point optimal. L'OL et le signal sont alors démultiplexés par un simple PBS.

Nous avons enfin réalisé un contrôle du contraste de l'interférence voie OL/voie signal, qui s'est révélé supérieur à 99,5 %, ce qui correspond à l'extinction (spécifiée à -22 dB) du PBS de démultiplexage. Un contraste limité ne se traduit de toute façon que par des pertes ajoutées, puisque les impulsions sont séparées temporellement.

5.4.3 Qualité de l'interférence OL/signal

Après démultiplexage, le signal et l'oscillateur local interfèrent dans le coupleur 50/50 de la détection homodyne. Or, la longueur d'onde de la diode dérive au cours de l'impulsion ; par conséquent, si l'interféromètre n'est pas à différence de marche nulle, les deux impulsions n'auront pas la même fréquence instantanée et n'interféreront pas correctement (dans l'optique d'une détection *homodyne*). Essayons de quantifier l'effet d'une différence de marche non-nulle et la précision nécessaire pour un fonctionnement correct :

- Nous considérons un modèle classique d'interférence, en ne prenant pas en compte l'effet du bruit de photon : on peut alors écrire les deux ondes qui interfèrent sous la forme [2] $E_S = A_S e^{i(2\pi t f(t)+\varphi)}$ et $E_{OL} = A_{OL} e^{2i\pi(t+\tau)f(t+\tau)}$.

2. Pour le calcul, nous avons besoin d'approcher la dérive de fréquence de la diode laser par ue expression analytique. Par exemple, la longueur d'onde $\lambda(t)$ correspondant à la figure 5.10 peut être approximée de façon *ad hoc* par $\lambda(t) = 1543{,}20 + 1{,}08\,t - 0{,}02\,e^{-0{,}03636\,t}$ où $\lambda(t)$ est exprimé en nm

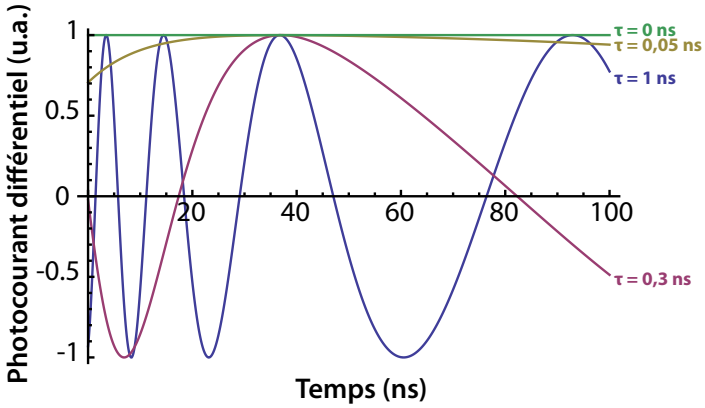


Figure 5.14 : *Modélisation du photocourant différentiel en entrée de l'amplificateur de charge en fonction de la position temporelle dans l'impulsion, pour une tentative de mesure de la quadrature X. Dans le cas d'une différence de marche nulle de l'interféromètre, la valeur du photocourant devrait être 1 pendant toute l'impulsion.*

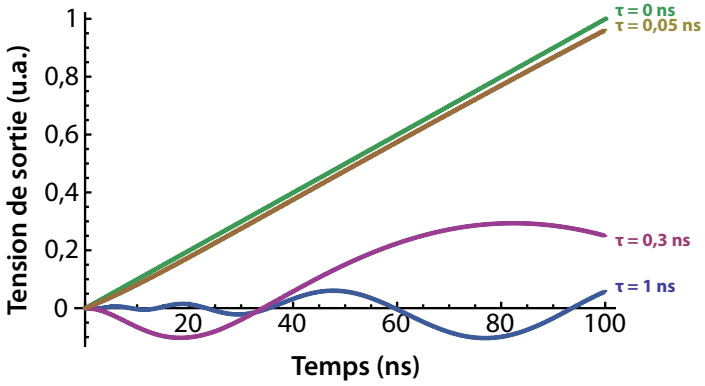


Figure 5.15 : *Modélisation de la tension de sortie de l'amplificateur de charge, en fonction de la position temporelle dans l'impulsion. À un facteur près, elle correspond au signal observé sur la détection homodyne.*

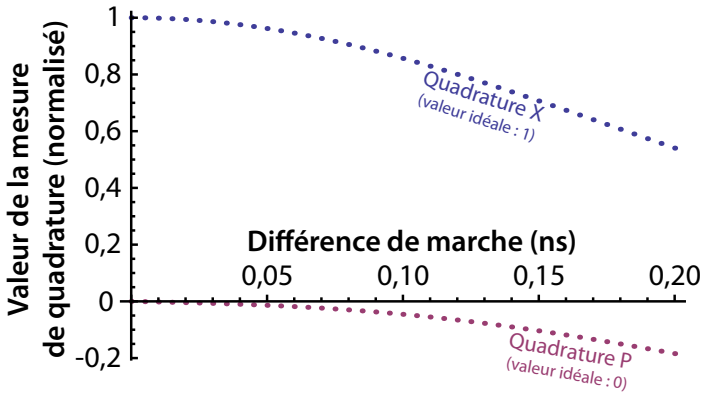


Figure 5.16 : *Modélisation de la tension mesurée à $t = 100$ ns, en fonction de la différence de marche de l'interféromètre, pour une tentative de mesure de X et P. Idéalement, la mesure de X devrait être de 1, et celle de P de 0.*

La différence des photocourants s'exprime

$$\Delta I(t,\tau) = A_S A_{OL} \cos\left(\varphi + 2\pi t f(t) - 2\pi(t+\tau) f(t+\tau)\right) \tag{5.1}$$

Bob impose une phase $\varphi = 0$ ou $\pi/2$ de façon à mesurer la quadrature X ou P. Pour une différence de marche nulle, la différence ΔI vaut $A_S A_{OL}$ pour $\varphi = 0$ et est nulle pour $\varphi = \pi/2$. En revanche, si la différence de marche n'est pas nulle, le signal et l'OL n'ont pas la même fréquence instantanée, et le détecteur se comporte de façon similaire à une détection hétérodyne : les franges d'interférences défilent pendant l'impulsion, d'autant plus vite que la différence de fréquence est grande. La figure 5.14 illustre ce défilement pour différentes différences de marche.

- Dans la détection homodyne, la différence des photocourants est intégrée sur 100 ns. Nous oublions ici le filtre RC qui coupe les fluctuations plus grandes que 10 MHz. Le signal de sortie s'écrit alors

$$U_{DH}(t,\tau) \propto \int_0^t \Delta I(t',\tau)\mathrm{d}t'$$

La figure 5.15 montre la forme du signal de sortie : pour une différence de marche nulle ou presque nulle, le signal est quasiment linéaire. En revanche, quand la différence de marche augmente, nous voyons apparaître des battements dans l'impulsion. Cette modélisation correspond assez bien à ce que nous observons en pratique sur le signal de détection homodyne.

- Enfin, notons que la mesure de quadrature est réalisée à la fin de l'impulsion. La figure 5.16 représente la valeur de cette mesure, pour les deux quadratures, en fonction de la différence de marche. Idéalement, la mesure de X devrait être de 1 ; l'écart à cette valeur avec la différence de marche peut être assimilée à une perte de contraste. On voit alors que pour assurer une mesure de X et P correcte à mieux que 99 %, il faut que la différence de marche soit plus petite que 0,02 ns, c'est-à-dire 4 mm de fibre.

Equilibrage des voies signal et oscillateur local

Nous retrouvons à travers cet équilibrage une difficulté majeure du travail avec des fibres optiques. Il faut ajuster la longueur de deux chemins optiques, chacun d'environ 100 m, à mieux que 4 mm ; or, la longueur des jarretières commercialisées est spécifiée à environ 3 cm.

Pour résoudre ce problème, nous avons tout d'abord dû trouver un réglage grossier des longueurs pour lequel l'interférence soit visible. Celle-ci commence à être perceptible pour une différence d'environ 40 cm. Nous avons ensuite acheté une trentaine d'échantillons de jarretières monomodes connectorisées, et aussi différentes en longueur que possible. En essayant différentes combinaisons de ces fibres et en optimisant le contraste de l'interférence, nous avons finalement trouvé une combinaison adéquate, fournissant une différence de marche résiduelle de 2 mm ($\pm$ 1 mm), soit un contraste d'environ 99,5 %.

et t en ns.

5.5 Détection homodyne

Le principe physique de la détection homodyne a été présenté dans la partie 2.2.2, et résumé dans l'équation de principe (2.27). Nous présentons dans cette section nos choix d'implémentation puis les résultats que nous avons obtenus.

5.5.1 Implémentation

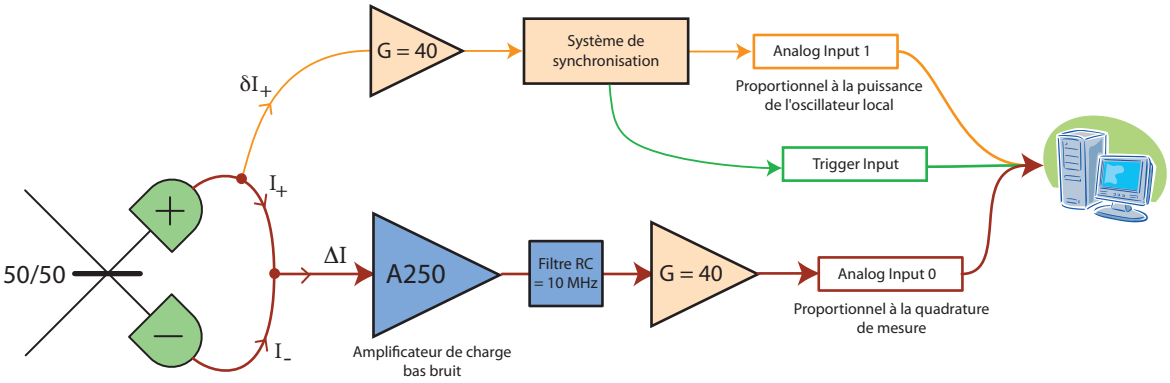


Figure 5.17 : *Réalisation expérimentale de la détection homodyne.*

La détection homodyne que nous utilisons est tout d'abord constituée d'un coupleur PM 50/50 de la société NovaWave, dans lequel interfèrent le signal et l'oscillateur local. Chacune des sorties de ce coupleur est connectée à une photodiode rapide. Il s'agit de photodiodes PIN (EPM 605) de la société JDS Uniphase, présentant une bande passante de 2 GHz, ce qui est largement suffisant pour résoudre nos impulsions de 100 ns, ainsi qu'une efficacité quantique d'environ 1,02 A/W à 1 550 nm, c'est-à-dire 82 %.

Les deux photodiodes sont montées de manière à ce que les photocourants soient soustraits électriquement, par une simple loi des nœuds. Le photocourant différentiel entre ensuite dans un préamplificateur de charge AmpTek A250 à très bas bruit, qui amplifie le signal d'un facteur[3] d'environ 3 000. Ce pré-amplificateur fonctionne de façon similaire à un circuit intégrateur à amplificateur opérationnel : la tension de sortie est proportionnelle à $\frac{Q}{C_f}$, où Q est la charge en entrée (c'est-à-dire l'intégrale de l'intensité), et C_f la capacité du circuit. Pour que l'amplification soit très forte, la capacité choisie pour l'intégration est très faible (typ. 1 pF). Néanmoins, cette faible capacité implique que l'amplificateur sature très rapidement, et ne peut donc

3. Calcul approximatif :
   - Le bruit de photon possède une amplitude de 60 mV en sortie de l'amplificateur MAX4107, chargé sur 50 Ω. Ceci nous fournit une intensité de 1,2 mA en sortie, c'est-à-dire 30 µA en entrée.
   - L'intensité du bruit de photon est de l'ordre de $\sqrt{I_{OL}} = \sqrt{10^8} = 10^4$ photons. La charge totale par impulsion est $10^4\, e \approx 10^{-15}$ C. Puisque l'impulsion dure 10^{-7} s, l'intensité en entrée de l'AmpTek est de l'ordre de $\frac{10^{-15}}{10^{-7}} = 10^{-8}$ A.
   - Le gain total est donc de l'ordre de $\frac{30\ \mu A}{10^{-8} A} \approx 3\,000$.

tolérer que des charges d'entrée très faibles, ce qui justifie son utilisation comme *pré*-amplificateur.

Après pré-amplification, le signal est ensuité filtré par un circuit passe-bas, de bande passante 10 MHz, puis ré-amplifié par un amplificateur standard Maxim MAX4107 présentant une bande passante d'environ 350 MHz pour un gain $G = 40$. Le circuit électronique, mis au point par André Villing (et détaillé dans la thèse de J. Wenger [4]), a été conçu pour limiter le bruit électronique, qui est d'environ 2 mV d'écart-type dans des conditions de laboratoire.

Puissance ou énergie ?

Les preuves de sécurité présentées dans le cadre des variables continues sont fondées sur les variances et bruits du signal ; néanmoins, il est tout aussi nécessaire de se prémunir contre des attaques de l'espion sur la forme de l'oscillateur local [58]. Par exemple, on peut imaginer que l'espion modifie le niveau d'oscillateur local sur la première moitié de l'impulsion et que la mesure de Bob ne porte que sur la deuxième moitié ; Bob ne se rend alors pas compte de l'espionnage.

Nous avons implémenté deux contre-mesures pour ce types d'attaques. Tout d'abord, pour éviter que l'espion ne manipule le niveau global de l'oscillateur local, celui-ci doit être contrôlé pour chaque impulsion. Ceci est réalisé en pratique grâce à la voie auxiliaire de la détection homodyne. Par ailleurs, pour rendre inutile une attaque de l'espion sur la forme de l'oscillateur local, la détection homodyne doit mesurer l'énergie globale de l'impulsion plutôt que sa puissance instantanée. Le circuit électrique que nous avons choisi permet de réaliser cette opération, puisqu'il se charge en continu pendant la durée de l'impulsion, de façon que le signal amplifié est proportionnel à la charge totale accumulée pendant l'impulsion. Nous réalisons donc la mesure 100 ns après le début de l'impulsion, de façon à ce qu'elle soit proportionnelle à $\int_0^{100\text{ ns}} \Delta I(t)$, c'est-à-dire proportionnelle à l'énergie totale dans l'impulsion.

5.5.2 Contrôle de la bonne soustraction des photocourants

Dans le modèle théorique présenté dans la section 2.2.2, les coupleurs et photodiodes de la détection homodyne sont parfaits, et les chemins optiques identiques pour chaque photodiode. Les photocourants se soustraient donc parfaitement, de façon à ne conserver que le terme proportionnel à la quadrature mesurée.

En pratique, les composants ne sont pas parfaits, et les transmissions et longueurs des deux voies de sortie du coupleur sont différentes. Les photocourants ne se soustraient donc pas parfaitement ; or, compte tenu du gain global de la détection homodyne (environ 120 000), celle-ci peut rapidement saturer si les réglages ne sont pas bons.

Signal d'une détection homodyne imparfaite

Pour modéliser les imperfections des coupleurs et des photodiodes, nous introduisons les transmissions h_+ et h_- des voies, sous la forme $h_+ = h\,(1+\varepsilon)$ et $h_- = h\,(1-\varepsilon)$, dans l'équation (2.27). Dans ces expressions, ε dépend *a priori* du temps, puisque les

photodiodes ne répondent pas instantanément à l'entrée optique. On trouve :

$$\frac{1}{h}\Delta\hat{I} \propto (1+\varepsilon)\,\hat{n}_+ - (1-\varepsilon)\,\hat{n}_- = (\hat{a}_+^\dagger\hat{a}_+ - \hat{a}_-^\dagger\hat{a}_-) + 2\varepsilon\,(\hat{a}_+^\dagger\hat{a}_+ + \hat{a}_-^\dagger\hat{a}_-) \quad (5.2)$$

$$= 2\,(\hat{a}_{OL}^\dagger\hat{a}_S + \hat{a}_S^\dagger\hat{a}_{OL}) + 4\varepsilon\,(\hat{a}_S^\dagger\hat{a}_S + \hat{a}_{OL}^\dagger\hat{a}_{OL}) \quad (5.3)$$

Si l'on suppose $I_S \ll I_{OL}$ et $N_0 \ll I_{OL}$, on trouve :

$$\frac{1}{h}\Delta\hat{I} = \underbrace{\frac{2\sqrt{I_{OL}}}{\sqrt{N_0}}\hat{X}_{S/OL}}_{\text{D.H. idéale}} + \underbrace{4\varepsilon(t)I_{OL}}_{\text{valeur moyenne}} \quad (5.4)$$

On voit donc que le signal de sortie de la détection homodyne, plutôt que de fluctuer autour de zéro, est centré sur un signal dépendant du temps, proportionnel à l'intensité de l'oscillateur local et au déséquilibre de la transmission des deux voies du coupleur. Il est possible d'évaluer la précision avec laquelle les deux voies doivent être équilibrées : si l'on considère que le niveau moyen ne doit pas dépasser l'écart-type du bruit de photon $\sqrt{N_0}$, on obtient $4|\varepsilon|I_{OL} \leqslant \frac{2\sqrt{I_{OL}}}{\sqrt{N_0}}\sqrt{N_0}$, c'est-à-dire

$$|\varepsilon(t)| \leqslant \frac{1}{2\sqrt{I_{OL}}} \quad (5.5)$$

Le niveau d'oscillateur local est de l'ordre de 10^8 photons par impulsion, ce qui donne $|\varepsilon(t)| \leqslant 10^{-4}$.

Rappelons que le déséquilibre peut être positif ou négatif, et que nous mesurons la tension de sortie de la détection homodyne 100 ns après le début de l'impulsion. Deux paramètres sont donc pertinents pour évaluer l'équilibrage. D'un côté, la valeur maximale du signal au cours de l'impulsion : $\varepsilon_m = \max|\varepsilon(t)|$; elle donne « l'amplitude » de la valeur moyenne du signal, et si ε_m est trop grand, l'amplificateur de charge sature. De l'autre, la valeur moyenne du signal mesuré en pratique, c'est-à-dire $\varepsilon_{100} = \int_0^{100\text{ ns}}\varepsilon(t)$, qui devrait être proche de zéro.

Équilibrage de la longueur des voies

Si les deux impulsions ne se recouvrent pas parfaitement, et sont décalées d'un temps τ, la première va charger l'amplificateur de charge pendant τ, ce qui déséquilibre le signal de détection homodyne. De la même façon que ci-dessus, le décalage maximal admissible correspond au temps nécessaire pour charger l'amplificateur avec l'équivalent de 10^4 photons, à comparer aux 10^8 photons contenus dans l'impulsion de durée $\tau_{\text{tot}} = 100$ ns. Au premier ordre, on peut donc exprimer $\tau_{\max} = \frac{10^4}{10^8}\tau_{\text{tot}} = 0{,}01$ ns.

Il faut donc équilibrer les voies à mieux que $\frac{c}{n}\tau_{\max} = 2$ mm. Avant équilibrage, les longueurs de fibre des deux voies de sortie du coupleur ne diffèrent en longueur que de 8 mm. Pour atteindre les 2 mm, nous avons le plus souvent utilisé la même technique que pour l'équilibrage de l'interférence signal/oscillateur local, à savoir l'insertion de fibres courtes de la bonne longueur.

Alternativement, pour limiter les pertes, il est aussi possible d'ajuster la longueur des fibres à environ 1 mm près en les coupant à la bonne longueur et en les soudant avec une soudeuse optique. Quoiqu'efficace, c'est une option plus définitive et assez délicate à réaliser, que nous n'avons choisie que pour l'une des détections utilisées lors de cette thèse.

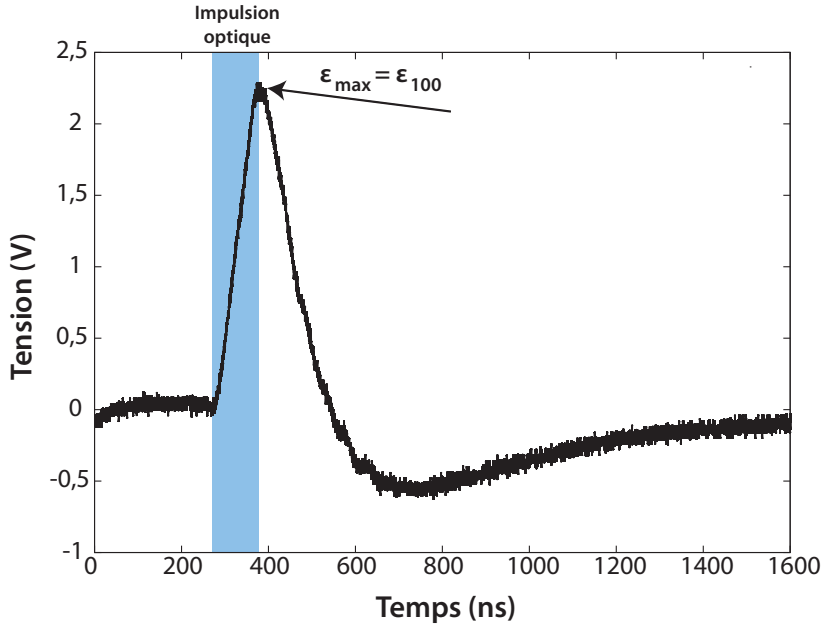


Figure 5.18 : *Signal de détection homodyne déséquilibrée. Pendant les 100 ns de l'impulsion optique, l'amplificateur de charge intègre l'impulsion, ce qui fournit un signal quasi-linéaire. L'amplificateur se décharge ensuite, pendant environ 1 µs. La valeur maximale atteinte par le signal, proportionnelle à ε_m, est alors atteinte à la fin de l'impulsion, c'est-à-dire que $\varepsilon_m = \varepsilon_{100}$.*

Équilibrage des transmissions

Une fois la longueur ajustée, nous devons rendre les deux voies aussi similaires que possibles pour obtenir une bonne qualité d'équilibrage (10^{-4}).

Dans un premier temps, il s'agit de choisir deux photodiodes aussi bien appariées que possible, en terme d'efficacité quantique et de réponse, ainsi qu'un coupleur aussi équilibré que possible. Lors de l'achat, le rapport de couplage des coupleurs que nous utilisons n'est spécifié qu'à environ 5 % près. Ainsi, notre coupleur 50/50 a un rapport réel de couplage de 54,4/45,6. Les photodiodes ont quant à elles des efficacités quantiques de 1,01 et 1,03 A/W, et des capacités internes de 0,50 et 0,55 pF. Ces caractéristiques conduisent à un déséquilibre initial $\varepsilon_m \approx 5\,10^{-2}$.

Nous pouvons ensuite équilibrer les puissances optiques en atténuant l'une des deux voies. Cet équilibrage est délicat, et l'une des meilleures méthodes que nous ayons trouvées est de simplement courber la fibre de façon à lui imposer un rayon de courbure suffisamment petit pour que des pertes apparaissent dans la fibre (voir partie 5.1.2). Fixer la fibre avec du ruban adhésif suffit à assurer une stabilité correcte des pertes en courbure. Nous obtenons ainsi un équilibrage à environ $\varepsilon_m \approx 10^{-2}$ et $\varepsilon_{100} \approx 10^{-4}$.

Le déséquilibre mesuré ε_{100} est acceptable, mais ε_m reste un peu grand, du fait de la différence de réponse des deux photodiodes. En espace libre, de nombreux degrés de liberté permettent de jouer sur cette réponse (focalisation, point d'impact, angle d'incidence) — mais ce n'est pas le cas en optique fibrée, la fibre étant sertie à la

photodiode. Le seul degré de liberté est la tension de polarisation des photodiodes, qui change très légèrement le temps de réponse des photodiodes. Le meilleur réglage que nous ayons observé est $\varepsilon_m \approx 3\ 10^{-3}$.

Avec ces réglages, nous obtenons un signal qui ne sature pas, mais dont la valeur moyenne peut fluctuer au cours de l'impulsion (voir figure 5.19).

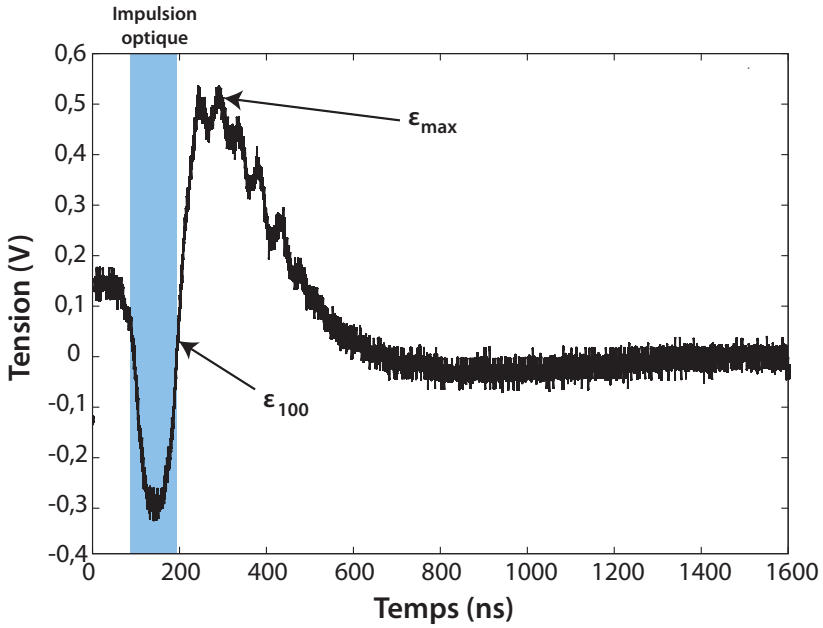


Figure 5.19 : *Signal de détection homodyne équilibrée au mieux. Les deux voies de la détection homodyne sont alors équilibrées de façon à ce que ε_{100} (et donc la mesure) soit proche de zéro. Nous voyons qu'il reste néanmoins une oscillation sur le signal, qui est due à la différence de réponse des deux photodiodes.*

5.5.3 Caractéristique de la détection

Pour pouvoir être utilisée dans un système de cryptographie quantique, la détection homodyne doit être limitée par le bruit de photon. Pour le vérifier, nous utilisons le fait que la variance du bruit de photon varie de manière linéaire avec l'intensité optique, alors que la variance d'un bruit classique est quadratique par rapport à celle-ci. De façon générale, la courbe caractéristique de la détection homodyne (variance du signal en fonction de la puissance d'oscillateur local) est la somme de trois bruits :

- Le bruit de photon, qui forme une droite passant par l'origine.
- Le bruit électronique, qui ne dépend pas du niveau d'oscillateur local, et correspond donc à l'ordonnée à l'origine.
- Le bruit optique classique, qui est quadratique du niveau d'oscillateur local.

La courbe caractéristique de notre détecteur est présentée dans la figure 5.20. On voit clairement que, pour les puissances que nous utilisons, il n'y a pas de composante quadratique dans la caractéristique : la détection est donc bien limitée par le bruit de photon.

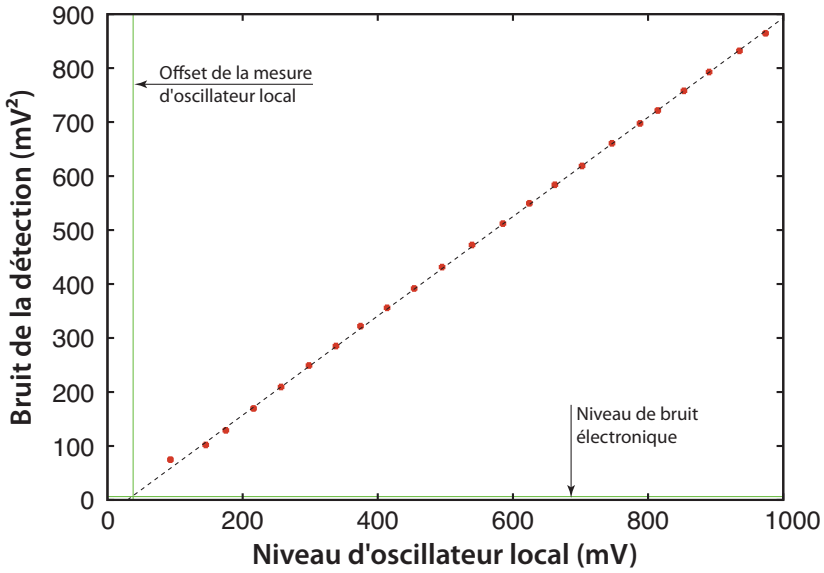


Figure 5.20 : *Caractéristique de la détection homodyne. Le circuit de mesure du niveau d'oscillateur local présente un offset de 36,1 mV, et la détection homodyne est affectée par un bruit électronique de 6,27 mV². L'expression de la variance du bruit de photon peut donc être approximée par la droite* N_0 = 0,924(U_{OL} - 36,1), *exprimés en mV et mV².*

6 Intégration et pilotage du système optique

> You see, wire telegraph is a kind of a very, very long cat. You pull his tail in New York and his head is meowing in Los Angeles. Do you understand this ? And radio operates exactly the same way : you send signals here, they receive them there. The only difference is that there is no cat.
>
> Albert Einstein

Une partie du travail de thèse a consisté à intégrer le système de cryptographie quantique de façon à obtenir un démonstrateur. Des contraintes de transportabilité et de compacité sont donc apparues, de même que la nécessité d'un programme unique, permettant à un opérateur peu averti de faire facilement fonctionner le système. Nous présentons dans ce chapitre les opérations d'intégration du montage fibré dans un boîtier rackable et détaillons la structure du programme complet de pilotage du système, qui assure en particulier le rétrocontrôle des différents composants.

6.1 Intégration du montage optique et électronique

Les systèmes d'Alice et de Bob ont chacun été intégrés dans deux boîtiers rackables de 19 pouces. Un boîtier 3U (5,25 pouces de hauteur) contient la totalité de l'optique, ainsi que l'électronique directement liée aux composants optoélectroniques : chez Alice, circuit d'alimentation de la diode, et circuit amplificateur de la photodiode ; chez Bob, électronique de la détection homodyne et circuit de régénération de la synchronisation. Un boîtier 4U (7 pouces) contient l'électronique restante, les ordinateurs de contrôle, les alimentations de l'électronique ainsi que les cartes d'acquisition et de génération de signaux.

6.1.1 Boîtier optique

Fibres et composants fibrés

Nous avons choisi de fixer la totalité des composants fibrés sur une plaque épaisse (1 cm) d'aluminium taraudé. Les fibres sont placées dans des roues de lovage fines vissées dans la plaque, ce qui permet de s'assurer de leur stabilité mécanique. Par ailleurs, la plaque joue le rôle d'un réservoir thermique relativement important ; elle n'empêche pas les fluctuations globales de température dans le boîtier, mais permet de limiter l'impact des fluctuations rapides sur les composants sensibles. En particulier, les modulateurs électro-optiques se sont montrés plus stables une fois fixés sur la plaque, métal contre métal.

Isolation mécanique et vibratoire de l'interféromètre

Notre montage optique peut être vu comme un grand interféromètre, délocalisé entre Alice et Bob, puisque le signal et l'oscillateur local sont séparés chez Alice mais interfèrent chez Bob. Pour que la génération de clés fonctionne correctement, il est nécessaire de s'assurer que l'interférence est bonne. Nous avons vu, dans le chapitre précédent, qu'un contrôle délicat de la longueur des deux chemins optiques et de la puissance des deux faisceaux est nécessaire. Dans la mesure où la phase du signal porte de l'information, il faut aussi s'assurer que la phase relative entre OL et signal est bien contrôlée.

Dans la fibre de transmission, les signaux sont multiplexés et leur phase relative ne change donc quasiment pas. Il reste environ 100 mètres pendant lesquels les signaux sont séparés, parmi lesquels 80 m correspondent aux lignes à retard de 400 ns. Cette longueur est assez importante, et toute fluctuation thermique ou mécanique peut induire une modification de la phase de l'impulsion qui s'y propage. En pratique, nous avons vu deux effets majeurs :

- une dérive lente mais très importante de la phase, du fait des fluctuations thermiques dans la fibre et la dilatation qui en découle. Cette dérive peut être considérée comme linéaire sur une seconde, et la vitesse de dérive est de l'ordre de 2π toutes les 30 secondes quand les boîtiers sont stabilisés thermiquement (5 à 10 minutes après leur fermeture).
- une fluctuation rapide de la phase, à des fréquences typiques de 100 – 1000 Hz. Cette fluctuation est directement reliée à l'environnement vibratoire des boîtiers, et peut être très forte (amplitude de plus de π, c'est-à-dire phase indéfinie à l'échelle de la seconde) si, par exemple, les boîtiers sont placés en suspension dans un rack contenant beaucoup de ventilateurs.

La dérive lente ne nous pose pas de problème, et la section 6.2.3 décrit le rétrocontrôle permettant sa gestion. En revanche, il ne nous est pas possible de contrôler la fluctuation de phase due aux vibrations, car elle est à la fois trop rapide et fortement non périodique. Nous avons donc dû isoler le montage optique, de façon à limiter l'impact des vibrations. Pour ce faire, nous avons soigneusement fixé les fibres à la plaque de support, et avons placé des plots anti-vibratoires entre la plaque et le boîtier qui la supporte. Les composants optiques sont ainsi mieux découplés du boîtier externe, ce qui a permis de faire fonctionner le démonstrateur dans des conditions réelles raisonnablement bruitées.

Isolation électrique de la détection homodyne

Le circuit électronique de la détection homodyne est très sensible aux perturbations électriques et électromagnétiques. En particulier, la partie du montage placée avant l'amplificateur de charge forme une boucle électrique d'environ 1 cm de longueur, mais le gain de l'amplificateur ($\approx 120\,000$) est tel que toute onde électromagnétique captée par cette micro-antenne produit un bruit significatif, voire très important, en sortie de détecteur.

Nous avons donc utilisé une alimentation linéaire de tension, plutôt qu'une alimentation à découpage, pour alimenter le circuit. Par ailleurs, nous avons placé toute l'électronique du détecteur, ainsi que les photodiodes, dans un boîtier métallique jouant le rôle de cage de Faraday, ce qui a permis d'éliminer les perturbations électromagnétiques. La variance standard du bruit électronique généré par le circuit est alors de 4 mV^2, quand seule la détection est alimentée. Quand les autres circuits de Bob (contrôleur de polarisation, circuit de synchronisation) sont aussi alimentés, le bruit électronique atteint 6 mV^2, de façon stable. Enfin, la qualité de la masse des circuits électronique joue un grand rôle dans la stabilité. Dans nos premières implémentations, des câbles trop longs ou mal isolés formaient des boucles de masse ou des antennes captant le bruit électromagnétique ambiant, et nous avons observé des pics de bruit atteignant 25 mV^2.

6.1.2 Boîtier électronique

Les boîtiers électroniques contiennent principalement les ordinateurs de contrôle. Ils sont conçus pour la fixation d'une carte-mère ATX standard, ainsi que des disques durs et périphériques de lecture habituels des ordinateurs de bureau. Nous avons donc monté une configuration minimale, construite autour d'un processeur Intel Core 2 Quad Q6600, qui comporte quatre cœurs (c'est-à-dire que le processeur est divisé en quatre « mini-processeurs » capables de fonctionner en parallèle).

Cartes de conversion analogique-numérique

L'acquisition des données est réalisée grâce à deux cartes d'acquisition/émission de données de la société National Instruments. La carte rapide PCI-6111 possède deux voies d'entrée en tension (résolution 12 bits), et deux voies de sortie en tension (résolution 16 bits), qui peuvent être synchronisées grâce à une voie *trigger* analogique, à des taux d'échantillonages allant jusque 5 MHz en théorie et 1 MHz en pratique pour notre système. Le temps de montée de la modulation en sortie est d'environ 200 ns, ce qui est suffisamment petit par rapport à notre fréquence d'émission de 500 kHz (2 µs).

Ces cartes ont un déclenchement simultané de toutes les acquisitions et émissions conditionnées au signal trigger. Ceci nous permet donc d'acquérir et d'émettre tous les signaux en même temps, ce qui est crucial car leur valeur change à chaque impulsion. Il s'agit, chez Alice, des tensions de consigne des modulateurs d'encodage et de la tension de sortie de la photodiode de contrôle (deux sorties, une entrée) ; chez Bob, de la tension de consigne du modulateur de phase et des deux tensions de sortie de la détection homodyne (une sortie, deux entrées).

La carte PCI-6704 est une carte analogique lente, permettant d'émettre des tensions quelconques entre −10 V et 10 V, variant avec des fréquences d'au plus 100 kHz. Nous l'utilisons pour émettre les tensions de commande variant lentement : la tension de commande du deuxième modulateur d'amplitude, qui contrôle le niveau fin de sortie d'Alice, et les quatre voies d'entrée du contrôleur de polarisation de Bob.

Ces cartes sont directement enfichées dans les cartes-mères, et reliées à des borniers placés dans les boîtiers optiques grâce à des câbles dédiés.

Générateur quantique de nombres aléatoires

Pour que le choix des amplitude, phase, et quadrature de mesure de chaque impulsion soit non biaisé, nous utilisons un générateur quantique de nombres aléatoires Quantis, de la société idQuantique. Ce générateur utilise une source de photons uniques transmise par une lame semi-réfléchissante suivie par deux détecteurs ; les deux bits 0 et 1 dépendent alors de la photodiode sur laquelle le photon est détecté. L'aléa de ce générateur repose sur un principe quantique, ce qui est crucial pour ne pas introduire de faille classique dans notre protocole quantique, et il est en mesure de générer jusque 16 Mbit/s de bits aléatoires. Nous avons besoin, en pratique, de 12 Mbit/s pour la génération des 2×12 bits d'amplitude et de phase codés à 500 kHz, et de 500 kbit/s pour le choix de la quadrature de mesure ; un générateur dans chaque ordinateur est donc suffisant pour fournir l'aléa nécessaire. Ce générateur est fourni avec une interface PCI, ce qui permet de l'enficher directement dans la carte-mère de l'ordinateur et de l'utiliser de façon native dans le programme de gestion du système.

6.2 Structure du programme de pilotage

Le programme global gérant le protocole de cryptographie quantique est composé de deux parties : un sous-programme de pilotage de la transmission quantique et un sous-programme de gestion des étapes classiques du protocole. Nous présentons ici la première partie du programme ; la seconde sera présentée dans la partie III de ce manuscrit, dédiée au traitement classique des données.

6.2.1 Emission des données

Les impulsions sont émises suivant une structure spécifique, présentée ci-dessous :

- La structure principale est composée de *blocs* de 50 000 impulsions (figure 6.1). Ces blocs correspondent à l'unité de traitement des cartes d'acquisition, c'est-à-dire que les données sont envoyées bloc par bloc aux cartes. La communication avec les cartes d'acquisition a donc lieu toutes les 100 millisecondes.
- Un bloc est subdivisé en 500 *datagrammes* de 100 impulsions.
- Un datagramme est subdivisé en 18 *impulsions-test* et 80 *impulsions de données*, séparées par 2 *impulsions-bornes* (figure 6.2).

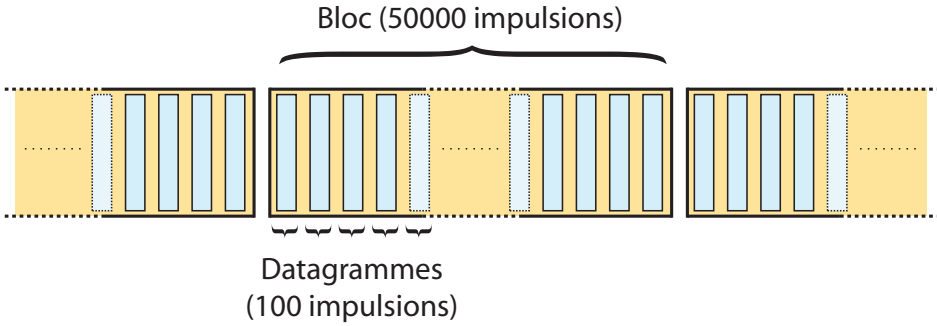


Figure 6.1 : *Structure d'un bloc de données de 50 000 impulsions*

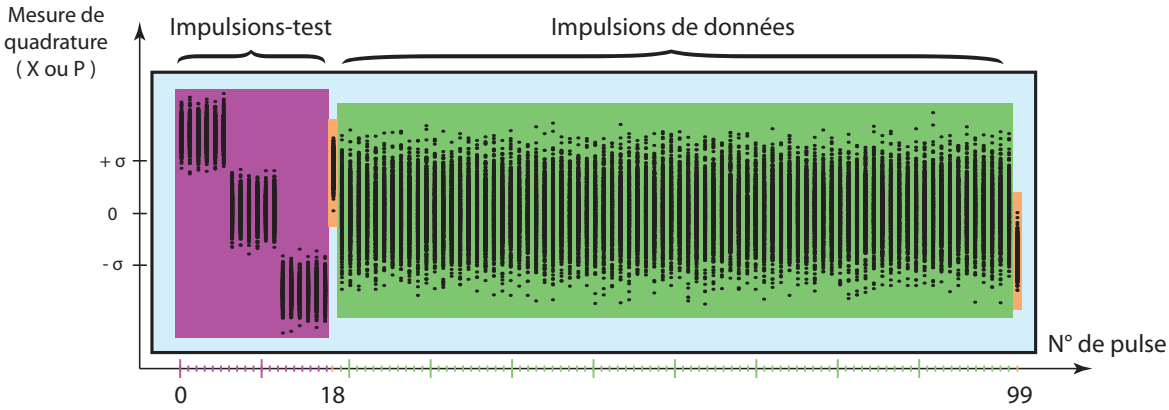


Figure 6.2 : *Structure d'un datagramme de 100 impulsions. Les points noirs correspondent aux 50 000 valeurs de quadrature mesurées dans un bloc, repliées ici modulo 100.*

Objectif des impulsions-test

Les 18 impulsions-test sont modulées avec une amplitude maximale ($4\sqrt{V_A N_0}$), et divisées en 3×6 impulsions modulées avec des phases de consigne respectives de 0, $2\pi/3$ et $4\pi/3$. Bob mesure ainsi, sur un bloc de données, $6\times500 = 3\,000$ échantillons de chacun des trois groupes d'impulsions-test, qui correspondent aux quadratures $\hat{X}_\theta$, $\hat{X}_{\theta+2\pi/3}$ et $\hat{X}_{\theta+4\pi/3}$, où θ est la phase relative entre le signal et l'oscillateur local.

Si θ ne varie pas trop vite, Bob peut moyenner ses mesures pour s'affranchir du bruit de photon. Il détermine ainsi trois tensions, qui correspondent aux mesures interférométriques *classiques* associées aux quadratures envoyées par Alice. Nous notons ces tensions $U_0 = \overline{v}_\theta$, $U_2 = \overline{v}_{\theta+2\pi/3}$ et $U_4 = \overline{v}_{\theta+4\pi/3}$.

Grâce à ces tensions, Bob peut déterminer, à l'aide de relations trigonométriques simples, trois paramètres importants du système :

- La tension moyenne M de la détection homodyne :

$$M = \frac{1}{2}(U_0 + U_2 + U_4)$$

- L'amplitude A des impulsions-test :

$$A = \left(\frac{2}{3} \left[(U_0 - M)^2 + (U_2 - M)^2 + (U_4 - M)^2 \right] \right)^{\frac{1}{2}}$$

- La phase relative φ entre signal et oscillateur local :

$$\cos(\varphi) = \frac{3}{A}(2U_0 - U_2 - U_4) \quad \text{et} \quad \sin(\varphi) = \frac{\sqrt{3}}{A}(U_4 - U_2)$$

Les impulsions-bornes, quant à elles, sont modulées avec des phases de $-\pi/6$ (impulsion 18) et $-\pi/2$ (impulsion 99), et sont utilisées dans l'algorithme de synchronisation entre Alice et Bob pour délimiter clairement les impulsions-test des impulsions de données.

Choix de la longueur des blocs

La taille des blocs a été choisie à 50 000 impulsions, pour répondre à plusieurs impératifs :

1. Elle doit être suffisamment longue pour que l'évaluation des variances ne soit pas trop affectée par les fluctuations statistiques.
2. Elle doit être suffisamment petite pour que la phase relative entre Alice et Bob ne fluctue pas trop pendant la durée du bloc. La fluctuation de phase est d'environ 2π en 30 secondes.
3. La taille physique que nécessite le stockage du bloc sur les ordinateurs doit être raisonnable.
4. Puisque les blocs sont envoyés un par un aux cartes d'acquisition, il faut s'assurer que le délai entre deux blocs soit supérieur à la latence de l'ordinateur. Cette latence correspond à un temps de processeur non disponible du fait de l'attribution des ressources à d'autres processus du système d'exploitation (par exemple communications réseau, affichage). Typiquement, celle-ci est comprise entre 10 et 50 ms pour un processeur standard.

Un bloc de 50 000 impulsions est émis en 100 ms, ce qui répond aux points 2 et 4. Puisque l'information est codée sur 32 bits, les données sont stockées sur 400 kilo-octets, ce qui est raisonnable. Restent les fluctuations statistiques, qui sont encore un peu trop importantes avec 50 000 échantillons. Pour les limiter, nous avons implémenté un système de moyenne glissante dans le programme, que nous utilisons quand nous souhaitons évaluer avec précision les paramètres importants pour les rétrocontrôles (voir dans la suite).

