HAL
open science

Dispositif de distribution quantique de clé avec desétats cohérents à longueur d'onde télécom

Jérôme Lodewyck

To cite this version:

Jérôme Lodewyck. Dispositif de distribution quantique de clé avec desétats cohérents à longueur d'onde télécom. Physique Atomique [physics.atom-ph]. Université Paris Sud - Paris XI, 2006. Français. NNT : . tel-00130680v1

HAL Id: tel-00130680

https://theses.hal.science/tel-00130680v1

Submitted on 13 Feb 2007 (v1), last revised 14 Feb 2007 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N ° d'ordre : 8543

Laboratoire Charles Fabry de l'Institut d'Optique

Thales Research and Technologies

Université Paris XI, UFR Scientifique d'Orsay

THÈSE

présentée pour obtenir le grade de
Docteur en Sciences de l'Université Paris XI

Spécialité : Laser et Matière

par
Jérôme LODEWYCK

Sujet :

DISPOSITIF DE DISTRIBUTION QUANTIQUE DE CLÉ AVEC DES ÉTATS COHÉRENTS À LONGUEUR D'ONDE TÉLÉCOM

Soutenue le 13 décembre 2006
devant la Commission d'examen :

Messieurs	Joseph BOUTROS,	Membre invité
	Nicolas CERF,	Membre invité
	Pierre CHAVEL,	Président
	Thierry DEBUISSCHERT,	Examinateur
	Noël DIMARCQ,	Examinateur
	Claude FABRE,	Rapporteur
	Nicolas GISIN,	Rapporteur
	Philippe GRANGIER,	Directeur de thèse

Table des matières

Remerciements **11**

Introduction **13**

I Distribution quantique de clé avec des variables continues **17**

1 Cryptographies classique et quantique **19**
1.1 Cryptographie classique et cryptographie quantique 19
1.2 Utilisation des variables continues en cryptographie quantique 22
1.3 Information classique 22
1.4 Information classique et variables gaussiennes 24

2 Distribution quantique de clé avec des états cohérents **27**
2.1 Variables quantiques continues 27
2.2 Modèle du canal gaussien 29
2.3 Informations mutuelles et protocoles à états cohérents 30
2.4 Protocole inverse 33
2.5 Attaques optimales 33
2.6 Implications expérimentales 37
2.7 Photons uniques ou variables continues ? 38

3 Distribution quantique de clé avec une détection hétérodyne **41**
3.1 Information mutuelle I_{AB} 41
3.2 Protocole inverse 42
3.3 Protocole direct 43
3.4 Recherche d'attaques optimales 48
3.5 Quel avantage pratique ? 50

4 Un aperçu de la sécurité des protocoles à variables continues **53**
4.1 Intrication virtuelle : schéma équivalent à la modulation d'Alice 54
4.2 Attaques individuelles gaussiennes 57
4.3 Attaques non gaussiennes et attaques collectives de taille finie 57
4.4 Attaques collectives gaussiennes 58
4.5 Attaques collectives générales 59
4.6 Attaques cohérentes 60
4.7 Attaques non quantiques 60
4.8 Autres protocoles de distribution quantique de clé utilisant des variables continues 61

II Réalisation expérimentale 63

5 Introduction 65

6 Démonstrateur de distribution quantique de clé 67
6.1 Faire de l'optique quantique avec des fibres optiques . . . 67
6.2 Composants utilisés . . . 69
6.3 Détection homodyne impulsionnelle limitée au bruit de photon . . . 71
6.4 Modulation gaussienne avec des modulateurs électro-optiques . . . 73

7 Excès de bruit et analyse de sécurité 79
7.1 Corrélations et information mutuelle I_{AB} . . . 79
7.2 Bruit à la sortie . . . 81
7.3 Bruit de photon . . . 81
7.4 Bruit électronique . . . 82
7.5 Bruit de phase . . . 83
7.6 Bruit d'amplitude . . . 85
7.7 Erreurs de modulation . . . 87
7.8 Bruit ramené à l'entrée et excès de bruit . . . 87
7.9 Information secrète . . . 89

8 Attaque non gaussienne 93
8.1 Attaque «interception-réémission » . . . 93
8.2 Réalisation expérimentale . . . 95
8.3 Analyse du bruit . . . 97
8.4 Attaque interception-réémission partielle . . . 97
8.5 Analyse du bruit de l'attaque interception-réémission partielle . . . 98
8.6 Informations accessibles à l'espion . . . 99
8.7 Attaque non gaussienne . . . 102

9 Multiplexage temporel 105
9.1 Multiplexage temporel avec une diode laser continue . . . 106
9.2 Multiplexage temporel avec une diode pulsée . . . 107
9.3 Démultiplexage . . . 110
9.4 Choix du coupleur de démultiplexage . . . 111
9.5 Multiplexage temporel dans une fibre de 25 km . . . 112

10 Intégration et prototypage 117
10.1 Pilotage des cartes d'acquisition . . . 117
10.2 Protocole de communication quantique . . . 119
10.3 Découpage en blocs . . . 122
10.4 Synchronisation entre Alice et Bob . . . 123
10.5 Gestion des blocs de données . . . 125
10.6 Programmation et dépendances logicielles . . . 126
10.7 Calibration de la transmission . . . 126
10.8 Génération de nombres aléatoires . . . 127
10.9 Intégration en rack 19 pouces . . . 129

III Distillation d'une clé secrète 131

11 Introduction 133

12 Réconciliation et codes correcteurs d'erreurs 137
12.1 Codes correcteurs d'erreurs . . . 138
12.2 Exemples de codes correcteurs d'erreurs . . . 138
12.3 Réconciliation et "side information" . . . 141
12.4 Décodage mou . . . 143
12.5 Décodage des codes LDPC . . . 146
12.6 Modulation codée . . . 148
12.7 Décodage multi-niveaux et réconciliation . . . 149

13 Réalisation et optimisation de la réconciliation 153
13.1 Compromis entre vitesse et efficacité . . . 153
13.2 Quel rapport signal à bruit choisir ? . . . 154
13.3 Paramètres de la discrétisation . . . 156
13.4 Taux des codes LDPC . . . 157
13.5 Décodage des codes LDPC : l'algorithme Message Passing et ses variantes . . . 159
13.6 Réduction du temps de calcul . . . 160
13.7 Turbo codes . . . 163
13.8 Perspectives d'amélioration . . . 164

14 Amplification de confidentialité 169
14.1 Familles de fonctions de hachage comme amplificateurs de confidentialité . . . 169
14.2 Une petite excursion dans le monde des corps finis . . . 170
14.3 Des exemples simples de familles de fonctions de hachage . . . 171
14.4 Multiplication rapide dans $GF(2^l)$ utilisant la transformée de Fourier discrète . . . 173
14.5 Famille de fonctions de hachage utilisant directement la NTT . . . 174
14.6 Universalité et paramètre de sécurité . . . 176

15 Distillation d'une clé à travers un réseau classique 179
15.1 Contraintes liées à l'utilisation d'un réseau . . . 179
15.2 Paradigmes de programmation appliqués à la réconciliation . . . 180
15.3 Protocoles de communication . . . 183
15.4 Implémentation d'un canal classique . . . 188
15.5 Interfaçage avec le logiciel de pilotage . . . 188

Conclusion 191

Annexes 192

A Turbo codes 193
A.1 Codes convolutifs . . . 193
A.2 Décodage mou des codes convolutifs . . . 194
A.3 Concaténation de codes . . . 195

B Décodage BCJR et "side-information" **197**

Bibliographie **200**

Table des figures

1.1 Espace des phases 26

2.1 Espace des phases 28
2.2 Effet du canal quantique gaussien 30
2.3 Information secrète, protocole direct 32
2.4 Information secrète, protocole inverse 34
2.5 Attaque de type lame séparatrice 34
2.6 Attaque de type cloneuse intriquante 35

3.1 Schéma de principe d'un protocole hétérodyne 42
3.2 Information mutuelle pour la réconciliation inverse 44
3.3 Modèle d'attaque hétérodyne 44
3.4 Modèle d'attaque générale 45
3.5 Information mutuelle pour la réconciliation directe 46
3.6 Meuilleures attaques hétérodynes 49
3.7 Efficacité de réconciliation minimale 51

4.1 Modélisations d'attaques 54
4.2 Schéma de modulation à intrication virtuelle 55
4.3 Information de Holevo 59

5.1 Schéma expérimental 66

6.1 Schéma de principe d'une détection homodyne 71
6.2 Tomographie 74
6.3 Modulation d'amplitude 75
6.4 Modulation d'amplitude corrigée 76
6.5 Effet de la troncature d'une modulation gaussienne 77

7.1 Schéma des bruits observés à la sortie 81
7.2 Électronique de la détection homodyne 82
7.3 Bruit de phase 83
7.4 Bruit symétrique 85
7.5 Erreurs de modulation 86
7.6 Schéma des bruits observés à la sortie 88
7.7 Schéma des bruits observés à l'entrée 89
7.8 Information secrète 91

8.1 Schéma de principe d'une attaque interception-réémission 94
8.2 Réalisation de l'attaque interception-réémission 95

8.3 Bruits ramenés à l'entrée pour une attaque interception-réémission 96
8.4 Schéma d'une attaque interception-réémission partielle 97
8.5 Excès de bruit pour une attaque interception-réémission partielle 98
8.6 Information I_{BE} pour une attaque interception-réémission partielle. 101

9.1 Multiplexage temporel 106
9.2 Préparation des impulsions 106
9.3 Fonctionnement de la diode laser pulsée 107
9.4 Signal de détection homodyne avec une diode pulsée 108
9.5 Signaux de déclenchement 110
9.6 Démultiplexage avec un coupleur 111
9.7 Signal de détection homodyne 112
9.8 Taux secret et pertes de la détection homodyne 113

10.1 Répartition des impulsions test 120
10.2 Décomposition d'un bloc de données 123
10.3 Synchronisation des blocs de données 124
10.4 Intensité du signal modulé 127
10.5 Topologie d'un réseau quantique 130

11.1 Étapes du processus de génération d'une clé secrète 134
11.2 Informations échangées dans le processus de génération d'une clé 134

12.1 Structure d'un mot code 138
12.2 Schéma des étapes de la réconciliation 141
12.3 Flux d'information molle dans un décodeur mou 144
12.4 Graphe représentatif d'un code LDPC 146
12.5 Schéma du décodage LDPC 147
12.6 Décodage multi-niveaux 149
12.7 Distribution des LLR intrinsèques 152

13.1 Efficacité de réconciliation 154
13.2 Optimisation de la variance de modulation 154
13.3 Portée du protocole de distribution quantique de clé 155
13.4 Discrétisation 157
13.5 Efficacité de discrétisation 158
13.6 Efficacité de discrétisation 160
13.7 Fonction $\phi(x) = -\ln(\tanh(x/2))$ 162
13.8 Carte EXIT pour des turbo codes 165

15.1 Protocole de communication CSKDP 184
15.2 Protocole de communication BSKDP 186
15.3 Protocole de communication BSKDP 187
15.4 Diagramme de classes de la réconciliation 189

A.1 Encodage d'un code convolutif 194
A.2 Principe du turbo décodage 195

Liste des tableaux

6.1 Pertes dans les fibres optiques 68

7.1 Excès de bruit 89

10.1 Caractéristiques des cartes d'acquisition 119

12.1 Performance des codes BCH 140

13.1 Caractéristiques de la réconciliation avec des codes LDPC 164

14.1 Liste de trinômes irréductibles de grand degré 172
14.2 Paramètres possibles pour la transformation NTT rapide 175
14.3 Vitesse de l'amplification de confidentialité 177

15.1 Usage du canal classique 180

Remerciements

Le travail effectué dans cette thèse est le fruit d'une collaboration entre le département d'Optique et d'Optronique de la société Thales Research and Technologies et le groupe d'Optique Quantique du Laboratoire Charles Fabry de d'Institut d'Optique. Durant mes années de thèse, j'ai été amené à côtoyer ces deux laboratoires, et ces courts remerciements ne suffiraient pas à citer touts les membres de mes deux familles. Toutefois, je tiens particulièrement à remercier Jean Paul Castera, de Thales, et Pierre Chavel, de l'Institut d'Optique, qui m'ont chaleureusement accueilli dans leurs laboratoires.

Mes remerciements vont également à Thierry Debuisschert, Philippe Grangier et Rosa Tualle-Brouri qui m'ont encadré pendant ces trois années. L'équilibre toujours juste entre encadrement et autonomie qu'ils ont su m'accorder m'a permis d'apprécier à tout moment mon travail de thèse. Leur talent et leur pragmatisme nous ont permis de mener à bien notre projet, jalonné par les échéances du projet européen SECOQC.

Merci à Claude Fabre et Nicolas Gisin qui ont accepté de rapporter ce manuscrit. Merci à Joseph Boutros, Nicolas Cerf, Pierre Chavel et Noël Dimarcq qui ont accepté de faire partie de mon jury.

Merci à ceux qui ont travaillé avec moi sur la manip crypto : Jérôme et Fred qui ont su si bien m'accueillir et Simon, Eleni et Anthony qui m'ont brillamment succédé. Travailler avec vous a toujours été agréable et enrichissant. Merci à Sébastien, Sylvain, Alexeï, Julien et Aurélien dont j'ai partagé le bureau et la bonne humeur.

Pendant ces trois années, j'ai eu l'occasion de collaborer avec Gilles Van Assche, Raúl García-Patrón et Matthieu Bloch. Leur compétence et leur amitié ont été pour moi une aide précieuse.

Enfin, mes plus chaleureux remerciements vont à mes parents pour avoir toujours eu confiance en moi, et à ma femme, Solène, qui m'a soutenu pendant ces trois années et qui a su placer et déplacer à la perfection chaque virgule de ce manuscrit.

Introduction

Ce manuscrit traite de la réalisation expérimentale d'un dispositif de distribution quantique de clé utilisant des états cohérents. La distribution quantique de clé se distingue des méthodes de cryptage classique usuelles par le caractère inconditionnel de ses preuves de sécurité. Cette propriété, fondée sur les lois de la physique quantique, permet de transmettre un secret sur un canal de communication sans avoir besoin de faire de supposition restrictive sur les capacités d'un éventuel espion qui voudrait connaître ce secret.

Le protocole de distribution quantique de clé avec des états cohérents se démarque des protocoles utilisant des photons uniques en ce que sa réalisation ne nécessite que des composants standards, optimisés par l'industrie des télécommunications. L'utilisation de ces composants ouvre la voie à des taux de transmission élevés.

Nous avons réalisé un dispositif cryptographique fibré utilisant ces composants télécom. Notre objectif est d'obtenir un système complet et autonome qui prenne en compte tous les aspects pratiques d'un système de cryptage. Ce travail a été réalisé dans le cadre du projet Européen SECOQC, qui aboutira à l'assemblage d'un réseau quantique dont notre système sera l'un des composants.

Contexte scientifique en début de thèse

Les premiers protocoles de distribution quantique de clé utilisant des variables continues sont apparus vers 2000 [1, 2, 3]. Ils nécessitaient la génération d'états comprimés de la lumière, ce qui les rendait peu attrayants sur le plan pratique. Le protocole utilisant des états semi-classiques de la lumière (ou états cohérents) que nous utilisons a été développé à l'Institut d'Optique par Frédéric Grosshans [4, 5]. Il se distingue des premiers protocoles par une grande simplicité, puisqu'il suffit à Alice et Bob d'utiliser des impulsions laser modulées. Une première démonstration de principe en espace libre a été réalisée par Jérôme Wenger [6]. Notre travail est parti de cette démonstration de principe pour aboutir à un dispositif complet de distribution quantique de clé.

Travail original

Le travail effectué au cours de cette thèse couvre plusieurs domaines :

- Nous avons construit une expérience de distribution quantique de clé avec des états cohérents, réalisée avec des fibres optiques et utilisant des composants standards des technologies des télécommunications. Nous avons caractérisé le bruit du dispositif qui quantifie la sécurité du système, puis nous avons distribué une clé secrète sur une fibre d'une longueur de 25 km.
- Nous avons réalisé concrètement diverses attaques afin de valider notre caractérisation.

- Nous avons étudié de façon théorique une adaptation du protocole à une double mesure de quadrature par Bob. Ce nouveau protocole produit un taux secret théorique supérieur au taux secret du protocole usuel.
- Nous avons réalisé et optimisé de nouveaux algorithmes permettant de corriger les erreurs introduites par la transmission quantique, et de distiller une clé secrète.
- Enfin, nous avons conçu un ensemble de logiciels permettant de piloter l'expérience de façon autonome, en contrôlant le protocole de communication classique entre Alice et Bob.

Collaborations scientifiques

Le travail présenté dans ce manuscrit a bénéficié de plusieurs collaborations scientifiques. Tout d'abord avec l'équipe de Nicolas Cerf de l'Université Libre de Bruxelles :

- Gilles Van Assche a conçu et mis en oeuvre la première version de l'algorithme de réconciliation par tranches [7]. Plusieurs notions et aspects techniques de ce protocole ont été conservés dans le protocole de réconciliation utilisé actuellement.
- Raúl García-Patrón a conduit plusieurs études théoriques liées à notre protocole, notamment concernant les taux secrets sûrs face aux attaques non-gaussiennes et collectives (voir chapitre 4).
- Cécile Neu, ingénieur en informatique et stagiaire dans le groupe de Nicolas Cerf, a conçu et programmé les fondements de notre mécanisme de canal classique permettant le dialogue entre Alice et Bob.

Nous avons également collaboré avec Matthieu Bloch, qui effectue sa thèse à Georgia Tech Lorraine (Metz), sous la direction de J.M. Mérolla et S. McLaughlin. Matthieu a développé un nouvel algorithme de réconciliation [8] spécifiquement conçu pour notre approche ; nous avons donc travaillé avec lui pour adapter cet algorithme à notre système expérimental, et pour optimiser le taux de clé secrète produite.

Organisation du manuscrit et guide de lecture

Ce manuscrit comporte trois parties. La première est consacrée à l'**étude théorique** des protocoles de distribution quantique de clé utilisant des variables continues. Les deux premiers chapitres de cette partie introduisent les notions que nous utiliserons par la suite, et expriment les taux d'information secrète produits par ces protocoles. Les deux chapitres suivants sont indépendants de la suite du manuscrit. Ils présentent respectivement une autre méthode de détection pour les protocoles à variables continues, et une revue des preuves qui garantissent la sécurité inconditionnelle de l'information secrète échangée.

La deuxième partie est consacrée à la **réalisation expérimentale** du protocole de distribution quantique de clé utilisant des états cohérents. Les deux premiers chapitres décrivent l'expérience et montrent comment calculer l'information secrète à partir des mesures expérimentales. Le chapitre suivant montre la robustesse de notre réalisation expérimentale face à une classe étendue d'attaques quantiques. Les deux derniers chapitres de cette partie traitent de l'adaptation de notre dispositif expérimental à des conditions réelles d'utilisation.

Les protocoles à variables continues nécessitent un **traitement classique** élaboré de l'information collectée pour extraire une clé secrète. La dernière partie de ce manuscrit est consacrée aux algorithmes classiques que nous employons pour obtenir une clé secrète à partir de nos données expérimentales. Cette partie est indépendante de la précédente. Ses deux premiers

chapitres décrivent le processus de réconciliation qui transforme les variables continues bruitées issues de l'expérience en bits sans erreurs. Le chapitre suivant présente une adaptation des algorithmes d'amplification de secret aux caractéristiques de notre protocole. Enfin le dernier chapitre est consacré à la réalisation d'un protocole de communication permettant de distribuer ces algorithmes sur plusieurs machines.

Première partie

Protocoles de distribution quantique de clé avec des variables continues

Chapitre 1

Cryptographies classique et quantique

Ce chapitre introduit les notions couramment utilisées dans le domaine de la cryptographie quantique. Nous décrirons les systèmes de cryptage classique usuels, puis nous expliquerons l'apport de la mécanique quantique dans le domaine de la cryptographie. Ensuite, nous définirons l'entropie et l'information mutuelle, éléments fondateurs de l'information classique que nous utiliserons pour quantifier la sécurité de notre système de cryptographie quantique. Munis de cette boîte à outils généraliste, nous pourrons aborder les concepts du système de cryptographie avec des variables continues.

1.1 Cryptographie classique et cryptographie quantique

La cryptologie, la science du secret, comporte deux parties complémentaires. D'une part, la cryptographie a pour objet de crypter de l'information confidentielle[1], afin de ne la rendre accessible qu'aux personnes autorisées. D'autre part, la cryptanalyse cherche à briser un cryptage pour révéler le message secret.

Remarquons tout d'abord que l'objet de la cryptanalyse n'est pas d'empêcher la transmission d'un message secret, mais d'en extraire le contenu. Ainsi, les analyses de sécurité que nous développerons supposent l'existence d'un canal de communication donné, non interrompu par l'espion, par lequel transite le message secret. Dans cette hypothèse, nous rechercherons l'information maximale accessible à un éventuel espion, compte tenu des caractéristiques du canal par lequel transite le message secret.

On distingue plusieurs classes de sécurité caractérisant les systèmes de cryptage. D'abord, la sécurité *inconditionnelle* garantit qu'il n'existe aucune méthode cryptanalytique capable de briser le procédé de cryptage. Il existe peu de primitives cryptographiques dont la sécurité inconditionnelle soit prouvée. Citons trois méthodes dont la sécurité inconditionnelle est prouvée par la théorie de l'information : l'authentification d'un canal, qui permet d'attester l'identité d'un interlocuteur, l'amplification de secret, qui permet de générer une chaîne de bits totalement secrète à partir d'une chaîne de bits partiellement secrète, et le cryptage par clé secrète décrit ci-dessous. Les protocoles de cryptographie quantique font usage de ces trois méthodes.