6.2.2 Synchronisation entre Alice et Bob

Signal de synchronisation

Dans la plupart des expériences en laboratoire, une horloge maîtresse est utilisée pour synchroniser la totalité des appareils. En cryptographie quantique appliquée, en revanche, Alice et Bob sont distants. Ils ne disposent pas d'un canal électrique de

synchronisation, et doivent donc utiliser la fibre optique de transmission pour distribuer une horloge commune.

Habituellement, les systèmes de cryptographie quantique utilisent une deuxième source laser, dans la bande de fréquence O (*original*) des fibres optiques, c'est-à-dire autour de 1 300 nm. Des impulsions de synchronisation à 1 300 nm sont multiplexées avec les données (transmises dans la bande C, *conventional*, à 1 530–1 565 nm) dans la fibre de transmission, et le démultiplexage est assuré chez Bob grâce à un filtre WDM séparant les longueurs d'ondes.

Dans le prototype à variables continues, nous disposons naturellement d'une source d'impulsions classiques, synchrones avec le signal quantique : l'oscillateur local. Nous utilisons donc la voie auxiliaire de sortie de la détection homodyne, proportionnelle à l'intensité de l'oscillateur local, pour recréer un signal de synchronisation de type TTL.

Habituellement, un simple comparateur à seuil proche de zéro permet de créer une synchronisation efficace. Néanmoins, dans notre cas, le signal de la détection est amplifié pour le rendre exploitable, et l'amplificateur utilisé « coupe le continu ». Le « pied » du signal de sortie, qui correspond à une intensité lumineuse nulle, n'est donc pas centré sur zéro (signal **1** de la figure 6.3), ce qui rendrait l'instant de déclenchement d'un comparateur très dépendant de l'amplitude du signal des photodiodes, et donc du niveau d'oscillateur local.

Un montage électronique de type *discriminateur à fraction constante* permet d'éviter cet effet. Le signal des photodiodes est séparé en deux, l'un est retardé de la durée de l'impulsion (100 ns), et les deux signaux **1** et **2** sont ensuite soustraits. Ceci permet d'obtenir un signal **3** de moyenne nulle, avec une pente importante au moment du changement de signe. Ce signal entre ensuite dans un comparateur, de seuil **5** théoriquement nul mais en pratique légèrement négatif, de façon à déclencher l'impulsion de synchronisation **6**.

Du fait de la forte pente du signal, l'instant de déclenchement du comparateur dépend très peu du niveau de l'impulsion **1**. Qui plus est, puisque le signal **3** est centré sur zéro, son niveau maximum est directement proportionnel à l'intensité de l'oscillateur local, et le circuit peut donc être utilisé pour une mesure précise du niveau d'oscillateur local (signal **4**), sous réserve de le décaler de 50 ns.

Détection de la modulation d'Alice

Une fois l'horloge commune établie, Alice et Bob doivent se mettre d'accord sur une numérotation de leurs impulsions. Nous avons opté pour une méthode ne nécessitant pas la modification du niveau global du signal, de façon à ne pas perturber la détection homodyne.

Initialisation du système Alice démarre tout d'abord le système de génération des impulsions (la diode laser). Puis, Bob démarre son système de mesure. Il ne mesure alors qu'un signal non modulé, d'amplitude fixe et de phase dérivant lentement. Alice initialise ensuite son système de modulation : elle génère un bloc « vide », c'est-à-dire qu'elle ne module que les impulsions-test et les impulsions-bornes, et éteint l'amplitude des impulsions de données. Nous appellerons le premier point de ce bloc vide *point d'entrée*, noté j_0. Elle module ensuite normalement tous les blocs suivants (voir figure

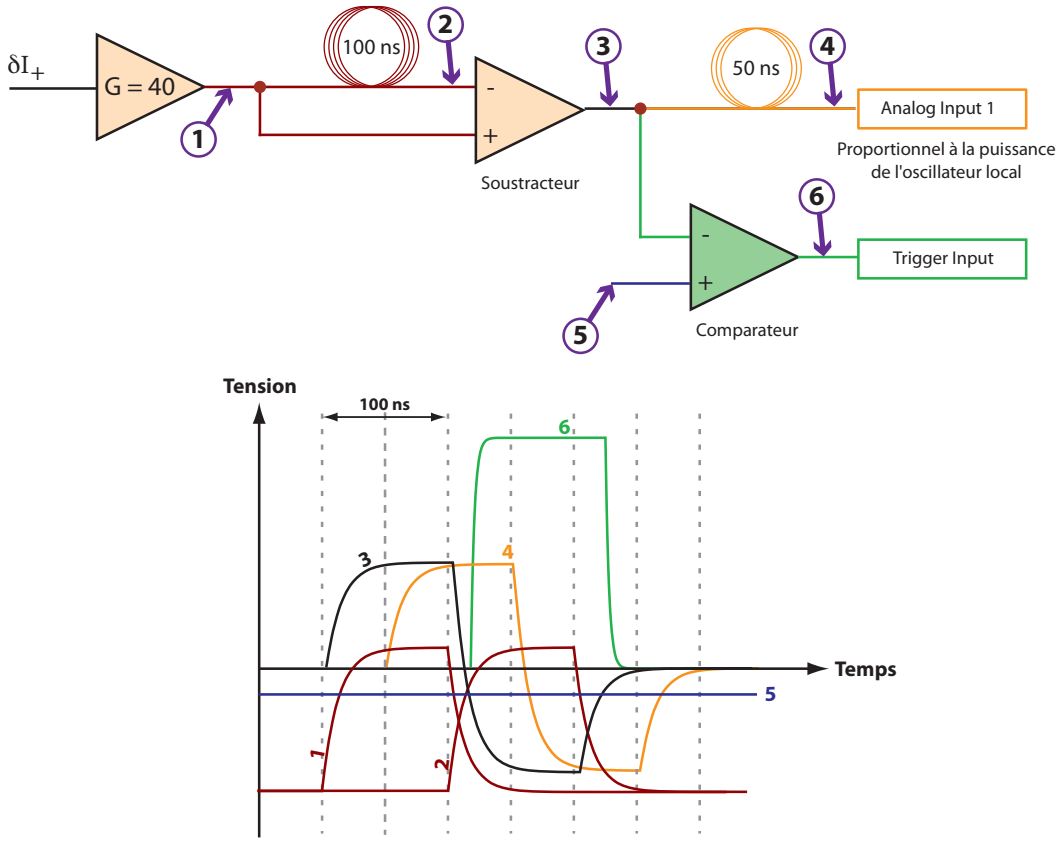


Figure 6.3 : *Schéma électrique de fonctionnement du signal de synchronisation. Le signal en entrée correspond au prélévement effectué sur l'une des photodiodes de la DH, proportionnel à l'intensité de l'oscillateur local.*

6.4). L'enjeu pour Bob est alors de déterminer le point d'entrée, c'est-à-dire la valeur exacte de j_0, à laquelle Alice a commencé sa modulation.

Vue globale de l'algorithme de détection Pour déterminer le début de la modulation, Bob utilise la procédure suivante, schématisée sur la figure 6.4 :

- **1. Détection du bloc d'entrée** : La communication entre le programme et la carte d'acquisition s'effectue par blocs. Ainsi, quand il démarre sa mesure, Bob définit des blocs de 50 000 points, de façon similaire à Alice. Puisque le début du premier bloc de Bob correspond à l'impulsion arbitraire à laquelle nous avons démarré le programme de Bob, Alice et Bob n'ont aucune raison d'avoir des blocs synchronisés. Le point d'entrée est donc situé au début d'un bloc chez Alice, mais peut être placé à n'importe quel niveau d'un bloc chez Bob. Par conséquent, pour chaque bloc qu'il reçoit, Bob essaie de déterminer s'il contient le point d'entrée, et de déterminer la position de ce point modulo 100, en utilisant l'algorithme suivant :

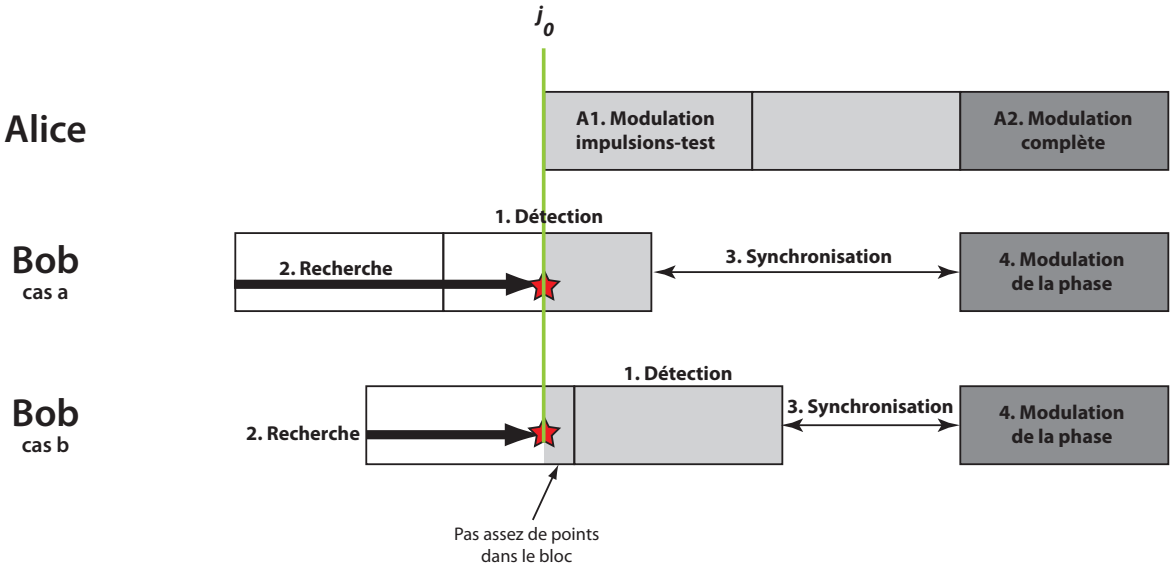


Figure 6.4 : *Représentation schématique de l'algorithme de détection du début de la modulation.*

- Bob utilise la distribution particulière des impulsions-test, qui présentent trois paliers de six impulsions dont la mesure de quadrature est identique, au bruit de photon près (cf. figure 6.2).
- Pour toutes les impulsions i de son bloc, il calcule la différence entre les mesures de quadrature Q_i et Q_{i+1} de i et $i+1$: $D_i = |Q_{i+1} - Q_i|$. Cette différence est donc petite si les impulsions i et $i+1$ sont deux impulsions-test du même type, c'est-à-dire s'il s'agit des impulsions 0 à 4, 6 à 10 ou 12 à 16 (relativement au datagramme).
- Puisqu'Alice et Bob sont décalés, l'impulsion 0 d'un bloc n'a pas de raison d'être le début d'un datagramme. Pour tout $j \in [\![0; 49\,899]\!]$, Bob calcule donc la somme

$$\begin{aligned} \mathrm{diff}_j = \quad & (D_j + D_{j+1} + D_{j+2} + D_{j+3} + D_{j+4} \\ & + D_{j+6} + D_{j+7} + D_{j+8} + D_{j+9} + D_{j+10} \\ & + D_{j+12} + D_{j+13} + D_{j+14} + D_{j+15} + D_{j+16}) \end{aligned} \tag{6.1}$$

  qu'il moyenne ensuite en regroupant diff_0 avec $\mathrm{diff}_{100}, \ldots, \mathrm{diff}_{49\,800}$; diff_1 avec $\mathrm{diff}_{101}, \ldots, \mathrm{diff}_{49\,801}$; etc. Il obtient ainsi 100 sommes de différences.
- La somme qui correspond, modulo 100, au premier point (dit *d'entrée*) j_0 des datagrammes d'Alice a une propriété particulière : toutes les différences intervenant dans la somme portent alors sur des impulsions-test successives du même type, et par conséquent la somme est petite. Au contraire, quand la somme correspond à $j_0 - 1$, $j_0 + 1 \ldots j_0 + 17$, l'un des D_i de la somme va correspondre au passage d'un groupe d'impulsions-test à un autre, et sera donc important. Notons néanmoins que, puisque les impulsions de données ne sont pas modulées, toutes les sommes correspondant à des j entre $j_0 + 20$ et $j_0 + 84$ peuvent elles aussi être petites : il nous faut donc distinguer j_0 par une autre propriété.
- C'est la deuxième propriété du point j_0 : il se distingue de tous les autres points par des valeurs diff_{j_0+1} et diff_{j_0-1} grandes, comme nous l'avons dit précédemment. Ce n'est en particulier pas le cas des impulsions de données, pour lesquelles les sommes précédentes et suivantes sont elles aussi petites.

- Bob peut alors déterminer le point d'entrée (modulo 100) de façon non équivoque, en calculant
$$j_0 = \max_{j \in [\![0;99]\!]} |\mathrm{diff}_j - \mathrm{diff}_{j-1}| + |\mathrm{diff}_{j+1} - \mathrm{diff}_j|$$
Il détecte bien entendu le bloc contenant j_0 du même coup.

- Au démarrage de Bob, les blocs sont non-modulés, et l'algorithme échoue donc. Quand Alice commence à moduler, deux cas peuvent se produire.

  Cas a. Le point d'entrée j_0 est situé au début ou au milieu du bloc de Bob. L'algorithme de détection voit un saut dans la variance des données, et renvoie alors un signal positif : j_0 est dans ce bloc.

  Cas b. Le point j_0 est trop proche de la fin du bloc de Bob. Puisque le bloc est essentiellement vide, le saut de variance entre le précédent et celui-ci est trop faible : l'algorithme de détection ne parvient pas à détecter la modulation. Le bloc suivant, en revanche, est entièrement modulé, et l'algorithme détecte alors le saut de variance sur ce bloc suivant. Dans ce cas, l'algorithme de détection s'est trompé d'un bloc.
- **2. Recherche précise du point d'entrée** : Puisque l'algorithme du point précédent ne fonctionne qu'à un bloc près, Bob ne sait *a priori* pas si le bloc qu'il a détecté est le bon bloc (contenant j_0) ou le suivant. Il démarre donc sa recherche précise du point d'entrée par le bloc précédant celui qu'il a détecté. Il parcourt ensuite chaque datagramme, de façon séquentielle, et calcule la variance des impulsions-test de chaque datagramme. Quand il atteint le point d'entrée, Bob observe un saut important de la variance entre deux datagrammes successifs. Il peut alors affirmer que la modulation a commencé : j_0 est alors le premier point de ce datagramme.
- **3. Synchronisation** : Comme nous l'avons dit, les blocs sont asynchrones, et Bob doit donc décaler son horloge pour que le début de ses blocs soit le même qu'Alice.
- **4. Modulation** : Enfin, Bob débute sa modulation de phase, et la distribution de clé peut commencer.

Zone de fonctionnement de l'algorithme Puisqu'il repose sur un saut de la variance des données, l'algorithme ne peut fonctionner que pour des variances de modulation suffisamment grandes. Dans l'implémentation précédente de ce protocole [5], quand Bob ne parvenait pas à détecter le début de la modulation des impulsions-test, il cherchait le début de la modulation des impulsions de données. Néanmoins, l'algorithme ne fonctionnait sans erreur que pour des variances de modulation (au niveau du détecteur) supérieures à 1,5 N_0 ; en dessous, le bruit de photon conduisait à des détections erronées.

Puisque les impulsions-test ont toujours une amplitude maximale, leur détection est plus aisée, même quand la variance de modulation est petite. Le nouveau protocole fonctionne sans erreur jusqu'à des variances de modulation chez Bob d'environ 0,1 N_0. Nous verrons que ce point est important pour le protocole quaternaire présenté dans le chapitre 10, qui nécessite des variances faibles. En dessous, la détection est aléatoire, car les impulsions-test n'ont qu'une variance de 16 fois celle des impulsions de données.

S'il était besoin de travailler à des variances de modulation plus faibles que $GV_A =$

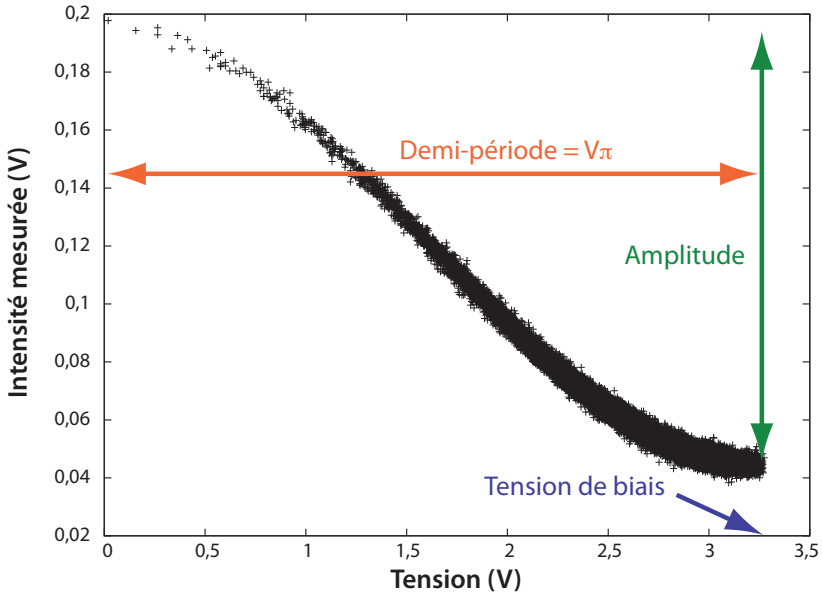


Figure 6.5 : *Intensité optique mesurée sur la photodiode de contrôle d'Alice, en fonction de la tension de commande du modulateur « dynamique » d'amplitude d'Alice. On voit que le tracé de la caractéristique permet de retrouver le V_π et la tension de biais du modulateur, ainsi que l'intensité de l'impulsion optique.*

0,1, comme pour le cas du protocole quaternaire décrit dans le chapitre 10, il nous faudrait générer un signal de synchronisation spécifique, ou augmenter temporairement la variance de modulation pour les deux premiers blocs. Dans le protocole actuel, nous travaillons avec des variances de modulation chez Bob de 1 à 4 N_0, et l'algorithme actuel est bien adapté.

6.2.3 Rétrocontrôles et automatisation

Modulateurs d'amplitude

Deux modulateurs d'amplitude sont présents dans le système d'Alice : l'un est utilisé pour coder l'information d'amplitude à chaque impulsion, l'autre pour ajuster finement le niveau de signal en sortie d'Alice. Ces modulateurs présentent des dérives au cours du temps, en particulier en ce qui concerne la tension de biais (c'est-à-dire la tension à appliquer pour obtenir une transmission minimale). Pour pallier ces effets majoritairement thermiques, il est nécessaire d'appliquer des mesures de rétrocontrôle réactives et ne nécessitant pas, si possible, l'interruption de la modulation.

Le rôle principal des impulsions de données est de transmettre l'information utilisée pour l'extraction de la clé, mais nous les utilisons aussi pour le rétrocontrôle des modulateurs d'amplitude d'Alice. Pour ce faire, un coupleur est placé dans la voie signal d'Alice, après les modulateurs, de manière à prélever une partie du signal. L'intensité de ce signal est ensuite mesurée par une photodiode du même type que dans la détection homodyne. Nous pouvons ainsi tracer la caractéristique des modulateurs d'amplitude (intensité du photocourant en fonction de la tension appliquée).

La figure 6.5 montre cette caractéristique. La modulation en amplitude n'est pas uniforme, et il y a donc beaucoup plus de points à faible amplitude qu'à grande amplitude. Néanmoins, on voit clairement se dessiner une demi-arche sinusoïdale, caractéristique du fonctionnement « Mach-Zehnder » du modulateur. Il est alors possible de déterminer les valeurs caractéristiques à partir de la sinusoïde :

- le V_π du premier modulateur est donné par la demi-période,
- la tension de biais est donnée par la phase,
- la variance de modulation, qui dépend du réglage du deuxième modulateur, est proportionnelle à l'amplitude.

Compte tenu du relativement faible nombre de données sur un bloc (40 000 mesures utiles), le bruit statistique sur ces valeurs est assez important, et il est donc nécessaire d'effectuer une moyenne sur 100 blocs pour obtenir une bonne précision, de l'ordre de quelques pour mille. Le temps typique de rétrocontrôle des modulateurs d'amplitude est donc de 10 secondes.

Modulateurs de phase

Deux modulateurs de phase sont présents dans les systèmes : l'un chez Alice, pour coder l'information de phase, et l'autre chez Bob, pour choisir aléatoirement une quadrature de mesure. La caractéristique tension-phase des modulateurs est linéaire ; pour gérer correctement la modulation, il est donc nécessaire de connaître le V_π de ceux-ci.

Pour ce faire, nous exploitons de nouveau les impulsions-test, en modulant les six premières impulsions de chaque datagramme de la façon suivante :

- Impulsions 1 et 2 : Tension 0 chez Alice, 0 chez Bob
- Impulsions 3 et 4 : Tension $2V_\pi$ chez Alice, 0 chez Bob
- Impulsions 5 et 6 : Tension 0 chez Alice, $2V'_\pi$ chez Bob

Si les V_π des modulateurs sont bien réglés, le déphasage entre une tension de consigne 0 et $2V_\pi$ est de 2π, et la quadrature mesurée en moyenne est donc $\langle X_\theta \rangle = A\cos\theta$ dans les deux cas, où A est l'amplitude des impulsions, et θ la phase relative entre signal et OL.

Dans le cas où Alice (par exemple) ne possède qu'une valeur approchée V_{approx} du V_π réel, la tension appliquée sur les impulsions 3 et 4 est $2V_{\text{approx}}$, et la quadrature mesurée est donc en moyenne $\langle X_{\theta+\Delta\theta} \rangle = A\cos(\theta + 2\pi\frac{V_{\text{approx}}}{V_\pi})$. Il est alors possible de retrouver efficacement la valeur réelle de V_π grâce à la relation trigonométrique

$$2\pi\frac{V_{\text{approx}}}{V_\pi} = \arccos\frac{\langle X_{\theta+\Delta\theta}\rangle}{A} - \arccos\frac{\langle X_\theta\rangle}{A} \tag{6.2}$$

Pour que les moyennes sur les quadratures soient suffisamment précises, il nous faut environ un million de mesures à raison de deux mesures par datagramme, soit environ 1000 blocs. Le temps typique de rétrocontrôle des modulateurs de phase est donc d'une minute.

6.2.4 Structure algorithmique du programme

Le sous-programme de gestion de la transmission quantique est programmé en C++, qui est un langage orienté objet. Ce type de langage repose sur la possibilité

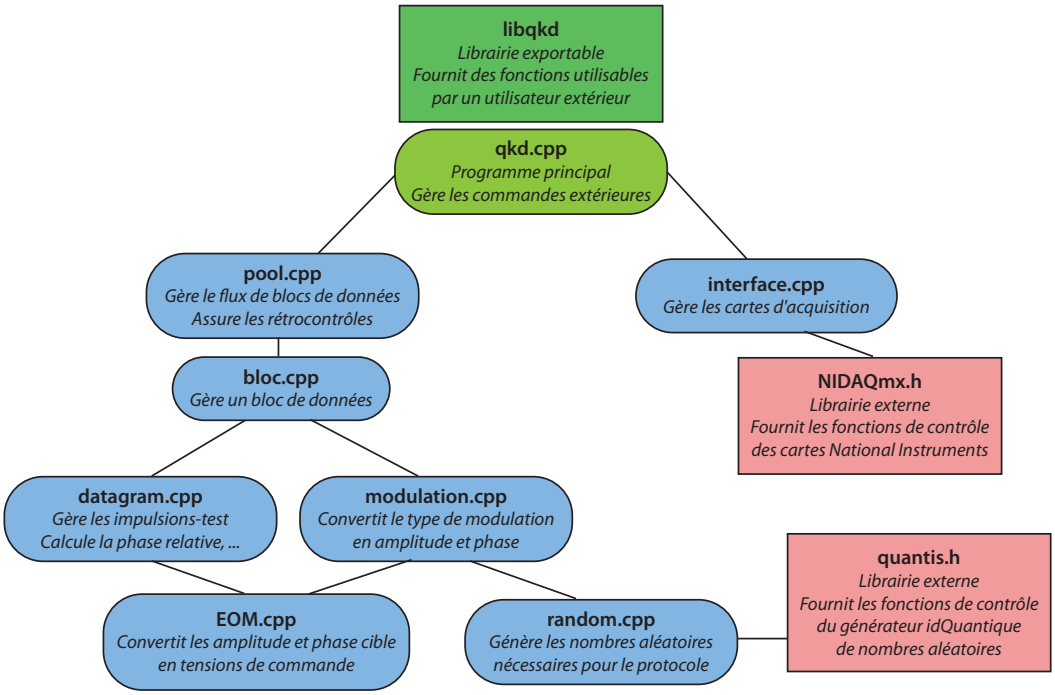


Figure 6.6 : *Vision schématique du sous-programme de gestion de la transmission quantique*

---

de définir des classes d'objets virtuels, qui ont chacun une tâche bien déterminée. Ces objets communiquent entre eux *via* des fonctions d'interface, et possèdent chacun des variables ou fonctions propres. Généralement, la classe B n'a pas de « droit d'ingérence » sur les variables détenues par la classe A : si elle veut accéder à ces variables ou les modifier, elle doit le lui demander grâce à une fonction d'interface, et la classe A est alors libre d'accéder ou non à la requête, en fonction des circonstances.

La figure 6.6 présente la structure générale du sous-programme. Il est organisé autour d'un fichier principal, `qkd.cpp`, qui gère les requêtes extérieures en les redirigeant vers les classes adaptées. Pour exposer le fonctionnement du système, prenons l'exemple de l'une des opérations fondamentales du protocole : la modulation d'un bloc d'impulsions.

- `qkd`, le programme principal, a la main, et demande à `pool` de remplir le buffer du prochain bloc avec des tensions de commande.
- `pool` récupère le numéro du bloc en cours, et transmet au bon `bloc` la requête de `qkd`.
- `bloc` demande d'une part à `datagram` de lui fournir les tensions de commande des impulsions-test, et à `modulation` de lui fournir les tensions de commande des impulsions de données.
- `modulation` connaît le type de modulation actuel, mais a besoin d'une liste de nombres aléatoires suivant une distribution gaussienne. Elle en fait la requête à `random`.
- `random` fait alors appel à la carte de génération Quantis à travers l'API de la librairie externe `quantis.h`. Une fois la liste aléatoire générée, il la fait remonter à `modulation`.

- `modulation` transforme cette liste de nombres aléatoires en liste d'amplitudes et de phases de commande. De son côté, `datagram` génère lui aussi la liste des amplitudes et phases des impulsions-test.
- Pour convertir ces amplitudes et phases en tensions de commande, `modulation` et `datagram` font appel à `EOM`, qui connaît les caractéristiques des modulateurs. Ce dernier leur renvoie donc la fameuse liste de tensions de commande.
- `modulation` et `datagram` renvoient la liste à `bloc`, qui la renvoie à `pool`, qui la renvoie à `qkd`.
- Avec ces données, `qkd` remplit le buffer de l'ordinateur. Il demande à `interface` d'envoyer ce buffer vers le buffer de la carte d'acquisition, ce qu'`interface` fait en appelant la bonne fonction de `NIDAQmx.h`, librairie externe pilotant les cartes d'acquisition. Le bloc d'impulsions suivant est prêt à être modulé !

En quelque sorte, un langage orienté objet permet de créer une structure administrative géante, dans laquelle chacun a ses prérogatives. Dès que les prérogatives d'une classe sont outrepassées par une autre classe, le programmeur s'expose à des conflits majeurs : le plus courant étant deux classes qui modifient la valeur d'une même variable en parallèle, sans qu'aucune ne le dise à l'autre.

Le programme est pour finir compilé, et « empaqueté » dans une librairie exportable appelée `libqkd`. Cette librairie fonctionne de façon similaire à un pilote d'imprimante : elle fournit quelques fonctions générales à l'utilisateur, qui peut donc démarrer le système, l'arrêter, récupérer des blocs, etc. Nous verrons dans le chapitre 9 que la librairie `libqkd` est utilisée par le programme d'extraction pour interagir avec le système physique.

6.3 Calibration du système

Après la transmission quantique, Alice et Bob doivent évaluer les différents paramètres de la transmission intervenant dans le calcul de sécurité, de façon à déterminer la quantité de secret présent dans leurs données corrélées. Il s'agit de la transmission totale $G = \eta T$ (incluant le gain T du canal et les pertes η du détecteur), du bruit en excès total $\xi = \varepsilon + \frac{v_{\mathrm{el}}}{\eta}$ (incluant le bruit en excès du canal ε et le bruit électronique v_{el}), et enfin de la variance de modulation d'Alice à l'entrée du canal, V_A.

Pour évaluer ces trois paramètres, Alice et Bob extraient tout d'abord, à partir de l'ensemble de leurs données corrélées, un échantillon choisi de façon aléatoire (dans notre cas, 50 % des données sont utilisées pour l'évaluation des paramètres). Ils comparent ensuite leurs données en révélant publiquement cet échantillon par un canal authentifié mais espionnable.

6.3.1 Lien avec les moments d'ordre 2

Les distributions gaussiennes corrélées d'Alice et de Bob peuvent être caractérisées par la variance V_A des données d'Alice, la variance V_B des données de Bob, et la corrélation ρ^2 entre les données d'Alice et de Bob. Rappelons que V_A est une valeur classique, non affectée par le bruit de photon et parfaitement connue par Alice, alors que V_B est la variance de la mesure de $\hat{X}_B$, qui inclut le bruit de photon.

En reprenant le modèle du canal gaussien, tel que $X_B = g(X_A + X_N)$, nous pouvons établir quelques relations utiles entre différents paramètres. Exprimons la corrélation ρ^2 :

$$\rho^2 = \frac{\langle X_A X_B \rangle^2}{V_A V_B} = \frac{\langle g X_A (X_A + X_N) \rangle^2}{V_A V_B} = \frac{G V_A}{V_B} \tag{6.3}$$

$$= \frac{V_A}{V_A + V_N} = \frac{\frac{V_A}{V_N}}{1 + \frac{V_A}{V_N}} \tag{6.4}$$

ce qui nous donne les trois équations

$$\boxed{G = \frac{\rho^2 V_B}{V_A}} \qquad \boxed{\rho^2 = \frac{\mathrm{SNR}}{1 + \mathrm{SNR}}} \qquad \boxed{\mathrm{SNR} = \frac{1}{1 - \rho^2}} \tag{6.5}$$

Notons qu'assez logiquement, une relation directe existe entre le rapport signal à bruit et la corrélation entre les données d'Alice et de Bob.

Nous pouvons aussi exprimer le bruit en excès ξ, en partant de (4.10) :

$$\xi = \chi_{\mathrm{tot}} + 1 - \frac{1}{G} = V_N - \frac{1}{G} \tag{6.6}$$

$$= \frac{V_B}{G} - V_A - \frac{1}{G} = \frac{V_B V_A}{\rho^2 V_B} - V_A - \frac{V_A}{\rho^2 V_B} \tag{6.7}$$

$$\boxed{\xi = \frac{V_A}{\rho^2}\left(\frac{V_B - 1}{V_B} - \rho^2\right)} \tag{6.8}$$

À un facteur près, l'excès de bruit peut donc s'exprimer comme une différence entre deux termes quasiment identiques : ρ^2, la corrélation des données, et $\frac{V_B - 1}{V_B}$, c'est-à-dire la fraction de la variance de Bob qui n'est pas due au bruit de photon. Dans le cas où $\xi = 0$, ces deux valeurs sont identiques.

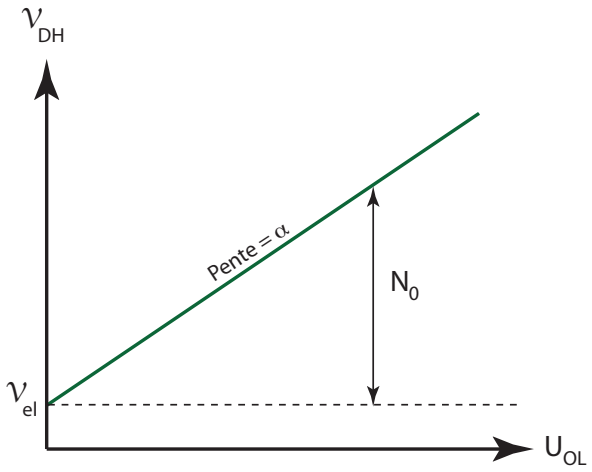
Finalement, on voit qu'il est possible d'exprimer les paramètres G et ξ en fonction des seuls moments des distributions d'Alice et Bob : V_A, V_B, ρ^2.

6.3.2 Calibration du bruit de photon

Dans les calculs ci-dessus, toutes les valeurs sont normalisées au bruit de photon. Or, en pratique, Alice ne connaît la variance de sa distribution qu'à travers l'amplitude qu'elle mesure sur sa photodiode, et Bob ne connaît de même que la variance de ses données en mV^2. Il est donc nécessaire pour Alice et Bob, avant toute communication, de calibrer le bruit de photon de façon à normaliser leurs distributions.

Bob

Le bruit de photon a la particularité d'être une fonction linéaire de la puissance de l'oscillateur local au niveau de la détection homodyne. Bob peut donc tracer la variance du signal principal de détection homodyne $\mathcal{V}_{DH}$ (en mV^2, voir figure 5.20) en fonction de la tension U_{OL} de la voie auxiliaire, proportionnelle à la puissance de l'OL. Grâce à la caractéristique, qui prend la forme $\mathcal{V}_{DH} = \alpha U_{OL} + \mathcal{V}_{\mathrm{el}}$, Bob détermine, d'une part, le niveau de bruit de photon, $N_0 = \alpha U_{OL}$, et d'autre part le niveau de bruit électronique, $v_{\mathrm{el}} = \frac{\mathcal{V}_{\mathrm{el}}}{N_0}$.



Bob peut alors normaliser sa distribution, en la centrant sur zéro et en divisant toutes les valeurs par $\sqrt{N_0}$. Rappelons que N_0 est déterminé pour une valeur donnée de U_{OL}, et que Bob doit donc ajuster sa calibration en fonction du niveau d'oscillateur local.

Alice

Pour Alice, normaliser sa distribution revient à déterminer le nombre moyen de photons par impulsion en sortie de son dispositif. Néanmoins, celui-ci est de l'ordre de 10 photons, et la puissance associée est donc trop faible pour être déterminée avec une photodiode classique. Au contraire, une détection homodyne est parfaitement adaptée pour ce type de calibration ; Alice est donc calibrée avant d'être séparée de Bob.

Tout d'abord, les pertes totales η du détecteur de Bob sont évaluées lors de la conception de Bob. Ensuite, Alice et Bob se placent à distance nulle pour effectuer une transmission quantique garantie sans espion. Bob détermine alors la variance de modulation au niveau de son détecteur V_B, normalisée au bruit de photon. V_A s'exprime, en fonction de V_B : $V_A = \frac{1}{\eta}(V_B - 1 - \eta\xi)$. Si Alice est en mesure d'évaluer ξ (sachant qu'il n'y a pas d'espionnage), elle peut donc calculer la variance de modulation V_A, et la mettre en relation avec la mesure de sa photodiode U_φ. Elle obtient ainsi une courbe de la forme $V_A = f(U_\varphi)$, qui reste valable une fois qu'Alice et Bob seront séparés pour une transmission réelle. La variance V_A est alors calibrée en unité de bruit de photon.

6.3.3 Paramètres du mode réaliste

Les calculs ci-dessus ne permettent pas de distinguer η de T, ni v_{el} de ε : puisqu'Alice et Bob ne disposent que de trois moments, ils ne pourront de toute façon en extraire plus de trois paramètres. Comme nous l'avons vu dans les calibrations d'Alice et Bob, les paramètres v_{el} et η sont évalués de manière indépendante. Néanmoins, pour ne pas surestimer la sécurité de la transmission, il faut s'assurer de ne pas attribuer au détecteur de Bob plus de pertes, ou plus de bruit électronique, qu'il n'en a réellement :

- Puisque nous soustrayons v_{el} de ξ pour obtenir ε, et que nous préférons surestimer ε, il nous faut sous-estimer v_{el} s'il existe une incertitude sur sa valeur.

- De même, puisque nous connaissons $G = \eta T$ et que nous préférons que les pertes soient attribuées à l'espion (c'est-à-dire que T soit sous-estimé), il nous faut sur-estimer η en cas de doute.

Ainsi, la fraction d'incertitude est attribuée à l'espion à travers T et ε, et la sécurité est conservée.

6.4 Résultats expérimentaux

6.4.1 Évaluation des bruits

Bruit de la diode laser Nous pouvons caractériser le bruit d'amplitude et le bruit de phase de notre diode laser avec la détection homodyne : ce sont des bruits classiques, et donc des fonctions linéaires de l'intensité lumineuse du signal. Ainsi, nous modulons linéairement l'amplitude du signal : le bruit d'amplitude correspond à la pente de la droite $V_B = f(I_{\text{signal}})$ lors d'une mesure de la quadrature d'amplitude $\hat{X}$; le bruit de phase correspond à cette même pente lors de la mesure de la quadrature de phase $\hat{P}$.

Jérôme Lodewyck a caractérisé le bruit de la diode au cours de sa thèse [5]. Le bruit d'amplitude est négligeable devant les autres bruits : en effet, dans les télécommunications, l'information est codée sur l'amplitude du signal et les diodes laser sont donc conçues pour avoir un très faible bruit d'amplitude. Par ailleurs, la diode laser est atténuée pour que le bruit de phase résiduel soit faible ; il est typiquement de $2{,}5.10^{-4} N_0$ par photon, ce qui correspond à un bruit de phase de 0,25 % du bruit de photon pour une variance de modulation $V_A = 20$.

Bruit électronique Le bruit électronique correspond au bruit mesuré en coupant le signal optique. Le bruit résiduel provient alors de l'électronique de la détection homodyne et des autres circuits électriques (contrôleur de polarisation, circuit de synchronisation). Le bruit intrinsèque au circuit de détection homodyne est de l'ordre de 4 mV^2. En présence des autres circuits électriques du système, et dans des conditions de laboratoire, le bruit est typiquement de 6,0 mV^2, de façon stable.

Bruit technique Le bruit technique correspond au bruit dû aux erreurs de modulation. Un mauvais réglage des V_π des modulateurs, en particulier, induit un bruit technique qui peut rapidement devenir important. Grâce aux rétrocontrôles mis en place dans le système, les tensions de contrôle des modulateurs sont corrigées à mieux que le millivolt, ce qui génère un bruit technique typique résiduel inférieur à $3.10^{-4} N_0$ par photon, soit moins de $0{,}3\,\% N_0$ pour une variance de modulation $V_A = 20$.

Bruit de phase dû aux vibrations Les vibrations mécaniques dues à l'environnement du système induisent une vibration des fibres dans nos systèmes, et donc une fluctuation rapide de la phase dans notre interféromètre. En isolant soigneusement nos systèmes, et dans des conditions de laboratoire, nous observons un bruit de phase résiduel dû aux vibrations de l'ordre de $1\,\% N_0$ pour $V_A = 20$.

Bilan des bruits Pour un bruit de photon typique de 600 mV^2 au niveau du détecteur de Bob, nous obtenons les contributions suivantes :

Bruit	Contribution pour $V_A = 20$
de phase (diode laser)	$0{,}25\,\%\,N_0$
électronique	$\frac{1}{G} \times 1\,\%\,N_0$
technique	$0{,}3\,\%\,N_0$
de phase (vibrations)	$1\,\%\,N_0$
Total en entrée	$(1{,}55 + \frac{1}{G})\,\%\,N_0$

6.4.2 Taux théorique

Paramètres du calcul

Avec les bruits ci-dessus, nous pouvons calculer les paramètres intervenant dans l'expression du taux secret :

- Nous avons mesuré des pertes du détecteur de Bob $\eta = 0{,}57$.
- Le bruit électronique s'écrit $v_{\text{el}} = 0{,}01$ pour $N_0 = 600$ mV^2.
- Nous considérons une transmission à travers une fibre de 25 km, soit une transmission mesurée $T = 0{,}31$.
- Le bruit en excès a été évalué à $\varepsilon = 0{,}0155$ pour $V_A = 20$.

Ces valeurs nous donnent des bruits :

- $\chi_{\text{hom}} = \frac{1}{\eta} - 1 + \frac{v_{\text{el}}}{\eta} = 0{,}771$
- $\chi_{\text{line}} = \frac{1}{T} - 1 + \varepsilon = 2{,}24$
- $\chi_{\text{tot}} = \chi_{\text{line}} + \frac{\chi_{\text{hom}}}{T} = 4{,}73$

Taux secret

Pour une transmission à 25 km, on utilise les expressions du taux secret de la section 4.3, et on trouve (exprimés en bit/impulsion) :

$$I_{AB} = 1{,}083, \quad I_{BE} = 0{,}935 \quad \Rightarrow \quad \mathbf{\Delta I}^{\text{Shannon}} = \mathbf{0{,}148}$$
$$I_{AB} = 1{,}083, \quad \chi_{BE} = 0{,}951 \quad \Rightarrow \quad \mathbf{\Delta I}^{\text{Holevo}} = \mathbf{0{,}132}$$

Compte tenu de notre taux d'émission de 500 000 impulsions/s, dont 20 % sont des impulsions-test, on trouve :

$$I_{AB} = 433 \text{ kbit/s}, \quad I_{BE} = 374 \text{ kbit/s} \quad \Rightarrow \quad \mathbf{\Delta I}^{\text{Shannon}} = \mathbf{59} \text{ kbit/s}$$
$$I_{AB} = 433 \text{ kbit/s}, \quad \chi_{BE} = 380 \text{ kbit/s} \quad \Rightarrow \quad \mathbf{\Delta I}^{\text{Holevo}} = \mathbf{53} \text{ kbit/s}$$

La figure 6.7 donne deux représentations de ces taux : l'une en fonction de la transmission T, l'autre en fonction de la distance de transmission, où l'on a choisi une transmission de 0,19 dB/km pour la fibre de transmission. On voit que, pour toute transmission T, l'information partagée par Alice et Bob I_{AB} reste toujours plus grande que l'information de l'espion I_{BE}. Il est donc théoriquement possible, même si l'excès de bruit n'est pas nul, de transmettre de l'information secrète entre Alice et Bob quelle que soit la distance de transmission. Néanmoins, nous verrons dans les chapitres 7 et 9 que l'efficacité avec laquelle l'information secrète est extraite des données corrélées d'Alice et Bob limite la distance de transmission dès qu'elle n'est pas de 100 %.

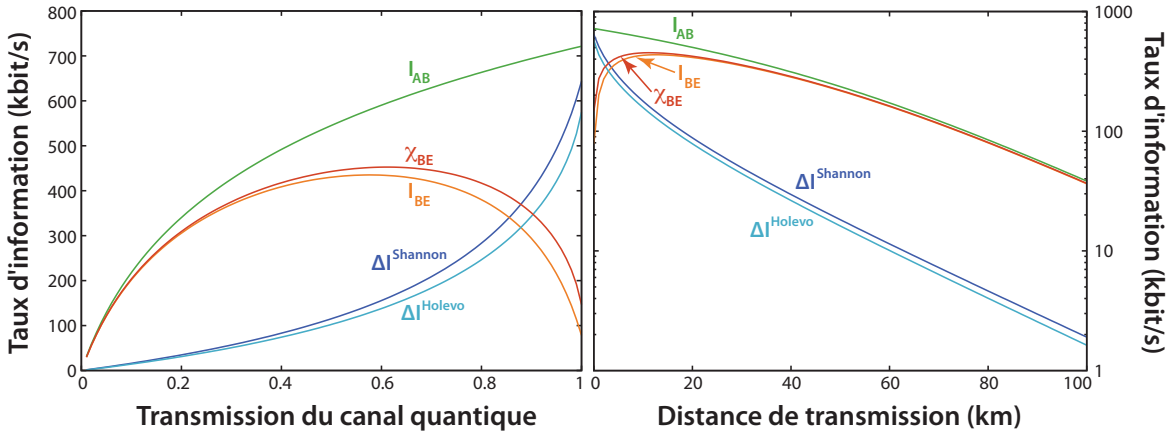


Figure 6.7 : *Quantités d'information et taux secrets théoriques, pour un débit d'impulsions de 400 000/s.*

6.4.3 Stabilité du sous-programme « transmission quantique »

Les ordinateurs utilisés pour la gestion du protocole utilisent le système d'exploitation OpenSuSE, fondé sur un noyau Linux. Ce choix d'une distribution Linux a été fait, d'une part, car la stabilité du système d'exploitation est cruciale lors du développement d'un programme fonctionnant en temps réel. Qui plus est, à travers le grand nombre d'outils libres permettant l'optimisation des programmes C++, Linux est bien adapté à notre application. D'autre part, notre système a pris part au réseau de cryptographie quantique SECOQC (voir chapitre 9), pour lequel il avait été décidé d'utiliser Linux pour tous les systèmes.