Les autres systèmes de cryptage fondent leur sécurité sur des hypothèses pratiques ou des conjectures d'ordre théorique. Par exemple, on peut supposer que la puissance de calcul d'un

[1] La cryptographie est un domaine plus vaste que le simple cryptage d'un message secret auquel nous nous consacrons. On peut citer les signatures électroniques, l'authentification, le partage de secret, etc. Tous ces aspects sont abordés en détail dans [9].

éventuel espion voulant décrypter un message est limitée. De plus, si nous pouvons borner inférieurement la complexité de la cryptanalyse d'un système cryptographique, nous pouvons utiliser cette hypothèse pour garantir la sécurité de ce système. Un exemple de conjecture théorique est la complexité de certains problèmes arithmétiques, comme la factorisation de nombres premiers, ou le calcul de logarithmes dans des ensembles de nombres entiers (nous rencontrerons cette notion d'opérations sur les nombres entiers au chapitre 14).

Suivant ces deux classes de sécurité, on distingue deux schémas génériques de cryptage :

le cryptage à clé privée Cette méthode de cryptage relativement récente (années 1970) est omniprésente dans les systèmes de cryptage actuels. Elle utilise un couple de clés, l'une publique, qui permet de crypter un message, l'autre privée et uniquement connue du destinataire, qui permet de décrypter le message. L'inviolabilité de cette méthode de cryptage n'est pas prouvée. Elle repose à la fois sur une hypothèse sur les capacités de calcul offertes à l'espion, et sur une conjecture sur la difficulté de retrouver la clé privée connaissant la clé publique. C'est pourquoi l'utilisateur du cryptage à clé privée doit admettre qu'un éventuel espion n'a ni les moyens intellectuels pour trouver un algorithme permettant de briser le cryptage, ni des moyens technologiques importants qui lui permettraient de surmonter la complexité du problème.

le cryptage à clé secrète Cette méthode, plus ancienne (XVIIIe siècle), consiste à combiner le message secret avec un chaîne aléatoire, appelée «clé secrète», produisant un message crypté aléatoire pour quiconque ignore la clé de cryptage. Claude Shannon a formalisé cette méthode [10] en montrant que ce cryptage peut être inconditionnellement sûr, si la clé secrète est aussi longue que le message à crypter, est à usage unique, et bien sûr est totalement inconnue d'un espion éventuel. Ainsi, la clé secrète doit être renouvelée au même rythme que le message est envoyé. C'est pourquoi cette méthode est peu utilisée en pratique, malgré sa sécurité inconditionnelle. En effet, il est difficile de *distribuer* une clé de grande taille en garantissant son caractère secret.

La cryptographie quantique répond au problème de la distribution de clé en offrant une méthode physique pour distribuer une clé secrète, méthode garantissant la sécurité inconditionnelle d'une transmission. D'abord, Alice et Bob[2] s'échangent une séquence de nombres aléatoires encodés dans des variables quantiques, transmises par un canal dit «quantique». À l'issue de la transmission, Alice et Bob partagent donc deux chaînes de symboles corrélés. Les lois de la physique quantique permettent ensuite de borner l'information sur cette chaîne accessible à un espion en fonction des paramètres de la transmission quantique. Pour obtenir une clé secrète à partir de leur échange quantique, Alice et Bob utilisent ensuite des algorithmes classiques pour corriger les erreurs de transmission et éliminer l'information connue par l'espion. Grâce à cette clé secrète, Alice et Bob peuvent s'échanger un message secret par un cryptage à clé secrète sur un canal classique non sécurisé.

Pour prouver la sécurité de la distribution quantique de clé, nous devons définir les variables quantiques transmises dans le canal quantique, ainsi qu'une procédure d'encodage de la clé secrète dans ces variables quantiques. Le premier protocole [11] de distribution quantique de clé utilise la polarisation d'impulsions lumineuses contenant un seul photon. Cette polarisation comporte deux états linéairement indépendants, définis dans une base donnée. Pour encoder

[2] Conformément à la terminologie consacrée, nous appelons «Alice» et «Bob» deux personnages désireux de s'échanger à distance un message secret. L'espion, Ève, de l'anglais "to eavesdrop" : espionner, tente de connaître le contenu du message.

l'information, Alice choisit une base aléatoire parmi deux bases distinctes, puis encode un bit aléatoire 0 ou 1 dans l'état de polarisation dans cette base. À l'autre bout du canal quantique, Bob mesure la polarisation dans une base aléatoire. Après sa mesure, Bob révèle ses choix de base, et dans les cas où les bases d'Alice et de Bob coïncident, un bit est échangé.

La théorie de la mesure quantique permet de détecter l'espionnage du canal quantique. Si un espion, intercalé entre Alice et Bob, tente de mesurer l'état de polarisation des photons envoyés par Alice, il utilisera avec une probabilité 1/2 la mauvaise base de polarisation. Ce faisant, il projettera l'état quantique sur l'état propre correspondant à la valeur de sa mesure. Si les choix de base d'Alice sont adaptés, cet état n'est plus un état propre de la base initialement choisie par Alice, et la valeur du bit envoyé aura une certaine *probabilité d'erreur* à son arrivée chez Bob.

De façon équivalente, on peut aborder la sécurité des protocoles de distribution quantique de clé du point de vue du clonage quantique. Sonder l'état quantique qui transite dans le canal quantique revient à le dupliquer en deux clones de l'état orignal. L'un des clones est envoyé à Bob, alors que l'autre clone est mesuré par Ève. Les relations de commutation introduites par la mécanique quantique empêchent de créer deux clones parfaitement identiques à l'état original. Dans le cas contraire, il serait possible de mesurer deux variables quantiques incompatibles à la fois. Quantitativement, nous pouvons relier le taux d'erreur minimal sur l'un des clones au taux d'erreur sur l'autre clone. Nous emploierons ce modèle pour établir la sécurité des protocoles utilisant des états cohérents (chapitre 2).

On évalue donc la quantité d'information qu'a obtenue l'espion sur la clé échangée entre Alice et Bob par la mesure du *rapport signal à bruit* de la transmission. Cette mesure permet d'évaluer à la fois la quantité d'information I_{AB} (définie à la fin de ce chapitre) échangée entre Alice et Bob, et de borner supérieurement l'information qu'obtient l'espion sur la clé envoyée par Alice (I_{AE}), ou sur la mesure de Bob (I_{BE}). Là s'arrête le traitement quantique de la distribution de clé. Un théorème classique, énoncé par Csiszar et Körner [12], calcule le taux de bits secrets ΔI qu'il est possible d'obtenir à partir d'un tel échange :

$$\Delta I = I_{AB} - \min(I_{AE}, I_{BE}). \tag{1.1}$$

Une étape d'amplification de secret permet finalement d'obtenir une clé secrète aléatoire de longueur ΔI totalement inconnue de l'espion.

La distribution quantique de clé est à ce jour la seule méthode de cryptage d'un message secret inconditionnellement sûre. Pourtant, cette propriété unique n'est pas toujours suffisante pour justifier son utilisation par rapport aux méthodes de cryptage traditionnelles. En effet, la cryptographie quantique est une méthode de cryptage physique, c'est-à-dire qu'elle nécessite du matériel dédié, tout en obtenant des débits largement inférieurs aux cryptages arithmétiques. Toutefois, citons deux avantages particuliers de la cryptographie quantique. D'abord, la sécurité inconditionnelle permet de garantir la pérennité du système de cryptage. Dans le milieu des cryptages arithmétiques, les avancées cryptanalytiques et technologiques obligent à régulièrement mettre à jour les systèmes, par exemple en utilisant des clés de cryptage plus longues, ou de nature différente. La sécurité des systèmes de distribution quantique de clé permet au contraire de garder un système sûr pour une durée arbitrairement longue, quelles que soient les avancées futures dans le domaine de la cryptanalyse, envisagées ou encore inconnues.

Un autre avantage de la cryptographie quantique est la pérennité du secret. Dans un système de cryptage classique, un espion peut conserver le message chiffré en espérant pouvoir un jour le décoder. Ainsi, la durée de vie d'un secret est égale à la durée de vie de la méthode qui sert à le chiffrer. En revanche, la cryptographie quantique se fonde sur la perturbation introduite par le simple fait d'espionner le canal quantique, que l'espion ait réussi ou non à interpréter la signification de l'information espionnée. Ainsi, un secret transmis par un système de cryptographie quantique a une durée de vie infinie.

1.2 Utilisation des variables continues en cryptographie quantique

Un nouveau champ de recherche a récemment émergé dans le domaine de l'information quantique. Il consiste en l'utilisation de variables continues, c'est-à-dire de variables comportant un continuum d'états indépendants pour encoder de l'information, par opposition aux variables discrètes, telle la polarisation de la lumière, qui ne comportent qu'un nombre fini d'états indépendants. Ces variables permettent de s'affranchir des contraintes technologiques qui accompagnent l'utilisation de variables discrètes portées par des photons uniques. Notamment, les mesures sur un système à variables continues se fondent sur une détection interférométrique appelée *détection homodyne* (abordée au chapitre 6), qui utilise des photodiodes rapides et efficaces, capables d'atteindre des taux de répétition élevés ; cette situation contraste avec les efficacités et vitesses limitées des compteurs de photons nécessaires aux protocoles à variables discrètes.

Des résultats théoriques sur le clonage quantique [13] ont initié l'utilisation de ces variables continues par des protocoles de distribution quantique de clé. Un premier protocole utilisant des états comprimés de la lumière a été développé dans le groupe de Nicolas Cerf [3]. Toutefois, la génération et la manipulation de ces états est peu pratique du point de vue expérimental, notamment en vue d'une application hors du laboratoire. Cependant, un nouveau nouveau protocole de distribution quantique de clé développé à l'Institut d'Optique [5, 14] encode l'information dans l'amplitude et la phase d'états semi-classiques de la lumière appelés «états cohérents». Ce protocole simplifie donc le système d'émission utilisé par Alice : une diode laser sert à générer des états cohérents qui sont ensuite modulés par des modulateurs électro-optiques d'amplitude et de phase. Finalement, l'ensemble du dispositif de distribution quantique de clé avec des états cohérents peut être réalisé à l'aide de composants standards des technologies télécom, ce qui ouvre la voie aux taux de répétition élevés offerts par ces technologies.

Avant d'aborder dans le détail ce protocole de distribution quantique de clé utilisant des états cohérents, nous allons établir quelques résultats élémentaires de la théorie de l'information classique qui nous permettront d'en étudier la sécurité.

1.3 Information classique

Dans le domaine de la cryptographie classique comme de la cryptographie quantique, beaucoup de résultats font appel à la théorie de l'information. Cette théorie a été introduite par Shannon en 1948 [15] dans le contexte des télécommunications.

La brique fondamentale de cette théorie est la notion d'entropie. Considérons une séquence X de symboles tirés d'un alphabet composé de n lettres $x_1, \ldots, x_n$. On peut se demander combien il faut de bits pour enregistrer cette information, ou à quel débit on peut la transmettre à travers un canal de communication. Bien entendu, la réponse à ces questions dépend du contexte du message, c'est-à-dire de l'*a priori* sur la nature du message. Par exemple, si l'on sait que le message est écrit en français, on sait que les mots qui le composent appartiennent à un ensemble restreint : le dictionnaire. Pour utiliser cette propriété, on peut par exemple envisager d'attribuer un code à chaque mot du dictionnaire, et le nombre de bits nécessaires pour désigner un mot sera inférieur au nombre de bits nécessaires à spécifier chacune des lettres indépendamment. Cet *a priori*, hors de tout contexte culturel comme celui introduit par notre exemple, est formalisé par Shannon par les probabilités d'occurrence $p(X = x_i)$ de chaque symbole. À partir de là, on définit l'entropie H d'une variable aléatoire X par :

$$H(X) = -\sum_{x_i \in X} p(X = x_i) \log_2 p(X = x_i). \tag{1.2}$$

Si nous savons qu'un et un seul symbole x_j va nous parvenir, alors $p(X = x_j) = 1$ et $p(X = x_i) = 0$ pour $i \neq j$, donc $H(X) = 0$. Au contraire, si tous les symboles sont équiprobables, $H(X) = \log_2(n)$ est maximale. L'entropie quantifie donc notre ignorance sur la variable aléatoire, c'est-à-dire l'information, exprimée en bits par symbole, qu'elle nous apporte quand on en prend connaissance.

Si nous considérons plusieurs variables aléatoires A et B, nous pouvons définir l'*entropie conjointe*. Elle s'exprime en considérant la probabilité conjointe $p(A, B)$:

$$H(A, B) = -\sum_{a_i \in A, b_i \in B} p(A = a_i, B = b_i) \log_2 p(A = a_i, B = b_i). \tag{1.3}$$

L'entropie conjointe est l'information qu'apporte la connaissance simultanée des deux variables aléatoires. Si les variables A et B sont indépendantes, on montre simplement que l'entropie totale est la somme des entropies individuelles :

$$H(A, B) = H(A) + H(B) \quad \text{si et seulement si } A \text{ et } B \text{ sont indépendantes.} \tag{1.4}$$

L'*entropie conditionnelle* quantifie notre ignorance de la variable aléatoire B connaissant la variable aléatoire A. Intuitivement, c'est l'information que nous apporte la connaissance conjointe de A et B, à laquelle on retranche l'information que nous avait déjà apportée la variable A. On l'écrit :

$$H(B|A) = H(A, B) - H(A). \tag{1.5}$$

À partir de cette définition, on peut exprimer l'entropie conditionnelle en fonction des probabilités d'occurrence :

$$\begin{aligned} H(B|A) &= \sum_{a_i \in A} p(A = a_i) H(B|A = a_i) && (1.6) \\ &= -\sum_{a_i \in A; b_i \in B} p(A = a_i, B = b_i) \log_2 p(B = b_i | A = a_i). && (1.7) \end{aligned}$$

L'*information mutuelle* $I(A; B)$, ou de façon plus compacte I_{AB}, est une mesure de la dépendance statistique entre deux variables aléatoires. Elle s'exprime :

$$I(A; B) = H(B) - H(B|A) = H(A) - H(A|B) = H(A) + H(B) - H(A, B). \tag{1.8}$$

L'information mutuelle est une grandeur positive. De plus, si les variables A et B sont indépendantes, alors $H(B|A) = H(B)$ donc $I(A;B) = 0$.

Munis de la notion d'entropie, nous pouvons énoncer les deux théorèmes de Shannon [16] :

Théorème du codage source k réalisations d'une variable aléatoire X d'entropie $H(X)$ peuvent être compressées en $kH(X)$ bits avec une probabilité arbitrairement petite de perdre de l'information. Si on tente de compresser davantage la source aléatoire, on est pratiquement assuré de perdre de l'information.

Théorème du codage de canal une variable aléatoire A est transmise à travers un canal de communication bruité qui transforme cette variable aléatoire en une autre variable aléatoire B. On définit la capacité du canal par

$$C = \max_{p(A)} I_{AB}. \tag{1.9}$$

Alors, il existe des techniques de codage qui permettent de transmettre sans erreur C bits par symbole transmis par le canal. De plus, il est impossible de transmettre sans erreur plus de C bits par symbole.

Les limites proposées par Shannon sont difficiles à atteindre avec des techniques de codage réalistes. En pratique, nous devrons nous contenter d'algorithmes sub-optimaux.

1.4 Information classique et variables gaussiennes

Comme nous le verrons dans la section suivante, le protocole de distribution quantique de clé avec des états cohérents manipule des variables aléatoires gaussiennes, c'est-à-dire des variables continues dont la distribution de probabilité suit la loi :

$$p(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-x_0)^2}{2\sigma^2}}, \tag{1.10}$$

où x_0 est la valeur moyenne, σ l'écart type et $V_X = \sigma^2$ la variance de la distribution. Nous devons donc définir la notion d'information mutuelle pour ces variables.

Un ensemble de variables aléatoires gaussiennes est entièrement caractérisé par ses moments d'ordre deux. Dans l'ensemble de ce manuscrit, nous considérerons uniquement des variables gaussiennes de valeur moyenne nulle. En effet, une simple translation permet de choisir $x_0 = 0$. On regroupe l'ensemble des moments d'ordre deux dans la *matrice de covariance* K, qui s'écrit, pour deux variables aléatoires A et B :

$$K = \begin{bmatrix} V_A & \langle AB \rangle \\ \langle AB \rangle & V_B \end{bmatrix}. \tag{1.11}$$

où $\langle AB \rangle = \int da \int db\, ab\, p(A = a, B = b)$ est la *corrélation* entre les variables A et B.

On généralise la distribution 1.10 à un ensemble de N variables aléatoires gaussiennes notées $\vec{X} = (X_1, \ldots, X_N)$:

$$p(\vec{X} = \vec{x}) = \frac{1}{\sqrt{(2\pi)^N \det(K)}} e^{-\frac{1}{2}(\vec{X}-\vec{X}_0)K^{-1}(\vec{X}-\vec{X}_0)^\top}, \tag{1.12}$$

qui s'écrit, pour deux variables A et B :

$$p(A = a, B = b) = \frac{1}{2\pi\sqrt{V_A V_B - \langle AB \rangle^2}} e^{-\frac{a^2 V_B - 2ab\langle AB \rangle + b^2 V_A}{2(V_A V_B - \langle AB \rangle^2)}}. \tag{1.13}$$

La notion d'entropie s'étend aux variables aléatoires continues. De façon analogue à l'entropie H d'une variable aléatoire discrète, on définit l'entropie différentielle S d'une variable aléatoire continue X :

$$S(X) = -\int p(X = x_i) \log_2 p(X = x_i) dx. \tag{1.14}$$

Cette entropie ne quantifie pas le nombre de bits que l'on peut encoder par symbole aléatoire, car une variable continue peut encoder une infinité de bits grâce à sa résolution infiniment fine. Toutefois, les intuitions que nous avons développées dans le cadre des variables discrètes redeviendront valables quand nous étendrons la notion d'information mutuelle aux variables continues. En effet, l'information mutuelle quantifie le degré de dépendance, ou corrélation, entre deux variables aléatoires. Or, une corrélation imparfaite est synonyme de bruit qui vient limiter la résolution d'un nombre réel.

Pour une variable aléatoire gaussienne, on calcule analytiquement l'entropie différentielle :

$$\begin{aligned} S(X) &= -\int p(x) \log_2 \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) dx - \int -\log_2(e) \frac{x^2}{2\sigma^2} p(x) dx & (1.15) \\ &= \frac{1}{2} \log_2(2\pi\sigma^2) + \frac{1}{2} \log_2(e) & (1.16) \\ &= \frac{1}{2} \log_2(2\pi e \sigma^2). & (1.17) \end{aligned}$$

Un calcul identique appliqué à la distribution 1.12 donne l'entropie conjointe de N variables gaussiennes corrélées de matrice de covariance K :

$$S(X_1, \ldots, X_N) = \frac{1}{2} \log_2 \left((2\pi e)^N \det(K) \right). \tag{1.18}$$

Avec cette expression, on peut calculer l'entropie conditionnelle pour deux variables aléatoires gaussiennes corrélées :

$$S(B|A) = S(A, B) - S(A) = \frac{1}{2} \log_2 \left(2\pi e \frac{\det(K)}{V_A} \right). \tag{1.19}$$

Dans cette expression, on définit la *variance conditionnelle* $V_{B|A}$ par :

$$V_{B|A} = \frac{\det(K)}{V_A} = V_B - \frac{\langle AB \rangle^2}{V_A}. \tag{1.20}$$

Le variance conditionnelle a une interprétation simple. Supposons que nous connaissions une réalisation a de la variable aléatoire A. L'équation 1.13 nous permet alors de calculer la distribution de probabilité de la réalisation b de la variable B associée, connaissant a :

$$\begin{aligned} p(B = b|A = a) &= \frac{p(A = a, B = b)}{p(A = a)} & (1.21) \\ &= \frac{1}{\sqrt{2\pi V_{B|A}}} e^{-\frac{\left(b - \frac{\langle AB \rangle}{V_A} a\right)^2}{2V_{B|A}}}. & (1.22) \end{aligned}$$

On constate que la variance conditionnelle est la variance de notre incertitude sur la valeur de b connaissant a. Cette variance est identique pour toutes les réalisations a de la variable aléatoire

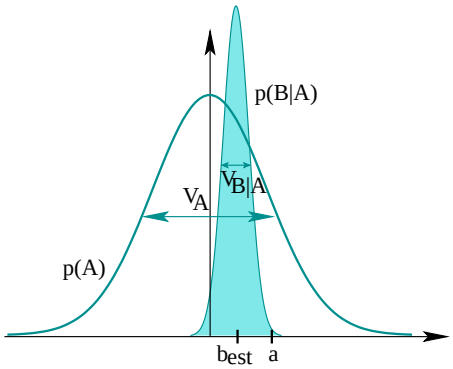


FIG. 1.1: Estimateur gaussien. On considère deux variables aléatoires gaussiennes corrélées. On connaît la réalisation a d'une de ces variables. La grandeur $b_{\text{est}} = \frac{\langle AB \rangle}{V_A} a$ est la meilleure estimation de la réalisation b de la variable B que nous puissions faire, connaissant a. Notre incertitude sur cette estimation est la variance conditionnelle $V_{B|A}$.

A (en d'autres termes : $\forall a, S(B|A) = S(B|A = a)$). La valeur moyenne de la distribution $p(B = b|A = a)$ est la valeur la plus probable pour b, connaissant a. On l'appelle l'«estimateur» (noté b_{est}), et on l'écrit :

$$b_{\text{est}} = \frac{\langle AB \rangle}{V_A} a. \tag{1.23}$$

Finalement, nous pouvons exprimer la variable b à partir de a :

$$b = b_{\text{est}} + \delta b, \tag{1.24}$$

où δb est un bruit gaussien de variance $V_{B|A}$. La figure 1.1 illustre ces notions.