Débogage

Notre programme de gestion de la transmission quantique doit pouvoir fonctionner en continu, pendant de longues périodes. Lors de la programmation d'un logiciel avec de telles contraintes, il faut prendre garde à toutes les causes de plantage (terme peu formel mais consacré : c'est une traduction de *crash*, correspondant à une interruption brutale, inattendue et anormale) :

1. les erreurs de segmentation, qui correspondent à l'appel d'un registre mémoire se trouvant hors de la zone autorisée. Typiquement, un vecteur `vec` est déclaré tel qu'il contienne 50 éléments, et on appelle `vec[50]`, qui est le 51ème élément du vecteur, et donc n'existe pas.
2. les dépassements d'entiers : un entier est codé sur un certain nombre de bits, qui conditionne la valeur maximale que cet entier peut prendre. Par exemple, pour un entier codé sur 8 bits (de type `short int`), la valeur maximale est $2^8 - 1 = 255$. Si l'on cherche à affecter à une variable `short int` une valeur plus grande que 255, le programme renvoie une erreur du type `integer overflow`.
3. de façon similaire, les dépassements de tas, de tampon, de pile, qui reviennent tous à dépasser la taille de l'objet considéré.

4. enfin, les fuites mémoires. Elles consistent en la déclaration explicite d'une nouvelle variable (par exemple, `new int foo`), mais à l'oubli de sa destruction (`delete foo`). Puisque la déclaration a été effectuée par `new`, le processeur n'est pas autorisé à supprimer de lui-même la variable, et la place que `foo` prend dans la mémoire vive n'est alors jamais libérée. Cet effet est particulièrement gênant quand le programme fonctionne selon une boucle : au fur et à mesure du fonctionnement, les nouveaux appels de `new` vont réserver de la place dans la mémoire vive, jusqu'à ce que celle-ci soit saturée. Le programme ne peut alors plus fonctionner, et plante.

Nous avons donc soigneusement débogué le programme pour éviter les trois premiers types d'erreur ; l'outil `gdb` [59] du projet GNU est particulièrement utile pour affiner le débogage. Par ailleurs, nous avons éliminé les fuites mémoires en nous aidant du logiciel de profilage `Valgrind` [60], qui trace le fonctionnement du programme et détecte toutes les variables non-libérées au fur et à mesure de son exécution.

Pilote de la carte d'acquisition

Après ces optimisations, le sous-programme ne présente plus de bogue significatif. Néanmoins, un dernier problème persiste : la carte d'acquisition National Instruments que nous utilisons numérote les impulsions qu'elle émet sur un entier signé de 32 bits. Ainsi, quand la carte atteint la 2^{31}ème impulsion, elle renvoie un message d'erreur et doit être réinitialisée. Or, $2^{31} = 2\,147\,483\,648$ impulsions sont émises à 500 kHz en 1 heure et 11 minutes. Nous avons donc dû mettre au point un système de réinitialisation régulier, qui permet de faire fonctionner le programme au delà de 71 minutes.

Néanmoins, lors de ces réinitialisations, il arrive que la carte d'acquisition se bloque, et que la seule solution pour la remettre en fonctionnement soit de redémarrer le programme manuellement. Ces blocages sont d'autant plus fréquents que le processeur de l'ordinateur est sollicité par ailleurs. De ce fait, la durée moyenne de fonctionnement autonome du sous-programme de transmission quantique est de l'ordre de 3 jours.

Pour éviter cet effet, il nous faudrait disposer d'un pilote National Instruments 64 bits, qui n'existe à l'heure actuelle que sous Windows. Nous pourrions alors numéroter les impulsions jusqu'environ 10^{19}, c'est-à-dire pendant plus d'un million d'années.

Troisième partie

Extraction de la clé secrète

Des corrélations au secret

Dans les parties précédentes, nous avons :

- transmis de l'information continue entre Alice et Bob, codée sur les quadratures d'états cohérents de la lumière ;
- optimisé le fonctionnement du système optique, et évalué les bruits ajoutés à la transmission ;
- montré qu'il était possible, compte tenu des paramètres de la transmission, d'extraire une quantité d'information secrète $\Delta I = I_{AB} - I_{BE}$ des données corrélées d'Alice et de Bob.

Cette information secrète, néanmoins, n'est pas accessible directement. En effet, elle n'existe que sous forme de corrélations entre les données continues d'Alice et de Bob. L'enjeu est donc de transformer ces deux ensembles de *données continues, corrélées, contenant du secret* en une chaîne de *données discrètes, identiques, partagées et totalement secrètes* — c'est-à-dire une clé secrète partagée.

Pour ce faire, les données corrélées subissent plusieurs opérations complexes, quoique complètement classiques :

- Binarisation des données, et correction des différences entre les chaînes d'Alice et de Bob : c'est le processus de *réconciliation*, décrit dans le chapitre 7 ;
- Élimination de toute l'information potentiellement espionnée : c'est l'*amplification de confidentialité*, décrite dans le chapitre 8.

Le chapitre 9 décrit enfin la structure du programme utilisé pour l'extraction de la clé, ainsi que les performances du système expérimental dans des conditions réelles de fonctionnement.

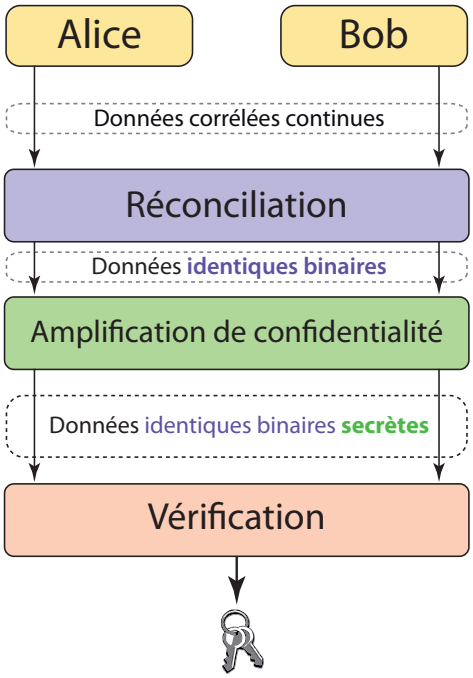


Figure 6.8 : *Processus d'extraction de la clé secrète*

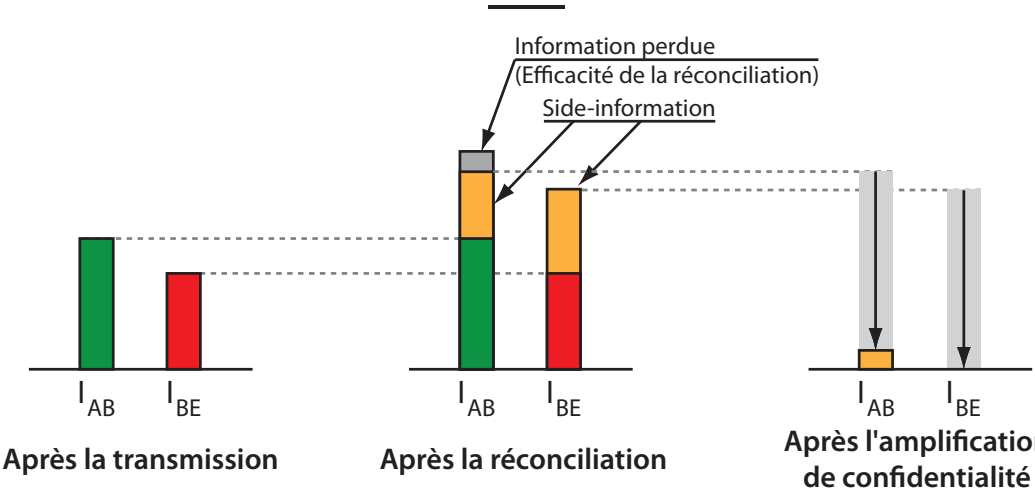


Figure 6.9 : *Informations d'Alice et Bob, et de l'espion, après chaque étape de l'extraction*

7 Réconciliation des données expérimentales

> Il y a des scientifiques sadiques qui se pressent de traquer les erreurs au lieu d'établir la vérité.
>
> Marie Curie

Après la transmission quantique, Alice et Bob ont chacun un ensemble de données, continues dans notre cas, qui sont corrélées. Ils doivent dans un premier temps en extraire deux chaînes binaires identiques, à l'aide d'un algorithme de *réconciliation*.

7.1 Protocoles de correction d'erreurs

Les protocoles de correction d'erreurs ont été développés dans le cadre des télécommunications classiques : une chaîne de référence (le message), générée par Claude[1], est transmise à Dominique dans un canal bruité. Il s'agit donc pour Dominique de corriger le bruit de la chaîne bruitée pour retrouver le message originel.

Le schéma général d'un algorithme de correction d'erreurs se décrit comme suit : Claude a la tâche de l'encodage, et utilise un *code correcteur* pour encoder le message dans un mot-code redondant. Après réception, Dominique doit décoder le message codé à partir du mot-code bruité, en utilisant un *algorithme de décodage*. Nous décrivons dans ce chapitre le fonctionnement de ces codes correcteurs et algorithmes de décodage, en insistant plus particulièrement sur les codes LDPC utilisés dans notre système.

7.1.1 Codes correcteurs

Théorème du codage de canal

Si l'on considère une variable aléatoire X, transmise par un canal bruité et se transformant en une variable aléatoire Y, on peut définir [8] la *capacité du canal*

1. Nous reprenons ici la notation utilisée par Gilles Van Assche [61]. Claude et Dominique sont des prénoms mixtes servant à désigner alternativement Alice ou Bob, en fonction de leur rôle : Claude est assigné(e) à l'encodage, Dominique au décodage.

comme :

$$C = \max_X I_{XY} \tag{7.1}$$

Cette capacité est aussi appelée *limite de Shannon*, et elle correspond au nombre maximal de bits par symbole pouvant être transmis sans erreur par le canal.

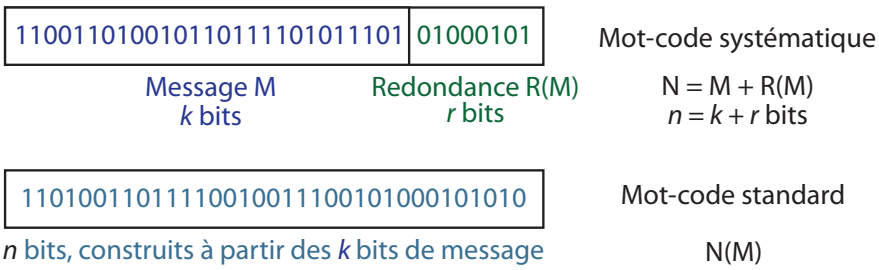
Transmission d'un message

On considère maintenant un canal bruité de capacité C, par lequel on cherche à transmettre un message de k bits. Selon le théorème de codage, un symbole envoyé par le canal ne peut transmettre au mieux que C bits d'information. Pour transmettre nos k bits, il faut donc transmettre au minimum $\frac{k}{C}$ symboles.

Dans le cas de symboles binaires, on a forcément $C \leqslant 1$, et il faut donc transmettre plus de k symboles pour transmettre k bits.

Codes, taux et redondance

Le but d'un code correcteur d'erreur est de permettre la transmission sans erreur d'un message de longueur k dans un canal de capacité C. Le code correcteur encode le message dans un *mot-code*, c'est-à-dire une chaîne de $n > \frac{k}{C}$ bits contenant les k bits du message, ainsi que $r = n - k$ bits redondants. Notons que les k bits utiles ne se retrouvent pas forcément de manière explicite dans le mot-code.



Le *code correcteur d'erreur* peut être défini formellement comme l'ensemble des mots-code, c'est-à-dire un sous-ensemble des mots de n bits. Néanmoins, en pratique, on peut l'assimiler à l'algorithme permettant d'encoder l'ensemble des messages de k bits dans des mots-codes de n bits.

Un *code systématique* est un code dans lequel chaque mot-code contient de manière explicite les k bits du message.

Enfin, le *taux du code* est défini comme la fraction du mot-code contenant le message, c'est-à-dire $R = \frac{k}{n}$. Pour qu'un code de taux R puisse être utilisé pour transmettre un message sans bruit dans un canal de capacité C, il faut logiquement que $R \leqslant C$, c'est-à-dire que le taux du code soit en dessous de la limite de Shannon. Un code de taux C est dit optimal pour ce canal (mais nous verrons qu'il ne peut pas être utilisé en pratique).

7.1.2 Correction des erreurs

Sans algorithme de décodage, un code correcteur n'est qu'une façon d'encoder de manière redondante de l'information. Un code correcteur et l'algorithme de décodage

adapté à ce code sont donc toujours conçus en parallèle, pour former un *protocole complet de correction d'erreur*[2]. Bien entendu, rien n'empêche de trouver plusieurs techniques de décodage pour un même code.

Réconciliation, ou correction d'erreurs ?

Dans le cadre de la distribution quantique de clés, il n'y a pas de message : l'information transmise est aléatoire. Le canal n'introduit donc pas d'erreurs à proprement parler, mais plutôt un bruit aléatoire. L'enjeu est donc de *réconcilier* les deux chaînes aléatoires d'Alice et de Bob, pour extraire une chaîne commune.

Pour ce faire, on peut employer plusieurs stratégies :

- On assimile les données d'Alice à un message de référence, et on applique un algorithme de correction des erreurs de Bob par rapport à Alice : c'est la *réconciliation directe*. Dans ce cadre, Alice est Claude, Bob est Dominique, et on doit écrire l'information secrète comme $\Delta I = I_{AB} - I_{AE}$.
- Puisque toutes les données sont aléatoires, les données de Bob peuvent servir de référence. Alice doit alors retrouver, à partir de ce qu'elle a envoyé, les données bruitées reçues par Bob : c'est la *réconciliation inverse*. Alice est Dominique, Bob est Claude, et on écrit l'information secrète $\Delta I = I_{AB} - I_{BE}$.
- En théorie, rien n'empêche Alice et Bob de choisir tout autre protocole de réconciliation qui conduirait à une chaîne intermédiaire entre celle d'Alice et Bob. Néanmoins, ce type de réconciliation demande un protocole *interactif*, dans lequel Alice et Bob révèlent tous deux de l'information sur leur clé, ce qui rend très difficile l'évaluation de l'information secrète. Par conséquent, en pratique, seuls des algorithmes de réconciliation directe ou inverse sont utilisés.

Side information

Dans les télécommunications classiques, les utilisateurs n'ont en général accès qu'à un seul canal de transmission, a priori bruité. Que le code soit systématique ou non, la totalité du mot-code est donc transmise par le canal, à charge pour Bob de décoder le mot-code bruité *a posteriori*.

Dans le contexte de la cryptographie quantique, les utilisateurs disposent d'un canal classique non-bruité — mais espionnable — en plus du canal quantique. Dans le cas où l'on utilise un code systématique, une stratégie simple pour Alice est donc d'envoyer le « message » aléatoire sur le canal quantique, et d'envoyer ensuite les bits de redondance par le canal classique, de manière à ce que Bob les reçoive de manière non bruitée. Ceci simplifie notablement le décodage, mais cette information auxiliaire, ou *side information*, est bien sûr accessible à l'espion. Il faut donc la prendre en compte dans l'expression du taux secret : nous détaillerons cette modification dans la section 7.2.3.

2. De ce fait, il arrive que le terme *code correcteur* soit utilisé abusivement, pour désigner un algorithme complet. Ceci est encore accentué par le fait que ces algorithmes sont souvent implémentés dans un ordinateur, sous forme de… *code*.

Qualités d'un bon protocole

Terminons par quelques qualités que devrait posséder un protocole de correction d'erreurs adapté à la cryptographie quantique. Tout d'abord, nous venons de le mentioner, la correction des erreurs nécessite la transmission sur le canal classique de redondances propres à aider au décodage. Le canal classique est non-bruité mais parfaitement accessible à l'espion, et un bon code correcteur doit donc révéler le moins d'information possible tout en assurant le décodage, c'est-à-dire avoir un taux aussi proche que possible de la limite de Shannon.

En pratique, un compromis doit être trouvé entre taux du code et temps de décodage : plus un protocole fonctionne près de la limite de Shannon, plus le nombre d'itérations nécessaires pour le décodage est important. Dans l'optique d'un système réel, les données doivent être corrigées plus vite qu'elles ne sont stockées (fonctionnement en temps réel), et un bon protocole de correction doit donc être aussi rapide que possible.

Enfin, il est préférable que les communications soient unidirectionnelles : dans ce cas, un bit transmis correspond à un bit accessible à l'espion, alors que l'analyse de sécurité d'un code bi-directionnel est plus complexe.

7.1.3 Quelques protocoles de correction d'erreurs

Code à répétition

C'est l'un des codes les plus simples à implémenter. Claude envoie sa chaîne aléatoire, en répétant chacun des bits n fois. Si chaque bloc de n bits ne contient pas plus de $\left[\frac{n-1}{2}\right]$ erreurs, Dominique est capable de retrouver le bit initial par vote majoritaire, c'est-à-dire en choisissant à chaque fois le bit le plus fréquent parmi les n. Ce type de code est très redondant, puisqu'un bit transmis ne contient que $1/n$ bit d'information, et fonctionne donc généralement très loin de la limite de Shannon.

Cascade

Cascade [62] est l'un des protocoles correcteurs les plus utilisés en cryptographie quantique. L'idée du protocole est de décomposer le message bruité en blocs, de façon à ce que chaque bloc ne contienne en moyenne que 0,73 erreur (valeur optimale, voir [63]). Le décodage s'effectue ensuite de façon itérative entre Alice et Bob, bloc par bloc, ce qui présente l'avantage de ne révéler que peu de bits pour la correction. Le protocole fonctionne pour des taux d'erreurs compris entre 0 et 25%.

Winnow

Winnow [64] est un protocole interactif similaire à Cascade, dans lequel le décodage s'effectue en utilisant un code de Hamming (voir figure 7.1). Winnow présente l'avantage d'être plus rapide que Cascade, mais il est généralement moins efficace, hormis pour des taux d'erreurs compris entre 10% et 18%, qui ne sont que très rares en cryptographie quantique (BB84 ne fournit plus de bits secrets pour un taux supérieur à 11 %).

Turbo codes

Les turbo codes ont été inventés en 1993 [65] ; il s'agit d'une concaténation de codes convolutifs, c'est-à-dire des codes dans lesquels les bits entrent dans un encodeur sous forme de flux, et les parités sont générées « à la volée ». Ces codes ont révolutionné la correction d'erreurs lors de leur invention, car ils peuvent être implémentés facilement sur des cartes programmables (de type FPGA par exemple), et présentent des performances surpassant de loin les autres codes correcteurs connus à l'époque.

7.2 Codes LDPC

Pour extraire une chaîne commune à partir des données transmises par le canal quantique, nous utilisons un protocole de réconciliation fondé sur des codes LDPC (pour *low-density parity-check*).

7.2.1 Codes *parity-check*

Matrice de parités

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

Figure 7.1 : *Matrice de parités d'un célèbre code de Hamming. Dans son interprétation systématique, les $k = 7$ colonnes correspondent aux bits du message à coder, les $r = 3$ lignes correspondent aux bits de parité. Le mot-code comporte donc $n = 10$ bits, et le taux du code est $R = \frac{7}{10} = 70\%$.*

Les codes à parité (plus couramment, *parity-check*) sont des codes correcteurs dans lesquels les redondances sont déterminées en effectuant des calculs de parités sur les bits du message. De façon générale, un code *parity-check* définit chacun des r bits de redondance, aussi appelés *syndrômes*, comme la parité d'un sous-ensemble des bits du message, c'est-à-dire la somme binaire des bits mis en jeu.

Il est donc possible de représenter le code sous forme d'une matrice $r \times k$ (voir figure 7.1), dans laquelle chaque ligne correspond à un bit de parité : un 1 dans la ligne j, colonne i, signifie que le bit i du message est pris en compte dans le calcul du bit de parité j. Comme son nom l'indique, un code LDPC est un code *parity-check* dans lequel la matrice de parités ne contient qu'une densité faible de 1.

Les codes LDPC que nous utilisons ont une matrice de parités contenant $k = 200\,000$ colonnes, et leur densité en 1 est de l'ordre de 10^{-4}. En pratique, nous utilisons donc 10 blocs de données, contenant 40 000 impulsions de données chacun ; 50 % d'entre elles sont utilisées pour évaluer les paramètres du canal, et nous concaténons les $10 \times 20\,000$ restantes pour former le message à corriger.

Graphe de Tanner

Une autre représentation intéressante est le graphe bipartite de Tanner (figure 7.2). Il met en vis-à-vis les bits de message et les bits de parité, en représentant les liens entre les nœuds de parité et les nœuds de message qui interviennent dans le calcul de ces parités.

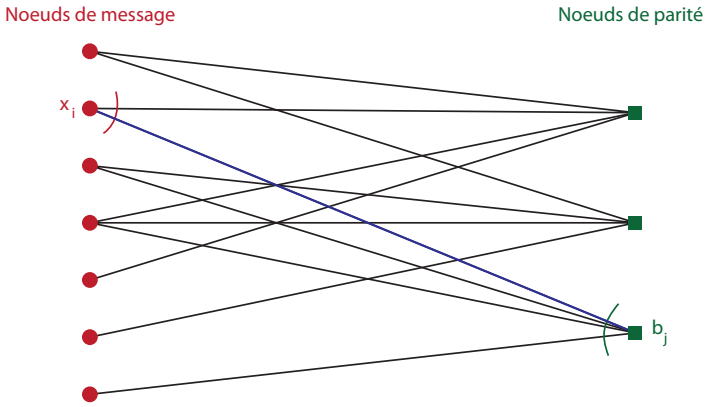


Figure 7.2 : *Graphe de Tanner, correspondant à la matrice de parités de la figure 7.1.*

Cette représentation est particulièrement pratique pour les codes LDPC, dans la mesure où leurs matrices de parités sont généralement trop grandes pour pouvoir être écrites explicitement.

7.2.2 Décodage des codes LDPC

Plaçons-nous, dans un premier temps, dans le cas où Claude envoie une modulation binaire, transmise par un canal bruité.

Les deux types de canaux bruités les plus usuels en télécommunications classiques sont :

- le canal binaire symétrique (BSC), qui intervertit la valeur des bits avec une probabilité p. C'est la version classique du canal considéré en cryptographie quantique à variables discrètes (par ex. BB84) ;
- le canal AWGN (Additive white Gaussian noise) qui ajoute au signal un bruit gaussien blanc de variance σ^2, indépendant du message. C'est une version classique de notre modèle du canal gaussien, présenté dans la section 3.3.1.

Le processus de réconciliation est indépendant des considérations quantiques liées à la sécurité. Il n'existe donc aucune différence, de ce point de vue, entre les versions classiques et quantiques des canaux.

Rapport de vraisemblance (LLR)

Après le passage par le canal bruité, Dominique obtient une chaîne de valeurs, qui sont discrètes si le canal ajoute un bruit discret (de type BSC), et réelles si le canal ajoute un bruit continu (de type AWGN).

Les algorithmes de *décodage dur*, tels que Cascade ou Winnow, binarisent directement le signal reçu par Dominique, et corrigent les erreurs binaires entre Claude et Dominique. Dans le cas d'un détecteur discret (tel qu'un détecteur de photons uniques), ou d'une modulation binaire passant par un canal BSC, l'information reçue par Bob est déjà binaire, et ces algorithmes sont donc bien adaptés.

En revanche, si le résultat de mesure prend des valeurs continues, ce type d'algorithme n'est pas optimal, puisqu'il nous fait perdre une partie de l'information. Prenons le cas d'une modulation binaire $\{-1,1\}$ traversant un canal AWGN. Si le résultat de mesure de Bob est 3,2 ou $-4{,}3$ on peut dire avec une bonne confiance que le bit envoyé était respectivement 1 ou -1. En revanche, si le résultat est proche de 0, il est difficile de conclure.

Le rapport de vraisemblance (LLR, pour *log-likelihood ratio*) permet de formaliser cette considération. Soit une variable aléatoire binaire X, on définit :

$$\mathrm{LLR} = \ln\left(\frac{\Pr(x=1)}{\Pr(x=0)}\right) \tag{7.2}$$

L'expression s'interprète comme suit :

- si le LLR est très positif, $\Pr(x=1) = \frac{e^{\mathrm{LLR}}}{1+e^{\mathrm{LLR}}} \to 1$
- si le LLR est nul, $\Pr(x=1) = \Pr(x=0) = 1/2$
- si le LLR est très négatif, $\Pr(x=1) \to 0$.

Le LLR est donc une expression de la certitude sur la valeur d'une variable binaire. À partir de ces LLR, on peut définir un autre type de décodeur, dit *mou* par opposition aux décodeurs durs, qui utilise le caractère continu des données pour optimiser le décodage.

Décodage mou

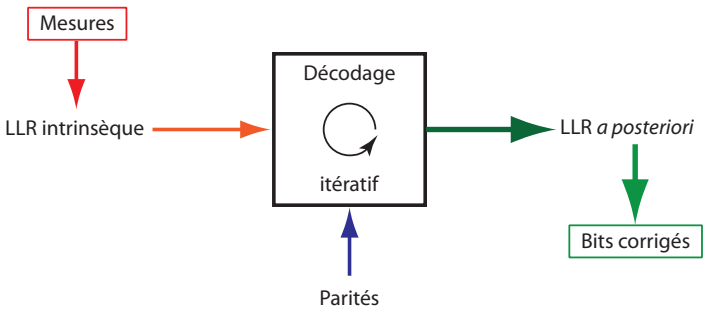


Figure 7.3 : *Schéma de principe d'un décodeur mou.*

Un décodeur mou peut être décrit par la figure 7.3. Il possède deux entrées : l'information intrinsèque, c'est-à-dire le degré de certitude sur la valeur du bit donné uniquement par le résultat des mesures, et l'information auxiliaire de décodage (les parités). En combinant ces deux entrées, le décodeur fournit une nouvelle information *a posteriori*, qui en principe améliore la certitude de Dominique sur la valeur des bits de Claude.

Un décodeur mou fonctionne en général sur un principe itératif. Dominique utilise tout d'abord ses résultats de mesure $\{y_i\}$ pour déterminer l'information intrinsèque sur les bits $\{x_i\}$, qui s'exprime sous la forme de LLR conditionnels :

$$\mathrm{LLR}_i^{\text{intrinsèque}} = \ln \frac{\Pr(x_i = 1|y_i)}{\Pr(x_i = 0|y_i)} \tag{7.3}$$

Ces LLR *a priori* sont ensuite combinés avec les parités fournies par Claude dans un sous-algorithme, qui fournit une nouvelle valeur affinée des LLR. Récursivement, ces LLR *a posteriori* sont réinjectés dans le sous-algorithme en tant que LLR *a priori*, combinés de nouveau avec les parités, etc.

Après un certain nombre d'itérations, les LLR *a posteriori* sont suffisamment grands en valeur absolue pour que Dominique puisse attribuer une valeur à chacun des bits, avec une bonne certitude. Les erreurs sont alors corrigées avec une bonne probabilité. Néanmoins, du fait même de la nature des LLR, un décodeur mou ne peut corriger les erreurs avec une efficacité de 100 %. Nous verrons dans la partie 7.3.2 comment prendre en compte les échecs de décodage.

Parités et LDPC systématiques

Dans l'interprétation non-systématique des codes LDPC, les mots-code M sont conçus de façon à ce que toutes leurs parités soient nulles : ils vérifient $H \cdot M = 0$, où H est la matrice de parités. Après une transmission bruitée, le mot-code peut avoir été modifié, auquel cas certaines des parités valent 1. Bob sait alors que le message a été corrompu, et il s'agit pour lui de retrouver le mot-code original, sachant que toutes les parités doivent valoir 0.

Dans notre cas (interprétation systématique), les parités ne sont pas modifiées par la transmission, puisqu'elles ont été transmises par un canal sans bruit. En revanche, les parités du message peuvent valoir 0 ou 1. Pour prendre en compte ce fait dans le décodage, nous devons introduire un nouveau nœud de variable x_{k+1}, auquel nous attribuons la valeur 1 avec une probabilité de 1 (c'est-à-dire un LLR = $+\infty$), et nous le relions à tous les nœuds de parité qui sont impairs. Ce nœud va donc « forcer », lors du décodage, la parité de tous les nœuds de parité auxquels il est relié.

Algorithme de décodage : *belief propagation*

Dans le cas des matrices LDPC, où la densité de 1 est suffisamment petite, Gallager a proposé en 1962 [66] un algorithme de décodage particulièrement efficace, fondé sur la représentation en graphe de Tanner. Cet algorithme est appelé indifféremment *belief propagation* ou *message passing*.

Introduisons d'abord quelques notations :

- q_{ij} est un message envoyé par le nœud de variable x_i au nœud de parité b_j. Ce message est un rapport de vraisemblance, exprimant la certitude du nœud x_i sur sa valeur, compte tenu de toutes les informations qu'il possède.
- r_{ji} est un message envoyé par le nœud de parité b_j au nœud de variable x_i. Ce message est un rapport de vraisemblance, exprimant la certitude du nœud b_j sur la valeur de x_i, compte tenu de toutes les informations qu'il possède.

L'algorithme proposé par Gallager [66] s'explicite alors comme suit :

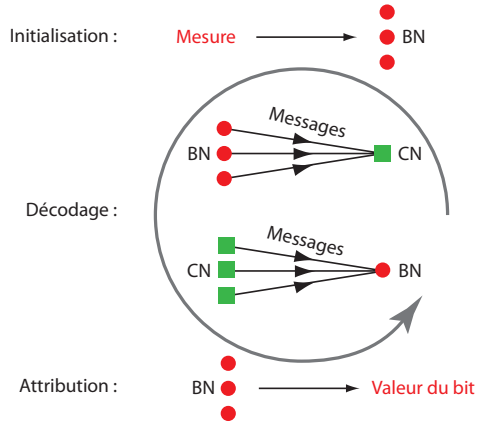


Figure 7.4 : *Vision schématique de l'algorithme Belief Propagation. BN et CN représentent respectivement les nœuds de variable (bit nodes) et de parité (check nodes).*

- **Initialisation :** Nous initialisons les variables avec les informations dont nous disposons : les LLR des nœuds de variable x_i sont initialisés avec la valeur des LLR intrinsèques correspondants. Les messages q_{ij} et r_{ji} sont eux tous mis à zéro.
- **Etape 1 : Message variable → parité.** Dans un premier temps, chaque nœud de variable envoie aux nœuds de parité connectés son message q_{ij}.

$$\underbrace{q_{ij}}_{\text{mis à jour}} \leftarrow \text{LLR}(x_i) - \underbrace{q_{ij}}_{\text{original}}$$

- **Etape 2 : Mise à jour des parités.** Chaque nœud de parité b_j a reçu plusieurs messages des variables. En les comparant, il calcule une valeur intermédiaire L_j :

$$L_j \leftarrow \prod_{\text{nœuds } i \text{ reliés à } j} \text{sign}(q_{ij}) \quad \varphi\left[\sum_{\text{nœuds } i \text{ reliés à } j} \varphi\left(|q_{ij}|\right)\right]$$

- **Etape 3 : Message parité → variable.** A partir de cette valeur intermédiaire, le nœud de parité b_j peut calculer les messages r_{ji} à renvoyer aux nœuds de variable :

$$r_{ji} \leftarrow \text{sign}(L_j)\ \text{sign}(r_{ji})\ \varphi\left[\varphi(L_j - \varphi(r_{ji})\right]$$

- **Etape 4 : Mise à jour des LLR de variable.** Enfin, chaque nœud de variable peut calculer son LLR *a posteriori*, à partir du LLR *a priori* ainsi que de tous les messages qu'il a reçus :

$$\text{LLR}^{a\ posteriori}(x_i) \leftarrow \text{LLR}^{a\ priori}(x_i) + \sum_{\text{nœuds } j \text{ reliés à } i} r_{ji}$$

- **On réitère (1),(2),(3),(4).**

Dans ces formules, la fonction φ doit répondre à l'équation fonctionnelle $\varphi \circ \varphi = \mathbb{I}$,

et s'écrit généralement

$$\varphi(x) = -\ln\left[\tanh\left(\frac{x}{2}\right)\right]$$

Après chaque itération, nous attribuons à chacun des nœuds de variable la valeur la plus probable compte tenu de son LLR *a posteriori*. A partir de ces valeurs de bit, nous calculons les valeurs des parités. Deux cas peuvent se produire :

- Les parités correspondent à celles envoyées par Claude. Nous sommes alors certains que les erreurs ont été corrigées, et arrêtons l'algorithme.
- Au moins une parité ne correspond pas. Il est alors impossible de savoir si nous sommes « loin » ou non de la correction totale : il peut arriver qu'une seule parité soit fausse mais que le mot-code corrigé soit très différent du mot-code original. Nous définissons donc un nombre maximal d'itérations : une fois ce nombre atteint, nous arrêtons l'algorithme. Le mot-code n'est pas parfaitement corrigé, mais si le code est bien conçu, il y a une bonne probabilité pour que le taux d'erreur résiduel soit très faible. Néanmoins, il y a toujours une probabilité, comme nous l'avons dit plus haut, que l'algorithme ait largement échoué à corriger le mot-code, auquel cas il peut rester beaucoup d'erreurs.

7.2.3 LDPC et variables continues

Revenons maintenant à la modulation gaussienne de notre protocole de cryptographie quantique. Puisque le protocole de Gallager est conçu pour des valeurs de référence discrètes, il faut l'adapter à une modulation continue. Gilles Van Assche a initialement développé dans [61] une technique de réconciliation par tranches, permettant la réconciliation de variables continues. Matthieu Bloch a ensuite adapté cette technique dans [67], en introduisant les concepts d'information molle et de décodage itératif, de façon à améliorer les performances du décodage.

Binarisation et décodage itératif

Dans un premier temps, nous devons établir un lien entre notre distribution réelle et un ensemble de mots binaires. Pour discrétiser notre distribution gaussienne, nous la découpons en 2^N intervalles réels. Nous attribuons ensuite à chaque valeur réelle y une représentation discrète $\hat{y}$, sous la forme d'une chaîne binaire de N bits, en fonction de la position de y dans la distribution. La variable aléatoire Y est donc décomposée en N variables aléatoires $\hat{Y}^{(i)}$, où i est le numéro de la tranche correspondante : $\hat{Y}^{(1)}$ correspond au bit de plus fort poids, alors que $\hat{Y}^{(N)}$ est le bit de plus faible poids.

L'idée du décodage itératif vient du fait que les $\hat{Y}^{(i)}$ ne sont pas indépendants, et le décodage d'une tranche apporte de l'information sur la suivante. Le décodage se fait donc avec l'algorithme de *message passing* binaire, tranche par tranche, en décodant d'abord les bits de plus faible poids $\hat{Y}^{(N)}$, et en remontant vers les bits de plus fort poids. Une fois que toutes les tranches ont été décodées une fois, on réitère le processus pour affiner le décodage.

Le taux des codes utilisés dans le décodage dépend de la tranche considérée. En effet, la tranche 1 a relativement peu d'erreurs, ce qui implique un taux proche de 1. Au contraire, les tranches de faible poids sont beaucoup plus soumises au bruit quantique :

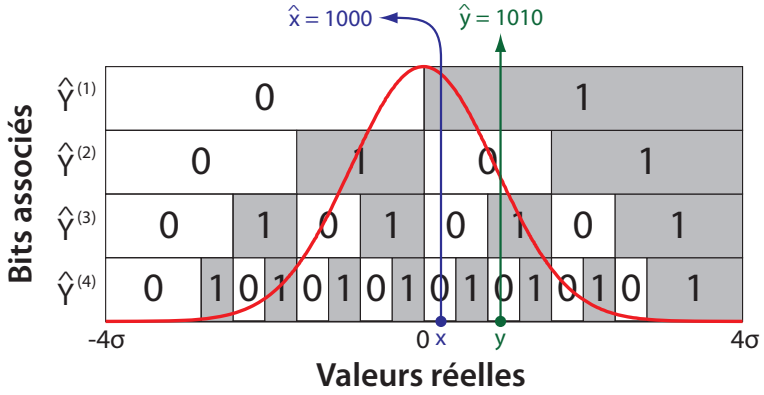


Figure 7.5 : *Schéma de binarisation des valeurs continues x et y en chaînes binaires $\hat{x}$ et $\hat{y}$.*

il faut donc beaucoup de redondance pour pouvoir corriger les erreurs, c'est-à-dire un code de faible taux.

Efficacité de réconciliation et information secrète

Pour pouvoir effectuer le décodage, nous avons dû révéler de l'information sur le canal classique, et nous avons perdu de l'information du fait de la binarisation de nos données continues. Pour modéliser cet effet, nous pouvons donc introduire le paramètre β, correspondant à une efficacité, de façon à ce que :

$$\Delta I = \beta I_{AB} - I_{BE} \tag{7.4}$$

Il s'agit maintenant d'exprimer cette efficacité en fonction des paramètres connus de la réconciliation. Après la réconciliation et la correction d'erreurs, Alice et Bob partagent une variable aléatoire discrète $\hat{X}$. L'information partagée entre Alice et Bob correspond donc à l'entropie de $\hat{X}$: $H(\hat{X})$. Or, cette information partagée correspond d'une part à l'information partagée par le canal quantique, I_{AB}, atténuée de l'efficacité de réconciliation β, et d'autre part à l'information révélée sur le canal classique, I_{rev}. On peut donc écrire $H(\hat{X}) = \beta I_{AB} + I_{\text{rev}}$, c'est-à-dire

$$\beta = \frac{H(\hat{X}) - I_{\text{rev}}}{I_{AB}} \tag{7.5}$$

Pour une modulation binaire, on peut écrire directement I_{rev} en fonction du taux R du code utilisé : $I_{\text{rev}} = 1 - R$. Pour notre modulation gaussienne discrétisée, on peut l'exprimer en fonction des taux $R^{(i)}$ des N codes utilisés pour les N tranches : $I_{\text{rev}} = N - \sum R^{(i)}$. Ce qui donne finalement :

$$\beta = \frac{H(\hat{X}) - N + \sum R^{(i)}}{I_{AB}} \tag{7.6}$$

Ce paramètre β est crucial en cryptographie quantique à variables continues. En effet, dans notre système, l'information partagée par Alice et Bob (I_{AB}) est très proche de l'information espionnée par Eve (I_{BE}, voir figure 6.7), et les deux courbes sont

tangentes à longue distance. Ainsi, si I_{AB} est diminuée d'un facteur β strictement différent de 1, les deux courbes se croisent pour une certaine transmission, et plus aucune information secrète ne peut être transmise entre Alice et Bob au delà. La section suivante est largement consacrée à l'optimisation de ce paramètre β, qui a été une partie importante de ce travail de thèse.

7.3 Optimisation de la réconciliation

7.3.1 Correction déterministe avec les codes BCH

Le décodage LDPC n'est pas déterministe, et il laisse souvent quelques erreurs entre les chaînes d'Alice et de Bob. Le plus fréquemment, il s'agit d'effets de bords, et le nombre d'erreurs est très faible (typiquement entre 2 et 10 sur 400 000) ; plus rarement, le décodage ne fonctionne pas du tout, et le nombre d'erreurs est alors beaucoup plus grand (quelques milliers). Il serait dommage de traiter ces deux cas de la même façon, et de défausser tous les blocs non parfaitement corrigés.

Les codes BCH, du nom de leurs inventeurs, Bose, Ray-Chaudhuri [68] et Hocquenghem [69], sont des codes binaires de faible complexité, utilisés pour décoder très simplement des chaînes ne contenant qu'un nombre réduit d'erreurs. Plus précisément, un code BCH d-correcteur est conçu pour pouvoir corriger jusque d erreurs : si le nombre d'erreurs résiduel est inférieur ou égal à d, celles-ci sont corrigées avec une très grande probabilité. Notons qu'un code BCH n'est pas capable de détecter la présence de plus de d erreurs : si c'est le cas, il revoie un succès de la correction alors même qu'il a échoué à corriger. [3]

Gilles Van Assche a implémenté dans notre programme la structure d'un code BCH permettant de décoder jusque 19 erreurs. Le message à décoder est composé des 400 000 bits sortant du décodage LDPC ; le décodeur nécessite 380 bits de redondance, ce qui donne un taux $R_{\mathrm{BCH}} = 0{,}99905$, et donc une excellente efficacité (typiquement $1 - 5.10^{-4}$). Grâce à ce code, nous corrigeons la majorité des blocs mal décodés par les codes LDPC. Le cas des quelques blocs non corrigés est géré grâce à un algorithme de vérification, présenté dans le chapitre 8.

7.3.2 Optimisation de l'efficacité des codes LDPC

L'efficacité de réconciliation globale β_{LDPC} des codes LDPC se décompose en deux efficacités partielles :

$$\begin{aligned} \beta_{\mathrm{LDPC}} &= \beta_{\mathrm{codes}}\beta_{\mathrm{tranches}} \\ \text{avec } \beta_{\mathrm{tranches}} &= \frac{H(\hat{X}) - N + \sum R^{(i)}_{\mathrm{optimal}}}{I_{AB}} \end{aligned} \tag{7.7}$$

où β_{codes} et $\beta_{\mathrm{tranches}}$ correspondent respectivement à l'efficacité des codes et à l'efficacité de discrétisation. L'efficacité β_{LDPC} tend vers 1 quand les deux sous-efficacités tendent vers 1, c'est à dire quand le taux $R^{(i)}$ de chaque code LDPC utilisé tend vers la capacité

3. En fait, il sera en mesure de détecter la non-correction s'il y a exactement $d+1$ erreurs dans le bloc.

$R^{(i)}_{\text{optimal}}$ du canal virtuel correspondant à la transmission du bit $\hat{Y}^{(i)}$, et que le nombre de tranches de binarisation devient grand.

Le choix des paramètres de décodage permet de jouer sur cette efficacité :

- β_{tranches} ne dépend que du schéma de binarisation : nombres de tranches et répartition des intervalles de binarisation.
- β_{codes} dépend de la différence entre le taux des codes choisis et la limite de Shannon.
- Enfin, nous fixons un taux d'échec maximal du décodage, qui conditionnera nos choix des autres paramètres.

Nous devons donc ajuster ces paramètres pour optimiser l'efficacité globable de réconciliation. Par ailleurs, puisque le rapport signal à bruit chez Bob dépend de la transmission du canal, il nous faut adapter tous les paramètres de façon dynamique, de façon à ce que l'efficacité de réconciliation ne varie pas trop si la transmission du canal fluctue au cours du temps.

Choix du nombre de tranches

De façon générale, le rapport signal à bruit est fixé par l'atténuateur d'Alice, et au cours de la transmission, il ne fluctue que peu (typiquement 1 %). Nous pouvons donc raisonnablement limiter l'optimisation au voisinage d'un rapport signal à bruit de 3, par exemple [2,4 ; 3,6]. Dans ces conditions, nous obtenons une information mutuelle I_{AB} de l'ordre d'un bit par impulsion (voir section 6.4.2). Ainsi, qualitativement, lors d'une discrétisation raisonnablement régulière, le premier bit de Y contiendra beaucoup d'information, le deuxième une quantité plus faible mais significative, et les suivantes très peu.

Nous avons donc choisi de discrétiser les données sur quatre bits, sur lesquels nous révélons les deux bits les plus bruités : ces derniers n'apportent donc pas d'information puisqu'ils sont presque entièrement « noyés » dans le bruit de photon, mais leur révélation permet de faciliter le décodage des deux bits utiles.

Notre chaîne réconciliée finale contient donc $2 \times 200\,000 = 400\,000$ bits.

Choix de l'intervalle de binarisation

Il nous reste ensuite à déterminer la façon dont la distribution réelle va être découpée en 2^4 intervalles : nous avons *a priori* 15 paramètres indépendants pour placer ces intervalles. Jérôme Lodewyck a etudié au cours de sa thèse [5] ce problème d'optimisation à $2^n - 1$ paramètres, et a abouti à plusieurs conclusions :

- La division en intervalles équiprobables, compte tenu de la distribution gaussienne, ne conduit qu'à des efficacités d'environ $\beta_{\text{tranches}} \approx 80\%$.
- Une division simple et assez proche de la division optimale est, curieusement, la distribution uniforme. L'intérêt de cette distribution est de ne dépendre que d'un paramètre, l'intervalle de binarisation, qui peut donc être optimisé plus aisément.

À partir de ces considérations, nous découpons notre distribution gaussienne (dont nous connaissons le SNR) selon un certain intervalle de binarisation. Le but est alors

de calculer le taux optimal de décodage de la tranche i ; il s'exprime [70, 67] :

$$R^{(i)}_{\text{optimal}} = 1 - \Big(H(\hat{X}^{(i+1)}|\hat{X}) - H(\hat{X}^{(i)}|\hat{X})\Big) + \Big(H(\hat{X}^{(i+1)}) - H(\hat{X}^{(i)})\Big) \tag{7.8}$$

Il nous faut alors calculer l'entropie des tranches $H(\hat{X}^{(i)})$ et l'entropie conditionnelle $H(\hat{X}^{(i)}|X)$. Les probabilités $\Pr(\hat{X}^{(i)})$ et $\Pr(\hat{X}^{(i)}|X)$ ne dépendent que de la position des tranches et du SNR de la distribution gaussienne $\Pr(X)$. Les entropies s'expriment ensuite à partir de la définition de l'entropie et de l'entropie conditionnelle présentée dans le chapitre 3 :

$$\begin{aligned} H(\hat{X}^{(i)}) &= \Pr(\hat{X}^{(i)}=0)\log_2\Pr(\hat{X}^{(i)}=0) + \Pr(\hat{X}^{(i)}=1)\log_2\Pr(\hat{X}^{(i)}=1) \\ H(\hat{X}^{(i)}|X) &= \int_{-4\sigma}^{4\sigma} \Pr(\hat{X}^{(i)}=0|X)\log_2\Pr(\hat{X}^{(i)}=0|X)\mathrm{d}X + \\ &\quad \int_{-4\sigma}^{4\sigma} \Pr(\hat{X}^{(i)}=1|X)\log_2\Pr(\hat{X}^{(i)}=1|X)\mathrm{d}X \end{aligned} \tag{7.9}$$

Une fois munis de ces taux optimaux et des entropies de tranches, la formule 7.7 nous permet de calculer l'efficacité de binarisation. Nous avons donc tracé, pour plusieurs SNR, l'efficacité de binarisation en fonction de l'intervalle de binarisation (voir figure 7.6).