Enfin, nous étendons la notion d'information mutuelle à deux variables gaussiennes aléatoires corrélées. Cette information mutuelle traduit le nombre de «bits communs» aux variables A et B :

$$\begin{aligned} I_{AB} &= S(B) - S(B|A) = \frac{1}{2}\log_2\left(\frac{V_B}{V_{B|A}}\right) && (1.25)\\ &= S(A) - S(A|B) = \frac{1}{2}\log_2\left(\frac{V_A}{V_{A|B}}\right) && (1.26)\\ &= S(A) + S(B) - S(A,B) = \frac{1}{2}\log_2\left(\frac{V_A V_B}{\det(K)}\right). && (1.27) \end{aligned}$$

Les concepts d'information mutuelle et de variance conditionnelle sont à la base des résultats théoriques prouvant la sécurité des protocoles de distribution quantique de clé utilisant des variables continues. Le chapitre suivant est consacré à l'étude de ces protocoles.

Chapitre 2

Distribution quantique de clé avec des états cohérents

Nous rassemblons dans ce chapitre les résultats théoriques obtenus par Frédéric Grosshans [4, 5, 14] concernant les protocoles de cryptographie quantique avec des variables continues. Ces résultats démontrent la possibilité de faire une distribution quantique de clé avec des états cohérents, et s'étendent aux états comprimés. Nous nous limiterons ici au cas pratique des protocoles à états cohérents. Ce chapitre définit les variables quantiques utilisées pour encoder l'information, puis définit le modèle du canal quantique gaussien transportant ces variables entre Alice et Bob. Dans ce modèle, nous déterminerons les taux d'information secrète générés par l'échange quantique. Nous terminerons ce chapitre par une comparaison entre les protocoles à états cohérents décrits dans ce chapitre et les protocoles à photons uniques usuels, de type BB84 [11].

2.1 Variables quantiques continues

Le protocole de distribution quantique de clé avec des états cohérents se distingue des protocoles utilisant des photons uniques par l'encodage de l'information dans des variables quantiques continues transmises sur le canal quantique. Ces variables, traditionnellement représentées par des opérateurs $\hat{X}$ et $\hat{P}$ en référence à la position et à l'impulsion d'un oscillateur harmonique, sont caractérisées par la relation de commutation

$$\left[\hat{X}, \hat{P}\right] = 2iN_0, \tag{2.1}$$

où N_0 est une constante de normalisation fonction de la dimension des variables $\hat{X}$ et $\hat{P}$. Ce commutateur est associé à une relation d'incertitude de Heisenberg qui traduit l'impossibilité de mesurer les deux variables $\hat{X}$ et $\hat{P}$ à la fois avec une précision arbitraire :

$$\Delta X \Delta P \geq N_0, \tag{2.2}$$

où ΔX (resp. ΔP) est l'incertitude sur la mesure de $\hat{X}$ (resp. $\hat{P}$).

La partie scalaire du champ électrique classique d'un mode de la lumière est caractérisée par son amplitude A et sa phase ϕ. De façon équivalente, nous pouvons exprimer les quadratures classiques :

$$X = A\cos(\phi) \tag{2.3}$$

$$\text{et} \quad P = A\sin(\phi), \tag{2.4}$$

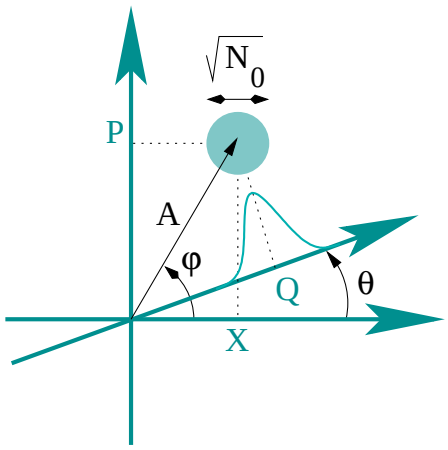


FIG. 2.1: Espace des phases dans lequel on repère un mode du champ électromagnétique par ses quadratures X et P, ou toute quadrature Q_θ. La quantification des variables de quadrature conduit à une zone d'incertitude de variance N_0 autour de la valeur classique d'une mesure de quadrature, appelée «bruit de photon».

qui sont les coordonnées cartésiennes associées aux variables A et ϕ dans l'espace des phases, ou encore toute quadrature arbitraire associée à une phase θ :

$$Q_\theta = A\cos(\phi - \theta) \tag{2.5}$$
$$= \cos(\theta)X + \sin(\theta)P. \tag{2.6}$$

La quantification du champ électromagnétique impose la relation de commutation 2.1 sur les quadratures quantiques $\hat{X}$ et $\hat{P}$ d'un mode de la lumière associées aux quadratures classiques X et P. Ce sont ces variables que nous allons utiliser pour encoder l'information quantique transmise entre Alice et Bob.

On représente un état quantique dans l'espace des phases par une densité de quasi-probabilité, la fonction de Wigner $W(x,p)$, telle que la probabilité associée au résultat q de la mesure d'une quadrature quelconque $\hat{Q}$ soit la projection de cette fonction sur la quadrature $\hat{Q}$ (figure 2.1). On écrit mathématiquement cette propriété [17] :

$$p(q) = \int W(R(q,q^\perp))dq^\perp, \tag{2.7}$$

où $q^\perp$ est une valeur de la mesure de la quadrature $\hat{Q}^\perp$ orthogonale à $\hat{Q}$, et R est la rotation d'angle θ permettant de transformer les quadratures $(\hat{Q},\hat{Q}^\perp)$ en les quadratures $(\hat{X},\hat{P})$.

Les états que nous utilisons en pratique sont des *états gaussiens*, c'est-à-dire des états dont la fonction de Wigner est une gaussienne à deux dimensions dans l'espace des phases. Ces états sont en effet les seuls états stables face à l'interaction du champ avec l'environnement (par exemple face aux pertes). Un état cohérent est un état gaussien dont la variance de la fonction de Wigner est N_0 selon toute quadrature $\hat{Q}$. Cet état, que nous représentons dans l'espace des phases par un cercle de diamètre $\sqrt{N_0}$ (figure 2.1), sature la relation de Heisenberg 2.2. Le point où la fonction de Wigner est maximale est la valeur classique du champ associée à l'état cohérent : la valeur moyenne d'une suite de mesures de la quadrature $\hat{Q}$ est la projection

de ce point sur la quadrature $\hat{Q}$. Autour de cette valeur moyenne, les mesures de quadrature sont réparties avec une distribution gaussienne de variance N_0. Cette dispersion est appelée le «bruit de photon». Comme la variance N_0 est une constante, le bruit de photon est de variance identique, quelle que soit l'amplitude de l'état cohérent. Notamment, si un état cohérent est atténué, son amplitude diminue mais son bruit reste constant : il y a une dégradation du rapport signal à bruit. Ce phénomène est l'ingrédient essentiel qui permet la sécurité des protocoles utilisant des états cohérents.

D'autres états gaussiens ne saturent pas l'inégalité de Heisenberg 2.2. On dit qu'ils ne sont pas limités au bruit de photon. Un état gaussien dont la variance du bruit est N_0 sur une quadrature, et est supérieure à N_0 sur l'autre quadrature est un exemple d'un tel état. Le bruit au-delà du bruit de photon est appelé «excès de bruit». Ce bruit est un bruit dit «classique» : contrairement au bruit de photon, son écart type est proportionnel à l'amplitude de l'état ; en d'autres termes, sa variance est proportionnelle à l'intensité de l'état. Notamment, si on atténue «suffisamment» un état présentant de l'excès de bruit, cet excès de bruit devient négligeable devant le bruit de photon et l'état devient un état cohérent. Ainsi, nous générons expérimentalement des états cohérents en atténuant une source laser. Nous quantifierons cet excès de bruit au chapitre 7.

2.2 Modèle du canal gaussien

Le canal gaussien modélise la transmission quantique des variables continues entre Alice et Bob. Il exprime les perturbations subies par un état cohérent envoyé par Alice dans un canal quantique ajoutant un bruit gaussien. Dans ce modèle, nous notons (X_A, P_A) la valeur classique de la modulation choisie par Alice, c'est-à-dire le centre de l'état cohérent envoyé dans le canal quantique. Les opérateurs quantiques de l'état sortant du dispositif d'Alice s'expriment donc[1] :

$$X = X_A + X_0 \tag{2.8}$$
$$P = P_A + P_0, \tag{2.9}$$

où X_0 et P_0 sont les quadratures du bruit de photon de variance N_0 associé à l'état cohérent. L'état quantique mesuré par Bob est modifié par le canal quantique éventuellement contrôlé par l'espion, et par les imperfections de la détection de Bob. On modélise ces perturbations par l'introduction d'un bruit gaussien[2] X_{CB} et par une transmission $g < 1$. On supposera que ces perturbations sont symétriques en les quadratures X et P. On écrit alors la quadrature X_B mesurée par Bob :

$$X_B = g(X + X_{CB}) = g(X_A + X_0 + X_{CB}) \equiv g(X_A + X_N), \tag{2.10}$$

où $G = g^2 = T\eta \leq 1$ est la transmission totale en intensité entre Alice et Bob, produit de la transmission T du canal quantique et de l'efficacité η de la détection de Bob. On appelle le bruit X_N le *bruit ramené à l'entrée*, sa variance est notée $V_N N_0$. Il est composé du bruit

[1] Jusqu'alors, nous avons désigné les opérateurs quantiques par l'adjonction d'un chapeau ($\hat{\cdot}$), pour les distinguer des variables classiques. Pour alléger la lecture, nous omettons à présent cette notation. Dans ce chapitre, seules les grandeurs X_A et P_A sont classiques.

[2] La notation X_{CB} indique que le bruit ajouté par le canal est sur la quadrature X, et qu'il est sur le signal mesuré par Bob, ramené à l'entrée.

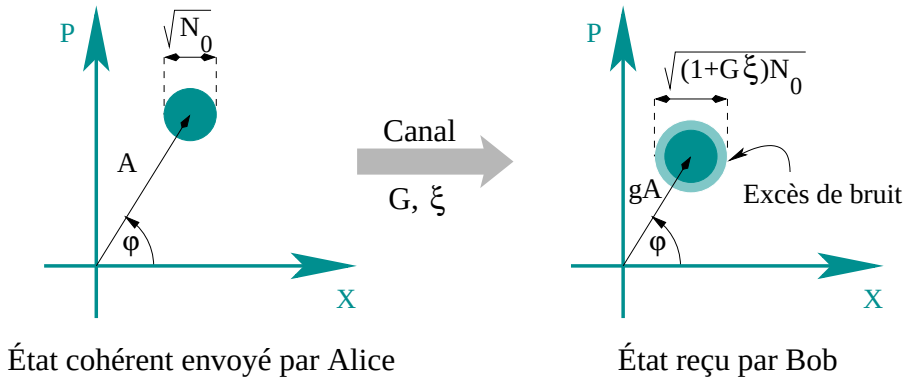


FIG. 2.2: Altération d'un état cohérent par un canal quantique gaussien de transmission $G = g^2$ et qui présente un excès de bruit ξ.

de photon X_0 initialement présent sur le signal envoyé par Alice, et du bruit X_{CB} ajouté par le canal quantique reliant Alice et Bob. On appelle ce dernier bruit le «*bruit ajouté* ramené à l'entrée» et on note sa variance χN_0. Comme les bruits gaussiens s'ajoutent en variance, on a

$$\begin{array}{ccccc} & \langle X_N^2 \rangle & = & \langle X_{CB}^2 \rangle & + & \langle X_0^2 \rangle \\ \Leftrightarrow & \underbrace{V_N}_{\substack{\text{bruit} \\ \text{ramené à l'entrée}}} & = & \underbrace{\chi}_{\substack{\text{bruit ajouté} \\ \text{ramené à l'entrée}}} & + & \underbrace{1}_{\substack{\text{unité de bruit de photon} \\ \text{initiale}}} \end{array} \tag{2.11}$$

Si le signal mesuré par Bob est limité au bruit de photon, on a $GV_N = 1$, donc $\chi = \frac{1}{G} - 1 \equiv \chi_0$. Ce bruit χ_0 est le *bruit ajouté dû aux pertes*. Il traduit le fait que les pertes en ligne font diminuer l'amplitude du signal sans diminuer la variance du bruit de photon.

Si la variance GV_N du bruit observé par Bob est supérieure au bruit de photon, on dit qu'il y a présence d'*excès de bruit*. La variance de ce bruit ramené à l'entrée est notée ξN_0. Finalement, on exprime le bruit ajouté par le canal quantique :

$$\underbrace{\chi}_{\text{bruit ajouté ramené à l'entrée}} = \underbrace{\chi_0}_{\text{bruit ajouté dû aux pertes}} + \underbrace{\xi}_{\text{excès de bruit}} \quad \text{avec} \quad \chi_0 = \frac{1}{G} - 1. \tag{2.12}$$

Le canal gaussien est donc caractérisé par sa transmission G et son bruit ajouté χ, ou de façon équivalente par son transmission G et son excès de bruit ξ. Ces deux derniers paramètres sont illustrés figure 2.2.

2.3 Informations mutuelles et protocoles à états cohérents

Nous supposons maintenant qu'Alice envoie une série d'états cohérents dans le canal quantique, répartis avec une modulation gaussienne dans les deux quadratures X_A et P_A, de variance $V_A N_0$. Cette modulation est l'encodage de l'information sur les états cohérents envoyés dans le canal quantique. De son côté, Bob mesure aléatoirement une quadrature X ou P de chaque état cohérent reçu. Après la mesure, Bob révèle publiquement la quadrature qu'il a choisie. On définit ainsi un protocole de communication quantique

à l'issue duquel Alice et Bob partagent une suite de variables continues corrélées. La suite de ce chapitre est consacrée à l'étude de la sécurité de ce protocole de communication utilisant des états cohérents. Nous évaluerons notamment la quantité d'information secrète, c'est-à-dire l'information inconnue de l'espion, fournie par ce protocole. L'obtention de cette information est ensuite assurée par une étape classique de distillation d'une clé secrète.

Nous pouvons simplement exprimer l'information mutuelle entre Alice et Bob I_{AB} contenue dans cette modulation gaussienne transmise par un canal gaussien grâce au théorème de Shannon :

$$I_{AB} = \frac{1}{2}\log_2(1+SNR) \tag{2.13}$$

$$= \frac{1}{2}\log_2\left(1+\frac{V_A}{V_N}\right), \tag{2.14}$$

où nous avons introduit le rapport signal à bruit $SNR = V_A/V_N$.

Démonstration.

$$V_B = G(V_A+V_N) \quad \text{et} \quad \langle X_A X_B\rangle^2 = GV_A^2 \tag{2.15}$$

$$\text{d'où} \quad V_{B|A} = V_B - \frac{\langle X_A X_B\rangle^2}{V_A} = GV_N \tag{2.16}$$

$$\text{puis} \quad I_{AB} = \frac{1}{2}\log_2\left(\frac{V_B}{V_{B|A}}\right) = \frac{1}{2}\log_2\left(1+\frac{V_A}{V_N}\right) \tag{2.17}$$

□

Avec nos paramètres du canal gaussien, nous exprimons :

$$I_{AB} = \frac{1}{2}\log_2\left(1+\frac{V_A}{1+\chi}\right). \tag{2.18}$$

Comme nous l'avons vu au chapitre précédent, cette information mutuelle est la limite de Shannon. Les chapitres 12 et 13 sont consacrés à une méthode pratique pour extraire cette information I_{AB} à partir des données expérimentales.

L'information I_{AE} est l'information mutuelle entre Alice et Ève. Elle quantifie la connaissance d'Ève sur la modulation d'Alice. Toujours dans le cadre du modèle du canal gaussien, on modélise la transmission entre Alice et Ève par un bruit ajouté X_{CE} (resp. P_{CE}) sur la quadrature X (resp. P) et une transmission en amplitude g_E :

$$X_E = g_E(X+X_{CE}) \tag{2.19}$$

$$P_E = g_E(P+P_{CE}). \tag{2.20}$$

Pour déterminer I_{AE}, on calcule le commutateur entre les variables quantiques X_B et P_E :

$$[X_B, P_E] = gg_E[X+X_{CB}, P+P_{CE}] \tag{2.21}$$

$$= gg_E\left([X,P]+[X_{CB},P_{CE}]\right), \tag{2.22}$$

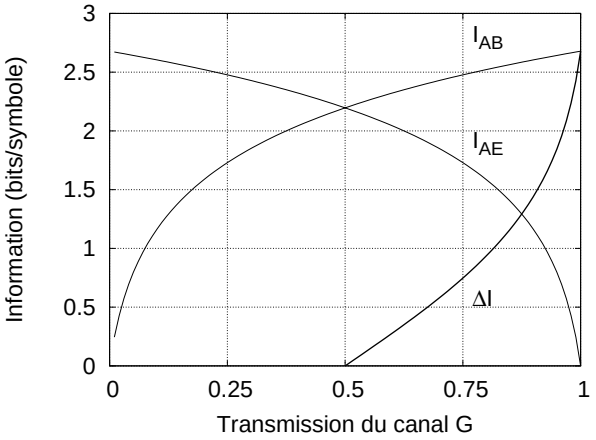


FIG. 2.3: Information secrète dans le cas d'un protocole direct, pour $V_A = 40N_0$ et $\xi = 0$. L'information secrète ΔI n'est positive que pour les transmissions supérieures à 1/2.

car les bruits introduits par les canaux sont indépendants de la modulation. De plus, les modes mesurés par Ève et Bob sont distincts, donc le commutateur $[X_B, P_E]$ est nul. Finalement, $[X, P] = -[X_{CB}, P_{CE}] = 2iN_0$, d'où la relation de Heisenberg :

$$\chi\chi_E \geq 1 \tag{2.23}$$

où χ_E est la variance de P_{CE}. Cette équation indique que le bruit introduit sur le canal d'Ève est borné inférieurement par $\frac{1}{\chi}$, où χ est le bruit ajouté par le canal sur la mesure de Bob. Connaissant le bruit χ, Alice et Bob peuvent donc borner supérieurement l'information mutuelle entre Alice et Ève :

$$I_{AE} = \frac{1}{2}\log_2\left(1 + \frac{V_A}{1+\chi_E}\right) \leq I_{AE}^{\max} = \frac{1}{2}\log_2\left(1 + \frac{V_A}{1+\frac{1}{\chi}}\right). \tag{2.24}$$

Enfin, selon les résultats de Csiszar et Körner [12], Alice et Bob disposent d'une borne inférieure sur l'information secrète transmise par le canal quantique :

$$\Delta I = I_{AB} - I_{AE} \geq I_{AB} - I_{AE}^{\max}. \tag{2.25}$$

La figure 2.3 trace les informations mutuelles et secrète. On constate que la transmission minimale du canal quantique permettant l'échange d'un secret entre Alice et Bob est $G = \frac{1}{2-\xi} \geq \frac{1}{2}$. Les calculs réalisés dans cette section supposent un canal symétrique en les quadratures X et P. Pour que cette condition soit vérifiée, il faut que l'attaque optimale pour Ève soit elle aussi symétrique en ces deux quadratures. C'est le cas si Bob ne privilégie pas l'une des deux quadratures lors de sa mesure ; pour ce faire, il doit mesurer une quadrature différente pour chaque état cohérent qui lui parvient. Après ses mesures, Bob révèle publiquement les bases qu'il a choisies. Ainsi, Ève ne sait pas *a priori* quelle quadrature Bob aura choisi de mesurer, et sera contrainte de faire une attaque symétrique. Nous verrons au chapitre suivant un autre protocole de distribution quantique de clé dans lequel Bob peut mesurer les deux quadratures de chaque état cohérent à la fois.

2.4 Protocole inverse

Le protocole que nous avons étudié dans la section précédente utilise la modulation choisie par Alice comme base de la clé secrète : Bob et Ève essaient chacun de deviner ce qu'Alice a envoyé. Nous l'appelons «protocole direct». Si la transmission du canal quantique est inférieure à 1/2, Ève obtient inévitablement plus d'information que Bob sur la clé d'Alice, empêchant ainsi toute transmission secrète.

Le protocole inverse permettent de dépasser cette limite. Son schéma optique est identique à celui du protocole direct : Alice envoie une série d'états cohérents avec une modulation gaussienne dans le plan complexe. Bob, de son côté, mesure une quadrature aléatoire du signal. La différence réside dans le traitement des données : la clé secrète est construite à partir des données mesurées par Bob. Ainsi, le théorème de Csiszar et Körner nous permet de calculer le taux secret :

$$\Delta I = I_{AB} - I_{BE}. \tag{2.26}$$

Pour borner inférieurement l'information secrète, il nous faut donc trouver une borne supérieure pour I_{BE}. On montre (voir section 4.1 et références [4, 18]) à l'aide d'inégalités de Heisenberg que la variance conditionnelle traduisant l'incertitude d'Ève sur la mesure de Bob est bornée par

$$V_{B|E} \geq V_{B|E}^{\min} = \frac{1}{G(\chi + \frac{1}{V})} N_0, \tag{2.27}$$

avec $V = V_A + 1$. D'où

$$I_{BE} \leq I_{BE}^{\max} = \frac{1}{2}\log_2\left(\frac{V_B}{V_{B|E}^{\min}}\right) = \frac{1}{2}\log_2\left(G^2\left(\chi + V\right)\left(\chi + \frac{1}{V}\right)\right), \tag{2.28}$$

puis

$$\Delta I \geq I_{AB} - I_{BE}^{\max} = -\frac{1}{2}\log_2\left(G^2\left(\chi + 1\right)\left(\chi + \frac{1}{V}\right)\right). \tag{2.29}$$

La figure 2.4 trace les informations mutuelles et secrète pour le cas inverse. On montre [4] que si l'excès de bruit est assez faible ($\xi \leq \frac{1}{2}\left(1 - \frac{1}{V}\right)$ pour $G \ll 1$), alors l'information secrète transmise entre Alice et Bob est positive. Le protocole inverse permet donc de distribuer une clé avec des états cohérents pour toute transmission du canal quantique.

2.5 Attaques optimales

Les informations mutuelles $I_{AE}^{\max}$ et $I_{BE}^{\max}$ que nous avons dérivées dans les sections précédentes sont des bornes supérieures pour l'information accessible à l'espion. On montre que ces bornes sont les meilleures possibles en exhibant une attaque physique qui les atteint. Nous allons aborder les attaques optimales en deux temps. D'abord, nous étudierons le cas simple d'une transmission sans excès de bruit ($\xi = 0$), c'est-à-dire une transmission pour laquelle le bruit ajouté est égale au bruit ajouté dû aux pertes : $\chi = \chi_0 = \frac{1}{G} - 1$. Ensuite, nous aborderons le cas général des transmissions avec excès de bruit.

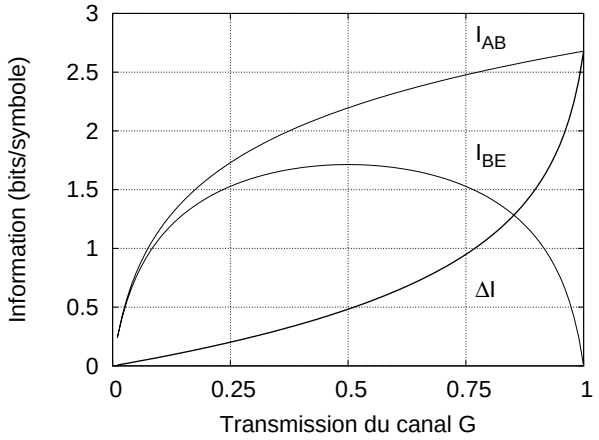


FIG. 2.4: Information secrète produite par un protocole inverse, pour $V_A = 40N_0$ et $\xi = 0$. La transmission d'une clé secrète est possible pour toute transmission du canal quantique.

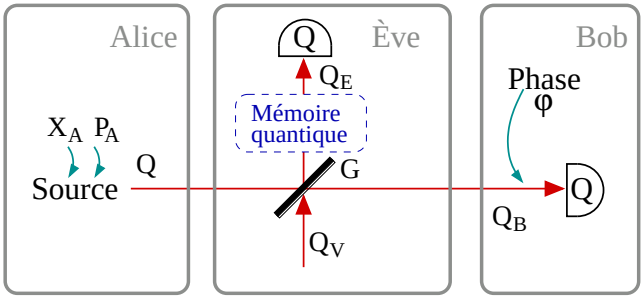


FIG. 2.5: Attaque de type lame séparatrice. Ève prélève une partie du signal par une lame de transmission G. Le mode vide Q_V à l'entrée inférieure de la lame ajoute du bruit sur les signaux de Bob et d'Ève. Les modes en sortie de la lame séparatrice sont toujours limités au bruit de photon : la lame séparatrice n'introduit pas d'excès de bruit. Cette attaque, simple en apparence, est techniquement difficile : Ève doit remplacer le canal quantique par un canal sans perte et doit mémoriser les états quantiques pour mesurer la même quadrature que Bob. En pratique, on simule une attaque de type lame séparatrice en introduisant des pertes entre Alice et Bob (voir chapitres 7 et 8).

En l'absence d'excès de bruit, l'attaque optimale est une attaque de type lame séparatrice. Dans cette attaque, Ève remplace le canal quantique reliant Alice à Bob par un canal idéal sans perte, et prélève une fraction $1 - G$ du signal transmis. Ève enregistre l'état quantique du faisceau prélevé dans une mémoire quantique. Quand Bob révèle ses choix de mesure, Ève lit sa mémoire quantique et mesure pour chaque état la quadrature X ou P choisie par Bob.

Le prélèvement du signal est réalisé par une lame séparatrice de transmission $G = g^2$ représentée figure 2.5, qui fait interférer la quadrature[3] Q envoyée par Alice avec un mode vide Q_V de variance N_0. Ève conserve la quadrature Q_E réfléchie, alors que la quadrature transmise

[3]La quadrature Q désigne ici la quadrature aléatoire mesurée par Bob.

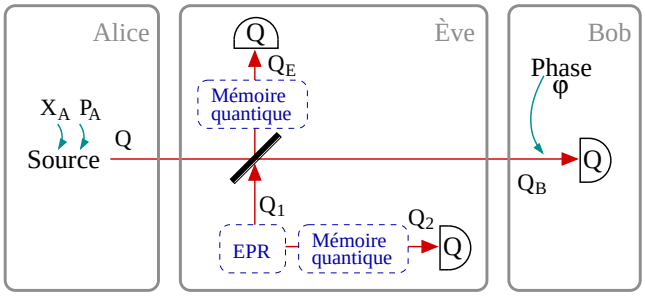


FIG. 2.6: Attaque de type cloneuse intriquante. Pour exploiter l'excès de bruit présent sur le canal, Ève injecte, via une paire EPR, un bruit en partie connu. Cette attaque est la plus générale dans le modèle du canal gaussien qui traite des attaques individuelles gaussiennes. Elle sature les bornes que nous avons établies sur l'information accessible à l'espion.

Q_B est envoyée à Bob. La transformation associée à la lame séparatrice s'écrit [17] :

$$Q_B = \sqrt{G}\,Q - \sqrt{1-G}\,Q_V = \sqrt{G}\,(Q - \sqrt{\chi_0}\,Q_V) \tag{2.30}$$

$$Q_E = \sqrt{1-G}\,Q + \sqrt{G}\,Q_V = \sqrt{1-G}\,(Q + Q_V/\sqrt{\chi_0}). \tag{2.31}$$

La première équation nous montre que la transmission du canal entre Alice et Bob est bien G, et que le bruit ajouté ramené à l'entrée est bien $\chi = \chi_0$, c'est-à-dire $\xi = 0$. Ève a donc correctement simulé le canal quantique sans excès de bruit. La deuxième équation nous permet de calculer le bruit ajouté sur la voie d'Ève $\chi_E = \frac{G}{1-G} = \frac{1}{\chi}$ qui sature l'inégalité 2.23 $\chi\chi_E \geq 1$: il s'agit bien de l'attaque optimale pour le protocole direct.

Pour traiter le cas du protocole inverse, nous calculons la variance conditionnelle $V_{Q_B|Q_E}$ à l'aide de la transformation de la lame séparatrice :

$$V_{Q_B|Q_E} = V_B - \frac{\langle Q_B Q_E\rangle^2}{V_E} = \frac{1}{G(\chi + \frac{1}{V})} N_0 \tag{2.32}$$

$$\text{avec} \quad V_B = \langle Q_B^2\rangle = G(V+\chi)N_0 \tag{2.33}$$

$$V_E = \langle Q_E^2\rangle = (1-G)\left(V + \frac{1}{\chi}\right) N_0 \tag{2.34}$$

$$\langle Q_B Q_E\rangle = \sqrt{G(1-G)}(V-1)N_0. \tag{2.35}$$

Cette variance conditionnelle sature la borne que nous avions établie dans la section précédente pour le protocole inverse : là encore, l'attaque de type lame séparatrice est optimale.

En présence d'excès de bruit, l'attaque optimale est une «cloneuse intriquante». Cette attaque utilise également une lame séparatrice pour simuler les pertes du canal quantique. Pour tirer parti de l'excès de bruit, Ève insère une source de bruit connue par la voie d'entrée vide de la lame séparatrice. Ce bruit est introduit par l'un des deux membres d'une paire EPR. Une paire EPR est un état bimode gaussien corrélé dont les quadratures sont notées Q_1 et Q_2, caractérisé par les moments d'ordre deux [19] :

$$\langle Q_1^2\rangle = \langle Q_2^2\rangle = \mathcal{V}N_0, \quad \langle Q_1 Q_2\rangle = \pm\sqrt{\mathcal{V}^2 - 1}N_0, \tag{2.36}$$

où $\mathcal{V} > 1$ est la variance de la paire EPR, et $\pm$ désigne le signe $+$ si $Q = X$ ou le signe $-$ si $Q = P$. L'autre membre de la paire EPR est enregistré dans une mémoire quantique (figure 2.6).

Ève mesure ensuite la quadrature Q_2 de ce mode après révélation des choix de base de Bob. Grâce aux corrélations entre les membres de la paire EPR, cette mesure donne de l'information à Ève sur l'excès de bruit introduit sur la ligne. En appliquant la transformation de la lame séparatrice (expression 2.30), on exprime les modes mesurés par Ève et Bob :

$$Q_B = \sqrt{G}\,Q - \sqrt{1-G}\,Q_1 = \sqrt{G}(Q - \sqrt{\chi_0}\,Q_1) \tag{2.37}$$
$$Q_E = \sqrt{1-G}\,Q + \sqrt{G}\,Q_1 = \sqrt{1-G}(Q + Q_1/\sqrt{\chi_0}), \tag{2.38}$$

d'où les moments d'ordre deux :

$$V_B = G(V + \chi_0\mathcal{V})N_0 \tag{2.39}$$
$$V_E = (1-G)(V + \mathcal{V}/\chi_0)N_0 \tag{2.40}$$
$$\langle Q_B Q_E\rangle = \sqrt{G(1-G)}(V - \mathcal{V})N_0 \tag{2.41}$$
$$\langle Q_E Q_2\rangle = \pm\sqrt{G}\sqrt{\mathcal{V}^2 - 1}N_0 \tag{2.42}$$
$$\langle Q_B Q_2\rangle = \mp\sqrt{1-G}\sqrt{\mathcal{V}^2 - 1}N_0. \tag{2.43}$$

L'équation 2.37 nous permet de calculer les paramètres du canal gaussien reliant Alice à Bob : le gain du canal est la transmission G de la lame séparatrice et le bruit ajouté χ vaut $\chi_0\mathcal{V}$. Pour simuler le canal quantique, Ève doit donc choisir la variance de sa paire EPR :

$$\mathcal{V} = \frac{\chi}{\chi_0}. \tag{2.44}$$

L'information acquise par Ève provient de ses mesures des deux quadratures Q_E et Q_2. C'est l'information mutuelle, notée $I(Q_E, Q_2; Q_B)$, entre les variables aléatoires Q_E et Q_2 d'une part, et la variable aléatoire Q_B d'autre part :

$$I_{BE} = I(Q_E, Q_2; Q_B) \tag{2.45}$$
$$= S(Q_B) - S(Q_B|Q_E, Q_2) \tag{2.46}$$
$$= S(Q_B) - (S(Q_B, Q_E, Q_2) - S(Q_E, Q_2)) \tag{2.47}$$
$$= \frac{1}{2}\log_2\left(\frac{V_B}{V_{B|E}}\right) \tag{2.48}$$
$$\text{avec} \quad V_{B|E} = \frac{\det(K)}{\det(K_E)}, \tag{2.49}$$

selon l'équation 1.18. K est la matrice de covariance des trois modes Q_B, Q_E et Q_2 regroupant les moments d'ordre deux écrits dans les équations 2.39 à 2.43 :

$$K = \begin{bmatrix} V_B & \langle Q_B Q_E\rangle & \langle Q_B Q_2\rangle \\ \langle Q_B Q_E\rangle & V_E & \langle Q_E Q_2\rangle \\ \langle Q_B Q_2\rangle & \langle Q_E Q_2\rangle & \mathcal{V} \end{bmatrix} \tag{2.50}$$

et K_E est la sous-matrice inférieure droite de K, qui ne considère que les modes Q_E et Q_2 d'Ève. Le calcul des déterminants $\det(K)$ et $\det(K_E)$ donne :

$$V_{B|E} = \frac{1}{G(\chi + \frac{1}{V})}N_0, \tag{2.51}$$

qui sature la borne que nous avions indiquée en section 2.4 : la cloneuse intriquante est donc l'attaque optimale pour le protocole inverse.

On traite le cas direct de la même façon, en considérant la matrice de covariance regroupant les quadratures Q_A, Q_E et Q_2, avec :

$$\langle Q_A Q_E \rangle = \sqrt{1-G} V_A N_0 \tag{2.52}$$

$$\langle Q_A Q_2 \rangle = 0, \tag{2.53}$$

qui donne

$$V_{A|E} = \frac{V_A(\chi+1)}{V\chi+1} N_0 \quad \Rightarrow \quad I_{AE} = \frac{1}{2}\log_2\left(\frac{V_A}{V_{A|E}}\right) = \frac{1}{2}\log_2\left(\frac{V+\frac{1}{\chi}}{1+\frac{1}{\chi}}\right), \tag{2.54}$$

qui est là aussi la borne supérieure sur I_{AE} que nous avions déterminée en section 2.3.

Finalement, nous avons exhibé une attaque inviduelle optimale pour les protocoles direct et inverse, et pour tout canal gaussien reliant Alice à Bob : les bornes supérieures sur les informations I_{AE} et I_{BE} que nous avons déterminées peuvent être atteintes.

Il existe d'autres attaques saturant les bornes sur l'information accessible à l'espion. Pour le cas direct, citons la cloneuse asymétrique [4, 20]. Pour le cas inverse, l'article [21] propose un montage utilisant un amplificateur optique.

2.6 Implications expérimentales

Résumons les étapes physiques du protocole de distribution quantique de clé dont nous venons de prouver la sécurité. Pour chaque symbole, Alice choisit deux variables aléatoires gaussiennes X_A et P_A de variance V_A. Elle déplace un état cohérent dans le plan complexe aux coordonnées (X_A, P_A). Cet état est transmis à Bob par un canal quantique bruité. Ce dernier mesure une quadrature aléatoire du signal reçu. À l'issue de la transmission, Alice et Bob possèdent deux chaînes de variables gaussiennes corrélées contenant un taux secret ΔI.

Les protocoles qui utilisent des variables continues sont donc particulièrement adaptés à une réalisation expérimentale dans le domaine des technologies télécom. En effet, la génération des symboles quantiques est une simple modulation gaussienne d'un état cohérent dans le plan complexe : elle peut être réalisée avec une diode laser et des modulateurs électro-optiques ; la mesure d'une quadrature du champ par Bob nécessite une détection homodyne (section 6.3) utilisant des photodiodes PIN efficaces. Tous ces composants standards ont été optimisés par l'industrie des télécommunications, et ont des bandes passantes supérieures à 10 GHz. Cette situation contraste avec les protocoles utilisant des photons uniques, qui nécessitent des compteurs de photons spécifiques et éventuellement des sources de photons uniques.

Toutefois, l'information mutuelle I_{AB} exprimée par l'équation 2.24 est une limite théorique. En pratique, l'obtention de bits à partir de nos variables continues nécessite l'utilisation d'algorithmes complexes. L'étape qui consiste à extraire cette information mutuelle I_{AB} est appelée «réconciliation». On parlera donc de réconciliation directe pour les protocoles directs, et de réconciliation inverse pour les protocoles inverses.

Nous introduisons ainsi les parties II et III de ce manuscrit, respectivement consacrées à la réalisation expérimentale des protocoles à variables continues et à l'extraction de l'information contenue dans ces variables.

2.7 Photons uniques ou variables continues ?

Le protocole de distribution quantique de clé avec des états cohérents n'est pas une simple évolution des protocoles utilisant des photons uniques. Les contraintes techniques sont en effet radicalement différentes. Dans cette section, nous détaillerons ces contraintes et leurs perspectives d'évolution afin d'apporter des éléments de réponse à la question ouverte : «Photons uniques ou variables continues ?».

Pour mener notre comparaison, nous écrivons le taux de clé final K, exprimé en bits par seconde, sous la forme

$$K = f\Delta I \tag{2.55}$$

où f est le taux de répétition des symboles traversant le canal quantique, et ΔI est la quantité d'information secrète que contient chaque symbole. Nous allons comparer individuellement ces deux paramètres pour les protocoles à photons uniques (p.u.) et pour les protocoles à variables continues (v.c.).

Étudions d'abord la quantité ΔI pour une configuration expérimentale parfaite, dans laquelle la seule perte d'information secrète entre Alice et Bob est la transmission T du canal quantique. On écrit :

$$\Delta I_{\text{p.u.}} = \frac{1}{2}T \tag{2.56}$$

$$\Delta I_{\text{c.v.}} = -\frac{1}{2}\log_2\left[T\left(\frac{1}{T} - 1 + \frac{1}{V}\right)\right] \tag{2.57}$$

$$\sim \frac{1}{2\ln(2)}T \simeq 0,72T \quad \text{pour } T \ll 1 \text{ et } V \gg 1. \tag{2.58}$$

L'information $\Delta I_{\text{c.v.}}$ est reproduite de l'équation 2.29 pour $\xi = 0$, et l'information $\Delta I_{\text{p.u.}}$ est simplement la fraction d'impulsions arrivant chez Bob, pondérée d'un facteur 1/2 rendant compte des choix de base incompatibles entre Alice et Bob. Dans la limite où $T \ll 1$, nous voyons que les deux types de protocoles ont un comportement linéaire, avec une pente comparable. On observe donc une décroissance exponentielle du taux en fonction de la distance pour les deux protocoles. En revanche, pour T proche de 1, les protocoles à photons uniques peuvent transmettre au plus 1/2 bit par impulsion, alors que les protocoles à variables continues ont un débit de $\frac{1}{2}\log_2 V$ bits par symbole qui peut être supérieur à 1, et potentiellement arbitrairement grand. C'est une première particularité de ces protocoles : l'exploitation du caractère continu des variables transmises permet de franchir le bit par symbole dans le régime des faibles pertes.

En pratique, la quantité ΔI est dégradée par les réalités expérimentales. Pour les photons uniques, on recense deux types d'imperfections [22] : les unes limitent le *débit* de la distribution de clé, les autres principalement sa *portée*.

Débit Pour obtenir une source de photons uniques, on a usuellement recours à une source cohérente atténuée[4] comportant en moyenne $\mu = 0,1$ photon par impulsion[5]. Ensuite, les

[4]Les sources de photons uniques existantes ont des efficacités comparables au nombre de photons par impulsion des sources atténuées [23].

[5]L'analyse de sécurité des protocoles utilisant des sources atténuées impose des contraintes plus restrictives sur la valeur de μ [24] afin de protéger le protocole face à des attaques de type PNS ("Photon Number Splitting"). Cependant, des récents protocoles ("decoy states" [25]) permettent de choisir $\mu \simeq 0,1$ [26].

détecteurs de photons uniques (photodiodes à avalanche en régime de comptage de photons, commercialisées notamment par les sociétés idQuantique ou Princeton LightWave) ont des efficacités limitées, de l'ordre de $\eta = 0,25$. Ces deux défauts ont pour effet de réduire le taux de clé secrète par symbole d'un facteur $\mu\eta$.

Portée Enfin, les détecteurs de photons uniques peuvent émettre un signal positif même en l'absence de photons. Ces coups d'obscurité (de l'ordre de 10^{-4} par nanoseconde) forment des erreurs sur la transmission entre Alice et Bob, et dégradent l'information secrète $\Delta I_{\text{p.u.}}$. Pour les faibles transmissions (correspondant à une distance d'une centaine de kilomètres), ils deviennent prédominants par rapport à la quantité de photons reçus, et annulent le taux de clé secrète.

On retrouve ces deux types d'imperfections dans les protocoles à variables continues.

Débit Les pertes des détecteurs réduisent le nombre de bits extractibles par symbole. Toutefois, cette réduction est limitée car les photodiodes utilisées ont de bonnes efficacités, de l'ordre 80%.

Portée Nous pouvons ré-écrire le taux d'information secrète $\Delta I = \beta I_{AB} - I_{BE}$, où β est l'efficacité d'extraction des bits contenus dans les variables continues (ou réconciliation, voir chapitres 12 et 13). Alors que $I_{AB} - I_{BE}$ est positif pour toute atténuation du canal entre Alice et Bob, ce n'est plus le cas de $\beta I_{AB} - I_{BE}$ qui devient négatif en dessous d'une certaine transmission du canal. L'efficacité de l'extraction des bits limite donc la portée des protocoles à variables continues. Pour une efficacité $\beta = 0,87$, on trouve une distance maximale de 30 km. On explique cette sensibilité des protocoles à variables continues à l'efficacité d'extraction des bits : pour les faibles gains, $\Delta I \ll I_{AB}, I_{BE}$ (voir courbe 2.4). Ainsi, une petite inefficacité dans l'extraction de l'information mutuelle entre Alice et Bob peut anéantir leur avantage. En revanche les protocoles à photons uniques sont moins sensibles à l'inefficacité de réconciliation car ΔI est plus proche de I_{AB} et I_{BE}. De plus, à transmission donnée, le taux d'erreur des protocoles à variables continues est plus important que pour les photons uniques, ce qui nécessite de meilleurs codes correcteurs d'erreurs. Ce taux d'erreur élevé pour les variables continues a une origine physique : pour les faibles transmissions, les protocoles à photons uniques concentrent l'information utile dans les quelques photons qui parviennent jusqu'à Bob. Les autres photons, perdus en lignes, sont simplement ignorés par le processus de réconciliation. En revanche, l'information transmise par les protocoles à variables continues est répartie équitablement sur chaque impulsion traversant le canal quantique. Pour les faibles transmissions, les protocoles de réconciliation doivent réussir à extraire une petite quantité d'information pour chaque impulsion, avec un mauvais rapport signal à bruit. Cette situation est bien entendu plus difficile à utiliser en pratique.