Il est remarquable de voir que, pour une division uniforme, l'efficacité de binarisation ne varie que peu autour de l'intervalle optimal. Pour un SNR de 3, l'efficacité est supérieure à 0,97 pour des intervalles compris entre 0,25 et 0,45, pour un maximum à 0,312 ($\beta_{\text{tranches}} = 0{,}977$).

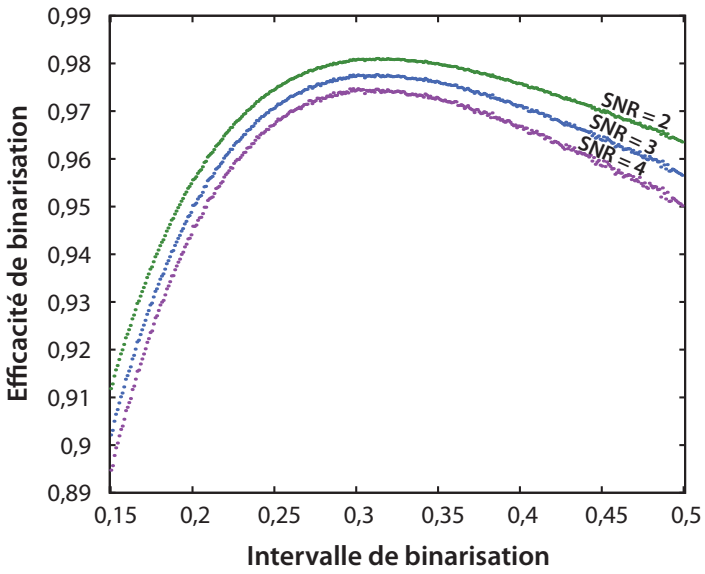


Figure 7.6 : *Efficacité de binarisation en fonction de l'intervalle de binarisation, pour des SNR de 2, 3 et 4.*

Ainsi, au premier ordre, changer l'intervalle de binarisation n'a que peu d'effet sur l'efficacité de binarisation, et revient principalement à modifier la limite de Shannon de

décodage, c'est-à-dire le taux optimal des codes à utiliser pour chacune des tranches. Pour un SNR de 3, nous avons tracé sur la figure 7.7 le taux optimal des codes des quatre tranches, en fonction de l'intervalle de binarisation. On voit que, pour un intervalle correspondant aux efficacités de binarisation optimales, les taux optimaux de la tranche 3 et 4 sont presque nuls. Plus précisément, l'information contenue dans la tranche 3 pour un intervalle de 0,3 est d'environ 0,01 bit. Il est donc plus intéressant de la révéler entièrement.

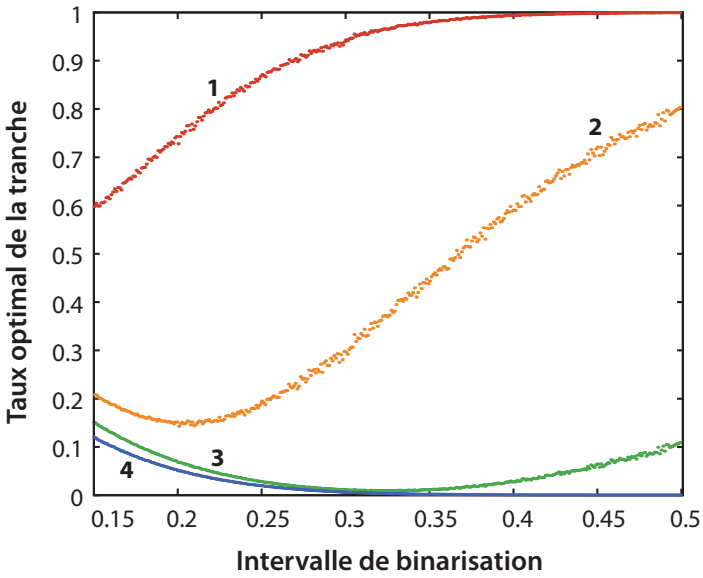


Figure 7.7 : *Taux optimaux (limites de Shannon) des quatre tranches de binarisation, en fonction de l'intervalle de binarisation.*

Nous nous concentrons donc sur l'optimisation de l'efficacité de décodage comme suit :

- En choisissant l'intervalle de binarisation, nous fixons la limite de Shannon.
- En choisissant le taux des codes, nous fixons la distance à cette limite.
- L'efficacité de décodage β_{codes} est alors déterminée par cette distance.

Choix du taux des codes

Comme l'ont montré Richardson *et al.* dans [71], les performances d'un code LDPC ne dépendent que de la distribution de ses connectivités, c'est-à-dire, pour tout m, de la proportion de nœuds de parité faisant intervenir m nœuds de variable. Ainsi, une équipe du LTHC de Lausanne a réalisé en 2001 une optimisation numérique de la distribution de ces connectivités, de façon à fournir des codes pour de nombreux taux discrets, allant de 5 % à 99 %, adaptés à un canal gaussien de type AWGN [72].

En utilisant ces distributions, ainsi qu'un programme écrit par Matthieu Bloch, Jérôme Lodewyck a généré des codes adaptés à nos besoins, en sélectionnant systématiquement les codes fournissant une vitesse de décodage élevée. Néanmoins, nous ne disposons de codes que pour certains taux discrets ; ainsi, quand le rapport signal

Taux	Distribution des connectivités			
42 %	70 %	des nœuds de parité sont reliés à	7	nœuds de variable
	30 %	"	8	"
	24,0 %	des nœuds de variable sont reliés à	2	nœuds de parité
	20,0 %	"	3	"
	13,7%	"	6	"
	4,1 %	"	7	"
	8,7 %	"	9	"
	9,8 %	"	10	"
	1,4 %	"	28	"
	12,2 %	"	32	"
	6,1%	"	35	"
95 %	100 %	des nœuds de parité sont reliés à	96	nœuds de variable
	13,1 %	des nœuds de variable sont reliés à	2	nœuds de parité
	25,9 %	"	3	"
	18,7%	"	7	"
	11,6%	"	8	"
	8,1%	"	19	"
	22,6 %	"	21	"

à bruit change, nous ne pouvons pas ajuster finement le taux du code pour rester à distance fixe de la limite de Shannon.

Plutôt que de conserver l'intervalle de binarisation constant et d'ajuster de façon imparfaite le taux des codes pour rester proche de la limite de Shannon, nous avons choisi de conserver le taux des codes constant et d'ajuster l'intervalle de binarisation afin de régler la limite de Shannon où nous le souhaitons. Après avoir testé plusieurs de nos codes, nous avons choisi deux codes particulièrement performants, de taux 0,95 pour la tranche 1 et 0,42 pour la tranche 2, que nous utilisons pour tous les SNR de [2,4 ; 3,6]. Nous décrivons dans le tableau ci-dessous les caractéristiques de ces deux codes.

Nombre d'itérations

Comme nous l'avons remarqué plus haut, l'idée du décodage itératif est de décoder une première fois toutes les tranches, en remontant vers les bits les moins bruités, puis de revenir à la tranche la plus bruitée et de réitérer le décodage (éventuellement plusieurs fois). En effet, les tranches ne sont pas indépendantes, et le décodage d'une tranche supérieure ou inférieure à la tranche i apporte de l'information utile pour le décodage de la tranche i.

Heuristiquement, nous avons remarqué que le décodage de la tranche 2 ne progressait presque plus à partir de la 40ème itération ; en revanche, en décodant la tranche 1 (ce qui est rapide) puis en revenant sur la tranche 2, le décodage se conclut très rapidement. Nous avons donc choisi le schéma de décodage suivant :

- Révélation sur le canal public de la totalité des tranches 4 et 3 les plus bruitées
- Décodage, pendant 40 itérations, de la tranche 2

- Décodage, pendant 15 itérations maximum, de la tranche 1
- Retour sur la tranche 2 : décodage pendant 10 itérations maximum

Optimisation de l'intervalle de binarisation en fonction du taux d'échec

Dans les paragraphes précédents, nous avons fixé : le nombre de tranches de binarisation, le taux des codes utilisés et le nombre d'itérations du décodage. Le problème d'optimisation est donc simplifié, puisqu'il ne nous reste plus qu'à optimiser l'intervalle de binarisation en fonction du SNR et du taux d'échec maximal.

Pourquoi un taux d'échec ? Lors d'un décodage, plusieurs effets se contrebalancent : plus un code travaille près de la limite de Shannon, plus il nécessite d'itérations (et donc de temps) pour faire aboutir le décodage. Au bout d'un nombre d'itérations fixé, un décodage trop proche de la limite de Shannon n'aura pas terminé le décodage, et des erreurs persisteront. Au contraire, si le code fonctionne loin de la limite, il décodera rapidement et son efficacité sera faible, mais il restera toujours une probabilité que le décodage échoue. Quel que soit le taux du code, on trouvera toujours des cas où le décodage n'aura pas fonctionné ; on sait juste que ce taux d'échec diminue quand la distance au taux optimal augmente.

Il nous faut donc choisir un taux d'échec maximal qui servira de condition supplémentaire pour l'optimisation du taux secret. Ce choix n'est pas évident : échouer à réconcilier un bloc revient à perdre des bits secrets, et donc à diminuer le taux de génération de clé ; à l'opposé, les informations mutuelles I_{AB} et I_{BE} sont souvent très proches, et l'augmentation de quelques pour cent (voire de quelques pour mille à longue distance) de l'efficacité de réconciliation peut améliorer significativement l'information secrète $\Delta I = \beta I_{AB} - I_{BE}$.

A faible distance, nous aurons donc intérêt à privilégier un taux d'échec bas ; à l'inverse, quand le taux secret est presque nul à longue distance, perdre 50 % des blocs peut être intéressant si β augmente en contrepartie.

Nous avons donc créé un programme optimisant de façon systématique l'efficacité de réconciliation (à travers l'intervalle de binarisation) en fonction du SNR, pour un taux d'échec de 1 %, ce qui est suffisamment petit pour ne pas trop diminuer le taux secret, et suffisamment grand pour pouvoir améliorer significativement l'efficacité de réconciliation. Pour chaque valeur de SNR, et pour chaque intervalle de binarisation, celui-ci effectue 1000 réconciliations, et calcule le nombre de réconciliations ayant échoué. Il peut ainsi déterminer le taux d'échec associé à chaque intervalle de binarisation et SNR, et enfin tracer une courbe « iso-échec », présentée sur la figure 7.8.

7.3.3 Variantes de l'algorithme *message passing*

L'algorithme Message Passing que nous avons présenté précédemment n'est bien sûr pas la seule façon de décoder un code LDPC, et il est utile de chercher d'autres techniques plus rapides ou plus efficaces. En particulier, Radosavljevic *et al.* ont proposé [73] trois algorithmes inspirés de l'algorithme Message Passing original :

- Nous reprenons les notations de la partie 7.2.2 ; l'algorithme Message Passing

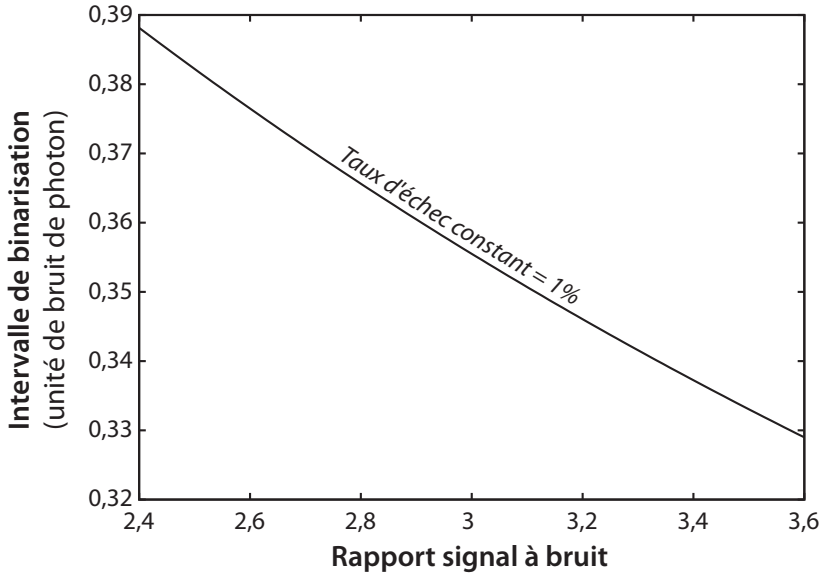


Figure 7.8 : *Intervalle de binarisation en fonction du rapport signal à bruit, optimisé de façon à ce que le taux d'échec de la réconciliation soit constant, fixé à 1 %. La courbe a été ajustée de façon* ad hoc *par la fonction* $-0{,}950863(\rho^2)^2 + 0{,}644345\rho^2 + 0{,}40710$ *; elle peut ainsi être utilisée par le programme pour ajuster la variance de façon dynamique, en fonction du SNR.*

---

original peut s'écrire de façon condensée :

$$\Big\{\forall i : \{\forall j : (1)\}\Big\} \rightarrow \Big\{\forall j : (2) \rightarrow \{\forall i : (3)\}\Big\} \rightarrow \Big\{\forall i : (4)\Big\}$$

- L'algorithme Row Message Passing (RMP) est alors décrit par :

$$\forall i : \quad \{\forall j : (1)\} \rightarrow (2) \rightarrow \{\forall j : (3) \rightarrow (4)\}$$

- L'algorithme Column Message Passing (CMP) est décrit par :

$$\forall j : \quad \{\forall i : (3)\} \rightarrow (4) \rightarrow \{\forall i : (1) \rightarrow (2)\}$$

- L'algorithme Row-Column Message Passing (RCMP) est décrit par :

$$\forall (i,j) : \quad (1) \rightarrow (3) \rightarrow (2) \rightarrow (4)$$

L'algorithme MP simple nécessite de parcourir d'abord tous les nœuds de variable reliés à chaque nœud de parité, et ensuite tous les nœuds de parité reliés à chaque nœud de variable. L'algorithme RMP, par exemple, est plus simple, puisque qu'il ne parcourt qu'une seule fois les nœuds de parité, en mettant à jour séquentiellement tous les nœuds de variable qui y sont reliés. Nous reproduisons ici (figure 7.9) le schéma présenté dans [73], résumant le fonctionnement des quatre algorithmes.

Intérêt Les trois algorithmes optimisés restent très proches de l'algorithme original, mais permettent de réaliser le décodage avec la même efficacité et en exactement deux fois moins d'itérations. Nous avons implémenté les trois algorithmes dans le programme

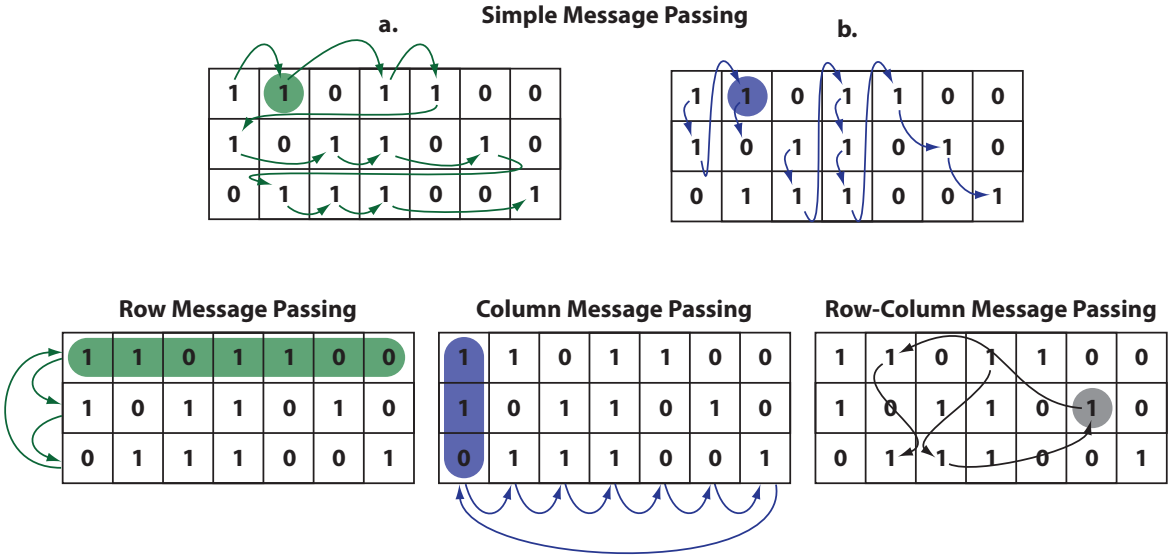


Figure 7.9 : *Principe de fonctionnement des algorithmes optimisés, en considérant l'exemple de la matrice de Hamming. Dans l'algorithme original, les liens sont parcourus deux fois, l'une pour mettre à jour tous les nœuds de variable, l'autre pour les nœuds de parité. Dans l'algorithme RMP (resp. CMP), les nœuds de parité (resp. variable) sont parcourus séquentiellement, en mettant à jour tous les nœuds de variable (resp. parité) qui y sont reliés. Enfin, dans l'algorithme RCMP, les liens variable/parité sont parcourus aléatoirement, en mettant à jour pour chaque lien les deux nœuds de parité et variable qui y sont reliés.*

---

d'extraction, et finalement opté pour le schéma RMP qui nécessite le moins de mémoire. Grâce à cette optimisation, le temps de décodage est diminué par deux, ce qui nous avait alors permis de franchir le seuil de taux de 1 kbit/s à 25 km, assez standard en cryptographie quantique.

8 Amplification de confidentialité et vérification de la clé

> Le plus terrible secret de ce monde serait qu'il n'y ait aucun secret.
>
> *Mémoires de 7 vies*
> Jean-François Deniau

Après le processus de réconciliation, Alice et Bob ont extrait une chaîne binaire partagée à partir de N mesures, et sont en mesure d'évaluer la quantité d'information secrète $z = N\Delta I$ présente dans cette chaîne. Pour obtenir une clé secrète, ils doivent trouver un algorithme qui comprime la chaîne réconciliée de longueur $n = 2N$ en une chaîne plus courte, de longueur $z \leqslant n$, de façon que cette dernière soit parfaitement inconnue à l'espion.

Pour que l'algorithme élimine l'information de l'espion, il doit (qualitativement) mélanger l'information connue par l'espion avec l'information qu'il ignore, de façon à ce que l'information connue soit distribuée de façon uniforme dans l'information inconnue. Cette étape est appelée *amplification de confidentialité*, et est mise en œuvre avec des fonctions de hachage. Les parties théoriques de ce chapitre s'inspirent largement du travail de Gilles Van Assche sur l'amplification de confidentialité [61].

8.1 Fonctions de hachage

8.1.1 Définition

Une fonction de hachage est définie de façon générale comme un procédé permettant, à partir de données fournies en entrée, d'obtenir une empreinte de ces données en sortie. L'empreinte est représentative des données, c'est-à-dire qu'elle permet de les identifier, bien qu'incomplètement, et de les distinguer rapidement d'un autre ensemble de données, même très proche.

L'une des utilisations les plus courantes du hachage est la vérification d'intégrité, dans la mesure où une bonne fontion de hachage produit avec une bonne probabilité

deux empreintes différentes pour deux entrées différentes, même quand un seul bit est altéré. Il est ainsi possible de vérifier que des données stockées sur un support détériorable, comme un disque optique, n'ont pas subi d'altération ; ou que des données transmises par un canal bruité ont été reçues sans erreur. Par exemple, les deux derniers chiffres du numéro de sécurité sociale sont une empreinte des 13 premiers. De même, le protocole TCP, permettant le contrôle du transfert des données sur Internet, inclut une étape de hachage pour vérifier le succès de la correction d'erreurs.

Les fonctions de hachage sont particulièrement utilisées en informatique et en cryptologie, et les données d'entrée sont alors des chaînes de bits, comme des fichiers, des messages ou des clés. L'empreinte est une chaîne de bits de longueur z, que l'on nomme condensat, somme de contrôle, ou encore empreinte cryptographique. Nous nous placerons dans ce cas dans ce chapitre, même si le hachage peut s'appliquer à des ensembles plus généraux, et nous considérons des fonctions de hachage allant de l'ensemble des chaînes de n bits vers l'ensemble des chaînes de z bits.

8.1.2 Familles de fonctions de hachage

Du fait de leurs propriétés, les fonctions de hachage peuvent être utilisées pour réaliser un algorithme d'amplification de confidentialité. Néanmoins, Alice et Bob ne peuvent pas se contenter de n'utiliser qu'une seule fonction f, définie *a priori*, pour effectuer tous les hachages. Il n'y aurait alors aucun aléa dans l'algorithme, et l'espion pourrait utiliser des effets de coïncidence (deux messages $B_1 \neq B_2$, mais une même empreinte $f(B_1) = f(B_2)$) pour acquérir de l'information sur les clés finales.

Pour éviter cet effet, Alice et Bob doivent donc définir une *famille* de fonctions de hachage, dans laquelle ils choisissent de manière aléatoire, et pour chaque extraction de clé, une fonction de hachage à utiliser. De la même façon que le choix aléatoire de la quadrature de mesure force l'espion à intervenir de façon symétrique lors de son attaque, ce choix aléatoire de la fonction de hachage permet d'empêcher que l'espion n'exploite les effets de coïncidence lors de son attaque.

Universalité

L'universalité des familles de hachage est une mesure de la qualité de leur hachage. Considérons un message B de longueur n, et observons l'ensemble des hachages de B par toutes les fonctions de la famille. Si la famille est bien conçue, toutes les empreintes devraient apparaître le même nombre de fois, à savoir 2^{n-z} fois. Si une empreinte apparaît plus que les autres, l'aléa du hachage n'est plus parfait. De la même façon, deux messages, même très peu différents, ne devraient fournir le même hachage qu'avec une probabilité de 2^{-z}.

Une famille de fonctions de hachage est alors dite *universelle* si, pour tout couple de messages d'entrée différents B_1 et B_2, la probabilité que $f(B_1) = f(B_2)$ est d'au plus $1/2^z$. La probabilité porte ici sur f, qui parcourt la famille de fonctions.

Par extension, une famille de fonctions est dite ε-universelle si la probabilité que $f(B_1) = f(B_2)$ est d'au plus $\varepsilon/2^z$.

8.1.3 Sécurité de l'amplification de confidentialité

Il a été prouvé formellement dans [74] que l'amplification de confidentialité, réalisée par une fonction choisie aléatoirement dans une famille universelle, est inconditionnellement sûre. Plus précisément, pour des blocs à amplifier suffisamment longs, l'information accessible à l'espion sur la clé amplifiée est bornée par $\frac{2^{-s}}{\ln 2}$, où s est appelé *paramètre de sécurité*. Ce paramètre de sécurité correspond au nombre de bits qu'Alice et Bob sacrifient pour assurer la sécurité de la clé, ce qui implique que l'information accessible à Eve peut être rendue arbitrairement petite si Alice et Bob sacrifient suffisamment de bits sur leur clé finale. Le choix du paramètre de sécurité fait donc l'objet d'un arbitrage entre perte de secret et risque de compromission avec l'espion.

Dans le cas d'une famille ε-universelle, l'information de l'espion peut aussi être bornée [61] par :

$$z - H(K|h,B_E) = \log_2 \varepsilon + \frac{2^{-s-\log_2 \varepsilon}}{\ln 2} \tag{8.1}$$

$$\xrightarrow[\varepsilon-1\to 0]{} \frac{2^{-s}+(\varepsilon-1)}{\ln 2} \quad \left(\xrightarrow[s\to\infty]{} \frac{\varepsilon-1}{\ln 2}\right) \tag{8.2}$$

On voit alors qu'une famille non-universelle ne permet pas l'inconditionnalité de la sécurité, c'est-à-dire le fait de pouvoir rendre l'information de l'espion arbitrairement petite, mais limite le paramètre de sécurité à $s < -\log_2(\varepsilon - 1)$.

8.2 Algorithmes d'amplification de confidentialité

8.2.1 Taille des blocs amplifiés

Nous avons vu dans le chapitre précédent que la taille des blocs réconciliées était de 400 000 bits. Néanmoins, nous sommes libres de réaliser l'amplification sur des blocs plus longs. En particulier, l'évaluation de l'information secrète ΔI, qui induit la longueur z de la chaîne amplifiée, doit se faire sur un nombre suffisamment grand de bits pour limiter les effets de fluctuation statistique. L'algorithme d'amplification a été implémenté de façon à s'adapter à toutes les tailles d'entrée n et de sortie z. En pratique, nous utilisons souvent une taille des blocs d'entrée de 4 000 000, soit 100 blocs de données, générés en 10 s.

8.2.2 Chaînes binaires et corps fini $\mathrm{GF}(2^l)$

Une chaîne binaire de l bits $\overline{b_1 b_2 \cdots b_l}$ peut être représentée de manière bijective par un polynôme de degré strictement inférieur à l, $b_1 X^{l-1} + b_2 X^{l-2} + \cdots + b_l$. On note l'ensemble de ces polynômes de degré strictement inférieur à l et à coefficients binaires : $\mathrm{GF}(2^l)$.

Algébriquement, cet ensemble est un corps fini, qui permet donc la multiplication et l'addition. Néanmoins, le produit usuel de deux polynômes de $\mathrm{GF}(2^l)$ ne donne généralement pas un polynôme de degré inférieur à l. Pour résoudre ce problème, la multiplication dans $\mathrm{GF}(2^l)$ est définie comme le produit usuel, réduit modulo un polynôme irréductible de degré l, à coefficients binaires.

Pour pouvoir définir la multiplication dans $\mathrm{GF}(2^l)$, il nous faut définir ce polynôme irréductible de degré l. Or, pour les valeurs de l que nous voudrions utiliser, à savoir $l = n = 400\,000$ ou $4\,000\,000$, il n'existe pas, pour le moment, de polynôme irréductible connu. En revanche, l'irréductibilité d'un certain nombre de trinômes dont les degrés correspondent aux nombres de Mersenne a pu être démontrée, en particulier par Brent *et al.*, et la recherche progresse toujours. Nous présentons ces trinômes dans la figure 8.1.

8.2.3 Algorithme 0 : Multiplication dans $\mathrm{GF}(2^l)$

Il est alors possible de définir un ensemble de fonctions de hachage reposant sur la multiplication dans $\mathrm{GF}(2^l)$ [74] :

- La chaîne à comprimer, de longueur n, est complétée par des 0 jusqu'à atteindre une longueur l, puis convertie en sa représentation polynômiale $P_{\mathrm{in}}[X]$.
- Une fonction de hachage est définie par Alice et Bob, qui génèrent aléatoirement une autre chaîne de longueur l, représentée par $P_{\mathrm{h}}[X]$.
- Alice et Bob calculent alors le polynôme-produit $P_{\mathrm{in}}[X] \times P_{\mathrm{h}}[X]$, et le convertissent en sa représentation binaire.
- La sortie de la fonction de hachage, c'est-à-dire la clé secrète, est définie comme cette chaîne-produit, tronquée à la longueur z voulue.

Il est alors possible de montrer [76] que cette famille de fonctions de hachage, qui contient autant d'éléments que de chaînes de l bits (soit 2^l), est universelle. Néanmoins, l'algorithme habituel utilisé pour réaliser la multiplication, appelé *Shift and Add*, nécessite un nombre d'opérations qui varie quadratiquement avec la longueur des polynômes. Pour des blocs de données de plus de quelques milliers de bits à amplifier, il n'est donc pas exploitable en pratique.

8.2.4 Algorithme 1 : Accélération utilisant la transformée de Fourier discrète

Pour accélérer l'algorithme ci-dessus, Gilles Van Assche a proposé une méthode numérique astucieuse, qui permet de multiplier rapidement des polynômes de $\mathrm{GF}(p)[X]/(X^L - 1)$ en utilisant la transformée de Fourier discrète.

Définissons d'abord $\mathrm{GF}(p)[X]/(X^L - 1)$: c'est l'ensemble des polynômes ayant

- des coefficients entiers compris entre 0 et $p - 1$,
- un degré strictement inférieur à L, et
- les règles de multiplication suivantes : le résultat d'un produit de deux éléments est réduit modulo $X^L - 1$ quand son degré dépasse L, et les coefficients sont ensuite réduits modulo p.

Nous ne pouvons pas choisir L et p arbitrairement : pour pouvoir exploiter l'algorithme rapide, L doit s'écrire sous la forme $L = 2^m$, p doit être premier, et doit s'écrire sous la forme $p = \nu L + 1$, où ν est entier. Sous ces conditions contraignantes, la multiplication peut s'effectuer de manière simple, en un nombre d'opérations variant en $L \log L$. Nous décrivons ci-dessous l'algorithme optimisé.

	l	Valeurs possibles de s	Découvert par
M12	127	1, 7, 15, 30, 63	Zierler (1969)
M13	521	32, 48, 158, 168	"
M14	607	105, 147, 273	"
M15	1 279	216, 418	"
M16	2 203	—	"
M17	2 281	715, 915, 1 029	"
M18	3 217	67, 576	"
M19	4 253	—	"
M20	4 423	271, 369, 370, 649, 1 393, 1 419, 2 098	"
M21	9 689	84, 471, 1 836, 2 444, 4 187	"
M22	9 941	—	Heringa *et al.* (1992)
M23	11 213	—	"
M24	19 937	881, 7 083, 9 842	"
M25	21 701	—	"
M26	23 209	1 530, 6 619, 9 739	"
M27	44 497	8 575, 21 034	"
M28	86 243	—	"
M29	110 503	25 230, 53 719	"
M30	132 049	7 000, 33 912, 41 469, 52 549, 54 454	"
M31	216 091	—	"
M32	**756 839**	**215 747**, 267 428, 279 695	Brent (2000)
M33	859 433	170 340, 288 477	Brent / Kumada (2000)
M34	1 257 787	—	Kumada *et al.* (2000)
M35	1 398 269	—	"
M36	2 976 221	—	"
M37	3 021 377	361 604, 1 010 202	Brent *et al.* (2000)
M38	**6 972 593**	**3 037 958**	Brent *et al.* (2002)
M39	13 466 917	—	—
M40[1]	20 996 011	—	—
M41[1]	24 036 583	8 412 642, 8 785 528	Brent *et al.* (2007)
M42[1]	25 964 951	880 890, 4 627 670, 4 830 131, 6 383 880	Brent *et al.* (2007)
M43[1]	30 402 457	2 162 059	Brent *et al.* (2007)
M44[1]	32 582 657	5 110 722, 5 552 421, 7 545 455	Brent *et al.* (2008)
M45[1]	37 156 667	—	—
M46[1]	42 643 801	—[2]	—
M47[1]	43 112 609	3 569 337, 4 463 337, 17 212 521, 21 078 848[2]	Brent *et al.* (2008-2009)

Figure 8.1 : *Liste de trinômes irréductibles, de la forme $X^l + X^s + 1$, et à coefficients dans GF(2). Les derniers polynômes de cette liste sont les plus grands polynômes irréductibles connus actuellement. Les lignes en gras correspondent aux valeurs que nous avons choisi pour des chaînes d'entrée de 400 000 et 4 000 000 bits.* ***Notes*** *: La recherche de trinômes irréductibles est intimement liée à la découverte de nouveaux nombres de Mersenne [75], qui sont les nombres premiers de la forme $2^l - 1$. En effet, des algorithmes ont été développés pour tester « facilement » l'irréductibilité de trinômes dans le cas où leur degré l est tel que $2^l - 1$ est un nombre de Mersenne. (1) Seuls 47 nombres de Mersenne sont connus à l'heure actuelle, ce qui se comprend quand on s'imagine que M47 a presque 13 millions de décimales, et que sa primalité n'est pas tout à fait triviale à tester. Les nombres de Mersenne sont traditionnellement numérotés de manière croissante, et la numérotation des nombres de Mersenne M40 à M47 n'est pas encore fixée, puisque la primalité de tous les candidats intermédiaires n'a pas encore été testée. (2) De la même façon, tous les trinômes n'ont pas encore été testés pour ces degrés, et il n'est donc pas exclu que quelques irréductibles résistent encore.*

Mise en forme des polynômes de $\mathrm{GF}(2^l)$**.** Pour utiliser la multiplication rapide, il faut tout d'abord transformer nos polynômes de $\mathrm{GF}(2^l)$ en éléments de $\mathrm{GF}(p)[X]/(X^L-1)$. Or, on peut montrer facilement que le produit (**usuel**) de deux polynômes de $\mathrm{GF}(2^l)$ ne peut avoir de coefficients supérieurs à $l-1$, ni un degré supérieur à $2l-2$. Ainsi, si l'on s'assure que $p \geqslant l$ et que $L \geqslant 2l$, les réductions modulo p et X^L-1 n'interviendront jamais, c'est-à-dire que $\mathrm{GF}(2^l)$ est alors un sous-ensemble de $\mathrm{GF}(p)[X]/(X^L-1)$. Il n'y a alors aucune mise en forme à réaliser : un polynôme de $\mathrm{GF}(2^l)$ s'écrit alors sous la même forme dans $\mathrm{GF}(p)[X]/(X^L-1)$, et le produit usuel de deux polynômes est le même que le produit « réduit ».

Il nous faut donc ensuite trouver des couples (L,p) tels que toutes les conditions — p premier, $L=2^m$, $p=\nu L+1$, $p \geqslant l$ et $L \geqslant 2l$ — soient respectées. Le tableau 8.2 donne quelques valeurs de ce couple utilisables pour des valeurs de l correspondant à notre cas ($\mathrm{l} > 400\,000$ ou $4\,000\,000$, en fonction du nombre de blocs utilisés).

Algorithme de hachage. Nous utilisons enfin la méthode numérique simplifiée pour calculer le produit des deux polynômes :

- Alice et Bob disposent d'une chaîne à amplifier de longueur l, *l'entrée*, qui provient de la chaîne réconciliée. Ils définissent par ailleurs la fonction de hachage en générant un *défi*, c'est-à-dire une autre chaîne, aléatoire, de longueur l.
- Ils calculent les transformées de Fourier discrètes (ou NTT, pour *Number Theoretic Transform*) de l'entrée E et du défi D, en choisissant des paramètres L et p adaptés. Ces NTT, $\mathcal{F}(E)$ et $\mathcal{F}(D)$, peuvent être calculées en un nombre d'opérations de l'ordre de $L \log L$ en utilisant l'algorithme de NTT rapide (c'est tout l'intérêt du choix particulier de L et p).
- En multipliant terme à terme les deux résultats des NTT, ils obtiennent le produit $\mathcal{F}(E) \cdot \mathcal{F}(D) = \mathcal{F}(E \times D)$. Ils effectuent ensuite la NTT inverse, et obtiennent finalement le produit $E \times D$.
- La suite s'effectue comme dans la section précédente : le polynôme ainsi obtenu est réduit par le polynôme irréductible de longueur l du tableau 8.2, et ses coefficients sont réduits modulo 2. Alice et Bob obtiennent ainsi une chaîne binaire de longueur l, dont ils prélèvent finalement z bits pour obtenir une clé secrète.

Par exemple, pour une taille de bloc de 400 000 bits, nous choisissons un trinôme tel que $l = 756\,839$, ce qui nous impose de choisir $L = 2^{21} = 2\,097\,152 > 2l$ et $p = 23\,068\,673 > l$.

Pour une taille de bloc de 4 000 000 bits, nous choisissons $l = 6\,972\,593$ et donc $L = 2^{24} = 16\,777\,216$ et $p = 167\,772\,161$.

8.2.5 Algorithme 2 : Multiplication directe dans $\mathrm{GF}(2^l)$ avec la NTT

L'algorithme précédent, quoique plus performant, ne semble pas optimal. En effet, non seulement il oblige à amplifier plus de bits que nécessaire, mais il nécessite aussi de traiter $L \times \log_2 p$ bits par la NTT. Pour des tailles de blocs de 400 000 et 4 000 000, ceci revient à manipuler une quantité de données de respectivement 50 mégabits et 450 mégabits !

Gilles Van Assche a proposé dans [61] une nouvelle famille de fonctions de hachage, qui utilise directement la NTT pour générer l'empreinte :

- Alice et Bob définissent un élément de la famille de fonctions, en choisissant aléatoirement une chaîne v de L nombres entiers compris entre 0 et $p-1$.
- Ils convertissent leur chaîne à amplifier de n bits en une chaîne d'entiers x, en regroupant les bits par paquets de $\lfloor \log_2 p \rfloor$. Ils obtiennent ainsi une chaîne d'entiers tous plus petits que p, qu'ils complètent par des 0 pour qu'elle contienne L éléments. Notons que, pour que les n bits puissent être convertis dans la chaîne entière, il faut que $L\lfloor \log_2 p \rfloor \geqslant n$.
- Ils appliquent la NTT inverse au produit des deux chaînes $v \cdot x$, et obtiennent une empreinte sous forme d'une chaîne d'entiers. Il ne leur reste alors plus qu'à extraire $\lfloor \log_2 p \rfloor$ bits de chaque entier jusqu'à atteindre les z bits nécessaires pour obtenir une clé secrète.

Contrairement à la famille précédente, cette famille de fonction de hachage n'est pas universelle : son facteur d'universalité s'exprime comme $\varepsilon = \left(\frac{p-1}{p}\right)^z \approx 1 + \frac{z}{p}$. Pour l'utiliser, nous devons donc choisir avec soin les valeurs de p et L. La condition $L\lfloor \log_2 p \rfloor \geqslant n$ doit de toute façon être respectée ; par ailleurs, le temps de calcul croît avec L, et la famille se rapproche de l'universalité quand p est grand. Nous avons donc tout à intérêt à choisir L petit et p grand, tout en gardant à l'esprit que, compte tenu du codage des entiers dans le système d'exploitation sur 32 bits, p ne peut dépasser 2^{32}.

Pour une taille de bloc de 400 000 bits, nous choisissons $L = 2^{14}$ et $p = 33\,832\,961$, de façon à ce que $L\lfloor \log_2 p \rfloor = 409\,600$. Avec ces paramètres et une clé finale de 4 000 bits, nous obtenons un paramètre d'universalité $\varepsilon \approx 1{,}0001$, c'est-à-dire un paramètre de sécurité $s_{\max} \approx 13$.

Pour une taille de bloc de 4 000 000 bits, nous choisissons $L = 2^{17}$ et $p = 2\,148\,794\,369$, de façon à ce que $L\lfloor \log_2 p \rfloor = 4\,063\,232$. Avec ces paramètres et une clé finale de 40 000 bits, nous obtenons un paramètre d'universalité $\varepsilon \approx 1{,}00002$, c'est-à-dire un paramètre de sécurité $s_{\max} \approx 15$.

8.2.6 Composition des algorithmes 1 et 2

L'algorithme 1 est universel, mais assez lent ; l'algorithme 2 n'autorise que des paramètres de sécurité faibles, mais est très rapide. Il est possible de définir un algorithme combiné qui exploite la rapidité de l'algorithme 2 tout en assurant un paramètre de sécurité suffisamment fort :

- Les n bits d'origine sont hachés par l'algorithme 2, pour obtenir une chaîne de i bits, où i correspond à une valeur du tableau 8.1 pour lequel nous connaissons un polynôme irréductible.
- L'algorithme 1 est ensuite utilisé pour amplifier les i bits jusqu'à obtenir une clé secrète de la longueur z voulue.

On montre [77] que l'universalité d'une telle famille, composition de deux familles de paramètres ε_a et ε_b, s'exprime sous la forme :

$$\varepsilon_{\text{tot}} = \varepsilon_a 2^{z-i} + \varepsilon_b \tag{8.3}$$

Si la famille b est universelle ($\varepsilon_b = 1$), il suffit de choisir un nombre de bits intermédiaire $i > z + s$ pour obtenir un paramètre de sécurité s aussi grand que souhaité.

Dans notre cas, si nous considérons une clé finale de 4 000 bits, nous pouvons par exemple choisir $i = 4\,423$ ou $9\,689$, qui permettent d'obtenir des paramètres de sécurité globaux respectifs d'environ $s = 400$ et $5\,500$. On voit que le choix de i doit être effectué de manière dynamique en fonction des paramètres du canal, de façon à optimiser pour chaque clé la vitesse et la sécurité de l'amplification de confidentialité.

8.3 Vérification de la clé secrète

Quelle que soit la distance à la limite de Shannon que nous choisissons pour le taux des codes LDPC, il existe toujours une probabilité que ceux-ci laissent des erreurs, et une probabilité encore plus petite (mais existante) que ce nombre d'erreurs soit supérieur à ce que le code BCH peut corriger. Dans ce dernier cas, les chaînes d'Alice et Bob sont différentes avant l'amplification, et rien ne nous permet de nous en rendre compte.

Or, dans un contexte réaliste, il n'est pas envisageable d'obtenir des chaînes finales différentes (des « clés ratées », en quelque sorte) : au delà de l'impossiblité de crypter un message, le fait de fournir deux clés différentes à une couche supérieure de gestion de clés risque très fortement de déstabiliser le réseau complet. Il est donc nécessaire de s'assurer contre ce type d'erreurs.

Pour détecter les erreurs résiduelles, nous avons implémenté un algorithme de vérification, qui utilise les propriétés d'universalité des fonctions de hachage utilisées lors de l'amplification de confidentialité. Considérons les deux chaînes mal réconciliées B_1 et B_2, qui sont donc légèrement différentes. Ces chaînes sont amplifiées en $f(B_1)$ et $f(B_2)$, et nous extrayons de ces deux chaînes les q premiers bits, pour obtenir les échantillons $f_q(B_1)$ et $f_q(B_2)$.

L'idée de la vérification est de considérer que les deux échantillons de clés $f_q(B_1)$ et $f_q(B_2)$ sont le résultat d'un hachage « plus fin » des chaînes B_1 et B_2 par le même algorithme. L'ε-universalité de l'amplification de confidentialité nous assure alors que la probabilité d'une identité « malheureuse » des deux échantillons de clés s'exprime :

$$\Pr\Big[f_q(B_1) = f_q(B_2) \;\Big|\; B_1 \neq B_2 \Big] = \frac{\varepsilon}{2^q} \approx 2^{-q} \tag{8.4}$$

Pour vérifier la bonne correction de la clé, avec une probabilité de faux positif de $2^{-s_{\text{corr}}}$, il nous suffit donc de prélever un échantillon de longueur s_{corr} de la clé finale, et de la comparer entre Alice et Bob. En pratique, nous prélevons 200 bits de chaque clé, ce qui nous donne un risque de faux positif d'environ 10^{-61}, c'est-à-dire virtuellement nul.

m	L	ν	p	$L\lfloor\log_2 p\rfloor$
$14^{(1)}$	16 384	4	65 537	262144
$14^{(2)}$	**16 384**	**2 065**	**33 832 961**	**409 600**
$15^{(1)}$	32 768	2	65 537	524 288
$15^{(2)}$	32 768	1 047	34 308 097	819 200
$16^{(1)}$	65 536	1	65 537	1 048 576
$16^{(2)}$	65 536	12	786 433	1 245 184
$16^{(2)}$	65 536	540	35 389 441	1 638 400
$16^{(2)}$	65 536	32 787	2 148 728 833	2 031 616
$17^{(1)}$	131 072	6	786 433	2 490 368
$17^{(2)}$	131 072	44	5 767 169	2 883 584
$17^{(2)}$	131 072	270	35 389 441	3 276 800
$17^{(2)}$	131 072	2 054	269 221 889	3 670 016
$17^{(2)}$	**131 072**	**16 394**	**2 148 794 369**	**4 063 232**
$18^{(1)}$	262 144	3	786 433	4 980 736
$19^{(1)}$	524 288	11	5 767 169	11 534 336
$20^{(1)}$	1 048 576	7	7 340 033	23 068 672
$21^{(1)}$	**2 097 152**	**11**	**23 068 673**	**50 331 648**
$22^{(1)}$	4 194 304	25	104 857 601	109 051 904
$23^{(1)}$	8 388 608	20	167 772 161	226 492 416
$24^{(1)}$	**16 777 216**	**10**	**167 772 161**	**452 984 832**

Figure 8.2 : *Liste de paramètres compatibles avec l'emploi de l'algorithme rapide, fondé sur la transformée de Fourier rapide. Les valeurs de m marquées d'un $^{(1)}$ correspondent à une valeur de p la plus petite possible, de façon à optimiser la vitesse de l'algorithme 1. Celles marquées de $^{(2)}$ correspondent à des valeurs de p plus grandes, spécifiquement choisies pour des tailles de chaînes n multiples de 400 000 de façon à optimiser l'universalité de l'algorithme 2. Les lignes en gras correspondent aux valeurs que nous choisissons pour des tailles de message de 400 000 et 4 000 000 bits.*

9 Performances de l'extraction

> Results ? Why, man, I have gotten a lot of results ! If I find ten thousand ways something won't work, I haven't failed. I am not discouraged, because every wrong attempt discarded is often a step forward...
>
> Thomas A. Edison

Dans ce chapitre, nous exposons dans un premier temps les aspects logiciels de l'extraction de clé, et les enjeux de vitesse d'extraction. Nous présentons ensuite les résultats expérimentaux que nous avons obtenus lors d'un fonctionnement complet du système, à la fois dans des conditions optimales de laboratoire, puis lors de tests sur le terrain effectués à Vienne en octobre 2008.

9.1 Logiciel d'extraction de la clé

9.1.1 Structure du programme

Le programme d'extraction de la clé, appelé QuantCOM (de *quantum communication*) est conçu comme une couche supérieure du programme de gestion de la transmission. Souvenons-nous que, lors de la compilation du sous-programme de transmission, une librairie `libqkd` a été générée, qui fournit au programme d'extraction une dizaine de fonctions simples pour réaliser les opérations liées à l'optique. Le nom de ces fonctions est transparent : `homodyne` (lance la mesure de Bob), `modulation` (lance la modulation d'Alice), `request_bloc`, `stop`, `hardware_check`, `redetect_datagram`, etc. Le programme d'extraction utilise donc cette librairie pour gérer le fonctionnement du système physique.

Par ailleurs, le programme d'extraction est conçu autour de deux pôles : le premier, `Node`, gère les communications classiques nécessaires pour l'extraction ; le deuxième, `CSKDP` (pour *Continuous Secret Key Distribution Protocol*), gère le bon fonctionnement des algorithmes d'extraction et des flux de données (à l'instar de `pool` dans l'autre sous-programme).

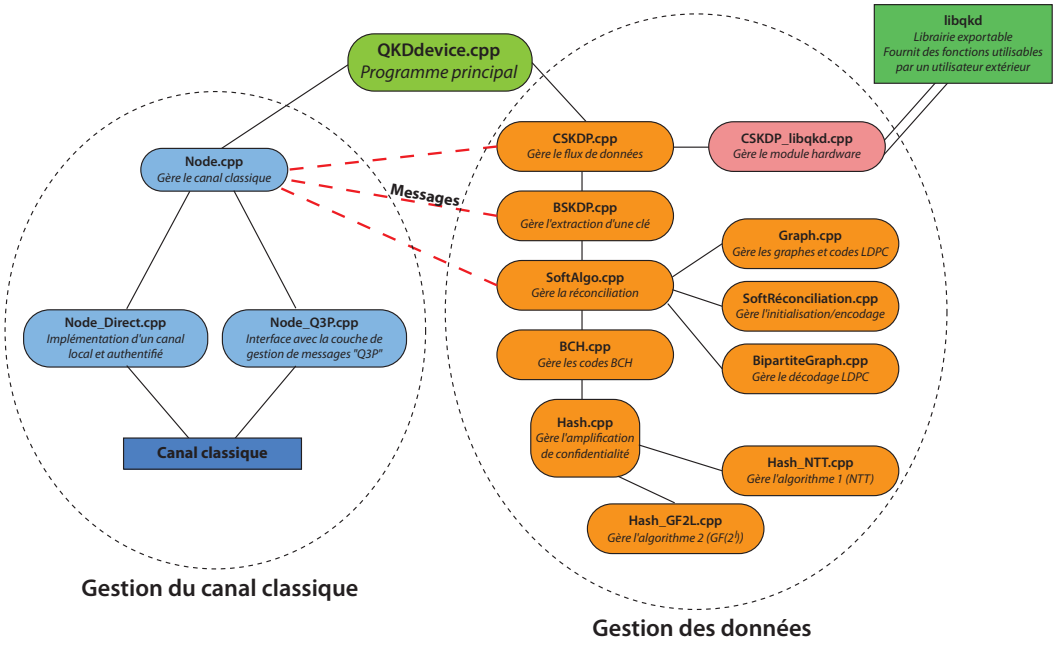


Figure 9.1 : *Vision schématique de l'architecture du programme de génération des clés secrètes. Elle peut se décomposer en deux parties : à gauche, les modules assurant la gestion du canal classique ; à droite, les modules assurant l'extraction de la clé secrète.*

La figure 9.1 présente la structure algorithmique du programme :

- `QKDdevice` est le fichier maître, qui génère l'exécutable général. Il autorise un certain nombre d'options, parmi lesquelles le choix de l'interface de communications classiques, les niveaux de messages d'erreur et d'information, le niveau de sécurité choisi (attaques individuelles/collectives), le fonctionnement en continu (ou non), etc. Une fois appelé, il ouvre le canal classique via `Node`.
- `Node` est le garant du bon fonctionnement des communications classiques. Nous présentons dans la section 9.1.2 le processus de synchronisation et de communication entre Alice et Bob. Deux choix sont possibles : une communication directe quand Alice et Bob sont dans un réseau local (éventuellement virtuel), ou une communication *via* une couche supérieure de gestion de messages, comme l'interface Q3P du projet SECOQC, que nous présentons plus loin.
- Une fois que la communication entre Alice et Bob est établie et qu'ils se sont authentifiés, une instance de `CSKDP` est appelée. La transmission quantique n'est alors pas encore initialisée, et `CSKDP` fait donc appel à l'interface `CSKDP_libqkd` pour démarrer les systèmes optiques d'Alice et Bob, ce qui est fait en appelant les fonctions adéquates dans la librairie `libqkd`.
- Quand la couche inférieure renvoie un signal positif (c'est-à-dire qu'elle est correctement démarrée), Alice et Bob appellent de nouveau la librairie `libqkd` pour récupérer des données continues issues de la transmission quantique. Ils doivent en particulier se mettre d'accord sur la liste des blocs de données à utiliser, ce qui donne lieu à une communication classique, qui passe donc par `Node`. Une fois ces blocs de données récupérés, Alice et Bob partagent les données corrélées dont

ils ont besoin pour débuter l'extraction de la clé secrète.

- `BSKDP` (*Block Secret Key Distribution Protocol*) est en charge de réaliser l'extraction de ce bloc donné. Dans un premier temps, il sacrifie 50 % des données, qu'il envoie à `dataset` de façon à déterminer les paramètres de la transmission pour ce bloc, et donc la quantité de bits secrets pouvant en être extraits. Si cette quantité est positive, il appelle la classe de gestion de la réconciliation, `SoftAlgo`.
- La réconciliation du bloc se fait en plusieurs étapes : dans un premier temps, les graphes des codes LDPC doivent être générés par `Graph`, en créant les structures de données assurant les connectivités entre nœuds de variable et de parité. Ensuite, le protocole proprement dit est mené par `SoftReconciliation` pour l'initialisation et l'encodage, et par `BipartiteGraph` pour le décodage.
- Après le décodage LDPC, `SoftAlgo` appelle l'algorithme `BCH`, qui corrige les dernières erreurs, puis initialise l'amplification de confidentialité en appelant `Hash`. Les deux algorithmes de hachage sont respectivement implémentés dans `HashGF2L` et `HashNTT`.
- Alice et Bob partagent alors une clé secrète, qu'ils stockent dans un module de gestion de clés appelé `KeyBroker`, géré en interne ou par la couche supérieure.

9.1.2 Gestion des communications classiques

Au cours du fonctionnement du programme, Alice et Bob ont besoin d'échanger des informations pour mener à bien l'extraction de la clé. Ces communications classiques doivent être très bien organisées, au risque de voir Alice et Bob se retrouver dans une situation bloquante : par exemple, tous deux en attente d'un message de l'autre partie. La figure 9.2 présente le schéma de ces communications, simplifié des signaux `OK_status` servant uniquement à passer la main à l'autre partie.

La plupart du temps, Alice et Bob effectuent leurs opérations de façon séquentielle, à la façon d'une discussion. Cependant, certaines opérations, telles que l'amplification de confidentialité, peuvent être effectuées en parallèle, ce qui permet de gagner du temps. Il faut simplement garantir que l'interlocuteur chargé de reprendre la discussion après ces étapes parallèles est bien défini. Dans notre cas, Alice a systématiquement la responsabilité d'initier les échanges.

Au final, environ 8 mégaoctets et une vingtaine de messages doivent être échangés par le canal classique pour la génération d'une clé secrète à partir de 100 blocs de données, ce qui correspond à un débit raisonnable sur le réseau classique : environ 5 Mbit/s au maximum (une connexion Ethernet standard transmet environ 100 Mbit/s).

Il est possible de réduire légèrement cette charge en codant plus astucieusement les deux messages les plus importants, la liste des phases et l'échantillon des données. Par exemple, puisque les cartes d'acquisition n'acquièrent les données que sur 12 bits, il est inutile de coder l'échantillon sur 16 bits. De la même façon, la liste des phases peut se réduire au choix de la quadrature de Bob, soit un bit par impulsion. Nous pouvons ainsi diminuer la charge du canal d'un facteur 2. Cette compression fait néanmoins appel à des types de données non-standard (on parle de multiplexage), ce qui complexifie un peu le programme. Dans notre cas pratique, le débit du canal classique est suffisant et nous n'avons donc pas implémenté ces structures dans le programme.

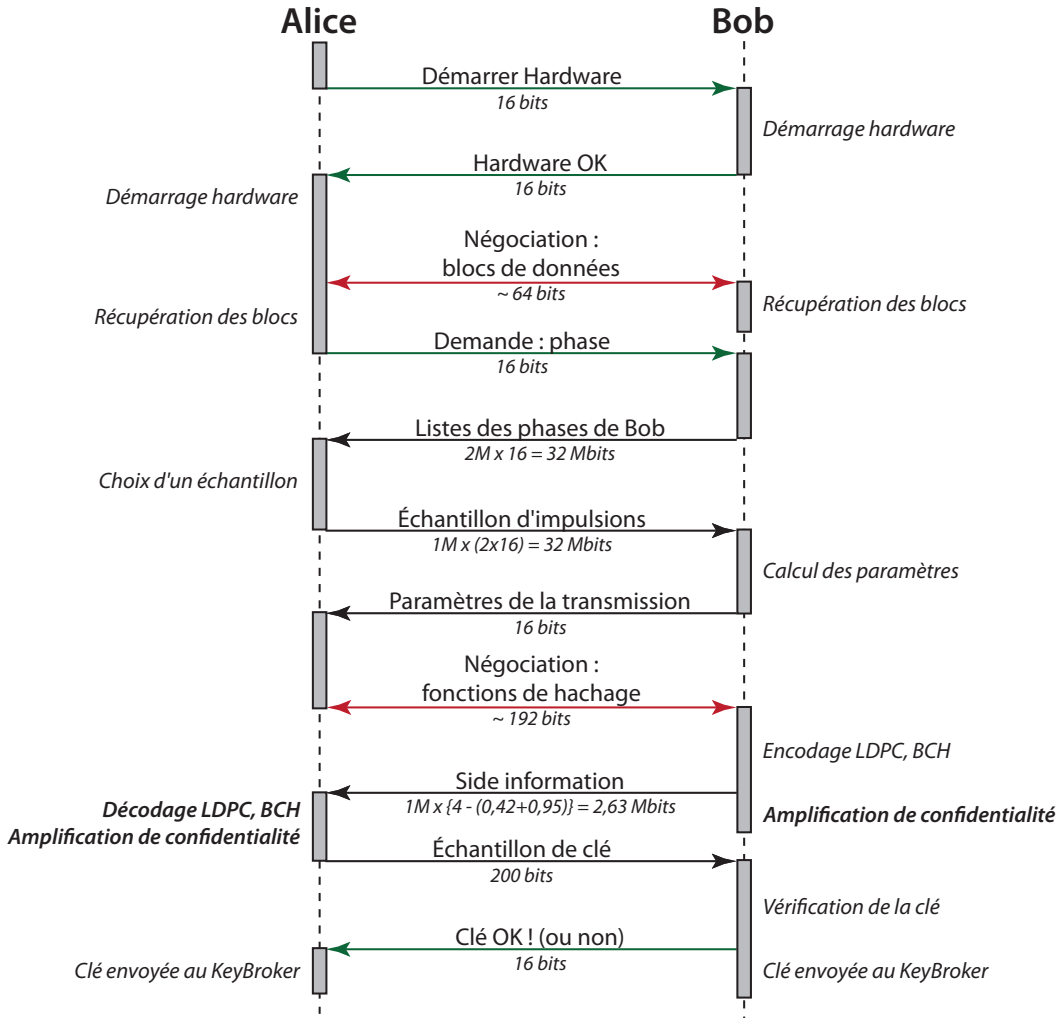


Figure 9.2 : *Schéma de synchronisation des communications classiques. Le nombre de bits échangés à chaque message correspond à une clé générée à partir de 100 blocs de données (2 millions d'impulsions).*

9.2 Temps d'extraction d'une clé

Pour chaque extraction de clé, Alice et Bob utilisent 2 millions d'impulsions utiles, correspondant à 100 blocs de données générées lors de la transmission, et ces 100 blocs sont émis en 10 secondes. Idéalement, il faudrait donc que l'extraction d'une clé se fasse en moins de temps, de façon à pouvoir traiter tous les blocs de données en temps réel. Or, les algorithmes de réconciliation et d'amplification de confidentialité nécessitent un temps de calcul non-négligeable ; nous détaillons dans cette partie les causes de ce temps de calcul et les optimisations permettant de le réduire.

9.2.1 Réconciliation

Du fait de la taille de la matrice de parités que nous utilisons, le décodage des codes LDPC nécessite de mettre à jour un nombre important de variables, et nécessite donc

une puissance de calcul assez importante. En particulier, l'algorithme Message Passing comporte des appels répétés à la fonction $\varphi = -\ln(\tanh(x/2))$: lors du décodage d'un bloc de 200 000 impulsions, la fonction φ est appelée plus de 100 millions de fois ! Cet algorithme a donc deux particularités :

- Il met en jeu de nombreuses variables (le nombre de liens parité-variable est typiquement de quelques millions), auxquelles nous devons accéder très fréquemment. Or, le temps d'accès à une variable dans la mémoire vive est de l'ordre d'une centaine de cycles de processeur, pendant lequel ce dernier est inactif.
- Il inclut un calcul de fonction non-triviale, qui est répété des millions de fois, et qui finit par prendre un temps cumulé important.

Puisque quelques lignes de code consomment la majorité du temps de décodage, l'optimisation de l'algorithme doit se concentrer sur celles-ci. Jérôme Lodewyck a réalisé un travail important d'optimisation du programme au cours de sa thèse [5], dont nous résumons ci-dessous les principaux aspects :

- La structure des données représentant le graphe de Tanner des codes a été choisie de façon à limiter les accès mémoire, en utilisant des pointeurs de façon récursive.
- La fonction φ a été tabulée et mise en mémoire, ce qui est plus rapide que de réaliser le calcul direct.
- Les structures de boucle redondantes ont été traquées et éliminées, avec l'aide d'un logiciel de profilage [60].
- Enfin, les calculs sur les réels ont été transformées en calculs sur des nombres à virgule fixe, plus rapides.

Pour ces lignes centrales dans l'algorithme, il n'apparaît alors plus d'optimisation aisée permettant de gagner un facteur significatif sur la vitesse d'exécution. La réconciliation de blocs de données de 200 000 impulsions dure typiquement 2,4 secondes sur un processeur Intel Core 2 Quad Q6600. Parmi ces 2,4 s, environ 800 ms sont consacrées au calcul de φ (soit 8 ns par appel), 500 ms au chargement des codes et à leur initialisation, et une seconde aux autres opérations de l'algorithme.

9.2.2 Amplification de confidentialité

La vitesse de l'amplification de confidentialité ne peut être négligée devant la vitesse de la réconciliation. Comme nous l'avons décrit dans le chapitre 8, l'algorithme est composé de deux hachages successifs, le premier étant rapide et le deuxième plus lent (mais universel). Or, la vitesse du second algorithme dépend principalement du degré du trinôme irréductible utilisé pour la multiplication rapide (figure 8.1), et le choix du degré dépend du nombre de bits final de la clé secrète. Des phénomènes de seuil apparaissent alors, qui peuvent dégrader sensiblement la vitesse d'amplification.

Nombre de bits dans la clé finale	Degré du trinôme choisi	Temps d'exécution
100 000	110 503	1 seconde
120 000	132 049	2 secondes
140 000	756 839	8,5 secondes
700 000	756 839	8,5 secondes

Pour certains paramètres, la durée de l'amplification de confidentialité peut devenir importante devant la durée de la réconciliation. Il peut alors être intéressant de faire

varier le nombre de blocs de données utilisés par clé de façon à ce que la taille de la clé secrète soit proche d'un trinôme connu, et d'optimiser ainsi la vitesse de l'amplification de confidentialité.

9.2.3 Parallélisation des extractions

Les programmes informatiques ont longtemps été conçus de façon linéaire : il n'existe alors qu'un fil d'exécution, et les instructions s'enchaînent de façon séquentielle, de sorte que l'instruction n ne peut être réalisée que si l'instruction $n-1$ est terminée. Ce type de programmation a l'avantage d'être relativement facile à concevoir pour le programmeur, puisque le programme suit les instructions ligne par ligne. Il n'est en revanche pas optimal : par exemple, lors d'un accès mémoire, le processeur reste inactif pendant plusieurs dizaines de cycles, qui pourraient être utilisés pour exécuter une instruction indépendante.

Depuis assez longtemps, une alternative à cette programmation linéaire a été mise en place dans les langages courants, appelée *programmation multi-fils*. Plusieurs fils d'exécution sont créés, et les instructions de chaque fil se déroulent de façon indépendante. Ainsi, pendant que l'un des fils doit mettre en mémoire une quantité importante de données, le processeur peut continuer l'exécution de l'autre fil. Ce type de programmation permet de réduire le nombre de cycles perdus par le programme.

Dans les dernières années, une nouvelle architecture de processeurs a été développée, appelée multi-cœurs, dans laquelle le processeur est constitué de plusieurs sous-processeurs pouvant travailler en parallèle. Nous avons intégré l'un de ces processeurs dans notre système : il s'agit d'un processeur Intel de type Core 2 Quad, qui contient 4 cœurs indépendants. Ces processeurs sont donc particulièrement adaptés à la programmation multi-fils, puisqu'un fil n'a pas besoin d'attendre que les autres fils aient terminé leurs instructions pour reprendre son exécution. À l'inverse, si un programme linéaire est exécuté sur un processeur multi-cœur, un seul des cœurs est mis à contribution, et les autres sont inactifs : une partie de la puissance est perdue.

Mise en place de la structure multi-fils

Pour que les étapes les plus chronophages puissent être exécutées en parallèle, notre programme global a été divisé en plusieurs fils. La figure 9.3 résume cette parallélisation. Le fil principal est celui lancé par l'utilisateur, c'est-à-dire le fil de gestion des communications et de l'extraction (`Node` et `CSKDP`). Dans un premier temps, un fil secondaire est créé au démarrage du programme, à travers la librairie `libqkd` : il s'agit du sous-programme de gestion de la transmission quantique. Par la suite, $n-1$ fils d'extraction de clé sont ouverts par `CSKDP`, de façon à ce que le programme effectue n réconciliations en parallèle. A un moment donné, nous pouvons donc avoir jusque $n+1$ fils fonctionnant en parallèle.

La programmation multi-cœur est donc très puissante mais délicate à réaliser proprement. En effet, les fils ne sont jamais complètement indépendants, et partagent un certain nombre de variables. Il faut alors s'assurer que deux fils ne puissent pas modifier une variable partagée en même temps, pour éviter les conflits. L'exemple le plus caractéristique dans notre système est celui des blocs de données. D'un côté, le programme de gestion de la transmission génère des blocs de données régulièrement, qu'il

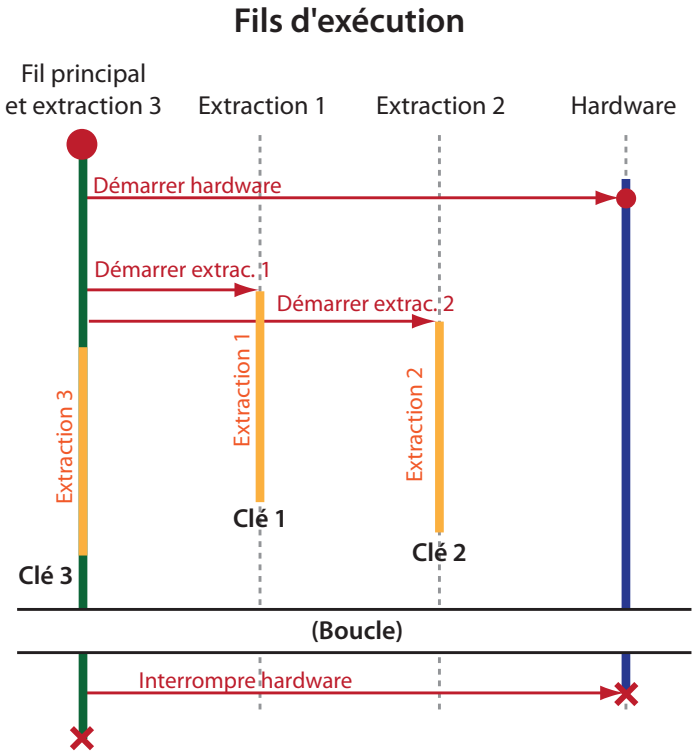


Figure 9.3 : *Schéma de la structure multi-fils, pour 3 extractions de clé. Après avoir démarré le fil hardware, le fil principal appelle un fil d'extraction de clé, puis un deuxième ; la troisième extraction est réalisée par le fil principal. Ainsi, trois clés sont produites en parallèle. Le fil principal a un contrôle total sur les autres fils.*

---

traite tout d'abord en interne, puis place dans une « banque de blocs ». De l'autre, le programme principal récupère ces blocs dans la banque quand il en a besoin, de façon à démarrer une extraction de clé. Or, la banque ne peut pas être modifiée en même temps par les deux parties, ce qui résulterait de façon à peu près certaine en une erreur de segmentation.

Il est donc nécessaire d'établir un protocole d'accès aux données partagées, qui utilise des structures d'exclusion mutuelle (ou `mutex`). Ces structures, quand elles sont déclenchées par un fil d'exécution, empêchent tous les autres fils d'accéder à la variable concernée. Ceux-ci doivent alors attendre que la variable soit libérée pour y accéder de nouveau. Qui plus est, les `mutex` doivent être placés de façon pertinente, pour qu'un fil n'empêche pas le fonctionnement des autres ; en particulier, quand le fil principal accède aux données, il ne doit absolument pas bloquer le fonctionnement du fil de gestion de la transmission, car ce dernier doit envoyer des données aux cartes toutes les 100 millisecondes. Des structures de buffer ont donc été mises en place pour éviter de bloquer les fils, afin de nous assurer que le système fonctionne correctement.

Conséquences sur la vitesse

La mise en parallèle des extractions de clé a permis d'accélérer la vitesse de traitement des données de façon significative. Le tableau ci-dessous donne les temps totaux

d'exécution de n réconciliations de clé (pour des paramètres typiques) lorsque n fils d'extraction fonctionnent en parallèle. Ces temps d'exécution sont calculés grâce aux structures `clock` et `time()`, qui sont un standard C++.

Nombre de fils d'extraction	Temps d'exécution	Temps d'exécution par clé
1	24 secondes	24 secondes
2	30 secondes	15 secondes
3	34 secondes	11,3 secondes
4	40 secondes	10 secondes
5	49 secondes	9,7 secondes
10	94 secondes	9,4 secondes

Puisque notre processeur ne comporte que 4 cœurs, il ne peut exécuter plus de 4 fils de façon complètement autonome. Au delà, deux fils seront exécutés sur le même cœur. Assez logiquement, ce résultat se retrouve dans le temps d'exécution du programme : jusque 3 fils d'exécution (plus le fil hardware), le temps d'extraction par clé diminue beaucoup, c'est-à-dire que la parallélisation est efficace. Au delà, non seulement le temps devient presque linéaire du nombre de fils, mais le fait que certains fils d'exécution soient effectués sur le même cœur que le fil hardware rend ce dernier très instable. Il n'est en pratique pas possible de faire fonctionner plus de trois extractions parallèles.

9.3 Résultats expérimentaux

9.3.1 Distance limite de transmission et taux de clé optimaux

Vitesse totale de l'extraction

Nous nous plaçons dans les conditions optimisées : trois extractions en parallèle, en utilisant l'algorithme RMP de réconciliation et en considérant une taille finale de clé secrète typique d'environ 100 000 bits finaux pour 2 millions d'impulsions réconciliées. Nous obtenons alors un temps d'extraction par clé d'environ 17 secondes, pour un temps de génération des impulsions de 10 secondes. Par conséquent, le taux de génération de clé n'est pas limité par la fréquence de répétion optique mais par la vitesse de traitement des données. Celle-ci est de l'ordre de 130 000 impulsions traitées par seconde, à comparer avec la fréquence de répétition optique utile : 200 000 impulsions modulées par seconde.

Distance limite de transmission

Comme nous l'avons introduit dans le chapitre 7, l'efficacité limitée du processus de réconciliation ne diminue pas le taux secret d'un facteur donné, ce qui n'aurait qu'un effet proportionnel. En effet, pour des longues distances, les courbes de I_{AB} et I_{BE} deviennent tangentes ; et puisque β ne joue que sur I_{AB}, les courbes βI_{AB} et I_{BE} vont se croiser à un certain point, c'est-à-dire que ΔI va devenir négatif, quelle que soit l'efficacité de réconciliation différente de 1. Ce point correspond donc à une

distance limite de transmission, au delà de laquelle Alice et Bob auront toujours moins d'information que l'espion.

Nous représentons dans la figure 9.4 l'évolution du taux secret, en prenant successivement tous les effets négatifs en compte : l'excès de bruit, l'efficacité et la vitesse de réconciliation.

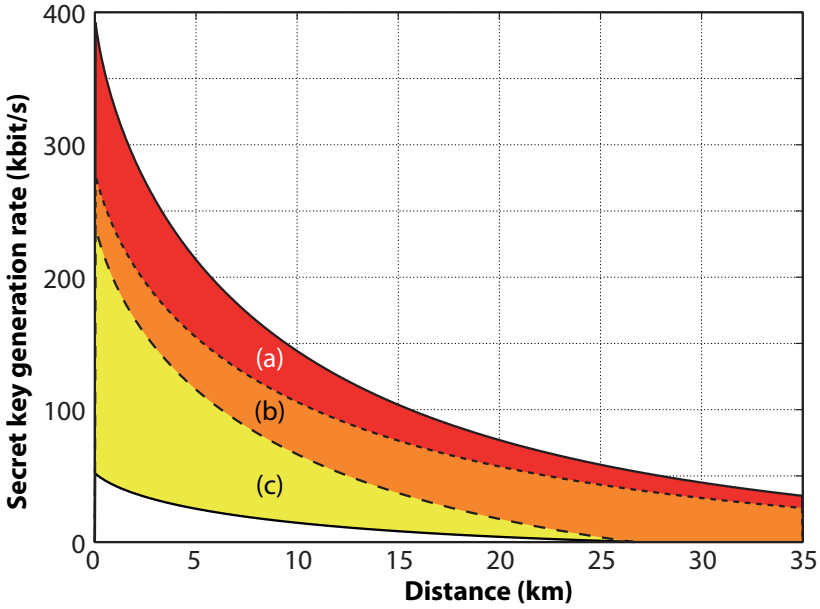


Figure 9.4 : *Résumé des différentes causes pratiques de la limitation du taux de clé secrète. La courbe supérieure, en trait plein, représente le taux maximal théoriquement accessible par le système, compte tenu de la fréquence optique. Les courbes inférieures représentent : (a) la chute due à un excès de bruit de 4 %, (b) la chute due à une efficacité de réconciliation limitée à 90 %, (c) la chute due à l'impossibilité de traiter toutes les données à la vitesse d'émission optique. La courbe pleine inférieure correspond au taux secret atteignable en pratique par le système.*

9.3.2 Fonctionnement dans des conditions optimales

Nous avons déjà présenté, dans le chapitre 6, les bruits typiques intervenant sur le système, et les taux secrets qui en découlent. Dans des conditions optimales de fonctionnement, les paramètres V_A, T, η et v_{el} de la transmission ne fluctuent presque pas au cours du temps. L'excès de bruit joue alors le rôle d'indicateur de la qualité de la transmission, puisqu'il augmente dès que la modulation est incorrecte, ou que des bruits non contrôlés viennent perturber l'interférence. La figure 9.5 présente l'évolution typique de l'excès de bruit au cours du temps quand le système est aussi bien isolé de l'environnement que possible.

Tout d'abord, nous voyons que la mesure de l'excès de bruit est très bruitée. Ce bruit est dû aux fluctuations statistiques sur l'évaluation des paramètres du canal : en effet, l'excès de bruit n'est évalué que sur un nombre fini d'impulsions. Pour réduire ce bruit, il suffirait d'augmenter encore plus la taille des blocs d'amplification de

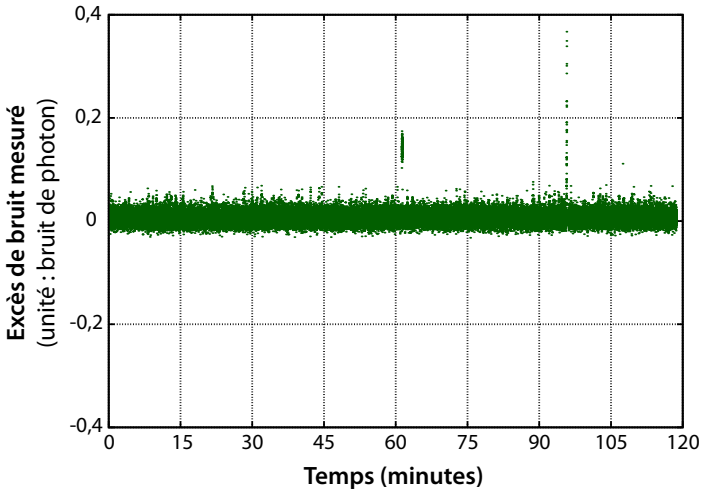


Figure 9.5 : *Excès de bruit du système, pour une distance de transmission équivalente à 15 km et dans des conditions optimisées (peu de vibrations, température stabilisée)*

confidentialité, mais ceci conduirait assez rapidement à des blocs trop grands pour la mémoire vive disponible.

En moyennant ce bruit, nous déterminons un niveau d'excès de bruit moyen d'environ 0,5 %, ce qui est plutôt faible. Par ailleurs, nous voyons l'excès de bruit varier légèrement au cours du temps (les « pics »dans la courbe), ce que nous pouvons principalement attribuer aux variations du bruit de phase dû aux vibrations de l'interféromètre, qui sont transmises aux boîtiers optiques par les câbles qui les relient aux boîtiers PC.

Dans des conditions contrôlées, le système présente donc une bonne stabilité et des performances très bonnes.

9.3.3 Fonctionnement dans des conditions réelles

Lors de cette thèse, nous avons participé au projet intégré européen SECOQC (*Development of a Global Network for Secure Communication based on Quantum Cryptography*). Ce projet, qui s'est achevé fin 2008, avait pour objectif de développer un réseau classique dont les échanges soient sécurisés par une couche inférieure de cryptographie quantique. Son architecture et les résultats de la démonstration ont été présentés dans [78].

Le réseau est présenté schématiquement sur la figure 9.6 : il présente 7 nœuds sécurisés répartis dans la ville de Vienne, entre lesquels des liens de fibre optique sont utilisés. Sur chaque lien du réseau, un système de cryptographie quantique (point à point) est installé, de façon à créer un maillage. Ce système génère alors des clés secrètes à travers le lien, et envoie ces clés à la couche supérieure. Cette couche supérieure comporte une structure de gestion des clés, ainsi que de gestion des communication classiques dans le réseau. Au sommet de la structure, une couche applicative est implémentée, pour permettre à une application d'utiliser les clés générées par la cryptographie quantique.

Ainsi, dans une telle structure, même si deux nœuds ne sont pas directement reliés

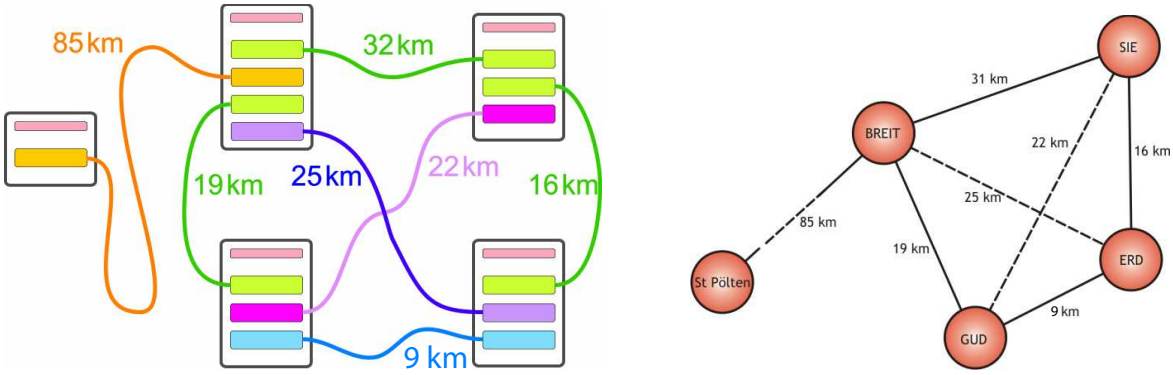


Figure 9.6 : *Structure du réseau de cryptographie quantique mis en place par le projet européen SECOQC.*

par un lien, il est possible de transférer un message crypté par des clés issues de la cryptographie quantique, en procédant par « sauts » de lien à lien. De même, si l'un des liens est coupé ou indisponible, le message peut emprunter un autre chemin pour parvenir au destinataire.

Liens quantiques du réseau SECOQC

Au niveau de la couche physique, le réseau SECOQC est composé de 8 liens :

- trois systèmes *Plug and Play* de la société idQuantique [79]
- un système utilisant un protocole de type BB84 avec codage sur la phase des impulsions, développé par Toshiba [80]
- un système à codage temporel développé par l'université de Genève [25]
- un système fondé sur le protocole à photons intriqués BBM92, développé par l'université de Vienne [81]
- un système à transmission en espace libre fondé sur BB84, développé par l'univeristé de Munich [82]
- ainsi que notre système.

Contraintes de fonctionnement dans un réseau

Puisque, dans des conditions réelles, Alice et Bob sont séparés, leurs communications classiques doivent passer par un réseau installé. Une partie du projet SECOQC avait pour but de développer une interface logicielle entre les systèmes physiques de distribution de clé et les utilisateurs finaux des clés. Cette interface, composée de QBB (pour Quantum BackBone) ainsi que Q3P (pour Quantum Point-to-Point Protocol) et développée par les équipes de Télécom ParisTech (Paris) et d'ARCS (Vienne, Autriche), permet d'implémenter un canal classique adéquat entre les systèmes quantiques, ainsi que de bâtir une structure de réseau entre plusieurs systèmes de cryptographie quantique.

La figure 9.7 représente la structure du nœud assurant l'interface classique : la couche Q3P détermine l'état des systèmes de cryptographie quantique, et assure les communications point à point ; la couche réseau détermine le meilleur chemin pour se

rendre d'un nœud A à un nœud B dans le réseau ; la couche de transport assure le transport de la clé secrète du nœud A au nœud B, par « sauts » de nœud en nœud. Sous cette stucture, nous trouvons les systèmes physiques de distribution de clé ; au dessus, la couche applicative.

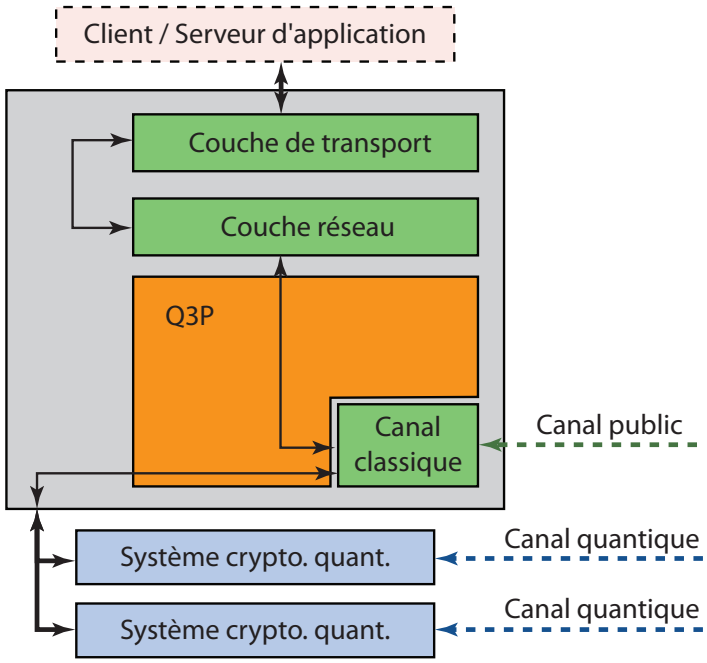


Figure 9.7 : *Structure des nœuds du réseau SECOQC, assurant l'interface entre les systèmes de cryptographie quantique et la couche applicative.*

En particulier, cette interface assure l'authentification des messages, nécessaire en cryptographie quantique. Cette procédure d'authentification nécessite une certaine puissance de calcul, et peut être longue quand les messages ont une taille importante. Enfin, comme dans toute communication classique, chaque message envoyé est soumis à une certaine latence (c'est-à-dire un délai fixe de traitement, dû au système d'exploitation), et l'envoi d'un grand nombre de petits messages nécessite donc plus de temps que l'envoi d'un seul message de grande taille.

Dans notre cas, les latences ont un effet minime, car Alice et Bob n'échangent qu'une vingtaine de messages par extraction de clé ; néanmoins, nous devons transmettre deux messages importants : l'échantillon de données et les choix de phase de Bob, soit environ 64 Mbit par clé. La transmission de ces deux messages par les couches classiques du réseau SECOQC donne lieu à un délai important de traitement, qui est en grande partie due aux étapes d'authentification des messages. Nous avons mesuré ce temps à environ 7 secondes dans les conditions standard de fonctionnement du réseau. Ce délai s'ajoute au temps d'extraction de la clé secrète, et diminue donc le débit de clé.

Résultats

Notre système a été installé sur l'un des liens de ce réseau, à chaque extrémité d'une fibre de 9 km. La transmission de cette fibre a été mesurée à −3,2 dB, ce qui correspond à des pertes en ligne de 0,35 dB/km, soit davantage qu'une fibre monomode standard (0,19 dB/km).

Les boîtiers ont été installés dans un rack comprenant deux autres systèmes de cryptographie quantique ainsi qu'un ordinateur (le « nœud ») assurant la gestion des couches classiques du réseau. Compte tenu du nombre important de ventilateurs dans le rack, nos boîtiers étaient soumis à des vibrations assez importantes. La température dans la pièce, quant à elle, a beaucoup fluctué, car il n'était pas possible de fermer totalement les fenêtres (la température devenant alors trop élevée pour certains détecteurs présents dans la pièce).

Pour nous prémunir contre des situations où le programme se bloque, ou contre le cas où il fonctionne normalement, mais où aucune clé n'est générée (par exemple dans le cas où une des deux cartes d'acquisition « rate » un point, auquel cas les données ne sont plus synchrones), nous avons implémenté un script de contrôle vérifiant régulièrement que des clés sont produites ; dans le cas contraire, le script tente de rédémarrer le programme, d'abord dans les règles de l'art (`SIGINT`), puis plus brutalement (`SIGKILL`), voire en redémarrant les ordinateurs d'Alice et de Bob automatiquement.

Au final, nous présentons dans la figure 9.8 le taux de clé et l'excès de bruit obtenus lors d'une transmission de 57 heures pendant la démonstration SECOQC, sans intervention d'un opérateur. Du fait des conditions plus difficiles que dans le laboratoire, l'excès de bruit, et donc le taux de clé, varient davantage. Nous obtenons un taux secret d'environ 8 kbit/s, avec des pics à 10 kbit/s. L'interprétation précise des fluctuations est difficile ; nous distinguons une structure globale sur 24 heures, qui peut probablement s'expliquer par les fluctuations de la température au cours de la journée.

Le taux de clé obtenu correspond à l'état de l'art actuel ; les performances des autres systèmes présents dans le réseau sont présentées dans [78], nous les résumons ci-dessous :

- Système idQuantique : 1 kbit/s pour 25 km
- Système Toshiba : 3,5 kbit/s pour 32 km
- Système Univ. Genève : 0,7 kbit/s pour 85 km
- Système Univ. Vienne : 2,5 kbit/s pour 16 km
- Système Univ. Munich : 15 kbit/s pour 80 mètres

Nous voyons que, malgré la présence de plusieurs technologies et protocoles de distribution de clé, les performances des systèmes de cryptographie quantique actuels sont assez comparables. Chaque système possède ses avantages et limitations, à la fois au niveau du protocole et de l'implémentation. De façon générale, les systèmes à variables discrètes ont des distances maximales de transmission plus importantes, alors que les systèmes à variables continues ne nécessitent pas de composants spécifiques, et sont plus efficaces à des distances de l'ordre de 15 à 20 km, ce qui correspond à la taille d'un réseau métropolitain.

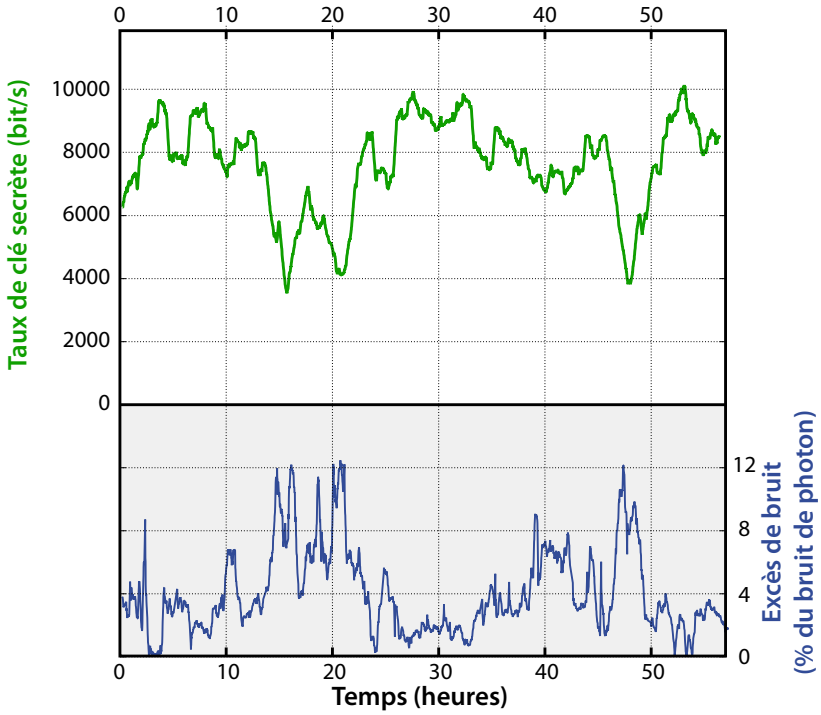


Figure 9.8 : *Taux de génération de clés secrètes et bruit en excès du système, en fonction du temps. Les deux courbes sont des moyennes glissantes : sur 100 clés successives pour l'excès de bruit, et sur une heure pour le taux secret.*

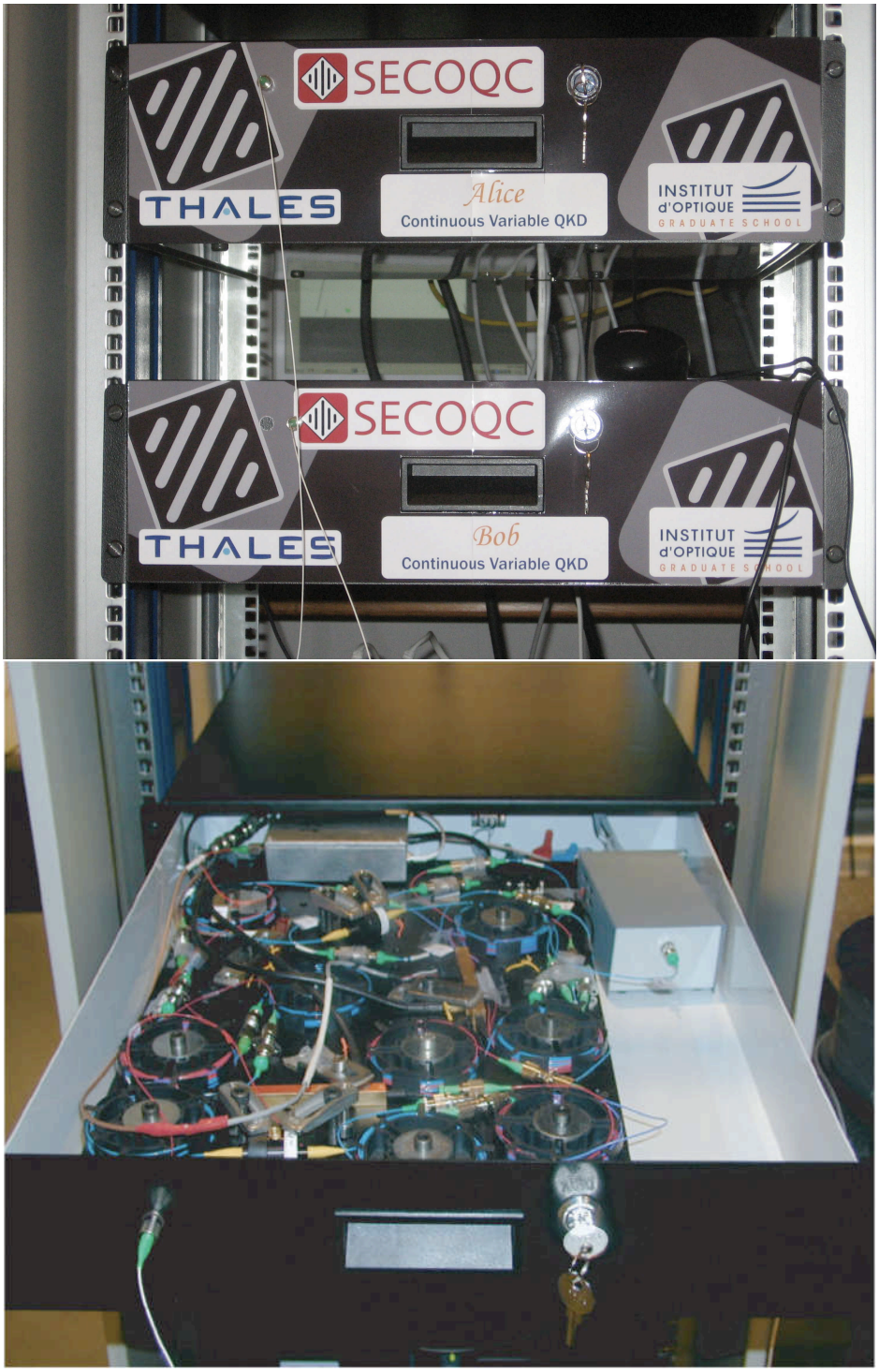


Figure 9.9 : *Quelques photographies de nos systèmes de cryptographie quantique*

Quatrième partie

Perspectives d'amélioration

10 Plus loin, plus vite : nouveaux outils

Plus vite, plus haut, plus fort.