Le taux de répétition f de l'expérience permet de traduire le taux secret ΔI en bits par seconde. Pour les protocoles à variables continues, nous distinguons trois régimes qui imposent des contraintes différentes sur le taux secret :

- L'optique est exclusivement réalisée avec des composants télécom standards, fonctionnant à des taux de répétition jusqu'à **10 GHz** ;
- L'électronique de la détection homodyne capable de mesurer une quadrature du signal fonctionne jusqu'à **10 MHz**. Une détection homodyne fonctionnant à 100 MHz a toutefois été démontrée [27]. L'électronique d'acquisition de notre dispositif expérimental est limitée à 2 MHz. Toutefois, certaines cartes électroniques, notamment commercialisées par

la société National Instruments, permettent l'acquisition et la modulation analogiques jusqu'à 100 MHz ;

- Dans notre réalisation expérimentale, le traitement des données continues est limité par la puissance des ordinateurs utilisés (Pentium D820). Nous verrons au chapitre 13 que ces ordinateurs permettent d'extraire l'information I_{AB} à un taux de **100 kHz** (c'est-à-dire 100 000 symboles par seconde).

Ce dernier point est le facteur limitant le taux de répétition global de l'expérience. L'utilisation de processeurs dédiés, ou des améliorations dans les algorithmes permettraient cependant d'atteindre des taux de répétition plus élevés.

Dans le cas des photons uniques, le taux de répétition est limité par la technologie des compteurs de photons : après détection d'un photon, ces détecteur souffrent d'un temps mort (typiquement de quelques microsecondes) pendant lequel ils sont inactifs.

Les protocoles à photons uniques et à variables continues présentent à l'heure actuelle des performances comparables, malgré la relative jeunesse des seconds. On remarque en effet que les faibles taux de répétition des protocoles à variables continues imposés par les algorithmes classiques font chuter le taux secret de façon analogue à l'inefficacité des sources et des détecteurs employés pour les protocoles utilisant des photons uniques. Quantitativement, les deux types de protocoles sont capables de délivrer un taux secret de l'ordre de quelques kilobits par seconde à une distance de 25 km. En dessous de de cette distance, les protocoles à variables continues atteignent des taux plus élevés, alors que les protocoles à photons uniques atteignent de plus longues distances. Toutefois, les barrières technologiques à franchir pour dépasser l'état de l'art actuel, résumées dans le tableau suivant, sont de natures bien différentes.

Limitations	**Photons uniques**	**Variables continues**
Portée	Coups d'obscurité des détecteurs	Efficacité des algorithmes de réconciliation
Débit (bits/symbole)	Efficacité des sources et des détecteurs	Efficacité des algorithmes
Taux de répétition (symboles/s)	Temps mort des détecteurs	Vitesse de réconciliation

Nous pouvons ainsi résumer les perspectives d'amélioration : des avancées technologiques (technologies à photons uniques, relais quantiques [28]) permettent d'améliorer les protocoles utilisant des photons uniques, alors que des avancées dans les algorithmes permettent d'améliorer les protocoles utilisant des variables continues. Ces avancées permettraient de rejoindre le taux théorique de l'équation 2.58. À ce moment, une augmentatin du taux de répétition de l'expérience, actuellement limité à 2MHz par l'électronique d'acquisition, devra être envisagée.

Chapitre 3

Distribution quantique de clé avec une détection hétérodyne

Dans ce chapitre, nous considérons un nouveau protocole utilisant des variables continues. La partie émission (Alice) est identique au schéma présenté dans le chapitre précédent. En revanche, à la réception, Bob mesure de façon conjointe les deux quadratures X et P, au lieu de mesurer une quadrature aléatoire. Cette mesure, appelée «hétérodyne», est réalisée en séparant le faisceau provenant du canal quantique par une lame séparatrice, puis en mesurant respectivement les quadratures X et P sur chacune des voies de sortie de la lame séparatrice. Ce schéma est représenté figure 3.1. Comme pour le protocole décrit au chapitre précédent, nous calculerons les taux d'information secrète pour les protocoles direct et inverse. Ensuite, nous évaluerons les avantages pratiques de ce protocole hétérodyne.

Notons que les résultats présentés dans la suite de ce manuscrit sont indépendants des résultats théoriques obtenus dans ce chapitre.

3.1 Information mutuelle I_{AB}

Commençons par évaluer l'information mutuelle I_{AB} pour le protocole hétérodyne dans le cadre du modèle du canal gaussien que nous avons introduit au chapitre précédent. On modélise ce canal par une transmission $G = g^2 < 1$ et l'ajout d'un bruit X_{CB} de variance χN_0 entre la quadrature X_A choisie par Alice et la quadrature X_B reçue par Bob :

$$X_B = g(X_A + X_0 + X_{CB}), \tag{3.1}$$

où X_0 est le bruit de photon de variance N_0 initialement présent sur le signal et X_{CB} est le bruit ajouté par le canal, de variance χN_0. Dans le protocole hétérodyne, la quadrature X'_B mesurée par Bob est issue de l'interférence sur la lame séparatrice de la quadrature X_B avec un mode vide X'_0 de variance N_0. Alors :

$$X'_B = \frac{1}{\sqrt{2}}(X_B - X'_0) = \frac{1}{\sqrt{2}}g(X_A + X_0 + X_{CB} - \frac{1}{g}X'_0). \tag{3.2}$$

La variance totale du bruit ramené à l'entrée s'exprime donc :

$$V'_N = \left(1 + \chi + \frac{1}{G}\right) N_0. \tag{3.3}$$

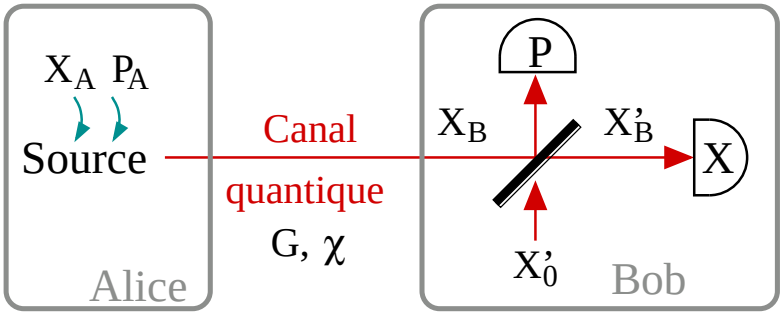


FIG. 3.1: Schéma de principe d'un protocole hétérodyne. Alice envoie à Bob une suite d'états cohérents modulés dans le plan complexe avec une variance V_A. Ce signal est modifié par la transmission G et le bruit ajouté ramené à l'entrée χ du canal quantique. Après réception, Bob divise le faisceau en deux parties et mesure respectivement les quadratures X et P sur chacune des voies. Chacune de ces mesures, symbolisées par un détecteur sur le schéma, est une mesure homodyne qui fait intervenir un faisceau oscillateur local non représenté.

Enfin, en reproduisant ce raisonnement avec la quadrature P, on déduit l'information mutuelle entre Alice et Bob :

$$\begin{aligned} I_{AB} &= 2 \times \frac{1}{2} \log_2 (1 + SNR) && (3.4) \\ &= \log_2 \left(1 + \frac{V_A}{V'_N} \right) && (3.5) \\ &= \log_2 \left(1 + \frac{V_A}{1 + \chi + \frac{1}{G}} \right). && (3.6) \end{aligned}$$

On constate deux différences avec le protocole homodyne. D'abord, l'unité de bruit de photon introduite par la lame séparatrice fait décroître l'information mutuelle. En contrepartie, la mesure des deux quadratures double l'information disponible. On montre simplement que pour toutes valeurs de G et de χ, le deuxième effet l'emporte sur le premier :

$$I_{AB}^{\text{hétérodyne}} > I_{AB}^{\text{homodyne}}. \tag{3.7}$$

Maintenant que nous avons calculé l'information mutuelle I_{AB}, nous devons estimer l'information acquise par l'espion pour déterminer le taux secret ΔI. Ainsi, nous allons borner l'information I_{BE} dans le cadre du protocole inverse (section 3.2), puis l'information I_{AE} dans le cadre du protocole direct (section 3.3).

3.2 Protocole inverse

Pour borner l'information accessible à l'espion dans le cadre du protocole inverse, nous imposons une restriction sur les pouvoirs qui lui sont accordés : nous supposons que le bruit X'_0 introduit par le détecteur de Bob est inconnu et incontrôlé par Ève. Nous rencontrerons à nouveau cette hypothèse dite «réaliste» au chapitre 7. Elle suppose qu'Ève n'a accès qu'à l'information qui transite sur le canal quantique, et qu'elle ne peut pas contrôler les systèmes d'émission d'Alice et de réception de Bob. Si elle le pouvait, elle pourrait simplement

lire la modulation choisie par Alice ou le signal mesuré par Bob, et le protocole ne fonctionnerait plus. Cette hypothèse est donc très générale et s'applique en pratique à tout protocole de cryptographie quantique.

Nous avions borné à la section 2.4 la variance conditionnelle $V_{B|E}$ pour le protocole homodyne :

$$V_{B|E} \geq \frac{1}{G\left(\chi + \frac{1}{V}\right)} N_0. \tag{3.8}$$

Nous en déduisons une borne inférieure sur la variance conditionnelle $V_{B'|E}$ pour le protocole hétérodyne :

$$V_{B'|E} = \frac{1}{2}\left(N_0 + V_{B|E}\right) \geq \frac{1}{2}\left(1 + \frac{1}{G\left(\chi + \frac{1}{V}\right)}\right) N_0. \tag{3.9}$$

Démonstration.

$$\begin{aligned} V_{B'|E} &= V'_B - \frac{\langle X_E X'_B \rangle^2}{V_E} && (3.10)\\ &= \frac{1}{2}\left(V_B + N_0\right) - \frac{1}{2}\frac{\langle X_E X_B \rangle^2}{V_E} \quad \text{car} \quad \begin{array}{l} X'_B = \frac{1}{\sqrt{2}}(X_B - X'_0) \\ \text{et } \langle X_E X'_0 \rangle = 0 \text{ (hypothèse réaliste)} \end{array} && (3.11)\\ &= \frac{1}{2}\left(N_0 + V_{B|E}\right). && (3.12) \end{aligned}$$

□

Nous reconnaissons dans cette expression l'unité de bruit de photon N_0 introduite par la lame séparatrice, et la diminution de l'intensité du signal d'un facteur deux.

Pour $\xi > 1 - 1/V$, nous devons ajouter une autre relation de Heisenberg :

$$V_{B'|E_X} \cdot V_{B'|E_P} \geq 1 \;\Rightarrow\; V_{B'|E} \geq 1. \tag{3.13}$$

Dans le cas pratique où $\xi < 1 - 1/V$, nous en déduisons une borne supérieure pour l'information accessible à l'espion I_{BE} et une borne inférieure pour l'information secrète ΔI :

$$\begin{aligned} I_{BE} &= 2 \times \tfrac{1}{2}\log_2\left(\tfrac{V_{B'}}{V_{B'|E}}\right) \\ &\leq \log_2\left[\frac{G\left(V+\chi+\frac{1}{G}\right)}{\frac{1}{G\left(\chi+\frac{1}{V}\right)}+1}\right] \end{aligned}, \quad \text{puis} \quad \begin{aligned} \Delta I &= I_{AB} - I_{BE} \\ &\geq -\log_2\left[\frac{G\left(1+\chi+\frac{1}{G}\right)}{\frac{1}{G\left(\chi+\frac{1}{V}\right)}+1}\right]. \end{aligned} \tag{3.14}$$

Nous représentons les informations mutuelles sur la figure 3.2. Le taux secret hétérodyne est supérieur au taux secret homodyne pour toute transmission du canal. Toutefois, la différence entre les deux protocoles diminue quand l'atténuation du canal augmente.

Nous verrons à la section 3.4 que nous ne connaissons pas d'attaque physique qui atteigne la borne que nous venons d'établir sur I_{BE}. Ainsi, il est possible qu'il existe une borne plus contraignante sur cette information. Mais d'abord, examinons le cas du protocole direct.

3.3 Protocole direct

Nous cherchons maintenant à borner l'information I_{AE} pour le protocole direct. Pour cela, commençons par modéliser une classe d'attaques restreinte dans laquelle

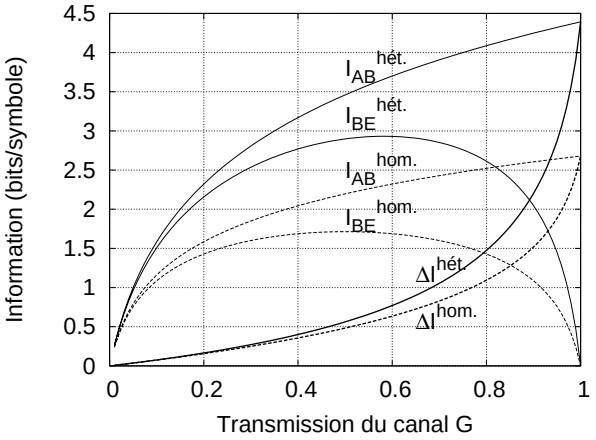


FIG. 3.2: Informations mutuelles pour une mesure hétérodyne en réconciliation inverse (courbes pleines), pour $V_A = 40N_0$ et $\xi = 0$. Ces informations sont comparées à celles obtenues avec une mesure homodyne (courbes pointillées). Les courbes grasses sont les informations secrètes : la mesure hétérodyne produit davantage de bits secrets que la mesure homodyne.

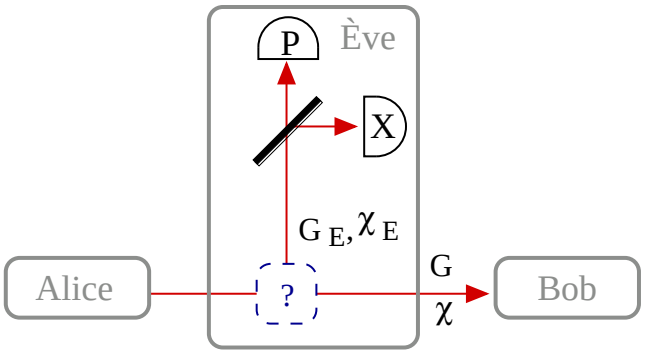


FIG. 3.3: Modèle d'attaque hétérodyne. Ève fait une mesure hétérodyne sur une copie du faisceau traversant le canal quantique. On note G_E et χ_E les gain et bruit ajouté en entrée de sa mesure hétérodyne.

Ève prélève une partie du signal traversant le canal quantique avec un dispositif arbitraire. Sur ce faisceau prélevé, Ève fait une mesure hétérodyne des quadratures X et P, comme représenté figure 3.3. Comme nous avions exprimé I_{AB} en fonction des paramètres du canal G et χ, nous pouvons exprimer I_{AE} en fonction des paramètres G_E et χ_E du canal vu par Ève :

$$I_{AE} = \log_2 \frac{V + \chi_E + \frac{1}{G_E}}{1 + \chi_E + \frac{1}{G_E}}. \tag{3.15}$$

Pour borner supérieurement cette information mutuelle, nous écrivons la relation de Heisenberg sur les bruits ajoutés, comme nous l'avions fait pour le cas homodyne (équation 2.23) :

$$\chi_E > \frac{1}{\chi}. \tag{3.16}$$

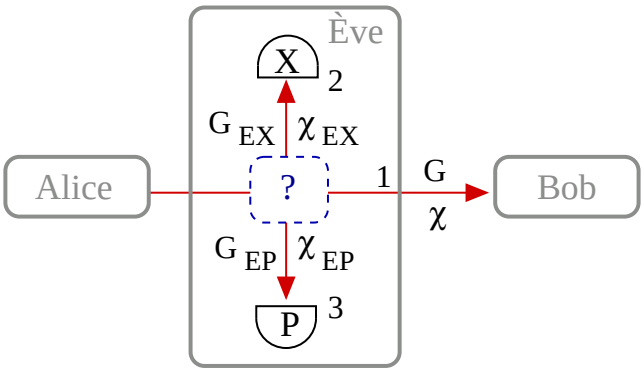


FIG. 3.4: Modèle d'attaque générale dans laquelle Ève réalise une cloneuse asymétrique $1 \to 3$

L'information I_{AE} est maximale quand $\chi_E + \frac{1}{G_E}$ est minimal. Ce minimum est atteint pour $G_E \longrightarrow \infty$ et $\chi_E = \frac{1}{\chi}$, *i.e.* $\chi_E + \frac{1}{G_E} = \frac{1}{\chi}$. L'information I_{AE} vaut alors asymptotiquement :

$$I_{AE} = \log_2 \frac{V + \frac{1}{\chi}}{1 + \frac{1}{\chi}}. \tag{3.17}$$

Comme pour le cas homodyne direct, on voit que si le bruit ajouté χ est supérieur à 1, cette borne est supérieure à I_{AB} : le protocole direct ne permet pas de transfert d'information secrète entre Alice et Bob si les pertes en lignes sont supérieures à 1/2. Par la suite, nous nous intéresserons donc uniquement aux faibles bruits ajoutés $0 < \chi < 1$.

Afin de trouver une borne plus générale sur I_{AE}, nous considérons maintenant une attaque individuelle gaussienne arbitraire. Cette attaque est un clonage asymétrique $1 \to 3$ du faisceau envoyé par Alice (figure 3.4). Le clonage quantique est une opération qui consiste à dupliquer M fois un état quantique incident. Par exemple, le modèle du canal entre Alice et Ève que nous avons utilisé en sections 2.3 et 3.3 est un de clonage $1 \to 2$ dans lequel un des clones est envoyé à Bob, alors que l'autre clone est conservé par Ève. Comme nous l'avons vu, le clonage introduit du bruit sur chacun des clones afin de respecter le principe d'incertitude de Heisenberg.

Un cas particulier simple de clonage est le clonage symétrique [29, 30], dans lequel tous les clones sont identiques. Pour modéliser une situation d'espionnage, nous devons considérer un clonage dissymétrique dans lequel le clone envoyé à Bob a un bruit χ, alors que le(s) clone(s) conservés par l'espion ont un bruit χ_E minimal (jusqu'à présent, nous avions borné $\chi_E > 1/\chi$).

La cloneuse $1 \to 3$ que nous envisageons maintenant (figure 3.4) est doublement dissymétrique. D'abord, et comme précédemment, le bruit sur le clone de Bob (noté 1) est différent du bruit sur les clones d'Ève. Ensuite, Ève mesure la quadrature X sur l'un des clones (noté 2) et la quadrature P sur l'autre clone (noté 3). Ce faisant, deux quadratures (X du clone 3 et P du clone 2) ne sont jamais mesurées par Ève. Elle a donc tout intérêt à introduire un bruit minimal sur les quadratures mesurées, aux dépens des quadratures non mesurées. La cloneuse optimale pour Ève est donc probablement dissymétrique en les quadratures X et P des clones 2 et 3. Nous quantifierons cette asymétrie au paragraphe suivant.

Commençons par borner l'information accessible à l'espion à l'aide des techniques que nous avons déjà rencontrées pour les protocoles homodynes. Les relations de Heisenberg qui bornent

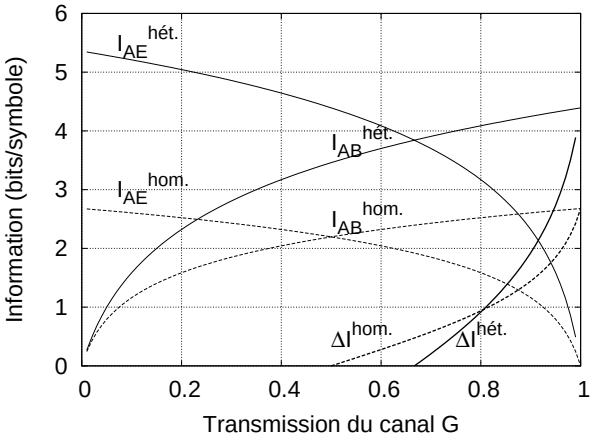


FIG. 3.5: Informations mutuelles pour une mesure hétérodyne en réconciliation directe (courbes pleines), pour $V_A = 40N_0$ et $\xi = 0$. Comme pour la réconciliation inverse, ces informations sont comparées à celles obtenues avec une mesure homodyne (courbes pointillées). Les courbes grasses sont les informations secrètes.

les variances des bruits sur les voies de sortie de la cloneuse s'écrivent (cf. équation 2.23) :

$$\begin{cases} \chi_{EX}\chi > 1 \\ \chi_{EP}\chi > 1 \\ \chi_{EX}\chi_{EP} > 1 \end{cases} \tag{3.18}$$

On remarque que pour notre domaine d'étude ($0 < \chi < 1$), la dernière contrainte est redondante avec les deux premières. Selon ces contraintes, les paramètres de la cloneuse qui minimisent le bruit sur le signal mesuré par Ève sont donc

$$\chi_{EX} = \chi_{EP} = \frac{1}{\chi}. \tag{3.19}$$

On en déduit l'information mutuelle I_{AE} maximale entre Alice et Ève :

$$I_{AE} = \frac{1}{2}\log_2 \frac{V + \chi_{EX}}{1 + \chi_{EX}} + \frac{1}{2}\log_2 \frac{V + \chi_{EP}}{1 + \chi_{EP}} = \log_2 \frac{V + \frac{1}{\chi}}{1 + \frac{1}{\chi}} \tag{3.20}$$

Cette expression est identique à l'information maximale asymptotique que nous avions trouvée en étudiant un type d'attaques restreint. Toutefois, Ève peut l'atteindre avec un gain fini.

Le figure 3.5 compare les informations mutuelles hétérodynes que nous venons d'établir dans le cas direct aux mesures homodynes démontrées au chapitre précédent. Le taux secret hétérodyne n'est supérieur au taux secret homodyne que pour de faibles pertes en ligne.