Henri Didon, devise des Jeux Olympiques

Quand Icare, en volant devenu téméraire
S'élève tout à coup au dessus de son Père
L'abandonne, et poussé d'un désir curieux
Tâche autant qu'il le peut à s'approcher des Cieux.
Sur lui, qui sent qu'alors ses plumes se détachent,
Les rayons du Soleil trop vivement s'attachent (...)
Il tombe, et de son Père implorant le secours,
Dans la Mer qui l'attend finit ses tristes jours.

Ovide, *Dédale et Icare*
Traduction de Pierre Corneille

Dans ce chapitre, nous présentons dans un premier temps un nouveau protocole à modulation discrète, qui peut théoriquement permettre de prolonger la distance de transmission de plusieurs dizaines de kilomètres, et que nous avons commencé à implémenter. Nous réfléchissons ensuite aux autres améliorations pouvant être mises en place sur le système de cryptographie quantique pour repousser ses limites en termes de taux et de distance maximale.

10.1 Protocole à modulation discrète — Théorie

10.1.1 Motivation

La nécessité d'un nouveau protocole à variables continues doit provenir des limitations du précédent. Nous l'avons dit, le facteur limitant en termes de distance maximale de transmission est l'efficacité de la réconciliation. Or, l'efficacité de la réconciliation des données fournies par le protocole gaussien dépend de manière cruciale du rapport

signal à bruit des données chez Bob : au dessus d'un SNR de 2, elle atteint environ 90 %, ce qui est assez bon ; en revanche, en dessous d'un SNR de 1, elle décroît très vite, et tend vers 0 quand le SNR devient nul.

Nous avons alors plusieurs choix :

- Garder la variance de modulation V_A fixée, de façon à ce que le SNR décroisse quand la distance de transmission augmente. Dans ce cas, le SNR devient assez rapidement petit, l'efficacité de réconciliation chute brutalement à un certain point, et ΔI devient alors nul.
- Adapter de façon dynamique la variance de modulation d'Alice en fonction de la distance de transmission, de façon à ce que le SNR reste constant chez Bob. Ceci permet de conserver une efficacité de réconciliation constante avec la distance, mais V_A augmente maintenant avec la distance. La distance limite de transmission est définie par $\Delta I = \beta I_{AB} - I_{BE} = 0$, c'est-à-dire par $I_{BE}/I_{AB} = \beta$. Pour des paramètres courants de transmission, la figure 10.1 présente l'évolution du rapport I_{BE}/I_{AB} en fonction de la variance de modulation. Nous nous apercevons que le rapport tend vers 1, et qu'il existera donc toujours un point auquel le rapport croisera la valeur de β. Au croisement, nous obtenons une nouvelle fois un point pour lequel ΔI devient nul, autour de 30 à 40 km.

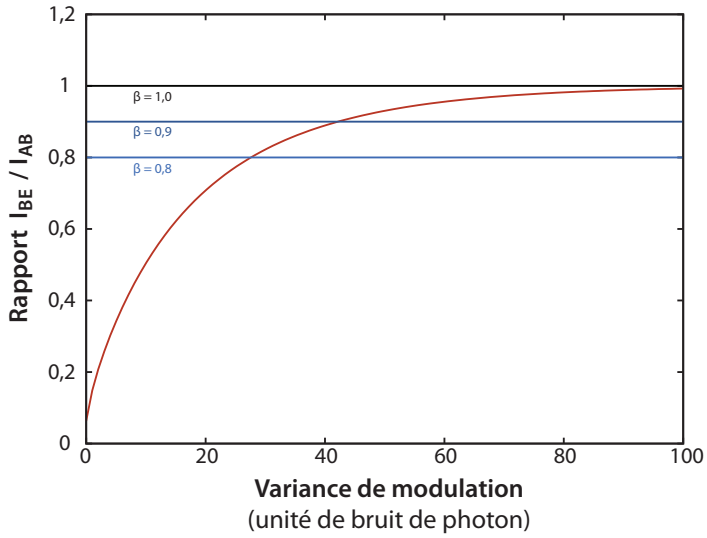


Figure 10.1 : *Rapport I_{BE}/I_{AB} en fonction de la variance de modulation. La distance limite de transmission est atteinte quand cette courbe croise la valeur de l'efficacité de réconciliation β.*

- Compte tenu des deux premiers points, nous nous apercevons que la seule façon de prolonger significativement la distance est de conserver V_A constant, de sorte que le SNR diminue avec la distance, et de trouver un codage de l'information permettant de réconcilier les données avec une efficacité correcte, même pour un SNR tendant vers zéro. Le protocole que nous présentons dans la suite a donc été conçu pour répondre à ces contraintes.

10.1.2 Description du protocole

Puisque les algorithmes de réconciliation sont optimisés pour un codage discret des données, la modulation gaussienne continue ne semble pas une bonne approche. Un protocole dans lequel les impulsions subissent une modulation discrète (mais toujours sur des variables continues) semble plus adapté. Cependant, comme toujours en cryptographie quantique, la conception d'un protocole est inutile s'il est impossible de prouver sa sécurité ; or, les preuves que nous avons présentées dans le chapitre 4 ne peuvent pas être appliquées directement à un protocole à modulation discrète, puisqu'elles utilisent le fait que la modulation est gaussienne, et qu'elle est donc équivalente au schéma à intrication virtuelle fondé sur une paire EPR.

En 2008, Anthony Leverrier et Philippe Grangier ont présenté dans [83] un nouveau protocole à modulation quaternaire, tout en en démontrant la sécurité contre des attaques cohérentes générales. Les aspects théoriques de cette section s'inspirent largement de cet article.

Le protocole est simple : plutôt que de moduler les impulsions de façon gaussienne, Alice définit une amplitude qu'elle impose à toutes les impulsions, et code l'information sur la phase de celles-ci. La phase peut prendre quatre valeurs : $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ ou $\frac{7\pi}{4}$, de façon aléatoire. Nous représentons sur la figure 10.2 ces quatre états dans l'espace des phases. Nous pouvons alors directement définir l'information contenue par les impulsions comme le signe de la projection de l'état sur les axes de quadrature : bit 0 si la projection est négative, ou 1 si elle est positive.

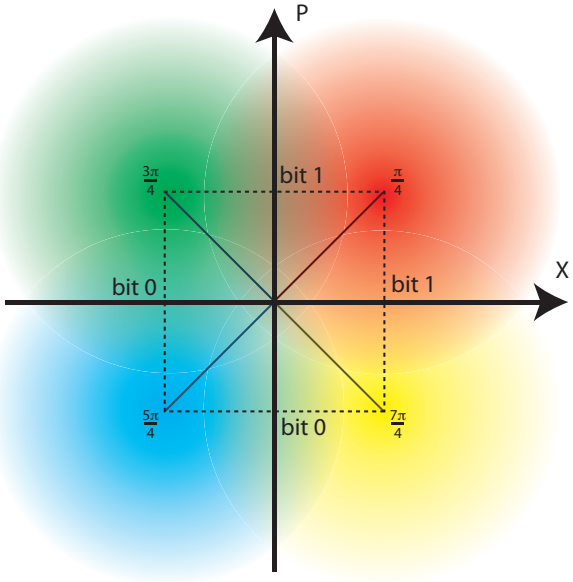


Figure 10.2 : *Les quatre états du protocole quaternaire, représentés dans l'espace des phases. Le protocole est conçu pour travailler à amplitude basse, ce qui explique l'importance du bruit de photon.*

10.1.3 Information secrète

Le raisonnement sous-jacent à la preuve de sécurité de ce protocole est le même que pour nos preuves collectives : le protocole prepare-and-measure est modélisé par un protocole à intrication virtuelle. L'enjeu des preuves de sécurité du protocole quaternaire est de trouver un état pur bi-mode adéquat tel que, quand Alice mesure l'un des modes, elle ne puisse obtenir que quatre résultats, et qu'en fonction de ce résultat le mode de Bob soit projeté dans l'un des quatre états souhaités.

Une fois que nous connaissons l'état intriqué adéquat, il suffit alors de calculer sa matrice de covariance : comme pour le protocole gaussien, l'extrémalité des états gaussiens [45, 46] peut être utilisée pour borner l'information de l'espion. Nous appliquons alors exactement le même raisonnement que pour les preuves gaussiennes ; en particulier, l'introduction du mode réaliste est immédiate.

Un état répondant aux conditions est présenté dans [83] :

$$|\Phi\rangle_{\mathrm{A,B}} = \frac{1}{4}\sum_{k=0}^{3} |\psi_k\rangle_{\mathrm{A}} \otimes |\alpha_k\rangle_{\mathrm{B}} \tag{10.1}$$

$$\text{où} \quad |\psi_k\rangle = \sum_{m=0}^{3}\sum_{n=0}^{\infty} L_{k,n,m}\frac{\alpha^{4n+m}}{\sqrt{(4n+m)!}}\ |4n+m\rangle \tag{10.2}$$

$$\text{et} \quad |\alpha_k\rangle = |\alpha e^{i\frac{(2k+1)\pi}{4}}\rangle \tag{10.3}$$

Les états $|\alpha_k\rangle$ sont les 4 états du protocole, intriqués avec les quatre états $|\psi_k\rangle$ — dont l'expression ressemble beaucoup à la décomposition de l'état cohérent $|\alpha\rangle$, mais avec des coefficients complexes $L_{k,m,n}$ devant les états de Fock. Ces quatre états $|\psi_k\rangle$ ont été conçus pour être orthogonaux, ce qui implique qu'Alice peut les distinguer en effectuant une mesure projective $\{|\psi_0\rangle\langle\psi_0|, |\psi_1\rangle\langle\psi_1|, |\psi_2\rangle\langle\psi_2|, |\psi_3\rangle\langle\psi_3|\}$. Elle prépare ainsi avec certitude l'un des états $|\alpha_k\rangle$ sur l'autre mode, de façon aléatoire, et le protocole à intrication virtuelle est donc équivalent au protocole prepare-and-measure.

Il ne reste alors plus qu'à calculer la matrice de covariance de $|\Phi\rangle$, qui s'écrit

$$\Gamma_4 = \begin{bmatrix} (V_A+1)\mathbb{I}_2 & Z_4\sqrt{T}\sigma_z \\ Z_4\sqrt{T}\sigma_z & (TV_A+1+T\varepsilon)\mathbb{I}_2 \end{bmatrix} \tag{10.4}$$

Danc cette expression, Z_4 remplace la corrélation de l'état EPR pour notre protocole gaussien, qui s'écrit $Z_{\mathrm{EPR}} = \sqrt{V^2-1}$. Pour le protocole quaternaire, l'expression est plus complexe [83] :

$$Z_4 = \frac{V_A}{2}\, e^{-\frac{V_A}{2}} \left(\sqrt{\frac{a_0}{a_1}} + \sqrt{\frac{a_1}{a_2}} + \sqrt{\frac{a_2}{a_3}} + \sqrt{\frac{a_3}{a_0}}\right) \tag{10.5}$$

où

$$\begin{aligned} a_0 &= \cosh(V_A/2) + \cos(V_A/2) \\ a_1 &= \sinh(V_A/2) + \sin(V_A/2) \\ a_2 &= \cosh(V_A/2) - \cos(V_A/2) \\ a_3 &= \sinh(V_A/2) - \sin(V_A/2) \end{aligned}$$

La figure 10.3 (gauche) présente les courbes de Z_4 et Z_{EPR} en fonction de V_A. L'état EPR étant l'état maximalement intriqué en quadratures, la corrélation Z_4 est nécessairement toujours moins bonne que la corrélation Z_{EPR}. Notons toutefois que la différence entre les courbes devient très petite quand V_A devient plus petit que 1, ce qui implique que la modélisation à intrication virtuelle devient alors quasiment la même.

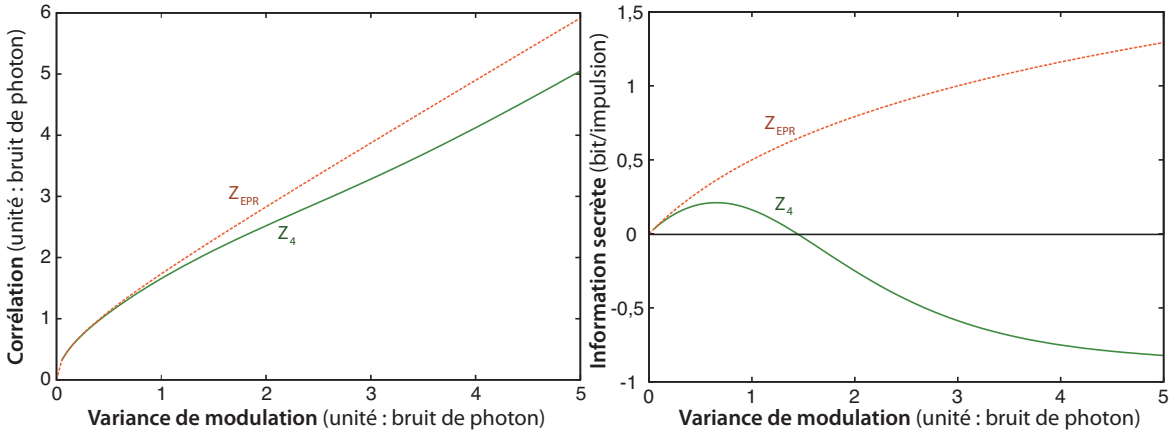


Figure 10.3 : *Corrélation* $\left\langle \hat{X}_A \hat{X}_B \right\rangle$ *entre les deux modes de l'état intriqué, ainsi que l'information secrète à distance nulle, dans les deux cas d'un état EPR ou de l'état intervenant dans le protocole quaternaire. Le système est ici parfait :* $\eta = 1, \varepsilon = v_{el} = 0$.

Pour obtenir l'expression du taux secret, nous appliquons le même raisonnement que pour les preuves de sécurité gaussiennes, en remplaçant la matrice γ_{AB_1} par la nouvelle matrice de covariance Γ_4, et obtenons une expression du taux secret. L'information secrète maximale disponible en fonction de la variance de modulation est présentée sur la figure 10.3 (droite). Nous voyons qu'à faible variance les deux protocoles sont presque identiques, et qu'à l'inverse, pour des variances élevées, l'information secrète devient négative.

Cet effet peut se comprendre de plusieurs façons : du point de vue du codage classique, le codage gaussien est optimal pour un canal de type AWGN, c'est-à-dire qu'il atteint la capacité du canal $C = \frac{1}{2}\log_2(1 + \mathrm{SNR})$; au contraire, le codage binaire est d'autant moins optimal que le SNR est élevé. Plus qualitativement, l'information mutuelle transmise par un canal de SNR élevé est supérieure à 1 bit ; or, l'information maximale codée par un codage binaire est de 1 bit, ce qui est nécessairement non-optimal. À bas SNR, en revanche, les tranches inférieures de la gaussienne ne contiennent presque pas d'information secrète, et seule la première tranche est donc pertinente : le codage gaussien devient presque identique au codage binaire.

Ceci se traduit par une « efficacité de codage », qui peut être incluse dans l'efficacité de réconciliation, et qui est égale à 1 pour le codage gaussien mais strictement plus petite que 1 pour le codage binaire. Par conséquent, l'information mutuelle $I_{AB,4}$ est strictement plus petite que $I_{AB,\mathrm{gauss}}$, alors que l'information de l'espion est la même dans les deux cas. Par conséquent, ΔI_4 s'éloigne de plus en plus de $\Delta I_{\mathrm{gauss}}$ quand V_A augmente. À l'opposé, quand la variance de modulation tend vers 0, les deux codages

deviennent presque identiques, et l'efficacité du codage tend asymptotiquement vers 1.

10.1.4 Réconciliation des données

Tout l'intérêt du protocole quaternaire vient de la possibilité de mieux réconcilier les données. Après la transmission, Bob possède des données qui correspondent à des données binaires ayant transité par un canal AWGN. Rappelons que l'information est alors codée par le signe de la mesure, signe qu'Alice doit donc retrouver à partir de ses données de modulation non bruitées. Par ailleurs, le protocole est conçu pour fonctionner à des SNR inférieurs à 1, ce qui veut dire que la quantité d'information mutuelle contenue dans chaque impulsion est très faible.

Pour décrire le schéma de réconciliation, considérons par exemple que l'information mutuelle I_{AB} est égale à 8.10^{-4}, soit un SNR de $1{,}6.10^{-3}$, et que la taille du bloc à réconcilier est de 2 000 000. Pour réconcilier nos données, il nous faudrait disposer d'un code de taux 0,08 %. Or, des codes de taux si bas n'ont jamais été optimisés, principalement parce que l'industrie des télécommunications ne travaille pas avec de telles conditions de bruit. En revanche, nous connaissons quelques codes d'assez bas rendement fournissant une bonne efficacité (par exemple R = 10 % pour $\beta \approx 0{,}8$, voir [84]), que nous aimerions donc utiliser.

Anthony Leverrier propose dans [83] une technique de réconciliation très efficace, utilisant des codes à répétition sur les données très bruitées. Dans un premier temps, Bob regroupe ses 2 000 000 impulsions par paquets de 100 ; nous notons b_k les impulsions du paquet, $k \in [\![0; 99]\!]$. Le but pour Alice est alors de trouver la valeur du signe de la première impulsion b_0 de chaque paquet. Pour l'aider, Bob lui envoie la *side information* suivante : la valeur absolue de ses mesures pour toutes les impulsions, et les valeurs de parités suivantes : $\text{sign}(b_0 \times b_1), \text{sign}(b_0 \times b_2), \cdots, \text{sign}(b_0 \times b_{99})$. Alice attribue alors à b_0 le signe le plus probable, par vote majoritaire, compte tenu de l'information envoyée par Bob.

Pour des SNR faibles, l'efficacité du code à répétition est très bonne : $\beta_{\text{repet}} \approx 1 - \text{SNR}_{\text{final}}/2$. Dans notre cas, la chaîne de sortie contient 20 000 valeurs et possède un SNR approximativement 100 fois plus grand que le SNR initial, c'est-à-dire d'environ 16 %, ce qui fournit une efficacité du code à répétition de 92 %. Nous décodons ensuite cette chaîne finale grâce à notre code LDPC de taux 10 % possédant une efficacité plus grande que 80 %, ce qui donne finalement une efficacité globale de l'ordre de 75 %.

Pour généraliser ce raisonnement à tous les SNR initiaux, il suffit de regrouper les 2 000 000 bits par paquets de taille l, telle que $l \approx 0{,}16/\text{SNR}$. De cette façon, le code LDPC de taux 10 % sera toujours utilisable, et l'efficacité totale de réconciliation quasi-constante avec le SNR.

10.1.5 Courbes théoriques de taux

Nous traçons tout d'abord la courbe d'information secrète brute (sans l'efficacité de réconciliation) pour le protocole quaternaire, et nous la comparons avec le protocole gaussien (figure 10.4). Sans surprise, l'information, pour le protocole gaussien, diminue de façon exponentielle de la distance, mais sans jamais atteindre de seuil. Au contraire, du fait de l'inefficacité du codage, l'information secrète du protocole quaternaire est

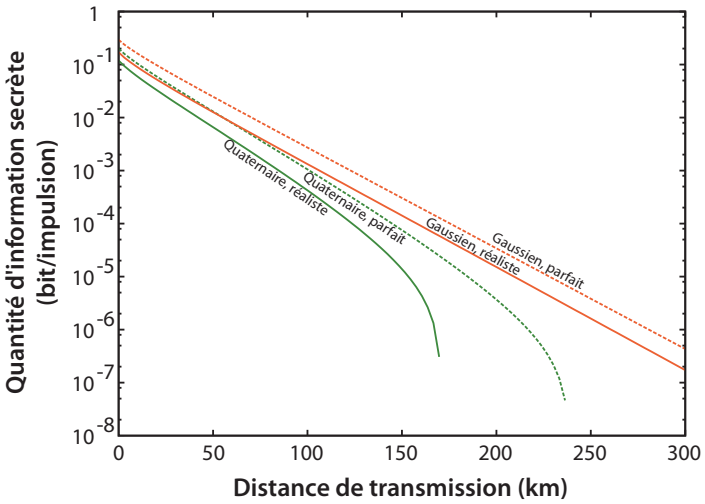


Figure 10.4 : *Information secrète des deux protocoles en fonction de la distance, pour un système parfait ($\eta = 1$; $v_{el} = \varepsilon = 0$) et pour un système réaliste ($\eta = 0{,}5$; $v_{el} = 0{,}01$; $\varepsilon = 0{,}005$). La variance de modulation est choisie à $V_A = 0{,}5$.*

toujours inférieure et atteint un seuil pour une certaine distance limite de transmission, même pour un système idéal.

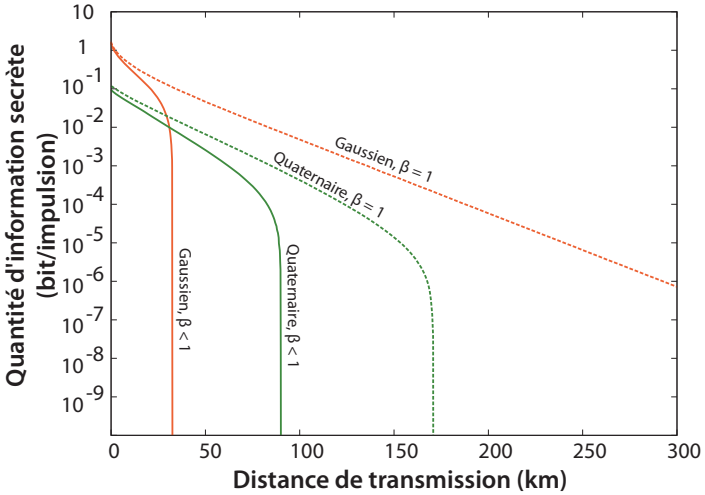


Figure 10.5 : *Information secrète des deux protocoles en fonction de la distance pour un système réaliste ($\eta = 0{,}5$,$v_{el} = 0{,}01$,$\varepsilon = 0{,}005$), en prenant en compte les efficacités de réconciliation pouvant être atteintes pour chacun des deux protocoles. La variance de modulation est choisie à $V_A = 0{,}5$ pour le protocole quaternaire, à 20 pour le protocole gaussien.*

Considérons ensuite l'effet de l'efficacité de réconciliation. La figure 10.5 montre les taux effectifs $\beta I_{AB} - \chi_{BE}$ pour les deux protocoles. Nous voyons alors que le désavantage du protocole quaternaire en terme de codage est largement compensé quand l'efficacité de réconciliation est prise en compte. Ainsi, le taux du protocole gaussien devient nul

autour de 30 km, du fait de la chute de l'efficacité de réconciliation à longue distance. Au contraire, le seuil du protocole quaternaire est beaucoup plus distant (environ 85 km), car l'efficacité de réconciliation reste élevée à longue distance.

Néanmoins, notons que le SNR diminue lui aussi exponentiellement avec la distance, et ces courbes théoriques supposent que nous sommes en mesure de réaliser une transmission quantique pour des SNR arbitrairement bas. Ceci n'est bien entendu pas possible en réalité, et nous détaillerons dans la section suivante les limitations sur le SNR imposées par le système expérimental.

10.2 Protocole à modulation discrète — Mise en œuvre

Au cours de la dernière année de ce travail de thèse, nous avons implémenté les structures nécessaires au protocole quaternaire sur le système expérimental, et réalisé une analyse préliminaire des performances du système.

10.2.1 Adaptation du système existant

Nous avons choisi de ne pas créer un système dédié uniquement au protocole quaternaire, dans la mesure où le système existant possède toutes les caractéristiques nécessaires au bon fontionnement de celui-ci. En effet, du point de vue optique, les différences entre les deux protocoles sont petites : chez Alice, il n'y a plus de modulation d'amplitude, et la modulation de phase devient discrète ; chez Bob, le protocole ne change pas.

Rétrocontrôle du modulateur d'amplitude

En conséquence du fait que, pour le protocole quaternaire, l'amplitude des impulsions reste fixe, l'algorithme que nous utilisions pour contrôler les tensions de commande des modulateurs d'amplitude n'est plus disponible. Nous pourrions imaginer d'ajuster l'amplitude de sortie d'Alice par des pas progressifs de tension, comme nous le faisions avec le deuxième modulateur dans le protocole gaussien, mais nous ne disposons que d'un signal de contrôle (l'amplitude de la photodiode) pour deux tensions de commande, ce qui rendrait la tâche difficile.

Nous avons donc choisi de recréer le signal que nous utilisions dans le protocole gaussien : tous les 100 blocs, une rampe de tension est appliquée au premier modulateur d'amplitude, de façon à recréer la caractéristique de celui-ci. Nous déterminons ensuite comme précédemment les tensions de contrôle du modulateur, et fixons l'amplitude à la valeur souhaitée.

Détection de la modulation d'Alice

Du fait du travail à bas SNR, le protocole de détection de la modulation d'Alice ne fonctionne plus correctement car l'amplitude des impulsions-test est trop petite. Pour que la synchronisation fonctionne, nous avons deux possibilités :

- Au début de la modulation d'Alice, moduler à grande amplitude les deux blocs utilisés par Bob pour la synchronisation, puis moduler les blocs suivants à faible amplitude.
- Dans chaque bloc, moduler les impulsions-test à grande amplitude et les impulsions de données à amplitude faible.

Nous avons choisi la deuxième solution car, bien qu'elle nous fasse perdre le lien entre la variance des impulsions-test et la variance des impulsions de données, elle permet un contrôle plus précis de la phase relative entre signal et oscillateur local. En effet, si la variance des impulsions-test est trop petite, le bruit résiduel après moyennage du bruit de photon n'est plus négligeable devant leur amplitude, et une incertitude sur la phase apparaît. Or, puisque l'information est codée sur la phase, ceci se traduit directement en excès de bruit.

Travail à bas SNR et à longue distance

Outre les problématiques de détection, le travail à bas SNR et/ou à longue distance pose trois limitations :

- Les cartes d'acquisition ne codent l'information que sur 12 bits, c'est-à-dire à peine quatre chiffres décimaux significatifs. Si nous fixons la plage de détection de notre carte d'acquisition à une amplitude de 4 fois l'amplitude du bruit de photon (de la même façon que nous la fixons à 4 fois l'amplitude de modulation pour le protocole gaussien), l'intervalle d'amplitude le plus petit que nous puissions mesurer est de $2^{-9}\sqrt{N_0}$, c'est-à-dire 0,2 % de $\sqrt{N_0}$. Cet intervalle peut être compris comme un bruit en excès dû à l'aquisition, exprimé au niveau du détecteur. Rapporté à la sortie du canal, il vaut donc $\varepsilon_{\text{acq}} N_0 = \frac{4.10^{-6}}{G} N_0$.
  Si nous fixons un bruit maximal autorisé (dû à l'acquisition) à $\varepsilon_{\text{acq, max}} = 0{,}01$, nous obtenons $G_{\text{min}} = 4.10^{-4}$, soit -34 dB (incluant les pertes du détecteur), c'est-à-dire une distance maximale de sécurité de 155 km — même en l'absence d'excès de bruit du reste du système.
- La précision de la calibration du bruit de photon devient cruciale. Nous atteignons typiquement des précisions relatives de calibration $\Delta N_0 = \varepsilon_{\text{calib}} \approx 2.10^{-3}$. Le calcul du point précédent reste alors valable, ce qui donne des pertes maximales admissibles d'environ -27 dB, soit une distance maximale de sécurité de 120 km.
- Puisque nous générons l'oscillateur local au niveau de la diode d'Alice, celui-ci est affecté par les pertes du canal. Ainsi, dans le cas de fortes pertes, il pourrait se révéler trop faible au niveau du détecteur de Bob. Le risque est alors que le bruit électronique ne soit plus négligeable devant le bruit de photon, ce qui réduirait notablement l'information secrète.

10.2.2 Résultats préliminaires

La totalité du protocole quaternaire a été implémentée dans le logiciel du système (modulation d'Alice, choix de quadrature de Bob et contrôle de la phase relative). Comme pour les résultats du protocole gaussien, le paramètre le plus pertinent à observer est l'excès de bruit du système, qui permet en particulier de vérifier que la

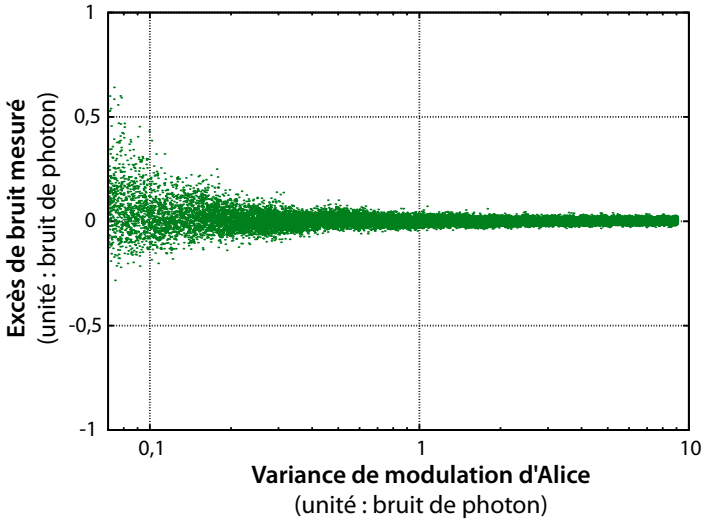


Figure 10.6 : *Excès de bruit, à distance de transmission nulle, en fonction de la variance de modulation d'Alice, pour le protocole quaternaire.*

modulation est correctement réalisée. La figure 10.6 présente l'excès de bruit de notre système, pour une modulation quaternaire.

En moyennant la courbe, nous obtenons un excès de bruit du même ordre que pour le protocole gaussien : moins de 2 % du bruit de photon. Cet excès de bruit moyen augmente un peu quand la variance de modulation diminue ; l'effet le plus significatif est néanmoins l'évolution du *bruit sur l'excès de bruit*, que nous voyons beaucoup augmenter quand la variance de modulation, et donc le SNR, diminue.

Il s'agit à nouveau d'un effet statistique, que nous essayons d'expliquer ainsi :

- Remarque préliminaire : les fluctuations statistiques sont à peu près constantes, car elles proviennent de l'évaluation de V_B, qui est toujours de l'ordre de 1. Elles ont une variance de l'ordre de 0,01 N_0, et leur effet sur un terme S n'est donc vraiment visible que si S est de cet ordre.
- L'excès de bruit peut s'exprimer $\varepsilon = V_A \left(\frac{V_B - 1}{V_B \rho^2} - 1 \right) - \frac{v_{\text{el}}}{G}$. Dans cette expression, V_A et v_{el} sont fixés, et $G \approx 1$: les fluctuations relatives sur G sont donc petites. Restent donc deux termes sur lesquels les fluctuations statistiques peuvent avoir un effet : le ρ^2 du dénominateur, et le terme $\frac{V_B - 1}{V_B}$.
- Puisque V_B est toujours de l'ordre de 1, les fluctuations sur l'évaluation de V_B sont à peu près constantes avec la variance de modulation. Le terme $\frac{V_B - 1}{V_B}$ est donc responsable du bruit que nous observons systématiquement, même à grande variance de modulation.
- En revanche, l'effet de ρ^2 n'est pas constant avec la variance de modulation. À forte variance de modulation, $\rho^2 \approx 1$ et ses fluctuations sont donc petites devant 1 : ε subit alors des fluctuations assez constantes, proportionnelles à $\frac{\delta V_B}{1 + \delta \rho^2} - 1$. À faible variance, au contraire, ρ^2 peut devenir très petit devant 1, et ε subit alors des fluctuations de l'ordre de $\frac{\delta V_B}{\delta \rho^2} - 1$, plus grandes qu'à forte variance de modulation. C'est ce que nous observons sur la courbe 10.6.

Interprétation

Compte tenu de ces résultats, le problème principal que nous aurons à régler à faible variance de modulation sera vraisemblablement l'effet statistique. Celui-ci est en théorie facile à diminuer, en augmentant la taille des échantillons ; en pratique, les fluctuations statistiques varient en $1/\sqrt{N}$, où N est la taille de l'échantillon, et nous risquons donc d'être limités par la mémoire disponible pour stocker l'échantillon dans l'ordinateur.

Hormis ces effets statistiques, moyenner la courbe nous donne un excès de bruit moyen assez faible (moins de 2 % de N_0), du moins jusqu'à un SNR chez Bob de l'ordre de 0,1. Si nous nous plaçons alors à un SNR de 0,1 chez Bob, sachant que la variance de modulation optimale chez Alice pour le protocole est de l'ordre de 0,5 N_0, nous obtenons un gain total de l'ordre de $\eta T \approx 0{,}2$, soit $T \approx 0{,}35$, c'est-à-dire une distance de transmission de l'ordre de 20 à 25 km.

Pour aller plus loin, il nous faudra (outre les effets statistiques) vérifier que l'excès de bruit reste petit avec l'augmentation de distance. Si c'est le cas, nous pouvons espérer une distance de transmission de l'ordre de 50 à 70 km ; au delà, il nous faudra trouver une solution pour régénérer la puissance de l'oscillateur local.

10.3 Pour aller encore plus vite...

Dans cette dernière section, nous réfléchissons aux possibilités d'amélioration de notre système à moyen terme. Actuellement, la vitesse du système est limitée par la vitesse d'extraction de la clé, et la fréquence actuelle d'émission des impulsions nous convient donc. Néanmoins, puisque ce problème est purement logiciel (ce qui ne veut pas dire qu'il soit simple à résoudre), il est raisonnable de penser que le développement d'une architecture de calcul dédiée est possible, et qu'elle puisse accélérer d'un facteur important la vitesse d'extraction des données.

Nous proposons donc tout d'abord une solution adaptée au type de calcul réalisé, à savoir le calcul massivement paralèlle sur un processeur graphique. Dans la suite, nous mettons de côté les enjeux de vitesse logicielle, pour essayer de lister les différents problèmes qui se poseront lors d'une accélération de la fréquence d'émission de données.

10.3.1 Réconciliation sur un processeur graphique

Depuis plus de 10 ans, les micro-ordinateurs contiennent tous un processeur graphique, sous forme de chipset sur la carte mère ou de carte spécifique. Un écran moderne contient un nombre important de pixels (environ 1 million), chacun étant mis à jour plus de 60 fois par seconde. L'écran est une matrice de grande taille, chaque point de la matrice étant mis à jour à chaque rafraîchissement en fonction de la matrice de l'itération précédente, et des nouveaux signaux en entrée (mouvement de la souris, déplacement d'un personnage, ...). Or, l'architecture des processeurs habituels n'est pas vraiment adaptée à la gestion de cet affichage, car elle est constituée d'un seul processeur [1] très puissant, conçu pour effectuer de nombreuses opérations sur un nombre limité de variables. Les processeurs graphiques, à l'inverse, sont constitués de centaines

1. En fait, les nouveaux processeurs en ont jusque 4 ou 8...

d'unités de traitement certes plus simples, mais capables d'effectuer en parallèle des opérations de complexité limitée sur de grandes matrices de variables. Ils sont donc très bien adaptés au calcul rapide de l'évolution de l'affichage.

La réconciliation par algorithme Message Passing correspond tout à fait à cette logique : nous disposons à chaque étape d'un ensemble de LLR (de variables ou parité), et nous mettons à jour chaque LLR de type opposé (resp. de parité ou variable) en fonction de cet ensemble. Par exemple, pour un code de taux 50 %, nous mettons à jour 100 000 LLR de parité, de façon *indépendante*, en fonction du vecteur des 200 000 LLR de variable. Par analogie avec les calculs d'affichage, ce type de traitement massivement parallèle semble alors très adapté à un processeur graphique !

Une implémentation OpenGL de l'algorithme sur une carte graphique Nvidia GTX 7950 par Jérôme Lodewyck avait déjà permis, en 2006, d'effectuer la réconciliation plus rapidement que sur un cœur d'un processeur Core 2 Duo. Depuis, une interface C++ spécialement dédiée au calcul sur carte graphique (GPGPU, pour *General Purpose GPU computation*) a été fournie par Nvidia. Qui plus est, la vitesse des processeurs graphiques a été multipliée par 8 entre temps, sans compter les améliorations sur la mémoire et le nombre de processeurs élémentaires. Il est donc très probable qu'une nouvelle implémentation permette de réduire le temps de réconciliation par un facteur de l'ordre de 10.

Notons pour finir que toutes l'extraction ne se résume pas à la réconciliation, et que (comme nous l'avons vu) l'amplification de confidentialité, la génération de nombres aléatoires et les rétrocontrôles nécessitent un temps de calcul très significatif. Or, ces opérations ne peuvent pas (au premier abord) être implémentées aisément sur un processeur graphique. Le problème de la vitesse de traitement est donc toujours ouvert.

10.3.2 Augmentation de la vitesse d'émission des impulsions

Oublions maintenant les problématiques de vitesse logicielle. La fréquence d'émission de nos impulsions est actuellement de 500 kHz, ce qui nous limite typiquement à des taux maximaux de 50 kbit/s à 25 km, quels que soient nos efforts pour optimiser la réconciliation. Quels seront donc les problèmes que nous rencontrerons en augmentant la fréquence d'émission ?

De 500 kHz à 5 MHz : peu de problèmes.

Nous émettons actuellement des impulsions de 100 ns, multiplexées en temps de 400 ns, avec un rapport cyclique de 20. Pour atteindre 5 MHz, aucun problème n'apparaît réellement : nous pouvons réduire facilement la durée des impulsions à 50 ns, multiplexées de 100 ns, et émises toutes les 200 ns. Il faudra bien entendu modifier la durée d'intégration de la détection homodyne en conséquence. Les cartes d'acquisition PCI-6111 ne fonctionneront plus à 5 MHz, mais nous pouvons utiliser le modèle supérieur PCI-6115, qui est spécifié à 10 MHz.

De 5 MHz à 50 MHz : envisageable

Pour atteindre 50 MHz, la tâche est plus délicate :

- La génération des impulsions est toujours possible : impulsions de 5 ns, multiplexées de 10 ns et émises toutes les 20 ns. Notons tout de même que nous

devrons prendre garde au temps de montée de notre générateur d'impulsions, qui est actuellement un peu trop grand (typiquement, 5-10 ns).

- Concevoir une détection homodyne impulsionnelle opérationnelle toutes les 20 ns devrait être possible [85] : le temps de montée de l'amplificateur de charge est de 2 ns, la bande passante de l'amplificateur de 350 MHz, et celle des photodiodes de 1 GHz. Ceci n'est en revanche pas trivial, et il faudra en particulier prendre garde au bruit de l'électronique.
- En revanche, les cartes d'acquisition que nous utilisons ne peuvent plus fonctionner. Il est particulièrement difficile de trouver des cartes d'acquisition rapides possédant toutes les caractéristiques que nous recherchons : simultanéité des acquisitions, synchronisation externe et profondeur mémoire suffisante. Nous avons pensé à des cartes du type de celles utilisées dans les oscilloscopes rapides, qui sont spécifiées à plus d'un GHz. Néanmoins, non seulement elles sont très chères, mais elles ne sont souvent pas en mesure d'effectuer une numérotation fiable des impulsions qu'elles acquièrent, et « ratent » des impulsions. Ceci étant, la technologie existe, et nous pouvons espérer trouver un modèle de carte qui nous convienne.

De 50 MHz à 500 MHz ou plus : difficile à concevoir

Il est difficile d'envisager un fonctionnement à 500 MHz. Tous les composants atteignent alors leurs limites : photodiodes, générateurs d'impulsions, amplificateurs, cartes d'acquisition, et sans doute aussi la diode laser. Seuls les modulateurs se distinguent, puisqu'ils ont une bande passante de plus de 30 GHz. Qui plus est, concevoir de l'électronique gigahertz à bas bruit est une tâche particulièrement ardue...

Bilan

Compte tenu de ces considérations, nous pouvons raisonnablement espérer une fréquence maximale de fonctionnement de l'ordre de 30 à 100 MHz. Pour celle-ci, nous atteignons un taux maximal de plusieurs mégabits par seconde, sous réserve que le traitement logiciel soit suffisamment performant. Néanmoins, l'étape la plus directe est le passage de 500 kHz à 5 MHz, ce qui nous permettrait déjà de gagner un ordre de grandeur sur le taux secret — sous réserve d'améliorer la vitesse de traitement, condition nécessaire.

11 Amplificateurs adaptés à la cryptographie quantique

11.1 Problématique

En télécommunications classiques, les transmissions se font sur de très longues distances, jusqu'à l'échelle de la Terre. Compte tenu de l'atténuation des fibres optiques utilisées (typiquement 50 % tous les 15 km), le signal doit être régénéré régulièrement. Ceci est effectué grâce à un répéteur, qui consiste principalement en une mesure et une correction de l'information binaire encodée, et une réémission du signal reçu avec un meilleur rapport signal à bruit.

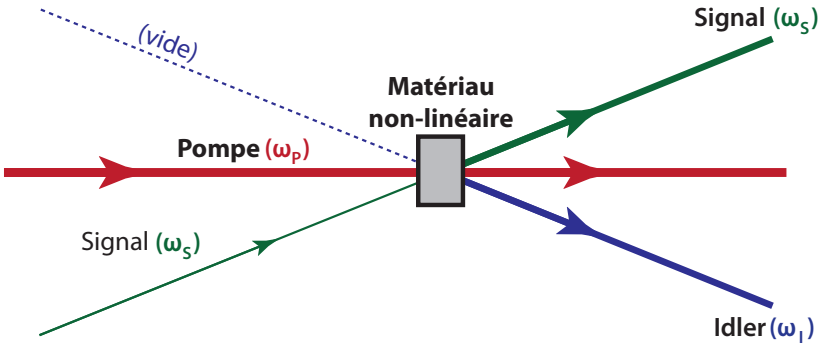
Dans le cas de la cryptographie quantique, l'effet des pertes du canal est encore plus critique, et il induit une distance limite de transmission de l'ordre de 20 à 200 km (en fonction des systèmes). Pour s'affranchir de cette limitation, il est donc intéressant de se demander s'il peut exister un tel type de système de régénération du signal pour la cryptographie quantique, c'est-à-dire un *répéteur quantique*. Les techniques classiques sont d'emblée rhédibitoires : une mesure complète de l'information détruit complètement les corrélations quantiques, ce qui se traduit dans notre cas par l'ajout d'un bruit équivalent à celui d'une attaque totale de l'espion. Il ne reste alors plus de secret dans les données partagées.

Des schémas de répéteurs quantiques ont été proposés il y a 10 ans [86, 87], mais ils requièrent en pratique l'utilisation de mémoires quantiques, permettant de stocker un état quantique pendant un temps relativement long (µs – ms). Le développement de ces mémoires n'en est encore qu'à un stade préliminaire [88, 89], et il n'est donc pas possible de développer de tels répéteurs quantiques à l'heure actuelle.

Malgré la non-disponibilité de ces répéteurs, nous avons cherché une technique permettant de prolonger la distance maximale de transmission. Assez naturellement, nous nous sommes tournés vers des schémas d'amplification optique, afin de tester leur effet sur notre protocole de cryptographie quantique. Nous présentons dans ce chapitre nos travaux concernant deux types d'amplificateurs : les amplificateurs paramétriques, étudiés et mis en œuvre depuis plus de 20 ans, et un nouveau modèle d'amplificateur non-déterministe proposé par Ralph et Lund en 2008 [90].