Les théorèmes qui traitent du clonage symétrique, proposent des bornes sur les bruits ajoutés plus contraignantes que les relations 3.18, en considérant de nouveaux commutateurs sur les quadratures des différents clones [29]. Afin de tenter de limiter davantage l'information accessible à l'espion, nous pouvons essayer d'appliquer ces méthodes à notre configuration de clonage asymétrique. On utilise les notations suivantes pour les bruits sur les trois voies de sortie de la cloneuse représentée figure 3.4, dans lesquelles la lettre C indique le canal reliant Alice à Bob, et les chiffres 2 et 3 indiquent les deux voies d'Ève :

- Les bruits sur les quadratures du faisceau arrivant chez Bob sont notés respectivement X_{CB} et P_{CB}. Ils sont de variance identique χ.
- Les bruits des quadratures mesurées par Ève sur ses deux voies de sortie sont notés X_{E_2} et P_{E_3}. Pour conserver la symétrie du problème en les quadratures X et P mesurées par Bob, nous choisissons ces bruits de variances identiques : $\chi_{EX} = \chi_{EP} = \chi_\alpha$.
- Les bruits des quadratures *non* mesurées par Ève sur ses deux voies de sortie sont notés X_{E_3} et P_{E_2}. Là encore, ces bruits sont choisis de variances identiques χ_β.
- On note les corrélations entre ces bruits : $\mathcal{C}_1 = \langle X_{E_2} X_{E_3} \rangle = \langle P_{E_2} P_{E_3} \rangle$, $\mathcal{C}_\alpha = \langle X_{CB} X_{E_2} \rangle = \langle P_{CB} P_{E_3} \rangle$ et $\mathcal{C}_\beta = \langle X_{CB} X_{E_3} \rangle = \langle P_{CB} P_{E_2} \rangle$

À l'image des techniques employées dans les théorèmes traitant du clonage $1 \longrightarrow M$ [4], nous allons utiliser l'opérateur :

$$\Lambda = X_{CB} + \lambda_2 X_{E_2} + \lambda_3 X_{E_3}. \tag{3.21}$$

où λ_2 et λ_3 sont deux constantes arbitraires. On calcule le commutateur de cet opérateur Λ avec le bruit P_{CB} sur la quadrature P reçue par Bob :

$$[\Lambda, P_{CB}] = -2iN_0(\lambda_2 + \lambda_3) \tag{3.22}$$

d'où la relation de Heisenberg :

$$\langle \Lambda^2 \rangle \chi \geq (\lambda_2 + \lambda_3)^2 \tag{3.23}$$

$$\text{avec} \quad \langle \Lambda^2 \rangle = \chi + \lambda_2^2 \chi_\alpha + \lambda_3^2 \chi_\beta + 2\lambda_2\lambda_3 \mathcal{C}_1 + 2\lambda_2 \mathcal{C}_\alpha + 2\lambda_3 \mathcal{C}_\beta \tag{3.24}$$

Corrélations symétriques. On suppose tout d'abord que toutes les corrélations sont identiques : $\mathcal{C}_1 = \mathcal{C}_\beta = \mathcal{C}_\alpha$. Afin d'obtenir une inégalité ne faisant intervenir que les bruits ajoutés, nous choisissons $\lambda_2 = \lambda_3 = -2$ pour faire disparaître les corrélations des expressions précédentes. On obtient alors l'inégalité :

$$\chi^2 + 4\chi(\chi_\alpha + \chi_\beta) - 16 \geq 0 \tag{3.25}$$

Si la cloneuse est totalement symétrique (*i.e.* si $\chi_\alpha = \chi_\beta = \chi$), on retrouve la condition $\chi \geq \frac{4}{3}$ qui est la borne inférieure sur les bruits imposée par un clonage symétrique $1 \rightarrow 3$ [29].

Si les deux voies d'Ève sont symétriques entre elles, mais différentes de la voie de Bob (*i.e.* si $\chi_\alpha = \chi_\beta = \chi_E \neq \chi$), on trouve

$$\chi_E \geq \frac{16 - \chi^2}{8\chi} > \frac{1}{\chi} \tag{3.26}$$

Cette condition est plus contraignante que les relations 3.18 : avec nos hypothèses restrictives sur l'attaque d'Ève (corrélations et bruits symétriques), Ève n'atteint pas la borne 3.20. Pour atteindre cette borne avec des corrélations symétriques entre les trois voies de sortie, il faut donc que les bruits sur les quadratures mesurées et non mesurées soient différents : $\chi_\alpha \neq \chi_\beta$. Quantitativement, pour une attaque optimale ($\chi_\alpha = \frac{1}{\chi}$), on a une limite inférieure pour le bruit χ_β sur les quadratures non mesurées :

$$\chi_\beta \geq \frac{12 - \chi^2}{4\chi} > \frac{16 - \chi^2}{8\chi} > \frac{1}{\chi} \tag{3.27}$$

Corrélations asymétriques. On considère maintenant des corrélations quelconques. Pour obtenir une borne simple sur les bruits ajoutés, on peut prendre $\lambda_2 = \lambda_3 = -\frac{\mathcal{C}_\alpha + \mathcal{C}_\beta}{\mathcal{C}_1}$.

Un protocole optimal ($\chi_\alpha = \frac{1}{\chi}$) est toujours possible, si la condition suivante est respectée :

$$\chi_\beta \geq \frac{3 - \mathcal{C}\chi^2}{\chi}, \quad \text{avec} \quad C = \left(\frac{\mathcal{C}_1}{\mathcal{C}_\alpha + \mathcal{C}_\beta}\right)^2 \tag{3.28}$$

Cette limite est d'autant moins contraignante que $\mathcal{C}$ est grand, c'est-à-dire que $\mathcal{C}_1$ est grand devant $\mathcal{C}_\alpha + \mathcal{C}_\beta$. On peut atteindre le configuration dans laquelle les quadratures mesurées et non mesurées par Ève sont interchangeables ($\chi_\alpha = \chi_\beta = \frac{1}{\chi}$) si

$$\mathcal{C} \geq \frac{2}{\chi^2} \tag{3.29}$$

Concluons notre recherche de bornes plus contraignantes que les équations 3.18 sur le bruit entachant la mesure d'Ève. Nous pouvons dire que :

- Nous n'avons pas de contrainte supplémentaire sur l'information acquise par Ève. Ceci est permis car il n'y a pas de contrainte sur les bruits des quadratures non mesurées par Ève.
- Mais nous avons obtenu des contraintes sur les symétries entre les quadratures mesurées et les quadratures non mesurées par Ève. Pour que notre attaque atteigne la borne 3.20, il faut ou bien que les bruits sur les quadratures non mesurées par Ève soient supérieurs aux bruits sur les quadratures mesurées (équation 3.27), ou bien que les bruits sur les quadratures mesurées et non mesurées soient assez corrélés (équation 3.29).

3.4 Recherche d'attaques optimales

Dans le cas inverse comme dans le cas direct, nous ne connaissons pas d'attaque atteignant les bornes 3.20 et 3.14. Cette observation peut avoir deux origines. D'abord, les attaques que nous allons étudier dans cette section peuvent ne pas être optimales. Dans ce cas, d'autres attaques plus puissantes doivent être envisagées. Ensuite, les bornes que nous avons établies peuvent ne pas être les plus contraignantes. Dans ce cas, d'autres contraintes sur les bruits ajoutés doivent être trouvées.

Pour ces deux cas, direct et inverse, nous connaissons trois attaques qui fournissent un taux secret identique. Nous ne connaissons par à l'heure actuelle de meilleure attaque. Ces trois attaques sont :

- La téléportation quantique dans laquelle Ève fait interférer le signal envoyé par Alice avec l'un des modes d'une paire EPR de variance $\mathcal{V}$ sur une lame séparatrice de transmission 1/2. Ensuite, elle mesure respectivement les quadratures X et P sur les voies de sortie de la lame. Enfin, elle translate les quadratures de l'autre mode de la paire EPR en fonction des résultats de ses mesures modifiés d'un facteur G en intensité. Ce mode translaté est envoyé à Bob. Les paramètres suivants maximisent l'information acquise par l'espion :

$$G = 2T \quad \text{et} \quad \mathcal{V} = \frac{s^2+1}{2s} \quad \text{avec} \quad s = \frac{T\left(\sqrt{T}+1\right)^2\left(\chi - \sqrt{\chi^2 - \left(\frac{1}{T}-1\right)^2}\right)}{(1-T)^2} \tag{3.30}$$

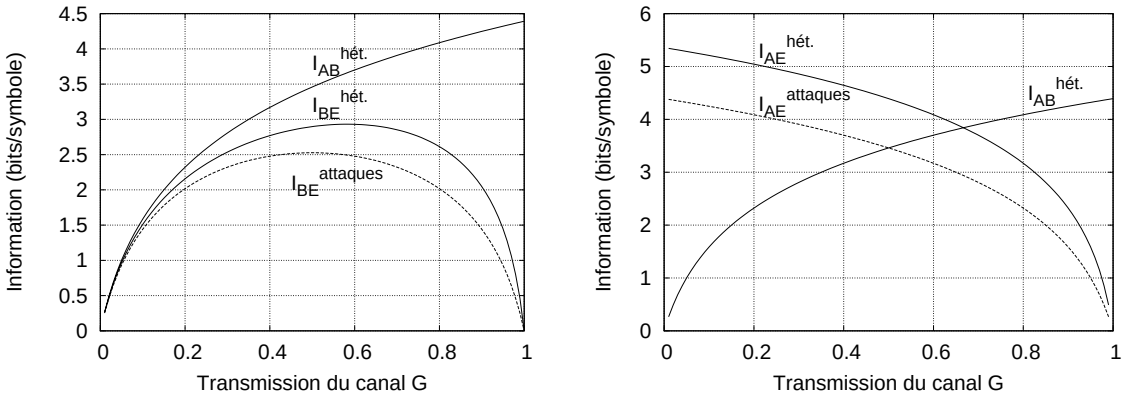


FIG. 3.6: Information accessible à l'espion pour les meilleures attaques connues dans les cas hétérodynes inverse (gauche) et direct (droite), pour $V_A = 40N_0$ et $\xi = 0$ (courbe pointillée). Cette information est comparée à la borne maximale sur l'information I_{BE} (gauche) ou I_{AE} (droite) que nous avons établie.

- La cloneuse optimale [21], dans laquelle Ève amplifie d'un facteur G le signal envoyé par Alice avec un amplificateur optique, puis atténue d'un facteur T_2 ce signal avant de l'envoyer à Bob. Ensuite, Ève fait interférer les deux autres modes issus respectivement de l'amplificateur et de la lame séparatrice, avant de mesurer respectivement les quadratures X et P dans chaque voie de sortie. Les paramètres suivants maximisent l'information acquise par l'espion :
$$T_2 = T(1-\xi/2) \quad \text{et} \quad G = \frac{1}{1-\xi/2}. \tag{3.31}$$
- L'attaque de type "Feed-Forward" [31, 32] qui utilise un schéma de cloneuse $1 \to 2$ sans amplification proposé dans [33]. Dans cette attaque, Ève prélève une fraction $1 - T_2$ du signal envoyé d'Alice à Bob, fait une mesure hétérodyne sur le signal prélevé puis translate les quadratures du faisceau envoyé à Bob en fonction des résultats de ses mesures modifiés d'un facteur G en intensité. Les paramètres suivants maximisent l'information acquise par l'espion :
$$T_2 = \frac{G\xi(2-\xi) - 2(\xi+2) + 4\sqrt{\xi(2-G(2-\xi))}}{4/G + 4\xi + G\xi^2} \quad \text{et} \quad G = T\xi, \tag{3.32}$$

Dans le cas inverse, l'information accessible à l'espion par ces attaques est donnée par :
$$V^a_{B|E} = \frac{T(2-\xi)^2(V-1)}{2(V+1)\sqrt{2-2T+T\xi}\sqrt{\xi} + \xi(T\xi - 3T + V(1+T)+1) + 2(V(1-T)+1+T)} + 1 \tag{3.33}$$
Cette variance conditionnelle ne coïncide pas en général avec les bornes que nous avions établies. Toutefois, il y a coïncidence pour des valeurs bien particulières du bruit ajouté :
$$\left\{ \begin{array}{rcl} \chi & = & \frac{1}{T}\sqrt{1-T+\frac{T}{V^2}} - \frac{1}{V} \\ \text{ou} \quad \xi & = & 2 \end{array} \right. \tag{3.34}$$

Dans le cas direct, on évalue l'information accessible à l'espion par :
$$\chi^a_E = \frac{T\xi^2 + \xi - 3\xi T + 2(T+1) + 2\sqrt{2-2T+T\xi}\sqrt{\xi}}{(\sqrt{2-2T+T\xi}+\sqrt{\xi})^2} \tag{3.35}$$

Là encore, on n'atteint pas en général les bornes que nous avions établies. Toutefois, il y a coïncidence pour :

$$\begin{cases} \chi = \sqrt{\frac{T-1}{T}} \\ \text{ou} \quad \xi = 2 \end{cases} \tag{3.36}$$

La figure 3.6 compare les taux d'information accessibles à l'espion pour ces attaques aux bornes que nous avons établies.

Concernant l'écart entre les meilleures attaques connues et les meilleures bornes connues, deux hypothèses sont à envisager : soit il existe de meilleures attaques, soit il existe des bornes plus contraingnantes. De récentes études nous laissent penser que la deuxième hypothèse semble être vérfiée.

3.5 Quel avantage pratique ?

Le protocole hétérodyne inverse semble bien adapté pour remplacer le protocole homodyne présenté au chapitre précédent : même avec la borne supérieure actuelle sur I_{BE}, il surpasse l'information secrète fournie par ce dernier. Pourtant, deux effets réduisent son intérêt pratique.

La mise en œuvre expérimentale d'un protocole hétérodyne est plus délicate. Pour réaliser un protocole homodyne, nous devons piloter un modulateur de phase capable de choisir la quadrature de mesure choisie par Bob. Cette modulation n'est que la duplication du dispositif de modulation de phase déjà réalisé chez Alice, et ne demande que l'installation d'un nouveau modulateur et la duplication de la gestion logicielle de la modulation. Le protocole hétérodyne nous affranchit de cette modulation de phase. En revanche, il nécessite la duplication de la détection homodyne capable de mesurer une quadrature du signal. Comme nous le verrons dans la partie suivante, l'effort de duplication d'une détection homodyne est de toute autre nature que l'effort de duplication d'un modulateur de phase. De plus, un contrôle actif de la phase relative entre les deux mesures est nécessaire pour garantir la mesure de deux quadratures orthogonales. Enfin, une détection homodyne nécessite une référence de phase intense. L'utilisation de deux détections homodynes divise de moitié la puissance de la référence pour chaque détection, et augmente ainsi l'effet du bruit électronique de la détection (voir chapitre 7).

Le protocole hétérodyne est moins robuste face aux imperfections des algorithmes de réconciliation. Pour obtenir une chaîne de bits secrets à partir de nos variables continues, nous devons utiliser des algorithmes classiques d'extraction de l'information mutuelle I_{AB}. L'efficacité β de ces algorithmes est limitée : nous verrons en troisième partie que $\beta = 0,87$ pour $G = 0,25$. Le nombre de bits secrets effectivement extraits par symbole transmis dans le canal quantique s'exprime alors :

$$\Delta I = \beta I_{AB} - I_{BE}. \tag{3.37}$$

Pour $\beta = 1$, ΔI est positif pour toute transmission du canal quantique. Cependant, si $\beta < 1$, il existe une transmission en dessous de laquelle plus aucun bit secret ne peut être transmis. La grandeur $\beta_{\min} = I_{BE}/I_{AB}$ est l'efficacité de réconciliation minimale pour que la transmission produise une clé secrète. On trace cette efficacité sur la figure 3.7, en utilisant les expressions de I_{AB} et I_{BE} que nous avons déterminées aux sections 2.4 pour le cas homodyne et 3.2 pour le cas hétérodyne. On remarque que le protocole hétérodyne est plus sensible à l'efficacité de

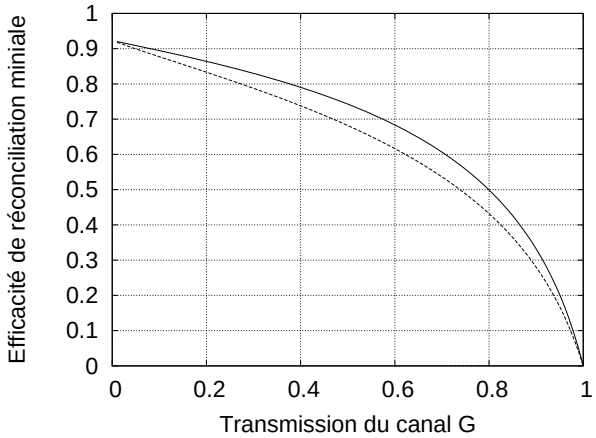


FIG. 3.7: Efficacité de réconciliation minimale pour extraire une clé secrète, en fonction de la distance, pour $V_A = 12N_0$ et $\xi = 0$. Le protocole homodyne inverse (courbe pointillée) supporte davantage les faibles efficacités que le protocole hétérodyne inverse (courbe pleine).

réconciliation que le protocole homodyne. Notamment, pour nos valeurs de l'efficacité, le protocole hétérodyne ne transmet plus de bits secrets pour $G = 0,25$. Cet effet a une interprétation physique : la lame séparatrice introduite chez Bob pour séparer le faisceau reçu introduit une unité de bruit de photon sur le signal. Or les performances des algorithmes de réconciliation sont moins bonnes pour les lignes plus bruitées. De plus, Bob doit traiter deux fois plus de données, puisqu'il dispose maintenant de deux mesures de quadrature : ce surcroît de données vient encore dégrader les performances, puisque la vitesse de traitement des données est à l'heure actuelle la limitation de notre dispositif.

Toutefois, le protocole hétérodyne reste intéressant du point de vue théorique. Il présente également des avantages pratiques pour d'autres réalisations expérimentales des protocoles de distribution quantique de clé utilisant des variables continues dans le domaine fréquentiel [34].

Chapitre 4

Un aperçu de la sécurité des protocoles à variables continues

Ce chapitre propose un tour d'horizon des preuves de sécurité des protocoles à variables continues qui ont été publiées au cours des dernières années. On distingue plusieurs modèles de sécurité, associés à différents types d'attaques quantiques schématisées figure 4.1 :

- Les attaques individuelles autorisent Ève à manipuler individuellement les états cohérents envoyés par Alice. Elle n'a pas le droit de faire des mesures quantiques faisant intervenir plusieurs états. On modélise ce type d'attaque de la façon suivante : Ève fait interagir chaque état cohérent transitant dans le canal quantique avec un signal sonde, qu'elle peut garder dans une mémoire quantique jusqu'à la révélation des quadratures que Bob a choisi de mesurer. Enfin, Ève mesure individuellement chacun de ces signaux sondes.
- Les attaques collectives autorisent Ève à faire une mesure quantique globale sur toute la séquence d'états envoyés par Alice. On modélise cette attaque de façon analogue aux attaques individuelles. La différence réside dans le fait qu'Ève a le droit de faire une mesure impliquant l'ensemble de ses signaux sondes, après le processus de réconciliation (c'est-à-dire l'établissement d'une clé secrète entre Alice et Bob), afin d'optimiser la stratégie d'espionnage sur toute la séquence transmise. Cette mesure collective est réalisée en appliquant une transformation unitaire qui a pour entrées l'ensemble des sondes, suivie d'une mesure individuelle des sorties de la transformation.
- Les attaques collectives de taille finie sont un sous-ensemble des attaques collectives, dans lesquelles Ève peut faire une suite d'attaques collectives sur des sous-ensembles d'états de petite taille devant le nombre total d'états échangés.
- Les attaques cohérentes sont les attaques les plus générales. On les modélise de façon similaire aux attaques collectives. En revanche, Ève est autorisée à intriquer toutes ses sondes avant de les faire interagir avec les états cohérents qui transitent dans le canal quantique.

Toutes les catégories énoncées ci-dessus se déclinent en deux familles : attaques gaussiennes et non-gaussiennes. Les attaques gaussiennes supposent que les opérations effectuées par Ève entrent dans le cadre du modèle du canal gaussien que nous avons développé au chapitre 2. Notamment Ève n'est autorisée à utiliser que des opérations ajoutant un bruit gaussien (séparation, amplification dépendante ou non de la phase, utilisation d'états EPR, mesures de quadrature...). Pour les attaques gaussiennes, la distribution des données mesurées par Bob est donc une gaussienne. Les attaques non gaussiennes autorisent Ève à effectuer tout

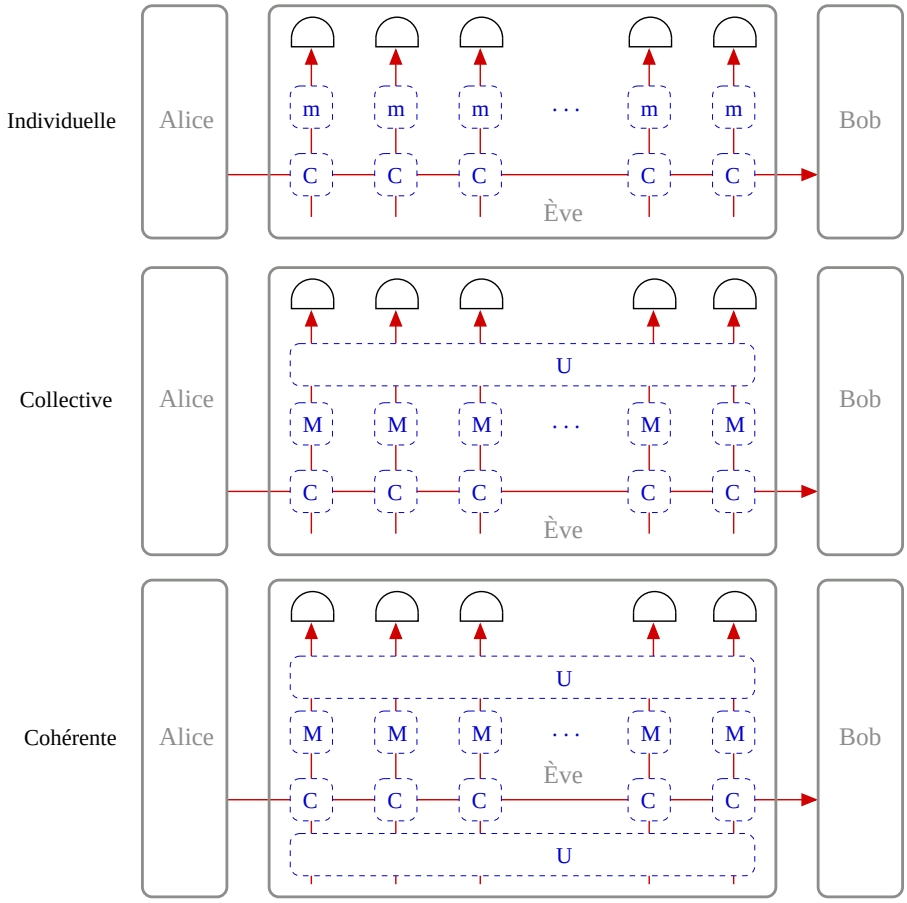


FIG. 4.1: Modélisation des attaques sur un protocole de distribution quantique de clé. Ève couple (C) chaque état traversant le canal quantique avec un faisceau sonde. Elle dispose de mémoires quantiques lui permettant de garder ses états quantiques en mémoire jusqu'à révélation des bases choisies par Bob (m), ou jusqu'à la fin du processus classique de distillation de la clé secrète (M). Les attaques collectives et cohérentes autorisent l'emploi de transformations unitaires faisant intervenir tous les symboles transmis sur le canal.

type d'attaque permis par la mécanique quantique et le modèle de sécurité considéré.

Enfin, toutes les analyses de sécurité sont asymptotiques : elles ne sont valables que pour des échanges quantiques de taille infinie. Les effets dus à la taille finie des séquences utilisées en pratique font l'objet d'études récentes [35].

4.1 Intrication virtuelle : schéma équivalent à la modulation d'Alice

Beaucoup de preuves de sécurité font appel à un schéma équivalent pour la production d'états cohérents modulés par Alice, appelé schéma à «intrication virtuelle» [18]. Ce schéma est plus facile à manipuler théoriquement que le schéma de type «préparation-émission» que nous avons considéré jusqu'alors. Alice produit sa modulation en

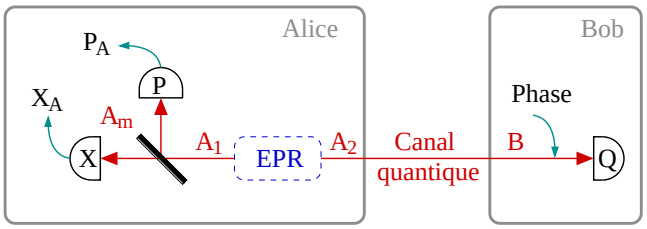


FIG. 4.2: Schéma de modulation à intrication virtuelle. Dans ce schéma, Alice fait une mesure hétérodyne sur un des deux faisceaux (mode A_1) d'une paire EPR de variance $V_A + 1$, alors que l'autre mode (A_2) est envoyé à Bob par le canal quantique. On note les résultats de sa mesure (X_{A_m}, P_{A_m}). Ce schéma est équivalent au schéma de modulation dans lequel Alice déplace un état cohérent dans le plan complexe vers la position (X_A, P_A) fonction de (X_{A_m}, P_{A_m}), avec une modulation gaussienne de variance $V_A N_0$

faisant une mesure hétérodyne sur l'un des modes (noté A_1) d'une paire EPR de variance $VN_0 = (V_A + 1)N_0$. L'autre mode de la paire A_2 est envoyé à Bob par le canal quantique (figure 4.2). Le centre de l'état cohérent (X_A, P_A) envoyé dans le canal quantique est fonction des résultats de la mesure hétérodyne (X_{A_m}, P_{A_m}) d'Alice.

Pour montrer l'équivalence entre les protocoles «intrication virtuelle» et «préparation-émission», calculons la matrice de covariance K_{A_m,A_2}. Nous avons vu (équation 2.36) que la matrice de covariance de la paire EPR s'exprimait :

$$K_{A_1,A_2} = \begin{bmatrix} VN_0 & \pm\sqrt{V^2-1}N_0 \\ \pm\sqrt{V^2-1}N_0 & VN_0 \end{bmatrix}, \tag{4.1}$$

où $\pm$ dépend de la quadrature X ou P choisie par Alice et Bob. On altère le mode A_1 par une lame séparatrice de transmission 1/2 pour obtenir la matrice de covariance K_{A_m,A_2} :

$$K_{A_m,A_2} = \begin{bmatrix} \frac{1}{2}(V+1)N_0 & \pm\frac{1}{\sqrt{2}}\sqrt{V^2-1}N_0 \\ \pm\frac{1}{\sqrt{2}}\sqrt{V^2-1}N_0 & VN_0 \end{bmatrix}. \tag{4.2}$$

Comme à la section 3.2, on reconnaît l'unité de bruit de photon N_0 ajoutée sur le mode A_m par la lame sépratrice, et l'atténuation d'un facteur $\frac{1}{\sqrt{2}}$ de l'amplitude ce mode. À partir de la matrice de covariance K_{A_m,A_2}, on déduit la variance conditionnelle $V_{A_2|A_m}$ qui traduit l'incertitude d'Alice sur le mode A_2 envoyé dans le canal quantique :

$$V_{A_2|A_m} = \frac{\det(K_{A_m,A_2})}{V_{A_m}} \tag{4.3}$$

$$= N_0. \tag{4.4}$$

Finalement, le mode envoyé par Alice a une variance $VN_0 = (V_A+1)N_0$ avec une incertitude N_0 : le protocole à «intrication virtuelle» est bien équivalent au protocole «préparation-émission» dans lequel Alice envoie un signal modulé avec une variance $V_A N_0$ auquel se rajoute le bruit de photon de variance N_0 inconnu d'Alice. À l'aide de l'estimateur défini par l'équation 1.23, on exprime les coordonnées (X_A, P_A) du centre de l'état cohérent envoyé dans le canal quantique

en fonction des résultats (X_{A_m}, P_{A_m}) de la mesure hétérodyne d'Alice :

$$(X_A, P_A) = \frac{\langle A_m A_2 \rangle}{V_{A_m}} (X_{A_m}, P_{A_m}) \tag{4.5}$$

$$= \frac{\sqrt{2(V^2-1)}}{V+1} (X_{A_m}, -P_{A_m}). \tag{4.6}$$

Dans ce schéma à «intrication virtuelle» équivalent, on peut simplement établir [4, 36] la borne sur la variance conditionnelle $V_{B|E}$ que nous avions annoncée en section 2.4 pour le protocole inverse homodyne (équation 2.27). Nous avions utilisé cette variance conditionnelle pour borner supérieurement l'information I_{BE} acquise par l'espion. Pour cela, on écrit les estimateurs des quadratures X_B et P_B connaissant respectivement X_{A_1} et la quadrature P_E mesurée par Ève[1] :

$$X_B = \alpha_{A_1} X_{A_1} + \delta X_{A_1} \tag{4.7}$$

$$P_B = \alpha_E P_E + \delta P_E. \tag{4.8}$$

où $\alpha_{A_1} = \frac{\langle A_1 B \rangle}{V_{A_1}}$, $\alpha_E = \frac{\langle EB \rangle}{V_E}$, $\langle \delta X_{A_1}^2 \rangle = V_{B|A_1}$ et $\langle \delta P_E^2 \rangle = V_{B|E}$ (voir équation 1.24). Ces relations d'estimation nous permettent de calculer le commutateur :

$$[\delta X_{A_1}, \delta P_E] = [X_B - \alpha_{A_1} X_{A_1}, P_B - \alpha_E P_E] \tag{4.9}$$

$$= [X_B, P_B] \quad \text{car les quadratures de deux modes distincts commutent} \tag{4.10}$$

$$= 2iN_0. \tag{4.11}$$

On en déduit la relation de Heisenberg reliant les variances de δX_{A_1} et δP_E :

$$V_{B|E} V_{B|A_1} \geq N_0^2, \tag{4.12}$$

qui borne la variance conditionnelle $V_{B|E}$ et donc l'information accessible à l'espion à propos des mesures de Bob. Pour obtenir l'expression de cette borne, on calcule la variance conditionnelle $V_{B|A_1}$ à l'aide de la matrice de covariance $K_{A_1,B}$ qui est obtenue après altération par le canal quantique de la matrice de covariance K_{A_1,A_2} :

$$K_{A_1,B} = \begin{bmatrix} VN_0 & \pm\sqrt{G}\sqrt{V^2-1}N_0 \\ \pm\sqrt{G}\sqrt{V^2-1}N_0 & G(V+\chi)N_0 \end{bmatrix}. \tag{4.13}$$

d'où

$$V_{B|A_1} = \frac{\det(K_{A_1,B})}{V_{A_1}} = G\left(\chi + \frac{1}{V}\right) N_0. \tag{4.14}$$

On en déduit une borne inférieure sur la variance conditionnelle limitant l'information accessible à l'espion :

$$V_{B|E} \geq \frac{1}{G\left(\chi + \frac{1}{V}\right)} N_0. \tag{4.15}$$

C'est la borne que nous avions annoncée à la section 2.4 et qui nous a permis de borner inférieurement l'information secrète ΔI dans le cas inverse.

[1] Là encore, nous considérons un protocole symétrique en les quadratures X et P. Pour cela, Bob mesure une quadrature aléatoire pour chaque état cohérent reçu.

4.2 Attaques individuelles gaussiennes

Les taux d'information que nous avons calculés au chapitre 2 sont sûrs face aux attaques individuelles gaussiennes. Dans cette analyse, nous avons en effet traité chaque état cohérent de façon indépendante, et nous avons supposé qu'Ève faisait une mesure de quadrature sur l'état envoyé par Alice auquel est adjoint un bruit gaussien. Cette preuve est une première étape dans notre analyse de sécurité. Notons que cette preuve est valable pour toutes valeurs de la transmission du canal G et de l'excès de bruit ξ : les taux que nous calculons prennent en compte tous les paramètres du canal gaussien, dont la présence d'un éventuel excès de bruit.

4.3 Attaques non gaussiennes et attaques collectives de taille finie

Frédéric Grosshans et Nicolas Cerf ont montré [36] que les attaques individuelles gaussiennes sont optimales parmi les attaques individuelles, gaussiennes ou non gaussiennes. Ce résultat est obtenu en remplaçant l'inégalité de Heisenberg sur les variances conditionnelles par une inégalité de Heisenberg informationnelle [37] :

$$S(X_B|E) + S(P_B|P_{A_1}) \geq 2S_0 \tag{4.16}$$

où S est l'entropie différentielle définie à la section 1.4, exprimée en bits par symbole, E est le résultat de la mesure (quelconque) d'Ève, et $S_0 = \frac{1}{2}\log_2(2\pi e N_0)$ est l'entropie associée à la distribution gaussienne du bruit de photon de variance N_0. Cette équation n'est autre que l'inégalité 4.12 généralisée aux distributions non gaussiennes. Elle permet de borner I_{BE} puis ΔI par des quantités connues d'Alice et de Bob :

$$\begin{aligned} I_{BE} &= S(X_B) - S(X_B|E) \leq S(X_B) + S(P_B|P_{A_1}) - 2S_0 && (4.17)\\ \text{puis}\quad \Delta I &= I_{AB} - I_{BE} \geq 2S_0 - S(X_B|X_A) - S(P_B|P_{A_1}) && (4.18)\\ \text{avec}\quad I_{AB} &= S(X_B) - S(X_B|X_A). && (4.19) \end{aligned}$$

La dernière étape consiste à utiliser l'extrémalité des distribution gaussiennes : $S(X_B|X_A) \leq S_{\mathcal{G}}(X_B|X_A)$ où $S_{\mathcal{G}}(X_B|X_A)$ est l'entropie de la distribution gaussienne ayant même matrice de covariance, c'est-à-dire les mêmes moments d'ordre deux (variances $V_A = \langle X_A^2 \rangle$, $V_B = \langle X_B^2 \rangle$, et corrélations $\langle X_A X_B \rangle$) que la distribution conjointe réelle des données entre Alice et Bob. On se référera à l'article [36] pour plus de détails.

On énonce finalement la sécurité face aux attaques individuelles non gaussiennes : si Alice et Bob mesurent les moments d'ordre deux (variances et corrélation) de leur distribution de données conjointe, alors ils disposent du taux secret $\Delta I = 2S_0 - S_{\mathcal{G}}(X_B|X_A) - S_{\mathcal{G}}(P_B|P_{A_1})$ sûr face aux attaques non gaussiennes, où les entropies $S_{\mathcal{G}}$ sont fonction des moments d'ordre deux mesurés. En calculant les paramètres G et χ du canal gaussien qui auraient produit ces mêmes moments d'ordre deux, on montre que ce dernier taux secret est identique à celui que nous avions exprimé au chapitre 2. Ce résultat est sans surprise, car si nous appliquons la relation de Heisenberg entropique 4.16 à des distributions gaussiennes à l'aide de l'expression 1.19, nous obtenons les relations de Heisenberg sur les variances conditionnelles 2.27 à partir desquelles nous avions bâti notre modèle de sécurité.

De plus, on peut remarquer que les informations mutuelles 4.18 sont des expressions de I_{AB} et I_{BE} prenant en compte les distributions de données effectives, c'est-à-dire non gaussiennes. Dans le cas général, ces expressions ne sont pas analytiques, mais s'écrivent sous forme d'une intégrale qu'on peut calculer numériquement. Ainsi, on peut envisager la recherche de modulations non gaussiennes particulières qui faciliteraient la procédure d'extraction de bits sans erreur à partir des variables continues, que nous aborderons en partie III. Toutefois, l'évaluation d'une information faisant intervenir des variables continues nécessite en théorie la connaissance d'une infinité de paramètres (par exemple l'ensemble des moments d'ordre $n \in [1, \infty[$ de la distribution des données). En pratique, il est impossible de mesurer tous ces moments, et nous calculons les taux ΔI à partir d'un ensemble fini de mesures. Il faut donc s'assurer que les preuves s'appliquent à ce cas, comme elles s'appliquent au cas où Alice et Bob ne mesurent que les moments d'ordre deux de leur distribution conjointe.

Enfin, nous avons réalisé expérimentalement (chapitre 8) une attaque individuelle non gaussienne qui consiste à entrelacer aléatoirement des attaques de type «lame séparatrice» et de type «interception-réémission». Le fait qu'une telle attaque soit possible expérimentalement justifie la nécessité que les taux secret avancés soient sûrs face aux attaques non gaussiennes.

L'optimalité des attaques individuelles gaussiennes s'étend aux attaques collectives de taille finie : les attaques individuelles gaussiennes sont meilleures que les attaques collectives de taille finie gaussiennes ou non gaussiennes [36]. Finalement, le taux secret ΔI que nous avons calculé au chapitre 2 est sûr contre les attaques collectives de taille finie gaussiennes ou non gaussiennes.

4.4 Attaques collectives gaussiennes

Deux articles [38, 39] ont simultanément fourni un moyen de calculer le taux secret face aux attaques collectives gaussiennes. Ces démonstrations fournissent de nouvelles expressions pour le taux secret qui utilisent l'entropie de Holevo χ. Pour une variable aléatoire discrète X, elle se définit par :

$$\chi(X) = S(\rho) - \sum_{x \in X} p(X = x) S(\rho_x), \tag{4.20}$$

où ρ_x est la matrice densité du symbole x, $\rho = \sum_{x \in X} p(X = x)\rho_x$ est la matrice densité du mélange statistique associé à la variable aléatoire X, et $S(\rho) = -\mathrm{Tr}(\rho \log_2 \rho)$ est l'entropie de von Neumann de la variable aléatoire X. L'entropie de Holevo est une adaptation de l'information de Shannon aux systèmes quantiques. Notamment, elle indique le nombre de bits classiques contenus dans un état quantique, et la quantité d'information classique qu'on peut transférer à travers un canal quantique [40]. Son utilisation garantit donc les taux d'information face à des opérations quantiques, notamment face aux attaques collectives.

Raúl García-Patrón a calculé l'information de Holevo dans le cadre du modèle du canal gaussien :

$$\chi_{BE} = G\left(\frac{\lambda_1 - 1}{2}\right) + G\left(\frac{\lambda_2 - 1}{2}\right) - G\left(\frac{\lambda_3 - 1}{2}\right), \tag{4.21}$$

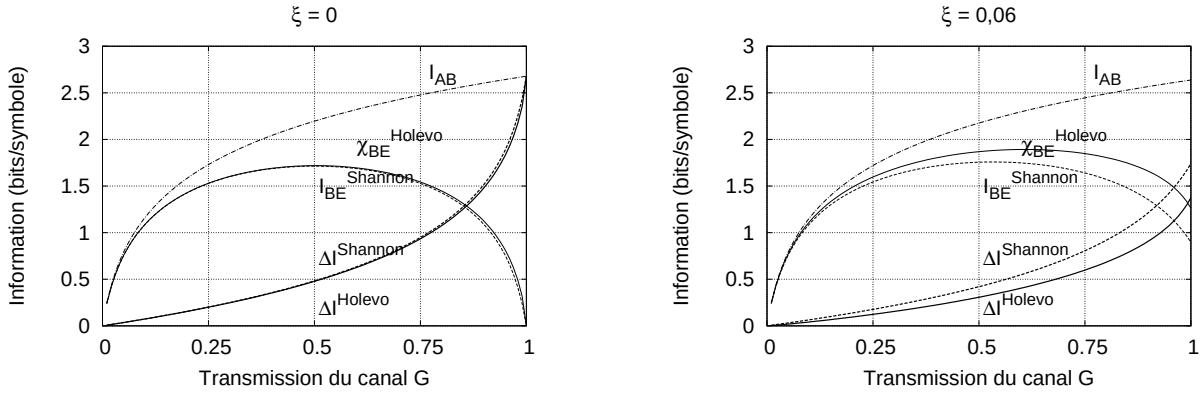


FIG. 4.3: Comparaison des informations de Shannon (courbes pointillées) applicables aux attaques individuelles et des informations de Holevo (courbes pleines) applicables aux attaques collectives, en fonction du gain du canal quantique pour une variance de modulation $V_A = 40N_0$ et un excès de bruit $\xi = 0$ (figure de gauche) et $\xi = 0,06$ (figure de droite). Comme Bob fait des mesures individuelles sur les symboles issus du canal quantique, l'information mutuelle I_{AB} est l'information de Shannon pour les deux types d'attaques envisagées. Comme les attaques collectives offrent davantage de pouvoir à Ève, l'information de Holevo χ_{BE} est supérieure à l'information de Shannon I_{BE}. Notons que l'information de Holevo est plus sensible à l'excès de bruit que l'information de Shannon : les deux informations sont presque confondues pour $\xi = 0$ et se détachent d'autant plus que l'excès de bruit est grand.

avec

$$
\begin{aligned}
G(x) &= (x+1)\log_2(x+1) - x\log_2(x) && (4.22)\\
\lambda_{1,2}^2 &= \frac{C \pm \sqrt{C^2 - 4D}}{2} && (4.23)\\
\lambda_3^2 &= B && (4.24)\\
V &= V_A + 1 && (4.25)\\
B &= V\frac{1 + V\chi}{V + \chi} && (4.26)\\
C &= V^2(1-2T) + 2T + T^2(V+\chi)^2 && (4.27)\\
D &= (T(V\chi + 1))^2 && (4.28)\\
& && (4.29)
\end{aligned}
$$

Avec cette expression, on peut calculer le taux secret $\Delta I = I_{AB} - \chi_{BE}$ sûr face aux attaques collectives. On remarque que l'expression de l'information mutuelle entre Alice et Bob est inchangée, car ils réalisent toujours des mesures individuelles. Comme l'entropie de Holevo est toujours supérieure à l'entropie de Shannon, le taux secret sûr face aux attaques collectives est inférieur au taux secret de Shannon dérivé au chapitre 2, sûr face aux attaques individuelles. La figure 4.3 compare les taux de Shannon et de Holevo.

4.5 Attaques collectives générales

La dernière brique de l'édifice des preuves de sécurité a été récemment apportée par deux articles [41, 42], qui utilisent des techniques différentes. De même que pour les

attaques individuelles, les attaques collectives gaussiennes sont optimales, et le taux secret se calcule à partir des paramètres du canal gaussien équivalent. Pour montrer ce résultat, le deuxième article se fonde sur un théorème d'«extrémalité de la gaussienne» [43]. Ce théorème montre que si une fonction f satisfait certaines propriétés, alors $f(\rho_G) \geq f(\rho)$ où ρ est la matrice densité regroupant les états d'Alice A_m et de Bob B, et ρ_G est la matrice densité de l'état gaussien qui a les mêmes moments d'ordre deux que l'état représenté par ρ. Notamment, l'information de Holevo qu'Alice et Bob calculent à partir de leur matrice densité satisfait ces propriétés. On en déduit donc que l'information de Holevo accessible à l'espion calculée à partir du modèle du canal gaussien est une borne supérieure à l'information de Holevo accessible à l'espion pour une distribution non gaussienne de même matrice de covariance. Cela prouve donc que le taux secret de Holevo est sûr face aux attaques collectives gaussiennes ou non gaussiennes.

4.6 Attaques cohérentes

La sécurité des protocoles à variables continues face aux attaques cohérentes reste à démontrer. Cependant, de récents résultats [44, 45] applicables aux variables discrètes prouvent que les attaques cohérentes n'apportent pas plus d'information à l'espion que les attaques collectives. Si ce résultat s'étend aux protocoles utilisant des variables continues, cela prouve la sécurité *inconditionnelle* du protocole à états cohérents.