11.2 Amplificateur paramétrique optique

11.2.1 Principe général



Le processus d'amplification paramétrique fait intervenir un faisceau signal S, qui interagit avec un faisceau de pompe intense P dans un milieu non-linéaire. En sortie du matériau non-linéaire, deux faisceaux sont produits : un faisceau signal S' de même fréquence que S, et un faisceau appelé *idler* I. On parle alors d'amplification car la puissance du faisceau S' peut devenir plus grande que la puissance de S. Par ailleurs, la conservation de l'énergie induit la relation

$$\omega_P = \omega_S + \omega_I$$

ce qui nous permet de définir deux cas :

Amplificateur paramétrique dépendant de la phase

L'amplificateur dépendant de la phase (*phase-sensitive amplifier*, ou PSA) est un amplificateur paramétrique optique dégénéré, c'est-à-dire dont les fréquences du signal et de l'idler sont identiques, telles que $\omega_S = \omega_I = \frac{\omega_P}{2}$. L'état généré par cette opération est un état comprimé monomode ou bi-mode, en fonction des polarisations des états d'entrée et du type de cristal non-linéaire. Dans le cadre d'un amplificateur, nous ne nous intéressons de toute façon qu'à un seul mode. Celui-ci est comprimé selon l'une de ses quadratures et anti-comprimé selon la quadrature orthogonale. Le choix de la quadrature comprimée est possible en contrôlant la phase relative entre signal et idler.

Nous pouvons résumer ce processus par la transformation

$$\begin{pmatrix} x \\ p \end{pmatrix}_{out} = \begin{pmatrix} e^s & 0 \\ 0 & e^{-s} \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix}_{in} \tag{11.1}$$

où $\sqrt{g} = e^s$ est le gain en amplitude de l'amplification. Le choix de phase correspond ici à une compression de la quadrature $\hat{P}_s$ (on parle en anglais de *squeezing*), et une amplification (ou *anti-squeezing*) de la quadrature $\hat{X}_s$.

Dans le cas d'une application qui n'utilise qu'une seule des deux quadratures de la lumière (c'est le cas pour notre protocole de cryptographie), l'amplificateur dépendant de la phase permet donc une amplification théoriquement sans bruit, c'est-à-dire conservant le rapport signal à bruit. De nombreuses implémentations ont été réalisées dans les 20 dernières années, généralement dans des conditions de laboratoire, et les gains restent modestes dans l'optique d'une amplification (3 dB en général, 10 dB au maximum) [91].

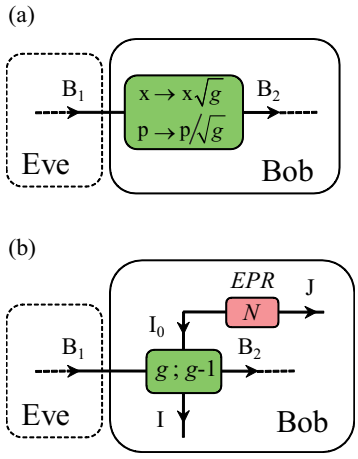


Figure 11.1 : *Modélisation d'un amplificateur (a) dépendant de la phase et (b) indépendant de la phase, placé à la sortie du canal quantique et dans le système de Bob. Les notations des modes correspondent à ceux de la figure 4.1.*

Amplificateur paramétrique indépendant de la phase

L'amplificateur indépendant de la phase (*phase-insensitive amplifier*, ou PIA) est un OPA non-dégénéré. Le signal est alors amplifié selon toutes les quadratures d'un facteur $\sqrt{g}$ en amplitude, mais couplé avec un mode de bruit. L'amplificateur est donc modélisé par la transformation

$$\hat{a}_{S,out} = \sqrt{g}\hat{a}S,in + \sqrt{g-1}\hat{a}^{\dagger}I,in \tag{11.2}$$
$$\hat{a}_{I,out} = \sqrt{g}\hat{a}I,in + \sqrt{g-1}\hat{a}^{\dagger}S,in \tag{11.3}$$
$$\tag{11.4}$$

ou alternativement par

$$\begin{pmatrix} x_S \\ p_S \\ x_I \\ p_I \end{pmatrix}_{out} = \begin{pmatrix} \sqrt{g} & 0 & \sqrt{g-1} & 0 \\ 0 & \sqrt{g} & 0 & -\sqrt{g-1} \\ \sqrt{g-1} & 0 & \sqrt{g} & 0 \\ 0 & -\sqrt{g-1} & 0 & \sqrt{g} \end{pmatrix} \begin{pmatrix} x_S \\ p_S \\ x_I \\ p_I \end{pmatrix}_{in} \tag{11.5}$$

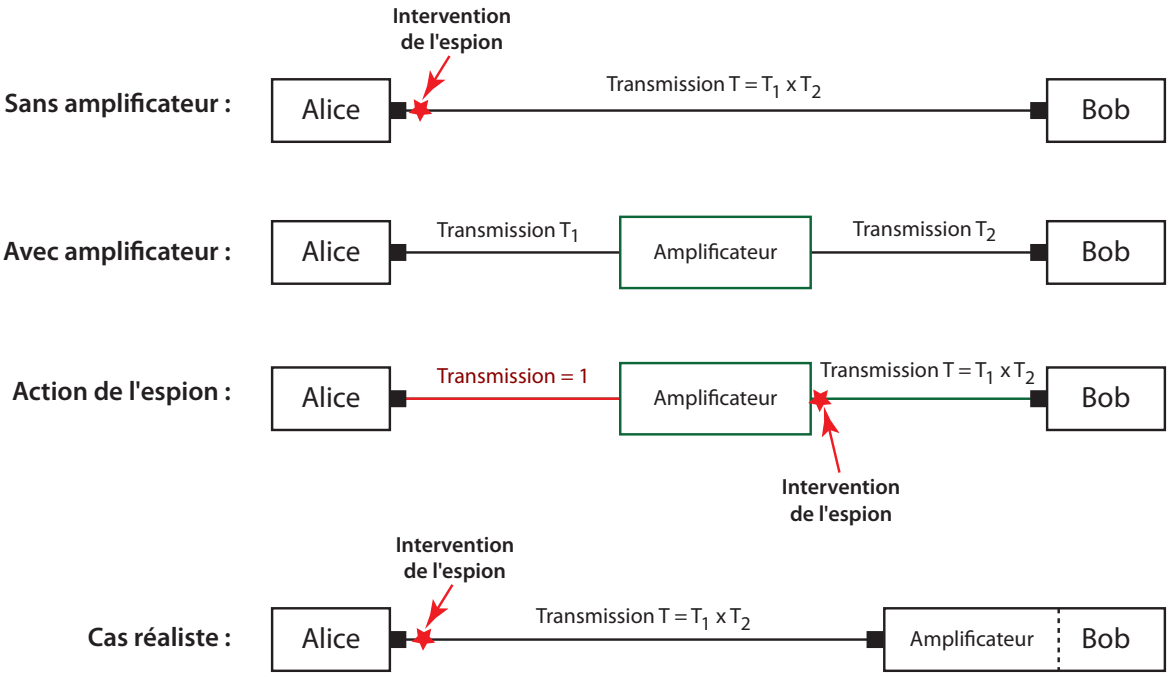
Les deux quadratures du mode signal x_S et p_S sont amplifiées d'un facteur $\sqrt{g}$, et couplées à un mode de bruit I, qui est un mode interne de l'amplificateur, au mieux égal au vide (d'où le nom d'*idler*, voir figure 11.1). En théorie, le bruit ajouté par un tel amplificateur, ramené à l'entrée de l'amplificateur, est $\frac{(g-1)V_N}{g}$, où $V_N N_0$ correspond à la variance de l'idler en entrée. V_N est égal à 1 si le mode 2 est vide (cas idéal), et le bruit ajouté par l'amplificateur tend alors vers N_0 aux forts gains. Un PIA rajoute donc au minimum 3 décibels de bruit au signal, c'est-à-dire qu'il dégrade le SNR d'un facteur 2.

Les amplificateurs à fibre dopée erbium [1] approximent bien le fonctionnement d'un PIA, en introduisant un bruit d'environ 4 à 6 dB, en fonction du gain de l'amplificateur.

1. Les amplificateurs à fibre dopée (*Doped Fibre Amplifier*, ou DFA) fonctionnent de la même façon

11.2.2 Intégration dans le système de distribution de clés

Il est conceptuellement inutile de placer un amplificateur paramétrique au milieu de la fibre de transmission, même dans un site sécurisé. En effet, nous supposons que l'espion a un contrôle total sur la fibre. Il serait donc en mesure de remplacer celle-ci par une fibre sans perte, et d'effectuer ses attaques sur la partie se trouvant après l'amplificateur. Ce cas reviendrait donc à la même sécurité qu'une transmission sans amplificateur, avec une variance de modulation différente et un bruit en excès plus grand ou égal, ce qui ne peut donc qu'être défavorable à Alice et Bob.



En revanche, dans le mode réaliste, il est possible de se placer après tout lieu d'attaque de l'espion, mais en amont des détecteurs. L'idée est alors d'ajouter un amplificateur en sortie de la fibre de transmission, qui amplifie alors le signal en entrée de Bob, de façon à ce que les pertes et bruits qui interviennent par la suite deviennent négligeables par rapport au niveau de signal en sortie de l'amplificateur.

Les caractéristiques des détecteurs homodyne et hétérodyne nous conduisent naturellement à favoriser certaines configurations. En effet, l'effet amplificateur est symétrique pour le PIA, et il semble donc plus adapté à un protocole à détection hétérodyne [2]. De même, la quadrature unique mesurée par la détection homodyne peut bénéficier de l'amplification théoriquement sans bruit du PSA. Nous explorons dans ce

qu'un laser. Une zone dopée, jouant le rôle d'un milieu à gain, est pompée optiquement par un autre laser. Quand le signal lumineux utile traverse la fibre, il stimule une émission cohérente du milieu à gain excité, et est donc amplifié. Le gain dépend de la longueur de la fibre et de l'intensité du pompage. Les amplificateur dopés à l'erbium [92, 93] sont les plus utilisés dans les télécommunications, car ils sont particulièrement efficaces dans les bandes C et L (1525-1610 nm) des fibres optiques, et peuvent être pompés avec des diodes à 980 ou 1480 nm, selon les applications visées. En pratique, le bruit de ces amplificateurs est de l'ordre de 4 à 6 décibels à fort gain, ce qui est assez proche de la valeur idéale (3 dB) autorisée par la physique quantique [94].

2. Dans tout ce chapitre, le terme *hétérodyne* est à comprendre comme une double mesure homodyne des quadratures $\hat{X}$ et $\hat{P}$, et non comme un oscillateur local de fréquence différente du signal.

chapitre ces deux configurations, et renvoyons en annexe A.3 les calculs pour les deux configurations croisées.

Détection homodyne et PSA

Nous commençons notre analyse par le cas des attaques collectives, quand un amplificateur idéal (sans bruit ajouté) est placé dans notre système de cryptographie quantique avec détection homodyne. Nous repartons de l'expression de χ_{BE} donnée dans l'équation (4.31). L'amplificateur dépendant de la phase ne rajoute pas de mode au calcul, et son insertion revient donc à modifier la valeur des matrices de covariance γ_{AB_1} et $\gamma_{AFG}^{x_B}$. Puisque l'amplificateur est placé après B_1, les expressions de γ_{AB_1} et des valeurs propres symplectiques associées $\lambda_{1,2}$ ne changent pas. Il nous faut en revanche modifier l'expression de $\gamma_{AFG}^{x_B}$. Celle-ci dérive de l'expression de γ_{ABFG}, que nous pouvons écrire :

$$\gamma_{ABFG} = (Y^{\mathrm{BS}})^T (Y^{\mathrm{PSA}})^T [\gamma_{AB_1} \oplus \gamma_{F_0 G}] Y^{\mathrm{PSA}} Y^{\mathrm{BS}} \tag{11.6}$$

Dans l'équation ci-dessus, les matrices $Y^{\mathrm{BS}}, \gamma_{AB_1}, \gamma_{F_0 G}$ sont les mêmes que dans la section 4.3.2, tandis que la matrice Y^{PSA} décrit la transformation induite par le PSA sur le mode B_1 :

$$\begin{aligned} Y^{\mathrm{PSA}} &= \mathbb{I}_A \oplus Y_{B_1}^{\mathrm{PSA}} \oplus \mathbb{I}_{F_0} \oplus \mathbb{I}_G, \text{ avec} \\ Y_{B_1}^{\mathrm{PSA}} &= \begin{bmatrix} \sqrt{g} & 0 \\ 0 & 1/\sqrt{g} \end{bmatrix} \end{aligned} \tag{11.7}$$

Le calcul des valeurs propres symplectiques $\lambda_{3,4,5}$ est alors direct, et nous obtenons un résultat similaire à celui de l'équation (4.31). La seule différence réside dans l'expression du bruit en excès dû à la détection χ_{hom} (et sa conséquence sur le bruit total en excès χ_{tot}), qui est modifiée de façon élégante pour inclure l'effet de l'amplificateur :

$$\chi_{\mathrm{hom}}^{\mathrm{PSA}} = \frac{(1-\eta) + v_{\mathrm{el}}}{g\eta} \tag{11.8}$$

Nous voyons apparaître le terme g au dénominateur, ce qui montre que l'effet du gain est de dissimuler le bruit ajouté par la détection homodyne.

Il est important de noter que nous avons implicitement supposé que Bob amplifie toujours la quadrature qu'il a choisi (aléatoirement) de mesurer. Ceci est clairement avantageux pour lui, et en théorie aisé à réaliser. Néanmoins, ceci nécessite en pratique d'effectuer un verrouillage en phase du faisceau de pompe avec l'oscillateur local de la détection homodyne, ce qui peut se révéler délicat à implémenter.

Dans le cas de l'information mutuelle entre Eve et Bob pour des attaques individuelles, ainsi que pour l'information mutuelle entre Alice et Bob dans tous les cas, nous nous apercevons aussi que les résultats (4.25) et (4.14) restent valables, en modifiant χ_{hom} et χ_{tot} comme dans (11.8).

Détection hétérodyne et PIA

Le raisonnement pour le cas de la détection homodyne et de l'amplificateur indépendant de la phase est assez similaire au cas précédent. En revanche, le modèle du

PIA nous force à ajouter 2 modes supplémentaires I et J dans les calculs, et donc à calculer l'expression de la matrice de covariance γ_{ABFGIJ} qui contient 12 colonnes et lignes. A partir de cette matrice, nous obtenons ensuite la matrice de covariance après la mesure hétérodyne, $\gamma_{AFGIJ}^{x_B,p_B}$, qui nous fournit cinq valeurs propres symplectiques $\lambda_{3,4,5,6,7}$, et une expression du taux similaire à celle de (4.31) avec deux termes supplémentaires. Pour ne pas surcharger le corps du manuscrit, et puisque l'interprétation du résultat est assez indépendante de la formule exacte du taux, nous renvoyons le lecteur à l'annexe A.1 et A.3 dans laquelle le calcul du taux pour un protocole hétérodyne, puis le calcul de ce taux avec un PIA, sont détaillés.

Le résultat du calcul se résume quant à lui très simplement. Comme dans le cas précédent, l'expression du taux pour un protocole hétérodyne dans lequel un PIA a été inséré est inchangé (par rapport au taux sans amplificateur), à une exception près : l'expression du bruit en excès dû à la détection χ_{het} doit être modifié de

$$\chi_{\text{het}}^{\text{original}} = \frac{1+(1-\eta)+2v_{\text{el}}}{\eta} \qquad \text{à :} \qquad \chi_{\text{het}}^{\text{PIA}} = \frac{1+(1-\eta)+2v_{\text{el}}+V_N(g-1)\eta}{g\eta} \tag{11.9}$$

Dans ce cas, nous voyons une nouvelle fois apparaître le terme g au dénominateur, qui tend à diminuer le bruit de la détection ; mais un terme proportionnel à $g-1$ apparaît au numérateur. Ainsi, le bruit effectif de la détection tend vers V_N quand g devient grand, c'est-à-dire qu'il tend vers le bruit présent dans le mode annexe entrant dans l'amplificateur.

Pour des attaques individuelles, les informations de Shannon sont aussi valables moyennant le changement ci-dessus.

11.2.3 Interprétation et effets pratiques

Taux secrets dérivés

Visualisons tout d'abord l'effet de ces modifications sur le taux de clé secrète. Nous avons choisi des paramètres courants pour les protocoles à variables continues : $\varepsilon = 0{,}005$; $\eta = 0{,}6$; $v_{\text{el}} = 0{,}05$. Le taux est représenté pour des valeurs de gain de 1 (équivalant à l'absence d'amplificateur), 3 ou 20, tandis que nous imposons un bruit ajouté dans le PIA de 1 (idéal) ou 1,5 (plus réaliste). La transmission du canal est quant à elle de 0,2 dB/km.

Dans les simulations suivantes, nous utilisons le SNR comme paramètre ajustable, en fonction duquel nous optimisons le taux de génération de clé secrète pour chaque distance. Ainsi, nous trouvons pour chaque longueur de fibre la variance de modulation qui maximise le taux secret, en remarquant que celle-ci est reliée au SNR par :

$$\begin{aligned} I_{AB} &= \frac{1}{2}\log_2(1+\text{SNR}) = \frac{1}{2}\log_2\frac{V+\chi_{\text{tot}}}{1+\chi_{\text{tot}}} \\ &= \frac{1}{2}\log_2\left(1+\frac{V_A}{1+\chi_{\text{tot}}}\right) \rightarrow V_A = \text{SNR}(1+\chi_{\text{tot}}) \end{aligned} \tag{11.10}$$

L'optimisation est réalisée pour des SNR dans l'intervalle [0,5 ; 15], qui correspondent à des valeurs typiquement accessibles dans notre système.

En plus des paramètres ci-dessus, nous devons prendre en compte l'efficacité de réconciliation β, qui dégrade le taux de clé secrète. L'efficacité dépend du rapport signal

à bruit ; grâce à une simulation numérique, nous avons approximé cette dépendance par :

$$\beta = \frac{\log(1 + \mathrm{SNR}^{1,2})}{1{,}29 \log(1 + \mathrm{SNR})} + 0{,}02 \tag{11.11}$$

Cette fonction reflète le fait qu'une efficacité élevée est possible en travaillant à haut SNR, alors que l'efficacité décroît très vite pour des rapports signal à bruit faibles. Dans l'intervalle de SNR que nous utilisons, β croît avec le SNR, et le rapport I_{BE}/I_{AB} croît lui aussi (voir figure 10.1). Par conséquent, l'évolution du taux secret ΔI avec le SNR n'est pas monotone : pour trouver le taux secret optimal à une distance donnée, il faut donc réaliser une optimisation non-triviale du SNR.

Les résultats de simulation sont donnés dans les figures 11.2 et 11.3 pour les deux cas présentés dans cette section.

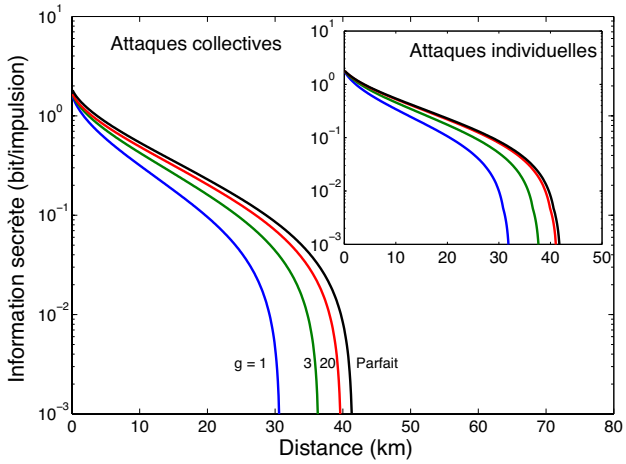


Figure 11.2 : *Taux de génération de clé secrète, en fonction de la distance, pour un protocole à* détection homodyne *et* amplificateur dépendant de la phase*, pour des attaques collectives et individuelles. Les courbes « parfait » correspondent à un détecteur parfait, en l'absence d'amplificateur.*

Interprétation

Nous remarquons que dans les deux cas, l'effet de l'amplificateur sur le taux secret est globalement le même pour les taux individuels et collectifs : le taux et la distance limite augmentent quand le gain de l'amplification augmente. Au contraire, une augmentation du bruit du PIA diminue logiquement le taux de clé secrète.

En observant les expressions modifiées des excès de bruit des détecteurs, nous pouvons affiner l'interprétation. Pour le cas de la détection homodyne et du PSA, nous voyons que le bruit ajouté tend vers zéro quand le gain de l'amplificateur augmente, quels que soient les défauts de la détection. Ainsi, un amplificateur paramétrique dépendant de la phase permet de pré-compenser les imperfections du détecteur homodyne, c'est-à-dire que du point de vue du système leur combinaison est équivalente à

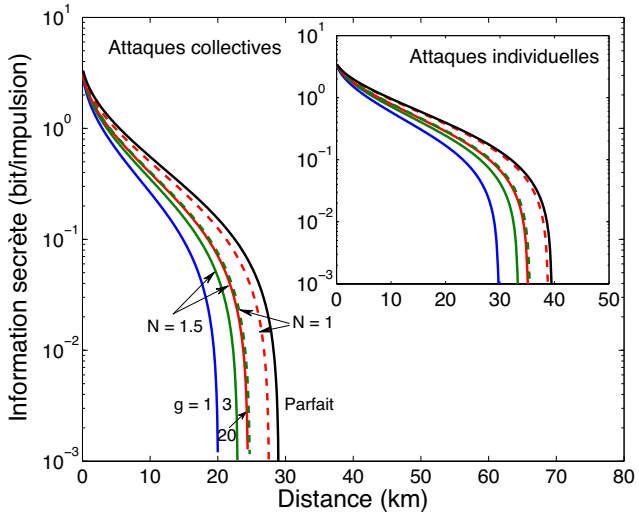


Figure 11.3 : *Taux de génération de clé secrète, en fonction de la distance, pour un protocole à* détection hétérodyne *et* amplificateur indépendant de la phase, *pour des attaques collectives et individuelles. L'amplificateur peut être optimal ($V_N = 1$: pointillés) ou réaliste ($V_N = 1{,}5$: trait plein). Les courbes « parfait » correspondent à un détecteur parfait, en l'absence d'amplificateur.*

---

un détecteur parfait. En effet, la figure 11.2 montre que, pour des g grands, le taux de clé secrète tend vers la courbe correspondant à un détecteur parfait.

Pour le cas de la détection hétérodyne et du PIA, l'expression de χ_{het} tend vers V_N dans la limite des forts gains. Nous pouvons alors modéliser la combinaison de l'amplificateur et du détecteur comme un détecteur hétérodyne ne présentant pas de bruit ajouté, mais des pertes η' telles que $[1+(1-\eta')]/\eta' = V_N$. Pour un PIA parfait, $V_N = 1$, ce qui induit $\eta' = 1$. Ainsi, un amplificateur paramétrique indépendant de la phase peut compenser toutes les imperfections d'un détecteur hétérodyne, de la même façon qu'un PSA pour une détection homodyne.

Pour un PIA plus réaliste, tel que $V_N > 1$, l'effet sur les performances du système dépend de la qualité *a priori* du détecteur. En particulier, si V_N devient plus grand que le bruit total ajouté par le détecteur, l'amplificateur dégrade les performances du système. Pour les paramètres que nous avons considérés, le seuil est d'environ $V_N = 2{,}5$.

D'un point de vue pratique, il est aussi intéressant de comparer directement les performances des protocoles homodyne et hétérodyne dans les configurations ci-dessus. La figure 11.4 illustre cette comparaison : nous avons effectué un calcul pour les deux cas homodyne et hétérodyne, en considérant des systèmes avec les mêmes imperfections. Pour un système idéal, les courbes homodyne et hétérodyne sont presques identiques, mais en présence d'imperfections, nous voyons que le protocole hétérodyne a une distance limite plus faible que le protocole homodyne. Cet avantage du protocole homodyne, en termes de distance, est particulièrement prononcé dans le cas des attaques collectives.

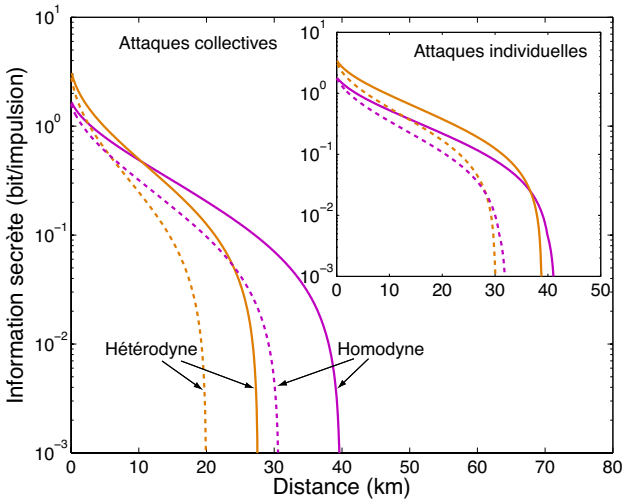


Figure 11.4 : *Taux de génération de clé secrète, en fonction de la distance, pour les deux configurations étudiées dans ce chapitre. Les courbes en trait plain correspondent aux taux avec amplification de gain 20 et de bruit minimal ($V_N = 1$), les courbes en pointillés aux taux sans amplification.*

11.3 Amplificateur non-déterministe

11.3.1 Contexte et idée

Nous venons de voir que l'apport d'un amplificateur paramétrique au taux secret est relativement faible, bien qu'existant. En particulier, quelle que soit la qualité de l'amplificateur, le taux secret ne pourra jamais dépasser le taux équivalant à un détecteur parfait. Ceci est directement dû aux propriétés de bruit des amplificateurs paramétriques, qui ne peuvent pas augmenter le rapport signal à bruit.

De fait, il est tentant d'essayer de concevoir un nouveau schéma d'amplification, qui ne rajoute pas (ou peu) de bruit sur un signal tout en amplifiant son amplitude moyenne, c'est-à-dire qui augmente son rapport signal à bruit. Malheureusement, la physique quantique interdit de réaliser une telle opération de façon déterministe[3]. La démonstration de cette impossibilité peut être réalisée de différentes façons, et elle revient fondamentalement à démontrer qu'une telle opération violerait le théorème de non-clonage. Nous utilisons ici la démonstration élégante proposée par Ralph et Lund dans [90], qui effectuent un raisonnement sur les états cohérents :

3. Ce n'est en fait pas si malheureux, car si ceci était possible, nos protocoles de cryptographie quantique ne fonctionneraient plus...

- Suppposons qu'il existe une transformation unitaire $\hat{T}$, qui permette d'amplifier un état $|\alpha\rangle$ en un état $|g\alpha\rangle$, $g \in \mathbb{C}^*$
- A partir de l'opérateur annihilation $\hat{a}$, nous définissons l'opérateur $\hat{b} = \hat{T}\hat{a}\hat{T}^\dagger$. Nous avons alors : $\hat{b}\,|g\alpha\rangle = (\hat{T}\hat{a}\hat{T}^\dagger)(\hat{T}\,|\alpha\rangle) = \hat{T}\hat{a}\,|\alpha\rangle = \alpha\,|g\alpha\rangle$.
  $|g\alpha\rangle$ est donc vecteur propre de $\hat{b}$ avec la valeur propre α. Par conséquent, nous pouvons écrire $\hat{b} = \frac{1}{g}\hat{a}$, et donc $\left[\hat{b}\,,\,\hat{b}^\dagger\right] = \frac{1}{g^2}$.
- Or, un calcul simple nous donne aussi $\left[\hat{b}\,,\,\hat{b}^\dagger\right] = \hat{T}\left[\hat{a}\,,\,\hat{a}^\dagger\right]\hat{T}^\dagger = 1$.
- **Conclusion :** une telle opération n'est possible que si $|g| = 1$, c'est-à-dire s'il n'y a pas d'amplification.

La façon habituelle de résoudre le problème est de rajouter un mode de bruit, à la façon d'un amplificateur paramétrique indépendant de la phase. Au contraire, l'interprétation de [90] consiste à considérer que l'amplification sans bruit est possible, sous réserve que $\hat{T}$ ne soit pas unitaire, mais plutôt une opération projective non-déterministe. L'amplification sans bruit peut alors fonctionner, mais avec une probabilité de réussite $p \leqslant \frac{1}{|g|^2}$ pour respecter les lois de la physique quantique.

Ainsi, dans la mesure où nous pouvons tout à fait accepter de perdre des impulsions entre Alice et Bob, ce type d'amplificateur peut être utilisé dans les mêmes conditions que les amplificateurs paramétriques pour améliorer notre protocole de cryptographie quantique.

11.3.2 Modèle théorique d'amplification

Amplification à N étages

Le principe de l'amplificateur proposé par Ralph et Lund est le suivant : l'état à amplifier $|\alpha_{\text{in}}\rangle$ est tout d'abord séparé en N états cohérents $\alpha/\sqrt{N}$ dans un N-splitter. Chacun de ces petits états cohérents entre ensuite dans un « étage » de l'amplificateur, qui est représenté dans la figure 11.5 et décrit ci-dessous.

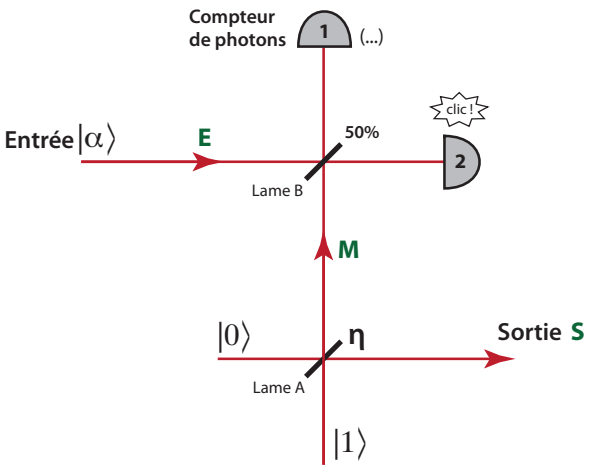


Figure 11.5 : *Schéma d'un étage de l'amplificateur proposé par Ralph et Lund dans [90].*

Dans un premier temps, un état de Fock à un photon $|1\rangle$ interfère avec un état vide $|0\rangle$ dans la lame séparatrice A, de transmission η. L'une des sorties de la lame constitue la sortie de l'amplificateur. L'autre sortie interfère avec l'état cohérent entrant dans l'étage, dans la lame B de transmission 1/2. Les deux sorties de la lame B sont finalement envoyées sur des compteurs de photons. Ce schéma est très similaire au protocole de troncature d'état, aussi appelé ciseaux quantiques (*quantum scissors*), qui a été introduit dans [95]. La différence réside dans la transmission de la lame A, qui n'est pas de 50 %.

Avant détection, nous disposons d'un état intriqué entre trois modes : la sortie libre de la lame A et les deux sorties de la lame B. Après la détection, cet état est projeté en fonction du résultat de mesure, ce qui conditionne le mode de sortie de l'amplificateur. Nous nous intéressons au résultat suivant : un photon détecté sur la photodiode 2 et aucune détection sur la photodiode 1. Dans ce cas, un calcul simple montre que l'état du mode de sortie s'écrit :

$$|\psi\rangle = e^{-\frac{\alpha^2}{2}}\sqrt{\frac{\eta}{2}}\left[|0\rangle + g\frac{\alpha}{\sqrt{N}}|1\rangle\right] \tag{11.12}$$

où $g = \sqrt{\frac{1-\eta}{\eta}}$.[4]

Pour finir, dans le cas où tous les étages ont fourni un bon résultat de mesure, les N états de sortie des étages sont recombinés dans un N-splitter. Un nouveau conditionnement est réalisé sur $N-1$ des N sorties : aucun photon ne doit être détecté sur les $N-1$ sorties. Sous toutes ces conditions, et si N est suffisamment grand, l'état de sortie est proportionnel à l'état $|g\alpha\rangle$: nous avons alors un amplificateur parfait.

Amplification à 1 seul étage

Le cas « N grand » n'est pas très intéressant en pratique : son implémentation nécessite $3N-1$ compteurs de photons, qui doivent tous donner un résultat adéquat en même temps, ce qui impose une probabilité de succès diminuant exponentiellement avec N.

Dans le reste de ce chapitre, nous nous concentrerons sur le cas d'un amplificateur à un seul étage, et donc deux compteurs. Dans ce cas, l'état de sortie est proportionnel à $|0\rangle + g\alpha|1\rangle$, ce qui correspond aux deux premiers termes de la décomposition de $|g\alpha\rangle$ dans la base de Fock. De fait, si les termes supérieurs sont négligeables, c'est-à-dire si $g\alpha$ est petit devant 1, l'état de sortie est très proche d'un état cohérent d'amplitude $g\alpha$.

11.3.3 Modélisation des imperfections expérimentales

La qualité de l'état de sortie de l'amplificateur dépend de notre capacité à produire des états à un photon corrects, ainsi que de la qualité des détecteurs : capacité à discriminer le nombre de photons, bruit ajouté et efficacité.

4. En réalité, nous pouvons aussi considérer le cas « clic sur 1, et rien sur 2 », qui conduit à un état de la forme $|0\rangle - g\alpha|1\rangle$. Un modulateur de phase peut alors corriger la phase de l'état de sortie. Ceci nous permet alors de gagner 2^N sur la probabilité de succès de l'amplificateur.

Détecteurs : nous notons λ l'efficacité de détection, que nous supposons égale pour les deux détecteurs. Compte tenu des différents filtres placés avant les détecteurs[5], celle-ci est de l'ordre de 15 %. Les détecteurs présentent environ 200 coups d'obscurité par seconde ; en synchronisant la détection avec le laser, nous sommes en mesure de définir une fenêtre de détection de 20 ns. Ces paramètres fournissent une probabilité de coups d'obscurité de 4.10^{-6}, suffisamment petite pour pouvoir la négliger par la suite. Enfin, nous ne disposons pas de détecteurs en mesure de compter le nombre de photons dans les impulsions, et nous devons le prendre en compte dans le calcul.

Etat $|\mathbf{1}\rangle$: nous retenons deux défauts principaux. D'un côté, l'état n'est pas parfaitement pur : il s'agit en fait d'un mélange statistique du type $p_0 |0\rangle\langle 0| + p_1 |1\rangle\langle 1| + p_2 |2\rangle\langle 2|$, où $p_0 = 0{,}2$; $p_1 = 0{,}7$; $p_2 = 0{,}1$. Par ailleurs, l'adaptation de modes spatio-temporels entre cet état et l'état cohérent entrant dans l'amplificateur n'est pas parfaite, et nous définissons un paramètre $\kappa \leqslant 1$ correspondant au recouvrement entre leurs deux enveloppes. Il vaut typiquement 0,85.

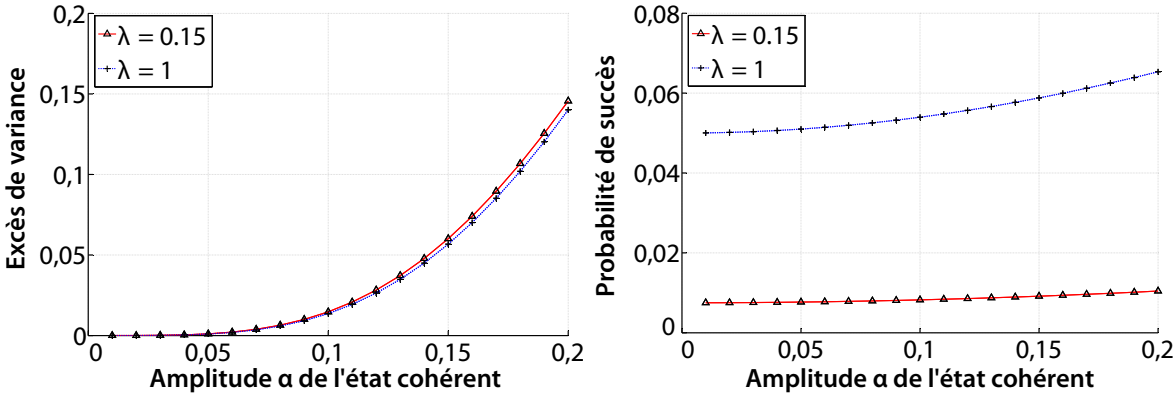
11.3.4 Propriétés de l'état de sortie

Nous ne détaillerons pas dans le corps du manuscrit le détail de notre calcul des états sortant de l'amplificateur, car il est assez fastidieux et difficile à interpréter directement. Nous présenterons plutôt dans cette partie des courbes caractéristiques de ces états.

Notons d'abord que, compte tenu des amplitudes des états d'entrée pour lesquelles l'amplificateur fonctionne (autour de 0,1), il semble parfaitement convenir aux paramètres du protocole à 4 états, pour lequel le SNR au niveau du détecteur doit être petit. Ainsi, dans l'optique d'une application à la cryptographie quantique, le paramètre qui nous intéresse particulièrement est l'excès de bruit de l'état de sortie de l'amplificateur, calculé par rapport à un état cohérent de même amplitude (d'excès de bruit nul). Puisque les 4 états du protocole quaternaire sont placés à 45 ° des quadratures X et P, les excès de bruit que nous considérons correspondent aux bruits à 45 ° des axes des quatre états. La probabilité de succès de l'amplificateur est elle aussi importante, car le taux secret sera directement diminué par cette probabilité. Nous avons donc tracé, pour les différents défauts de l'amplificateur, ces deux paramètres. À noter : dans les cas qui suivent, nous considérons que les détecteurs ne discriminent pas le nombre de photons dans l'impulsion, et nous fixons le gain théorique $g = \sqrt{\frac{1-\eta}{\eta}}$ à 3.

5. Les paramètres numériques de cette section correspondent aux paramètres expérimentaux de l'expérience de génération d'états non-classiques menée dans le groupe d'Optique Quantique du LCFIO, présentés dans la thèse d'Alexei Ourjoumtsev. [96]

Cas 1 : Photodiodes inefficaces

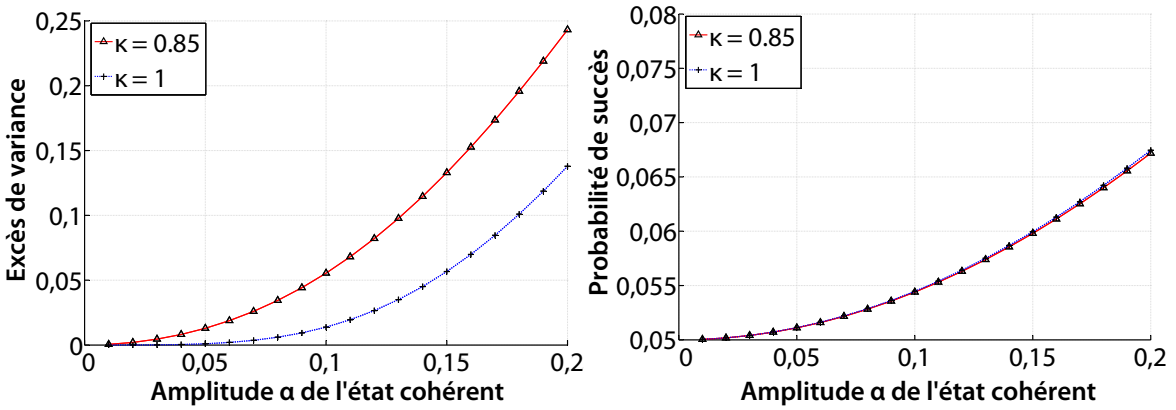


Un défaut n'a un effet sur l'excès de bruit que s'il induit une validation du résultat des détecteurs alors que le nombre de photons dans les voies avant détection n'était pas le bon : par exemple, deux photons sont présents dans la voie 2 mais le détecteur 2 ne peut en détecter qu'un ; ou alors, un photon est présent dans la voie 1, mais le détecteur 1 ne le détecte pas (et renvoie donc « 0 »). Dans ces deux cas, l'état de sortie est mauvais, et il augmente donc l'excès de bruit.

Le cas de l'inefficacité des détecteurs est de ce point de vue significatif : nous voyons que malgré une efficacité basse, l'effet sur l'excès de bruit est faible. Ceci peut être interprété de la façon suivante : d'une part, l'état d'entrée ne contient en moyenne que 0,005 photon, pour $\alpha \approx 0{,}1$). Sachant qu'un photon au plus arrive de l'autre voie de la lame B, il est très peu probable que deux photons soient présents dans la voie 2. D'autre part, la probabilité que la voie 1 contienne un photon, sachant que la voie 2 doit en contenir un, est tout aussi faible.

Ainsi, les détecteurs ne « cliquent » correctement que très rarement quand l'efficacité est basse, mais quand ils cliquent, la configuration est la bonne avec une très bonne probabilité. L'excès de bruit est donc faible.

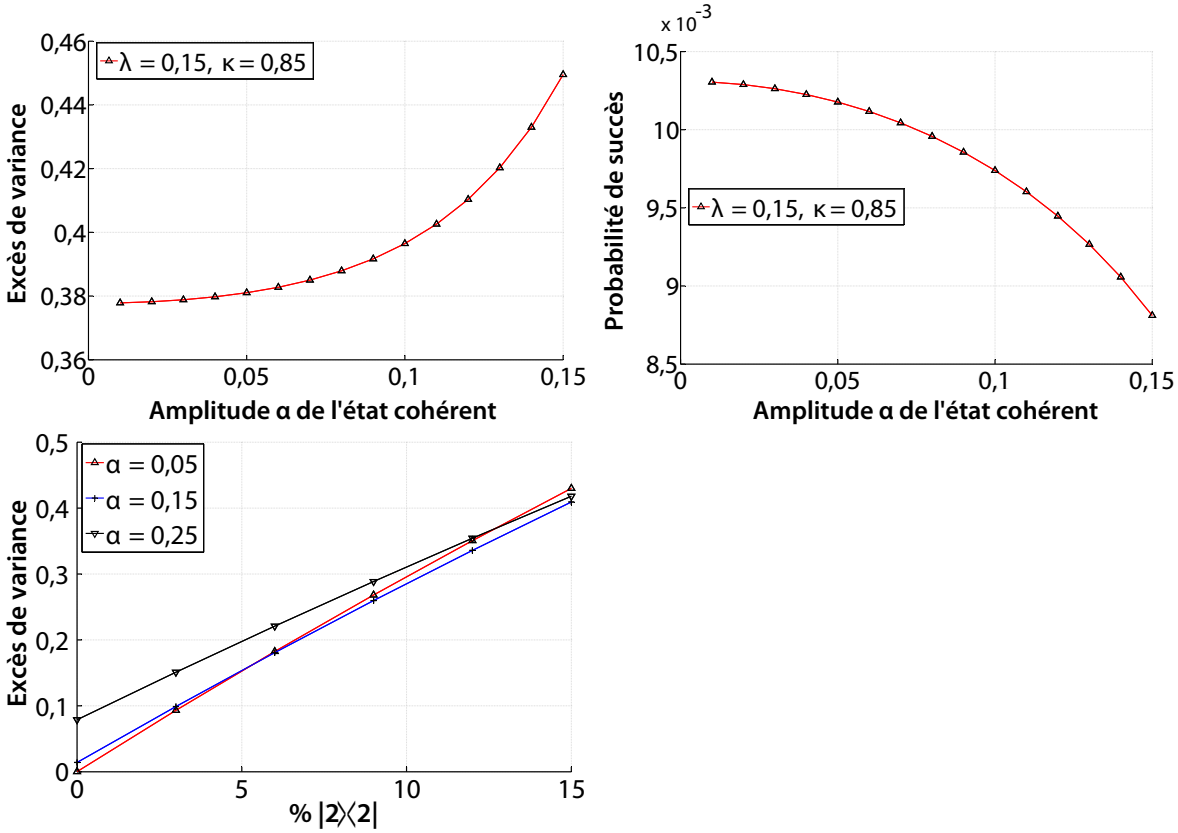
Cas 2 : Modes mal adaptés



Dans le cas d'une mauvaise adaptation de modes, l'état de sortie n'est plus un état pur ; il est donc décrit par une matrice densité (ne contenant que des termes en 0 ou 1). Le paramètre alors pertinent est la corrélation en $|0\rangle\langle 1|$: plus celle-ci s'éloigne de celle d'un état pur ($= \frac{g\alpha}{1+g^2\alpha^2}$), plus l'excès de bruit associé devient grand. L'effet d'une mauvaise adaptation de modes est alors la diminution du gain effectif d'amplification,

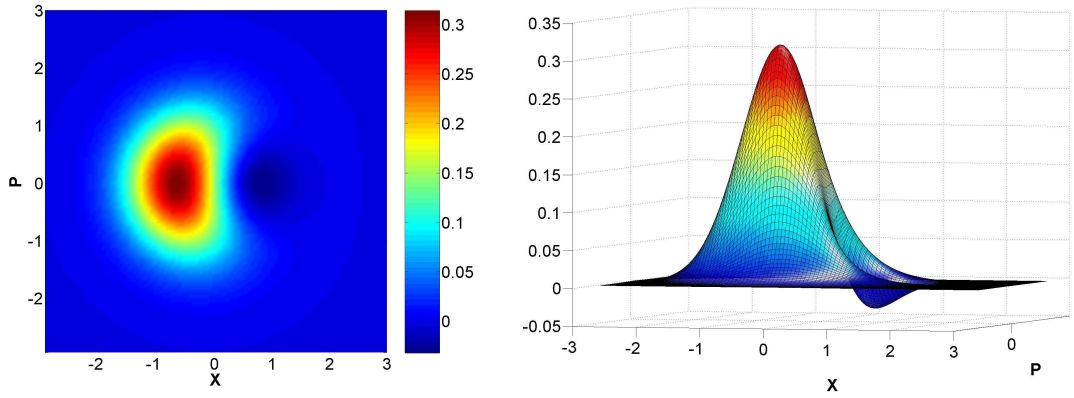
et une augmentation significative de l'excès de bruit.

Cas 3 : Amplificateur réaliste



Nous considérons enfin des paramètres réalistes pour l'amplificateur ; en particulier, l'imperfection de la génération de $|1\rangle$, qui contient une part non-négligeable de $|2\rangle\langle 2|$. Cette part d'état à deux photons est très problématique : en effet, jusqu'ici, si l'état d'entrée était vide, au plus un photon pénétrait dans la lame B, et l'excès de bruit était alors nul. Dans la configuration réaliste, il est tout à fait possible, par exemple, que deux photons soient présents dans la voie 2 alors même qu'aucun état ne rentre dans l'amplificateur ; ou bien qu'un photon soit dans la voie 1 (situation idéale), mais qu'un photon soit présent en même temps dans la voie de sortie, ce qui est anormal.

Dans ce cas, le gain est fixé à $g = 2$. Pour des imperfections réalistes, nous voyons apparaître un bruit *plancher*, présent même en l'absence de signal, qui est largement non-négligeable (38 % de N_0 pour un gain de 2). Nous voyons en fait sur la troisième courbe que l'excès de bruit dépend principalement de la proportion de $|2\rangle\langle 2|$. Les figures suivantes montrent la forme de la fonction de Wigner sortant de l'amplificateur. Celle-ci reste malgré tout assez proche d'un état cohérent amplifié, mais possède un « trou » qui dénote l'imperfection de l'état.



11.3.5 Application à la cryptographie quantique

Comparaison avec un amplificateur paramétrique

Dans les exemples précédents, nous avons travaillé avec des gains $g = 2$ ou 3. Pour un amplificateur paramétrique, de tels gains induisent nécessairement des bruits ajoutés supérieurs à $g^2 - 1$, c'est-à-dire respectivement 3 N_0 et 8 N_0. Dans notre cas réaliste, le bruit pour un gain de 2 est de 0,4 N_0, c'est-à-dire plus de sept fois inférieur au bruit minimal autorisé par la mécanique quantique pour une amplification déterministe. Notons que la probabilité de succès est alors seulement d'environ 1 %.

Il est de même assez dérangeant de considérer l'évolution du SNR de l'ensemble des impulsions ayant été amplifiées. Celui-ci passe de $\frac{0{,}01}{1} = 10^{-2}$ à $\frac{0{,}04}{1{,}4} = 2{,}9.10^{-2}$: il a donc été augmenté d'un facteur proche de 3. L'information mutuelle disponible à Alice et Bob *par impulsion amplifiée* augmente alors d'un facteur 3 ; mais il faut garder en mémoire que l'information mutuelle sur l'ensemble des impulsions est bien plus petite, car nous perdons 99 % des impulsions (puisque la probabilité de succès est de l'ordre de 1 %).

Information secrète

Comme pour l'amplificateur paramétrique, il serait alors intéressant de déterminer l'information secrète pouvant être extraite des impulsions si nous plaçons l'amplificateur en sortie de canal. Puisque l'amplificateur améliore le rapport signal à bruit avant la mesure de Bob, et qu'il n'est pas accessible à l'espion, nous pouvons imaginer qu'il permette à Alice et Bob d'améliorer leur avantage sur Eve.

Si nous considérons que le canal quantique ne présente que des pertes et pas d'excès de bruit, l'état entrant dans l'amplificateur est un état cohérent, et nous savons alors calculer l'état de sortie de l'amplificateur. En revanche, si le canal présente de l'excès de bruit, l'état d'entrée n'est plus un état cohérent, et le calcul devient d'autant plus difficile que le bruit en excès provient de l'espion, et que nous n'avons *a priori* pas d'information sur celui-ci (hormis sa variance, que nous mesurons).

Quoi qu'il en soit, pour pouvoir établir une preuve de sécurité, nous devons utiliser une description de type *intrication virtuelle*. Or, jusqu'ici, toutes les opérations que nous avons réalisées (pertes, bruit, amplificateur paramétrique) ont pu être décrites par des processus déterministes et des opérations unitaires ; ce n'est pas le cas de l'amplificateur non-déterministe ! A l'heure actuelle, nous n'avons pas trouvé de façon

vraiment satisfaisante pour intégrer au calcul de sécurité le processus non-déterministe. Par exemple, nous pourrions essayer d'intégrer directement dans l'expression de χ_{hom} les paramètres de l'amplificateur non déterministe, ce à quoi se ramenaient les calculs avec amplificateurs paramétriques :

$$\chi_{\mathrm{hom}}^{\mathrm{nd}} = \frac{1 + (1-\eta) + 2v_{\mathrm{el}} + \varepsilon_{\mathrm{nd}}(g-1)\eta}{g\eta} \tag{11.13}$$

Ceci étant, d'une part, l'application numérique nous donne parfois des valeurs négatives de $\chi_{\mathrm{hom}}^{\mathrm{nd}}$. Ces valeurs négatives peuvent se comprendre qualitativement par l'effet d'augmentation du SNR, qui n'est pas habituel. D'autre part, pour certaines valeurs, $\chi_{\mathrm{tot}}^{\mathrm{nd}} = \chi_{\mathrm{line}} + \chi_{\mathrm{hom}}^{\mathrm{nd}}$ est lui aussi négatif, et ceci devient vraiment étonnant...

Pour résumer, il est assez clair que l'amplification non-déterministe est une voie très prometteuse qui sort du cadre des systèmes étudiés habituellement en mécanique quantique, et il est probable que des applications inattendues d'un tel amplificateur apparaissent dans les années à venir. En particulier, son utilisation dans un dispositif de cryptographie quantique semble intéressante, mais il reste encore à en effectuer une étude détaillée.

Bilan et perspectives

Nous avons élaboré au cours de cette thèse un démonstrateur complet de cryptographie quantique, dans lequel l'information est codée sur des variables continues de la lumière, à savoir les quadratures du champ électromagnétique. Le système, entièrement composé de composants télécom standard, permet de générer entre deux parties distantes (Alice et Bob) une clé partagée, dont la sécurité peut être démontrée par la théorie de l'information.

Le système est composé d'un montage optique fibré permettant à Alice de générer des états cohérents de la lumière grâce à une diode laser impulsionnelle, puis de moduler les quadratures de ces états suivant une distribution gaussienne. Elle transmet ensuite ces états à Bob par une fibre optique monomode. La mesure de l'information par Bob est réalisée grâce à une détection homodyne fonctionnant en régime impulsionnel et limitée par le bruit de photon, fonctionnant à un taux de répétition de 500 kHz.

Nous avons par ailleurs développé un logiciel complet de gestion de la distribution de clé, qui assure d'une part la gestion de la transmission quantique et les rétrocontrôles nécessaires pour la stabilité de celle-ci, et qui permet d'autre part l'extraction d'une clé secrète à partir des données continues partagées par Alice et Bob après la transmission quantique. En particulier, nous avons implémenté un ensemble d'algorithmes comprenant des techniques de correction d'erreurs et d'amplification de confidentialité, et optimisé leur fonctionnement pour améliorer le taux de génération de clé secrète.

Enfin, le système a été intégré dans des boîtiers rackables, ce qui permet un transport aisé du matériel. Nous avons en particulier effectué une démonstration en vraie grandeur, dans le cadre du réseau de cryptographie quantique mis en place dans le cadre du projet européen SECOQC, sur des fibres installées dans la ville de Vienne (Autriche). Lors de cette démonstration, notre système a fonctionné en continu pendant 57 heures, à un taux moyen de 8 kbit/s sur 9 km (3,2 dB d'atténuation).

D'autre part, nous avons exploré les différentes possibilités d'amélioration du protocole. Tout d'abord, nous avons implémenté sur le système un nouveau protocole se distinguant par une modulation discrète des quadratures. Ce protocole peut théoriquement permettre de prolonger la distance maximale de transmission de plusieurs dizaines de kilomètres ; nos résultats préliminaires montrent que, sous réserve de pouvoir limiter les effets de fluctuation statistique, une distance de 50 km pourrait être atteinte.

Nous avons aussi étudié la possiblité d'amplifier le signal à la sortie de la fibre de transmission, de façon à améliorer le niveau de signal sur les détecteurs de Bob.

Nous avons tout d'abord considéré le cas d'amplificateurs paramétriques optiques, et montré que ceux-ci permettaient de compenser les défauts du détecteur de Bob, ce qui permet d'augmenter à la fois le taux de distribution de clé, et la distance de sécurité. Nous avons ensuite étudié le cas d'un amplificateur non-déterministe (introduit récemment dans [90]), c'est-à-dire un système permettant l'amplification d'un état quantique en augmentant son rapport signal à bruit, mais ne fonctionnant qu'avec une certaine probabilité de succès. Cet effet pourrait probablement permettre à Alice et Bob d'augmenter leur avantage sur l'espion.

Dans les années à venir, plusieurs perspectives s'offrent à la cryptographie à variables continues. La vitesse de distribution de clé est toujours largement limitée par la puissance de calcul nécessaire aux algorithmes classiques d'extraction de clé. Il s'agit néanmoins d'un problème classique lors d'une étape de prototypage, et plusieurs solutions sont envisageables (par ex. informatique dédiée, ou calcul parallèle). Le taux de répétition optique, lui aussi, peut encore être significativement amélioré.

Des efforts importants doivent encore être consacrés à la stabilité du système, et aux problématiques d'évaluation des paramètres du système. Ceux-ci sont les garants de la sécurité du protocole de distribution de clés secrètes, et doivent donc être aussi précis que possible.

Enfin, il reste encore une marge importante d'amélioration des protocoles eux-mêmes ; quelques pistes ont été explorées dans ce manuscrit, et il est probable que de nouvelles idées apparaissent dans les années à venir.

Annexe

A Encore plus de taux secrets...

Dans cette annexe, nous explicitons l'expression des informations secrètes d'un certain nombre de cas pratiques que nous n'avons détaillés dans le corps du manuscrit : cas non-réaliste, réconciliation directe, protocole hétérodyne et assemblages « non-canoniques » de couples détecteur-amplificateur.

A.1 Protocole à détection hétérodyne

L'information mutuelle entre Alice et Bob correspond au double de l'information obtenue par une détection homodyne :

$$I_{AB}^{\text{het}} = 2 \times \frac{1}{2} \log_2 \frac{V_B}{V_{B|A}} = \log_2 \frac{V + \chi_{\text{tot}}}{1 + \chi_{\text{tot}}} \tag{A.1}$$

avec $V_B = \eta T(V + \chi_{\text{tot}})/2$, $V_{B|A} = \eta T(1 + \chi_{\text{tot}})/2$, et où χ_{tot} prend l'expression adaptée à une détection hétérodyne, à savoir :

$$\chi_{\text{het}} = \frac{1 + (1 - \eta) + 2v_{\text{el}}}{\eta}, \text{ et } \chi_{\text{tot}} = \chi_{\text{line}} + \frac{\chi_{\text{het}}}{T} \tag{A.2}$$

Nous exprimons maintenant l'information accessible à l'espion.

A.1.1 Cas individuel

L'information I_{BE} s'écrit $I_{BE}^{\text{het}} = \log_2(V_B/V_{B|E})$. Une borne sur $V_{B|E}$, quand Eve est autorisée à avoir accès au détecteur de Bob (mode non-réaliste), a été calculée dans [97, 98] :

$$V_{B|E} = \frac{1}{2}\left(\frac{V\chi_E + 1}{V + \chi_E} + 1\right)$$

avec

$$\chi_E = \frac{T(2 - \varepsilon)^2}{(\sqrt{2 - 2T + T\varepsilon} + \sqrt{\varepsilon})^2} + 1$$

Pour étendre cette expression au cas réaliste, nous devons prendre en compte le fait que le détecteur n'est pas accessible à Eve, et n'ajoute que du bruit non-corrélé

au signal. Les commutateurs signal-bruit intervenant dans l'expression sont donc tous nuls, et le calcul est équivalent à corriger les différentes variances intervenant dans les expressions, de façon à prendre en compte les paramètres de la détection. Nous trouvons alors

$$V_{B|E} = \eta \frac{\frac{V\chi_E+1}{V+\chi_E} + \chi_{\text{het}}}{2} \tag{A.3}$$

et donc

$$I_{BE}^{\text{het}} = \log_2 \frac{V_B}{V_{B|E}} = \log_2 \frac{T(V+\chi_{\text{tot}})(V+\chi_E)}{V\chi_E + 1 + \chi_{\text{het}}(V+\chi_E)} \tag{A.4}$$

A.1.2 Cas collectif

Le raisonnement est exactement le même que dans le cas homodyne : la seule différence vient de la transformation symplectique « mesure hétérodyne » qui doit être appliquée à la matrice γ_{ABFG}, de cette façon :

$$\gamma_{AFG}^{x_B,p_B} = \gamma_{AFG} - \sigma_{AFG;B}^T(\gamma_B + \mathbb{I}_2)^{-1}\sigma_{AFG;B} \tag{A.5}$$

Le calcul est plus ardu mais tout aussi direct, et nous trouvons finalement :

$$\boxed{\chi_{\text{BE}} = G\left(\frac{\lambda_1 - 1}{2}\right) + G\left(\frac{\lambda_2 - 1}{2}\right) - G\left(\frac{\lambda_3 - 1}{2}\right) - G\left(\frac{\lambda_4 - 1}{2}\right)} \tag{A.6}$$

avec

$$\begin{aligned}
\lambda_{1,2}^2 &= \frac{1}{2}\left(A \pm \sqrt{A^2 - 4B}\right) \qquad \lambda_{3,4}^2 = \frac{1}{2}\left(C \pm \sqrt{C^2 - 4D}\right) \\
A &= V^2(1-2T) + 2T + T^2(V+\chi_{\text{line}})^2 \qquad B = T^2(V\chi_{\text{line}}+1)^2 \\
C_{\text{het}} &= \frac{A\chi_{\text{het}}^2 + B + 1 + 2\chi_{\text{het}}\left(V\sqrt{B} + T\left(V+\chi_{\text{line}}\right)\right) + 2T\left(V^2-1\right)}{\left(T\left(V+\chi_{\text{tot}}\right)\right)^2} \\
D_{\text{het}} &= \left(\frac{V+\sqrt{B}\chi_{\text{het}}}{T(V+\chi_{\text{tot}})}\right)^2
\end{aligned} \tag{A.7}$$

Seuls les expressions de C et D changent donc par rapport au cas homodyne, ce qui est assez logique dans la mesure où seuls $\lambda_{3,4}$ proviennent d'une matrice faisant intervenir la détection.

A.2 Cas non-réaliste

Dans tout ce manuscrit, nous avons effectué l'hypothèse dite réaliste, supposant que le détecteur de Bob n'est pas contrôlé par Eve. Il est néanmoins possible de dériver les taux secrets d'un cas plus contraignant, que nous avions appelé « paranoïaque » dans un certain nombre d'articles. Néanmoins, cette dénomination laisse entendre que le cas réaliste ne serait pas totalement sûr, ce qui est faux sous réserve que les bruits et pertes réalistes sont bien évalués. Nous préférerons donc la dénomination « non-réaliste », ce qui renvoie plus directement au fait que l'hypothèse de contrôle du détecteur par Eve ne correspond pas à un scénario envisageable pour un système pratique.

Pour exprimer les taux secrets, il suffit alors d'attribuer tous les bruits du détecteur χ_{hom} à Eve, c'est-à-dire poser $\chi'_{\text{hom}} = 0$, $\chi'_{\text{line}} = \chi'_{\text{tot}} = \chi_{\text{line}} + \chi_{\text{hom}}/T$. Dans tous les cas, l'information mutuelle d'Alice et Bob ne change pas, mais l'information accessible à l'espion s'écrit alors :

Cas homodyne — individuel

$$I_{BE} \leqslant \frac{1}{2}\log_2\left[T^2\left(V+\chi_{\text{tot}}\right)\left(\frac{1}{V}+\chi_{\text{tot}}\right)\right] \tag{A.8}$$

Cas homodyne — collectif

$$\chi_{\text{BE}} \leqslant G\left(\frac{\lambda_1-1}{2}\right)+G\left(\frac{\lambda_2-1}{2}\right) \tag{A.9}$$

$$\lambda^2_{1,2} = \frac{1}{2}(A \pm \sqrt{A^2-4B}) \tag{A.10}$$

$$A = V^2(1-2T)+2T+T^2(V+\chi_{\text{tot}})^2 \qquad B = T^2(V\chi_{\text{tot}}+1)^2$$

Cas hétérodyne — individuel

$$I^{\text{het}}_{BE} \leqslant \log_2 \frac{T(V+\chi_{\text{tot}})(V+\chi_E)}{V\chi_E+1} \tag{A.11}$$

Cas hétérodyne — collectif

La formule est exactement la même que pour le cas homodyne — collectif, en remplaçant χ_{tot} par l'expression hétérodyne.

A.3 Amplificateurs paramétriques — la suite

A.3.1 Cas hétérodyne + PIA : détail du calcul

Nous reprenons le raisonnement ayant conduit à l'expression des informations pour des attaques collectives. Compte tenu de la modélisation du PIA, il est clair que nous devons rajouter deux modes I et J dans le calcul. Par conséquent, χ_{BE} est calculé à partir des équations suivantes, qui remplacent (A.6) :

$$\chi_{BE} = S(\rho_{AB_1}) - S(\rho^{x_B,p_B}_{AIJFG}) \tag{A.12}$$

$$\chi_{BE} = \sum_{i=1}^{2} G\left(\frac{\lambda_i-1}{2}\right) - \sum_{i=3}^{7} G\left(\frac{\lambda_i-1}{2}\right) \tag{A.13}$$

Il est donc nécessaire d'exprimer les valeurs propres symplectiques de la matrice de covariance γ^{x_B,p_B}_{AIJFG}, ce qui implique la résolution d'un polynôme de degré 5. En procédant comme précédemment, nous écrivons

$$\gamma^{x_B,p_B}_{AIJFG} = \gamma_{AIJFG} - \sigma^T_{AIJFG;B} H_{\text{het}} \sigma_{AIJFG;B} \tag{A.14}$$

où $H_{\text{het}} = (\gamma_{\text{B}} + \mathbb{I}_2)^{-1}$. La matrice identité dans cette expression représente le coupleur inclus dans la détection hétérodyne. Les matrices de cette équation peuvent être extraites de la décomposition de la matrice γ_{AIJFGB}, qui peut elle même être exprimée en calculant tout d'abord la matrice

$$\gamma_{AB_2IJF_0G} = (Y^{\text{PIA}})^T[\gamma_{AB_1} \oplus \gamma_{I_0J} \oplus \gamma_{F_0G}]Y^{\text{PIA}} \tag{A.15}$$

puis en la réarrangeant pour obtenir $\gamma_{AB_2F_0IJG}$, en en calculant enfin

$$\gamma_{ABFIJG} \;=\; (Y^{\text{BS}})^T\gamma_{AB_2F_0IJG}Y^{\text{BS}} \tag{A.16}$$

ce qui permet d'obtenir le résultat cherché grâce à un dernier réarrangement de la matrice. Ici, la transformation du coupleur s'écrit $Y^{\text{BS}} = \mathbb{I}_A \oplus Y^{\text{BS}}_{B_2F_0} \oplus \mathbb{I}_I \oplus \mathbb{I}_J \oplus \mathbb{I}_G$ pour inclure les nouveaux modes. La matrice γ_{I_0J} décrit l'état EPR de variance V_N utilisé pour modéliser le bruit intrinsèque de l'amplificateur, et s'écrit

$$\gamma_{I_0J} = \begin{bmatrix} V_N \cdot \mathbb{I}_2 & \sqrt{V_N^2 - 1} \cdot \sigma_z \\ \sqrt{V_N^2 - 1} \cdot \sigma_z & V_N \cdot \mathbb{I}_2 \end{bmatrix} \tag{A.17}$$

Finalement, la transformation induite par l'amplificateur indépendant de la phase sur les modes B_1 et I_0 est décrite par la matrice

$$\begin{aligned} Y^{\text{PIA}} &= \mathbb{I}_A \oplus Y^{\text{PIA}}_{B_1I_0} \oplus \mathbb{I}_J \oplus \mathbb{I}_{F_0} \oplus \mathbb{I}_G, \text{ avec} \\ Y^{\text{PIA}}_{B_1I_0} &= \begin{bmatrix} \sqrt{g} \cdot \mathbb{I}_2 & \sqrt{g-1} \cdot \sigma_z \\ \sqrt{g-1} \cdot \sigma_z & \sqrt{g} \cdot \mathbb{I}_2 \end{bmatrix} \end{aligned} \tag{A.18}$$

A l'aide de tous ces éléments, un calcul assez fastidieux mais direct conduit aux mêmes expressions que dans l'équation (A.7) pour les valeurs propres $\lambda_{3,4}$, et donne $\lambda_{5,6,7} = 1$.

De la même façon que dans le cas d'une détection homodyne couplée avec un PSA, l'effet du PIA peut être inclus en modifiant le bruit ajouté par la détection hétérodyne de la façon suivante :

$$\chi^{\text{original}}_{\text{het}} = \frac{1 + (1-\eta) + 2v_{\text{el}}}{\eta} \qquad \rightarrow \qquad \chi^{\text{PIA}}_{\text{het}} = \frac{1 + (1-\eta) + 2v_{\text{el}} + V_N(g-1)\eta}{g\eta} \tag{A.19}$$

A.3.2 Cas non-canonique 1 : Détection homodyne et PIA

L'analyse pour les attaques collectives est sensiblement la même que dans le cas d'une détection hétérodyne avec ce type d'amplificateur, excepté pour l'expression de γ_{F_0G}, dans laquelle v doit prendre la valeur appropriée, ainsi que pour l'expression (A.14), qui doit prendre la forme

$$\gamma^{x_B,p_B}_{AIJFG} = \gamma_{AIJFG} - \sigma^T_{AIJFG;B}H_{\text{hom}}\sigma_{AIJFG;B} \tag{A.20}$$

où $H_{\text{het}} = (X\gamma_{\text{B}}X)^{\text{MP}}$.

En prenant en compte ces changements, nous trouvons les mêmes valeurs de $\lambda_{3,4,5,6,7}$, moyennant de nouveau une modification de l'excès de bruit :

$$\chi^{\text{PIA}}_{\text{hom}} = \frac{(1-\eta) + v_{\text{el}} + V_N(g-1)\eta}{g\eta}$$

Interprétation. L'interprétation est alors la même que pour une détection hétérodyne : puisque $\chi_{\text{hom}}^{\text{PIA}}$ tend vers V_N à fort gain, le couplage détection-amplificateur est équivalent à une détection homodyne sans bruit, mais présentant des pertes η' telles que $(1-\eta')/\eta' = V_N$. Puisque $V_N = 1$ quand l'amplificateur est idéal, nous trouvons $\eta' = 0{,}5$, ce qui revient à dire qu'un amplificateur idéal permet de simuler une détection homodyne sans bruit, mais présentant 3 dB de pertes. Par conséquent, même pour un scénario optimal, introduire un tel amplificateur dégrade les performances du système, sauf si le détecteur présente des pertes ou bruits particulièrement importants. Cet effet négatif est présenté dans la figure A.1.

A.3.3 Cas non-canonique 2 : Détection hétérodyne et PSA

Dans le cas de ce protocole, l'analyse est plus complexe, car l'effet de l'amplificateur n'est pas le même sur les deux quadratures. Nous devons donc séparer les expressions des bruits ajoutés pour $\hat{X}$ et $\hat{P}$ comme suit :

$$\begin{aligned} \chi_{\text{het}}^{\text{PSA},x} &= \frac{1+(1-\eta)+2v_{\text{el}}}{g\eta} \\ \chi_{\text{het}}^{\text{PSA},p} &= g\frac{1+(1-\eta)+2v_{\text{el}}}{\eta} \end{aligned} \tag{A.21}$$

Le bruit total est alors défini de façon correspondante : $\chi_{\text{tot}}^{x,p} = \chi_{\text{line}} + \chi_{\text{het}}^{\text{PSA},x,p}/T$, et les expressions de C et D deviennent alors :

$$\begin{aligned} C_{\text{het}}^{\text{PSA}} &= \frac{1}{\left(T\left(V+\chi_{\text{tot}}^{x}\right)\right)\left(T\left(V+\chi_{\text{tot}}^{p}\right)\right)} \times \Big[A\chi_{\text{het}}^{\text{PSA},x}\chi_{\text{het}}^{\text{PSA},p} + B + 1 + \\ &\quad + \left(\chi_{\text{het}}^{\text{PSA},x} + \chi_{\text{het}}^{\text{PSA},p}\right)\left(V\sqrt{B} + T\left(V+\chi_{\text{line}}\right)\right) + +2T\left(V^2-1\right)\Big] \\ D_{\text{het}}^{\text{PSA}} &= \left(\frac{V+\sqrt{B}\chi_{\text{het}}^{\text{PSA},x}}{T(V+\chi_{\text{tot}}^{x})}\right)\left(\frac{V+\sqrt{B}\chi_{\text{het}}^{\text{PSA},p}}{T(V+\chi_{\text{tot}}^{p})}\right) \end{aligned}$$

Les informations mutuelles de Shannon sont modifiées de la même façon, en utilisant

$$V_B = \frac{\eta T}{2}\left[(V+\chi_{\text{tot}}^{x})(V+\chi_{\text{tot}}^{p})\right]^{\frac{1}{2}} \tag{A.22}$$

$$V_{B|A} = \frac{\eta T}{2}\left[(1+\chi_{\text{tot}}^{x})(1+\chi_{\text{tot}}^{p})\right]^{\frac{1}{2}} \tag{A.23}$$

$$V_{B|E}^{x} = \frac{g\eta}{2}\left(\frac{V\chi_E+1}{V+\chi_E} + \chi_{\text{het}}^{\text{PSA},x}\right) \tag{A.24}$$

$$V_{B|E}^{p} = \frac{\eta}{2g}\left(\frac{V\chi_E+1}{V+\chi_E} + \chi_{\text{het}}^{\text{PSA},p}\right) \tag{A.25}$$

$$V_{B|E} = \left(V_{B|E}^{x}V_{B|E}^{p}\right)^{\frac{1}{2}} \tag{A.26}$$

Interprétation. Pour ce schéma, l'interprétation est plus subtile, du fait de l'asymétrie entre quadratures. L'effet sur le taux secret dépend en fait des paramètres du système, et peut être positif ou négatif. Néanmoins, le taux secret reste assez loin de celui fourni par un détecteur parfait, comme illustré sur la figure A.2.

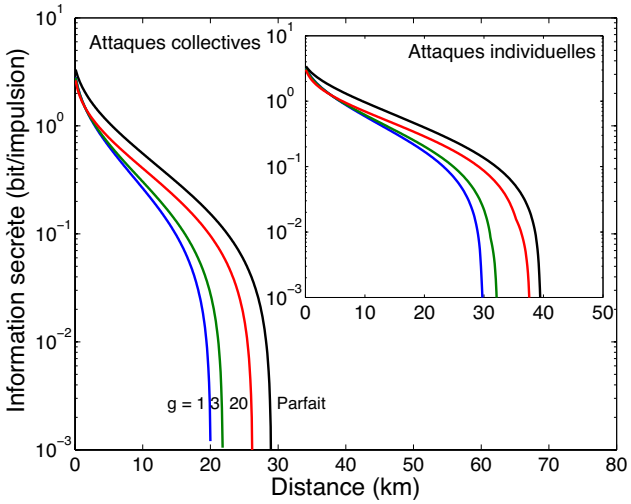


Figure A.1 : *Taux de génération de clé secrète, en fonction de la distance, pour un protocole à* détection homodyne *et* amplificateur indépendant de la phase, *pour des attaques collectives et individuelles. L'amplificateur peut être optimal ($V_N = 1$: pointillés) ou réaliste ($V_N = 1{,}5$: trait plein). Les courbes « parfait » correspondent à un détecteur parfait, en l'absence d'amplificateur.*

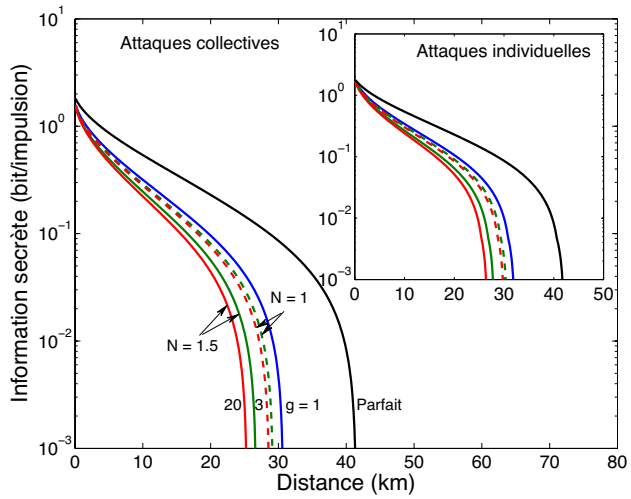


Figure A.2 : *Taux de génération de clé secrète, en fonction de la distance, pour un protocole à* détection hétérodyne *et* amplificateur dépendant de la phase, *pour des attaques collectives et individuelles. Les courbes « parfait » correspondent à un détecteur parfait, en l'absence d'amplificateur.*

Bibliographie

[1] F. Grosshans et P. Grangier. "Continuous Variable Quantum Cryptography Using Coherent States". *Phys. Rev. Lett.* **88**, 057902 (2002).

[2] F. Grosshans, G. van Assche, J. Wenger, R. Brouri, N. Cerf et P. Grangier. "Quantum key distribution using gaussian-modulated coherent states". *Nature* **421**, 238 (2003).

[3] F. Grosshans. *Communication et cryptographie quantiques avec des variables continues*. Thèse de doctorat, Université Paris-Sud 11 (2002).

[4] J. Wenger. *Dispositifs impulsionnels pour la communication quantique à variables continues*. Thèse de doctorat, Université Paris-Sud 11 (2004).

[5] J. Lodewyck. *Dispositif de distribution quantique de clé avec des états cohérents à longueur d'onde télécom*. Thèse de doctorat, Université Paris-Sud 11 (2006).

[6] B. Schneier. *Applied Cryptography* (John Wiley & Sons, 1996).

[7] A. Kerckhoffs. "La cryptographie militaire". *Journal des sciences militaires* **IX**, 5 (1883).

[8] C. Shannon. "A Mathematical Theory of Communication / Communication Theory of Secrecy Systems". *Bell System Technical Journal* **27** / **28**, 379 / 623 (1948 / 1949).

[9] Standard FIPS PUB 46 du NIST (1977).

[10] Standard ANS X9.52 de l'ANSI (1993).

[11] B. Schneier. "The Blowfish Encryption Algorithm". *Dr Dobbs Journal* **19**, 38 (1994).

[12] V. R. Joan Daemen. Proposé sous le nom *Rijndael* (1998), standardisé par le NIST sous le standard FIPS PUB 197 (2001).

[13] R. L. Rivest, A. Shamir et L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". *Commun. ACM* **21**, 120 (1978).

[14] Standard FIPS PUB 186 du NIST (1993).

[15] G. S. Vernam. "Secret Signaling System". Brevet déposé aux États-Unis sous le numéro 1,310,719 (1919).

[16] C. Bennett et G. Brassard. "Quantum Cryptography : Public Key Distribution and Coin Tossing". *IEEE Conf. on Computers, Systems and Signal Processing, Bangalore, India* p. 175 (1984).

[17] C. H. Bennett. "Quantum cryptography using any two nonorthogonal states". *Phys. Rev. Lett.* **68**, 3121 (1992).

[18] D. Bruss. "Optimal Eavesdropping in Quantum Cryptography with Six States". *Phys. Rev. Lett.* **81**, 3018 (1998).

[19] A. K. Ekert. "Quantum cryptography based on Bell's theorem". *Phys. Rev. Lett.* **67**, 661 (1991).

[20] K. Inoue, E. Waks et Y. Yamamoto. "Differential Phase Shift Quantum Key Distribution". *Phys. Rev. Lett.* **89**, 037902 (2002).

[21] P. C. Sun, Y. Mazurenko et Y. Fainman. "Long-distance frequency-division interferometer for communication and quantum cryptography". *Opt. Lett.* **20**, 1062 (1995).

[22] Y. T. Mazurenko, R. Giust et J. P. Goedgebuer. "Spectral coding for secure optical communications using refractive index dispersion". *Optics Comm.* **133**, 87 (1997).

[23] J.-M. Mérolla, Y. Mazurenko, J.-P. Goedgebuer et W. T. Rhodes. "Single-Photon Interference in Sidebands of Phase-Modulated Light for Quantum Cryptography". *Phys. Rev. Lett.* **82**, 1656 (1999).

[24] T. Debuisschert et W. Boucher. "Time coding protocols for quantum key distribution". *Phys. Rev. A* **70**, 042306 (2004).

[25] D. Stucki, N. Brunner, N. Gisin, V. Scarani et H. Zbinden. "Fast and simple one-way quantum key distribution". *Applied Physics Letters* **87**, 194108 (2005).

[26] W. Boucher et T. Debuisschert. "Experimental implementation of time-coding quantum key distribution". *Phys. Rev. A* **72**, 062325 (2005).

[27] J. Bonneau et I. Mironov. "Cache-collision timing attacks against AES". *Lecture notes in computer science* **4249**, 201 (2006).

[28] D. J. Bernstein. "Cache-timing attacks on AES". `http://cr.yp.to/papers.html#cachetiming` (2004).

[29] `http://www.iota.u-psud.fr/~sequre`.

[30] E. Wigner. "On the Quantum Correction For Thermodynamic Equilibrium". *Phys. Rev.* **40**, 749 (1932).

[31] R. L. Hudson. "When is the Wigner quasi-probability density non-negative ?" *Rep. Math. Phys.* **6**, 249 (1974).

[32] C. Piquet. "Fonctions de type positif associées à deux opérateurs hermitiens". *C. R. Acad. Sc. Paris* **279 A**, 107 (1974).

[33] A. Einstein, B. Podolsky et N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete ?" *Phys. Rev.* **47**, 777 (1935).

[34] W. H. Zurek. "Decoherence, einselection, and the quantum origins of the classical". *Rev. Mod. Phys.* **75**, 715 (2003).

[35] J. Williamson. "On the algebraic problem concerning the normal forms of linear dynam- ical systems". *Amer. J. of Math.* **58**, 141 (1963).

[36] R. García-Patrón Sanchez. *Quantum Information with Optical Continuous Variables : from Bell Tests to Key Distribution.* Thèse de doctorat, Université Libre de Bruxelles (2007).

[37] W. Vogel, D.-G. Welsch et S. Wallentowitz. *Quantum Optics, An Introduction* (Wiley-VCH, 2001).

[38] R. A. Horn et C. R. Johnson. *Matrix Analysis* (Cambridge University Press, 1985).

[39] T. C. Ralph. "Continuous variable quantum cryptography". *Phys. Rev. A* **61**, 010303 (1999).

[40] M. Hillery. "Quantum cryptography with squeezed states". *Phys. Rev. A* **61**, 022309 (2000).

[41] N. J. Cerf, M. Lévy et G. van Assche. "Quantum distribution of Gaussian keys using squeezed states". *Phys. Rev. A* **63**, 052311 (2001).

[42] C. Weedbrook, A. M. Lance, W. P. Bowen, T. Symul, T. C. Ralph et P. K. Lam. "Quantum Cryptography Without Switching". *Phys. Rev. Lett.* **93**, 170504 (2004).

[43] N. J. Cerf et C. Adami. "Quantum extension of conditional probability". *Phys. Rev. A* **60**, 893 (1999).

[44] J. von Neumann. *Mathematical Foundations of Quantum Mechanics* (Princeton University Press, 1996).

[45] R. García-Patrón et N. J. Cerf. "Unconditional Optimality of Gaussian Attacks against Continuous-Variable Quantum Key Distribution". *Phys. Rev. Lett.* **97**, 190503 (2006).

[46] M. Navascués, F. Grosshans et A. Acín. "Optimality of Gaussian Attacks in Continuous-Variable Quantum Cryptography". *Phys. Rev. Lett.* **97**, 190502 (2006).

[47] A. S. Holevo. "The capacity of the quantum channel with general signal states". *IEEE Trans. Info. Theory* **44**, 269 (1998).

[48] I. Devetak et A. Winter. "Distillation of secret key and entanglement from quantum states". *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science* **461**, 207 (2005).

[49] I. Csiszár et J. Körner. "Broadcast channels with confidential messages". *IEEE Trans. on Inf. Theory* **24**, 339 (1978).

[50] R. Renner et J. I. Cirac. "de Finetti Representation Theorem for Infinite-Dimensional Quantum Systems and Applications to Quantum Cryptography". *Phys. Rev. Lett.* **102**, 110504 (2009).

[51] J. Lodewyck, M. Bloch, R. García-Patrón, S. Fossier, E. Karpov, E. Diamanti, T. Debuisschert, N. J. Cerf, R. Tualle-Brouri, S. W. McLaughlin et P. Grangier. "Quantum key distribution over 25 km with an all-fiber continuous-variable system". *Phys. Rev. A* **76**, 042305 (2007).

[52] R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. Oemer, M. Fuerst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter et A. Zeilinger. "Entanglement-based quantum communication over 144km". *Nature Physics* **3**, 481 (2007).

[53] T. Schmitt-Manderbach, H. Weier, M. Fuerst, R. Ursin, F. Tiefenbacher, T. Scheidl, J. Perdigues, Z. Sodnik, C. Kurtsiefer, J. G. Rarity, A. Zeilinger et H. Weinfurter. "Experimental demonstration of free-space decoy-state quantum key distribution over 144 km". *Phys. Rev. Lett.* **98**, 010504 (2007).

[54] A. Cozannet, M. Treheux et R. Bouillie. "Étude de la propagation de la lumière dans les fibres optiques à gradient d'indice". *Annals of Telecommunications* **24**, 219 (1974).

[55] J. P. Pocholle, M. Papuchon, J. Raffy et C. Puech. "Effets non linéaires dans les fibres optiques. Applications". *Revue Phys. Appl.* **21**, 673 (1986).

[56] J. Lodewyck, T. Debuisschert, R. Tualle-Brouri et P. Grangier. "Controlling excess noise in fiber-optics continuous-variable quantum key distribution". *Phys. Rev. A* **72**, 050303 (2005).

[57] S. Fossier, E. Diamanti, T. Debuisschert, A. Villing, R. Tualle-Brouri et P. Grangier. "Field test of a continuous-variable quantum key distribution prototype". *New Journal of Physics* **11**, 045023 (14pp) (2009).

[58] H. Haseler, T. Moroder et N. Lütkenhaus. "Testing quantum devices : Practical entanglement verification in bipartite optical systems". *Phys. Rev. A* **77**, 032303 (2008).

[59] `http://www.gnu.org/software/gdb/`.

[60] `http://valgrind.org/`.

[61] G. van Assche. *Quantum Cryptography and Secret-Key Distillation* (Cambridge University Press, 2006).

[62] G. Brassard et L. Salvail. "Secret-Key Reconciliation by Public Discussion". In "Advances in Cryptology", pp. 410–423 (Springer-Verlag, 1994).

[63] C. Crépeau. "Réconciliation et Distillation publiques de secret".

[64] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue et C. G. Peterson. "Fast, efficient error reconciliation for quantum cryptography". *Phys. Rev. A* **67**, 052303 (2003).

[65] C. Berrou, A. Glavieux et P. Thitimajshima. "Near Shannon limit error-correcting coding and decoding : Turbo codes". In "IEEE International Conference on Communications", pp. 1064–1070 (Geneva, 1993).

[66] R. G. Gallager. "Low Density Parity Check Codes". *Trans. of the IRE Professional Group on Information Theory* **49**, 21 (1962).

[67] M. Bloch, A. Thangaraj et S. W. McLaughlin. "Efficient Reconciliation of Correlated Continuous Random Variables using LDPC Codes". *arXiv* **abs/cs/0509041** (2005).

[68] R. C. Bose et D. Ray-Chaudhuri. "On A Class of Error Correcting Binary Group Codes". *Information and Control* **3**, 68 (1960).

[69] A. Hocquenghem. "Codes correcteurs d'erreurs". *Chiffres (Paris)* **2**, 147 (1959).

[70] U. Wachsmann, R. F. Fischer et J. B. Huber. "Multilevel Codes : Theoretical Concepts and Practical Design Rules" (1999).

[71] T. Richardson, M. Shokrollahi et R. Urbanke. "Design of capacity-approaching irregular low-density parity-checkcodes". *IEEE Trans. on Inf. Theory* **47**, 619 (2001).

[72] `http://ipgdemos.epfl.ch/ldpcopt/`.

[73] P. Radosavljevic, A. de Baynast et J. Cavallaro. "Optimized Message Passing Schedules for LDPC Decoding". *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers* p. 591 (2005).

[74] C. Bennett, G. Brassard, C. Crepeau et U. Maurer. "Generalized privacy amplification". *IEEE Trans. on Inf. Theory* **41**, 1915 (1995).

[75] `http://mersenne.org/`.

[76] J. L. Carter et M. N. Wegman. "Universal classes of hash functions". In "STOC '77 : Proceedings of the ninth annual ACM symposium on Theory of computing", pp. 106–112 (ACM, New York, NY, USA, 1977).

[77] D. R. Stinson. "Universal Hashing and Authentication Codes". In "CRYPTO '91 : 11th Annual International Cryptology Conference on Advances in Cryptology", p. 74 (London, 1992).

[78] M. Peev, C. Pacher, R. Alleaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Furst, J.-D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hubel, G. Humer, T. Langer, M. Legre, R. Lieger, J. Lodewyck, T. Lorunser, N. Lutkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J.-B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki, M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden et A. Zeilinger. "The SECOQC quantum key distribution network in Vienna". *New Journal of Physics* **11**, 075001 (2009).

[79] `http://www.idquantique.com/`.

[80] J. Dynes, Z. Yuan, A. Sharpe et A. Shields. "Decoy pulse quantum key distribution for practical purposes". *IET Optoelectron.* **2**, 195 (2008).

[81] H. Hübel, M. R. Vanner, T. Lederer, B. Blauensteiner, T. Lorünser, A. Poppe et A. Zeilinger. "High-fidelity transmission of polarization encoded qubits from an entangled source over 100 km of fiber". *Opt. Express* **15**, 7853 (2007).

[82] H. Weier, T. Schmitt-Manderbach, N. Regner, C. Kurtsiefer et H. Weinfurter. "Free space quantum key distribution : Towards a real life application". *Fortschritte der Physik* **54**, 840 (2006).

[83] A. Leverrier et P. Grangier. "Unconditional Security Proof of Long-Distance Continuous-Variable Quantum Key Distribution with Discrete Modulation". *Phys. Rev. Lett.* **102**, 180504 (2009).

[84] T. Richardson et R. Urbanke. "Multi-Edge Type LDPC Codes". *Workshop honoring Prof. Bob McEliece on his 60th birthday* (2002).

[85] A. Zavatta, S. Viciani et M. Bellini. "Quantum-to-Classical Transition with Single-Photon-Added Coherent States of Light". *Science* **306**, 660 (2004).

[86] H.-J. Briegel, W. Dür, J. I. Cirac et P. Zoller. "Quantum Repeaters : The Role of Imperfect Local Operations in Quantum Communication". *Phys. Rev. Lett.* **81**, 5932 (1998).

[87] W. Dür, H.-J. Briegel, J. I. Cirac et P. Zoller. "Quantum repeaters based on entanglement purification". *Phys. Rev. A* **59**, 169 (1999).

[88] M. Fleischhauer et M. D. Lukin. "Quantum memory for photons : Dark-state polaritons". *Phys. Rev. A* **65**, 022314 (2002).

[89] B. Julsgaard, J. Sherson, J. Cirac, J. Fiurášek et E. Polzik. "Experimental demonstration of quantum memory for light". *Nature* **432**, 482 (2004).

[90] T. C. Ralph et A. P. Lund. "Nondeterministic Noiseless Amplification of Quantum Systems". *arXiv :quantph/0809.0326* (2008).

[91] J. A. Levenson, I. Abram, T. Rivera et P. Grangier. "Reduction of quantum noise in optical parametric amplification". *J. Opt. Soc. Am. B* **10**, 2233 (1993).

[92] R. Mears, L. Reekie, I. Jauncey et D. Payne. "Low-noise erbium-doped fibre amplifier operating at 1.54 µm". *Electronics Letters* **23**, 1026 (1987).

[93] E. Desurvire, J. R. Simpson et P. C. Becker. "High-gain erbium-doped traveling-wave fiber amplifier". *Opt. Lett.* **12**, 888 (1987).

[94] C. M. Caves. "Quantum limits on noise in linear amplifiers". *Phys. Rev. D* **26**, 1817 (1982).

[95] D. T. Pegg, L. S. Phillips et S. M. Barnett. "Optical State Truncation by Projection Synthesis". *Phys. Rev. Lett.* **81**, 1604 (1998).

[96] A. Ourjoumtsev. *Étude théorique et expérimentale de superpositions quantiques cohérentes et d'états intriqués non-gaussiens de la lumière*. Thèse de doctorat, Université Paris-Sud 11 (2007).

[97] J. Lodewyck et P. Grangier. "Tight bound on the coherent-state quantum key distribution with heterodyne detection". *Phys. Rev. A* **76**, 022332 (2007).

[98] J. Sudjana, L. Magnin, R. Garcia-Patron et N. J. Cerf. "Tight bounds on the eavesdropping of a continuous-variable quantum cryptographic protocol with no basis switching". *Phys. Rev. A* **76**, 052301 (2007).