4.7 Attaques non quantiques

Toutes les preuves de sécurité que nous avons énoncées jusqu'à présent reposent sur l'hypothèse selon laquelle la seule source d'information que puisse obtenir Ève sur l'échange de clé entre Alice et Bob est l'ensemble des symboles envoyés par Alice traversant le canal quantique. Si l'espion a accès à de l'information qui n'est pas encodée dans ces variables quantiques, nous avons une fuite d'information qui n'est pas couverte par les théorèmes de sécurité inconditionnelle. Quand nous réalisons expérimentalement un échange quantique, nous devons donc nous assurer que cette hypothèse est vérifiée. Nous identifions plusieurs fuites possibles.

Les canaux cachés ou "side channels" portent de l'information non mesurée par Bob à travers le canal quantique. Cette information est portée par un mode ou une variable quantique non mesurés par Bob. Prenons l'exemple de la polarisation : une imperfection expérimentale peut faire que Bob ne mesure pas les quadratures X ou P dans le même mode de polarisation que celui envoyé par Alice. Pour parer à cette éventualité, nous pouvons calibrer la variance du signal envoyé dans le canal quantique en mesurant l'intensité lumineuse indépendante de la polarisation en sortie d'Alice. De cette façon, une mauvaise mesure de Bob sera considérée comme des pertes supplémentaires sur le canal entre Alice et Bob. La polarisation peut être la cause d'une effet plus grave. Nous verrons que les modulateurs utilisés pour générer la modulation gaussienne d'Alice sont sensibles à la polarisation : un mauvais alignement de la polarisation en entrée de ces modulateurs produit une modulation de polarisation corrélée à notre modulation d'amplitude et de phase. Maintenant, l'information n'est plus seulement portée par un autre mode, mais par une autre variable quantique – la polarisation –, qui n'est pas soumise aux théorèmes de sécurité que nous avons énoncés. Dans ce cas, nous ne pouvons

plus considérer cette fuite d'information comme de simples pertes et nous devons prendre des mesures pour éliminer cette modulation, notamment en polarisant la lumière avant et après passage dans les modulateurs.

De façon générale, pour éviter les fuites d'information par des canaux cachés, nous devons donc identifier ces fuites, puis les considérer comme une inefficacité de détection dans le cas d'une mauvaise adaptation entre les modes envoyés par Alice et ceux reçus par Bob, ou les éliminer s'il s'agit de variables quantiques non mesurées.

Les attaques de type «cheval de Troie» autorisent Ève à sonder les dispositifs d'Alice et de Bob via le canal quantique. Par exemple Ève peut envoyer un faisceau sonde à une longueur d'onde différente de celle utilisée par Alice et Bob, puis examiner les signaux réfléchis par les interfaces des composants qui constituent les dispositifs d'Alice et de Bob. Ainsi, Ève peut déterminer les paramètres des modulateurs d'Alice et de Bob sans interférer avec le signal envoyé par Alice. Ces attaques ont été étudiées dans le cadre des protocoles à photons uniques [46]. On peut s'en prémunir en plaçant des filtres fréquentiels couvrant la gamme de transparence des fibres optiques entre les dispositifs d'Alice et de Bob et le canal quantique. Pour empêcher l'intrusion d'un signal chez Alice, ou le retour d'un signal chez Bob, on peut placer des isolateurs optiques de part et d'autre du canal quantique. Ces isolateurs peuvent supprimer tout retour de la lumière jusqu'à 60 dB. Pour sonder les dispositifs d'Alice et de Bob, Ève devrait alors utiliser une telle puissance optique qu'elle endommagerait notre matériel !

De façon analogue, Ève peut tenter de s'introduire chez Alice et Bob par le canal classique. Nous sortons ici du domaine de la physique, et des systèmes informatiques sécurisés doivent être conçus.

Un attaque non quantique plus spécifique aux protocoles utilisant des variables continues est la manipulation de la référence de phase transmise par le canal quantique. Nous verrons que la mesure d'une quadrature par Bob nécessite la transmission d'une référence de phase entre Alice et Bob cohérente avec le signal, utilisée pour amplifier le signal reçu par Bob (voir section 6.3). Ève peut alors tenter de moduler l'intensité de cette référence de phase pour dissimuler une attaque sur le signal. Pour contrer cette attaque potentielle, nous observons l'intensité de la référence de phase pour chaque état traversant le canal quantique.

4.8 Autres protocoles de distribution quantique de clé utilisant des variables continues

D'autres protocoles de distribution quantique de clé utilisant des variables continues ont été proposés [34, 47]. Ces protocoles reposent sur l'observation que l'information I_{AB} transmise entre Alice et Bob est répartie sur l'ensemble des états cohérents traversant le canal quantique. Cette répartition rend le processus de réconciliation ardu, car, pour les faibles transmissions, il faut réussir à extraire peu d'information pour chaque symbole. Toutefois, certains symboles dont la probabilité d'occurrence est faible portent davantage d'information que les symboles typiques. Suivant cette remarque, on peut concevoir des protocoles de type «post-sélection» dans lesquels Alice et Bob sélectionnent *a posteriori* quelques symboles rares parmi la séquence envoyée pour construire leur clé secrète. Cette opération est analogue à la sélec-

tion «naturelle» des photons par les pertes du canal quantique dans les protocoles à photons uniques : pour de faibles transmissions, peu de photons parviennent à Bob, mais ceux que Bob reçoit contiennent beaucoup d'information.

Tout comme les protocoles inverses, les protocoles de type post-sélection permettent la transmission d'une clé secrète pour des transmissions inférieures à 1/2. De plus, ils permettent de simplifier la délicate étape d'extraction de l'information secrète I_{AB}. Mais la sécurité inconditionnelle de ces protocoles est plus délicate à établir [48]. Des preuves de sécurité existent pour des sous-ensembles restreints d'attaques individuelles, comme les attaques de type «lame séparatrice» [34] ou des attaques particulières ajoutant un excès de bruit [49].

Deuxième partie

Réalisation expérimentale

Chapitre 5

Introduction

La deuxième partie de ce manuscrit est dédiée à notre réalisation expérimentale de distribution quantique de clé avec des états cohérents, fonctionnant à longueur d'onde télécom (1,55 μm), et réalisée avec des fibres optiques. Conformément aux principes théoriques énoncés au chapitre 2, l'expérience consiste en la modulation aléatoire en amplitude et en phase d'une série d'impulsions optiques pour Alice, puis en la mesure d'une quadrature aléatoire par Bob. Cette dernière mesure nécessite une référence de phase à partir de laquelle Bob choisit sa quadrature de mesure. C'est pourquoi notre expérience est un interféromètre de Mach-Zender délocalisé entre Alice et Bob. La figure 5.1 montre le schéma expérimental global :

- Nous générons des impulsions de largeur 100 ns qui forment les symboles du canal quantique en découpant le faisceau optique d'une diode laser continue avec un modulateur électro-optique ou en pulsant électriquement une diode (section 9.2). On peut choisir le taux de répétition des impulsions jusqu'à 1 MHz.
- Le système de modulation permet d'appliquer une modulation d'amplitude et de phase arbitraires au signal quantique à l'aide de modulateurs électro-optiques (section 6.4). En particulier, la distribution quantique de clé nécessite une modulation gaussienne dans le plan complexe.
- La mesure d'une quadrature aléatoire est réalisée par une détection homodyne impulsionnelle, limitée au bruit de photon. Elle consiste à faire interférer une référence de phase intense (oscillateur local) avec le signal mesuré (section 6.3).
- Un système de multiplexage permet de transmettre le signal et l'oscillateur local sur une fibre optique de plusieurs dizaines de kilomètres, tout en conservant leur phase et leur polarisation relatives (chapitre 9).
- Des cartes d'acquisition pilotées par un logiciel assurent un fonctionnement continu et automatique de l'expérience (chapitre 10).
- Divers capteurs permettent de calibrer la transmission quantique (section 10.7).

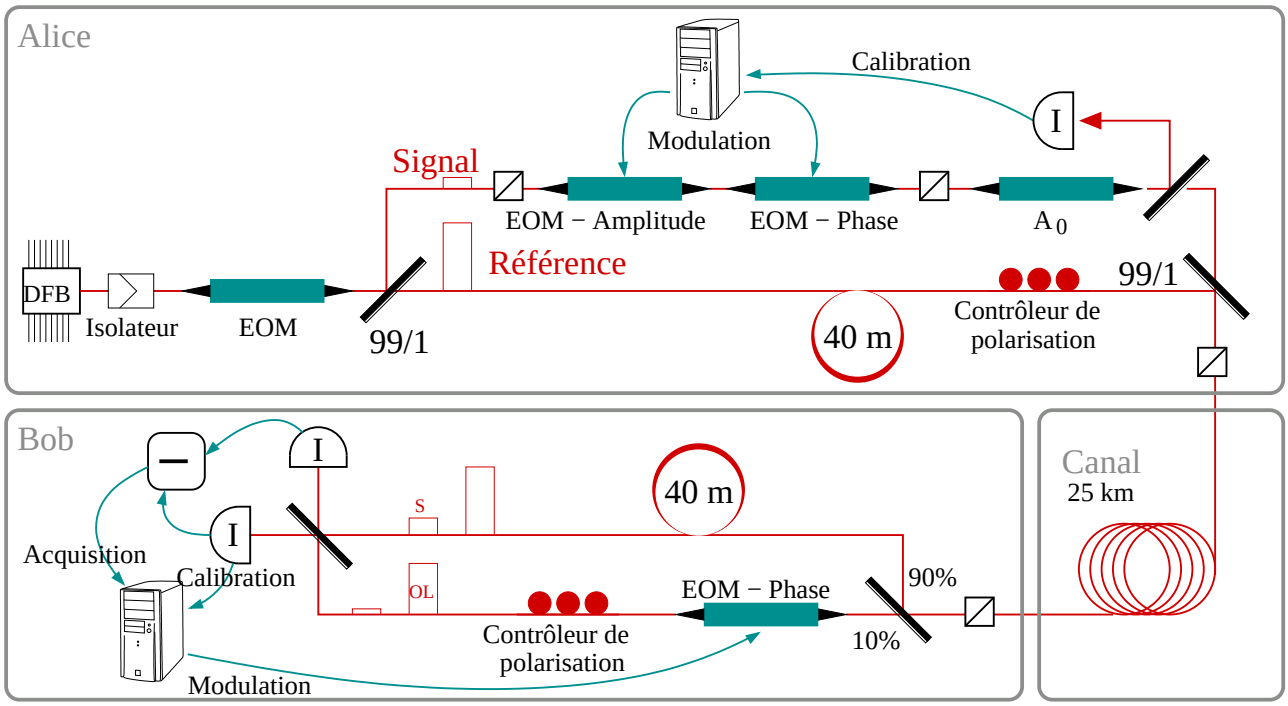


FIG. 5.1: Schéma expérimental montrant chaque sous-sytème de l'expérience.

Chapitre 6

Démonstrateur de distribution quantique de clé avec des états cohérents

Ce chapitre décrit les aspects généraux de notre expérience de distribution quantique de clé. Il expose quelques particularités liées à l'utilisation de fibres optiques et détaille les deux parties principales du dispositif : la modulation d'un état cohérent et la mesure d'une des quadratures de cet état.

6.1 Faire de l'optique quantique avec des fibres optiques

Une des originalités expérimentales de notre système est sa réalisation en fibres optiques exclusivement. Nous décrivons ici quelques aspects spécifiques des fibres par rapport à l'optique en espace libre couramment utilisée dans les expériences d'optique quantique.

La polarisation dans les fibres optiques

Le comportement de la lumière polarisée dans une fibre optique est sans doute un des aspects les plus critiques de notre expérience. Nous utilisons deux types de fibres optiques : les fibres «monomodes» (SM) et les fibres «monomodes à maintien de polarisation» (PM).

Les fibres monomodes ne conservent pas la polarisation. Une torsion de la fibre peut introduire une biréfringence et modifier la polarisation de la lumière. Ainsi, il est possible, en tordant ou en écrasant une fibre monomode, de parcourir l'ensemble de la sphère de Poincaré. C'est le principe des contrôleurs de polarisation passifs que nous utilisons. D'autre part, des dérives thermiques ou des vibrations mécaniques introduisent des fluctuations de la biréfringence qui nuisent à la stabilité des montages fibrés SM. Toutefois, les fluctuations de polarisation dans les fibres monomodes peuvent être maîtrisées simplement : si la fibre est correctement fixée et protégée, sa biréfringence reste stable sur l'échelle de la journée. De plus, si la fibre revient à sa position initiale après une torsion, la polarisation revient elle aussi à sa configuration initiale.

Les fibres à maintien de polarisation possèdent des structures qui introduisent une biréfringence intrinsèque importante devant la biréfringence introduite par les torsions de la fibre. Ainsi, on distingue deux axes propres, appelés axe rapide et axe lent, dont les indices diffèrent de $\Delta n = 10^{-3}$ autour de l'indice de la silice ($n = 1,5$). Cette biréfringence intrinsèque permet une isolation des deux axes supérieure à 25 dB sur 100 m. Suivant une convention établie dans le domaine des télécommunications, nous avons choisi d'utiliser l'axe lent des fibres PM. Autant

Composant	Pertes
Modulateur d'amplitude	3 dB
Switch	2,5 dB
Modulateur de phase	2 dB
Polariseur	0,5 dB
Filtre fréquentiel	0,5 dB
Connexions	0,3 dB
fibre de 1 km	0,2 dB
Coupleur	0,05 dB

TAB. 6.1: Pertes typiques des composants utilisés dans notre expérience de cryptographie quantique.

que faire se peut, nous avons opté pour des fibres PM, qui offrent un important confort d'utilisation. Toutefois, si la polarisation est bien conservée lors de la propagation dans une fibre PM, les défauts d'alignement entre les axes propres des fibres PM au niveau des connecteurs (typiquement 3 °) peuvent s'accumuler. Nous compensons ces défauts en plaçant des polariseurs le long de notre chemin optique. L'ensemble du schéma expérimental représenté figure 5.1 est composé de fibres à maintien de polarisation, à l'exception des lignes à retard (dans une première version de l'expérience), et de l'extrémité de la détection homodyne.

Dérives de phase dans les fibres optiques

Comme nous l'avons vu pour la polarisation, l'indice des fibres peut varier, ce qui occasionne une dérive de la phase relative entre le signal et l'oscillateur local, qui vient perturber notre mesure interférométrique. Cette dérive est d'origine mécanique et thermique. Pour l'éliminer, nous devons fixer les fibres optiques et isoler thermiquement notre système en le plaçant dans une boîte. Finalement, nous observons une dérive de la phase relative entre les deux voies de notre interféromètre de l'ordre de 1 rad/s, ce qui est négligeable devant les durées d'acquisition (voir chapitre 10).

Contrôle des pertes

Les composants fibrés souffrent souvent de plus de pertes optiques que leurs équivalents en espace libre. Nous distinguons plusieurs classes de composants. D'abord, les composants passifs (coupleurs, polariseurs, filtres...) qui occasionnent des pertes modérées. Ensuite, les composants actifs en niobate de lithium (modulateurs, switch) qui induisent de larges pertes au niveau du couplage entre la fibre et le guide d'onde qui les constitue. Enfin, chaque connexion optique introduit des pertes. Le tableau 6.1 résume les pertes de ces composants.

Nous devons contrôler les pertes à plusieurs égards :

- Chez Bob, sur la voie signal, toute perte optique se traduit par une dégradation du rapport signal à bruit et donc par une chute du taux secret. Heureusement, ces pertes sont limitées : sur la voie signal, on ne trouve que le coupleur de démultiplexage et le coupleur de la détection homodyne. Il faut aussi ajouter aux pertes optiques l'efficacité limitée des photodiodes de la détection homodyne (80%). En revanche, chez Alice, les pertes sur le signal ne jouent aucun rôle puisque seule la variance de modulation en entrée du canal est utilisée pour l'évaluation du taux secret. Au contraire, nous devons atténuer

suffisamment le signal pour atteindre les quelques photons par impulsion qui maximisent le taux secret[1]. Pour cela, nous séparons et recombinons le signal et l'oscillateur local avec deux coupleurs 99/1 qui laissent inchangée la puissance de l'oscillateur local tout en atténuant suffisamment le signal (voir section 6.4 pour un bilan complet).
- Les pertes sur la voie oscillateur local n'ont pas d'effet direct sur le taux secret. Pourtant, la variance du bruit de photon croît avec la puissance d'oscillateur local, donc un oscillateur local intense réduit l'importance relative du bruit électronique constant du circuit de détection par rapport au bruit de photon. Nous verrons au chapitre suivant qu'une puissance d'oscillateur local d'au moins 10^7 photons par impulsion est nécessaire.
- Enfin, les pertes sur le canal de transmission sont imposées par la longueur de ce canal. Elles sont typiquement de 6 dB pour un canal de 25 km[2].

Reproductibilité et inspection des connexions

Une difficulté des montages fibrés réside dans l'impossibilité d'accéder au faisceau optique sans modifier le montage. Il est facile de mesurer ou d'atténuer la puissance d'un faisceau optique libre sans perturber le montage optique. Sur un montage fibré, ces opérations nécessitent de débrancher un connecteur. Lorsqu'on le rebranche après la mesure, les pertes au couplage ainsi que l'alignement entre les axes propres des fibres peuvent avoir changé.

Ensuite, les fibres ont une longueur fixe, ce qui rend peu aisé l'équilibrage des voies d'un interféromètre. Pour cette opération, nous disposons d'un jeu de fibres de longueurs variées, comprises entre 10 cm et 1 m. Une combinaison adéquate de ces fibres nous permet d'atteindre une résolution de 1 cm. Pour atteindre des équilibrages plus précis, de l'ordre du centimètre, il est possible de cliver puis souder des fibres, au prix de l'irréversibilité du processus. Nous verrons que notre interféromètre requiert deux équilibrages :
- l'équilibrage des voies signal et oscillateur local. Nous le réalisons en trouvant une combinaison idoine de fibres.
- l'équilibrage des voies de sortie de l'interféromètre, entre le coupleur de sortie et les photodiodes de la détection. Nous atteignons une précision de 5 mm en soudant soigneusement les fibres des photodiodes aux fibres du coupleur. En raccourcissant les broches des photodiodes, nous obtenons une précision inférieure au millimètre.

Terminons notre tour d'horizon par quelques avantages des fibres optiques. Tout d'abord, nous utilisons des fibres monomodes, qui éliminent tout problème d'adaptation de la taille des faisceaux optiques ou de mode spatial, notamment lors de l'interférence entre le signal et l'oscillateur local. Ensuite, les fibres optiques ne nécessitent pas d'alignement, ni de pièces mécaniques : nos fibres sont fixées par du ruban adhésif. L'utilisation de fibres permet également de cloisonner l'expérience : nous pouvons agir sur une partie précise du dispositif sans perturber les parties aval. Enfin, hormis au niveau des connecteurs, les composants fibrés sont sertis et permettent un bonne reproductibilité.

6.2 Composants utilisés

Nous avons utilisé les composants suivants :

[1]Nous aborderons la variance de modulation optimale au chapitre 13.
[2]Le rouleau de fibre que nous avons utilisé est de bonne qualité, avec des pertes de 5 dB pour 25 km.

Fibres optiques (ou jarretières) commercialisées par la société Jenoptec. Les fibres monomodes ont un cœur en silice de diamètre 9 μm, recouvert d'un cladding de 125 μm. Les fibres à maintien de polarisation, de type PANDA, sont manufacturées par la société Fujikura. Elles ont un cœur de 8 μm et une isolation en polarisation de 25 dB par 100 m. Les propriétés de ces fibres sont garanties pour un rayon de courbure supérieur à 3 cm.

Canal de transmission Rouleau de fibre monomode, de la société Draka (anciennement Alcatel, division fibres optique), de longueur 25 km, introduisant des pertes de 5 dB.

Coupleurs fournis par la société britannique SIFAM, de type SM ou PM. Ces coupleurs ont des pertes intrinsèques particulièrement faibles (0,05 dB). Le rapport de couplage des coupleurs monomodes ne dépend pas de la polarisation d'entrée (dépendance supérieure à -30 dB).

Modulateurs de la société EOSpace, présentant des pertes modérées, une excellente extinction ($>$ 30 dB), un polariseur intégré et une photodiode de contrôle. Ces modulateurs ont des tensions de commande particulièrement stables (dérive de 0.1 V pour 30 minutes)

Photodiodes de la société JDS-Uniphase (anciennement Epitaxx). Ces photodiodes ont une réponse de 1,0 A/W (soit un rendement quantique de 80%), sélectionnées dans un lot de 15 photodiodes de réponse moyenne 0,9 A/W. Elles ont un diamètre actif de 0,1 mm pour une capacité parasite de 5 pF. Nous les polarisons en inverse avec une tension de 6 V. Nous avons opté pour la version "pigtail", c'est-à-dire des photodiodes couplées en usine à une fibre monomode, plutôt que la version "réceptacle" pour fibre, plus instable et moins reproductible.

Polariseurs de la société NovaWave Technologies. Ils présentent une isolation de 25 dB. Certains modèles peuvent séparer les polarisations selon les axes lent et rapide de la fibre d'entrée vers deux fibres de sortie distinctes. Nous avons également utilisé des polariseurs de la société General Photonics, dont le corps est solidaire des connecteurs, c'est-à-dire sans fibres optiques souples en entrée et en sortie.

Atténuateur variable Pour équilibrer finement notre système de détection homodyne, nous avons utilisé un atténuateur variable de la société IDIL. Cet atténuateur introduit des pertes en courbant une fibre optique montée sur une platine de translation. Ce modèle a été spécifiquement modifié pour permettre de petites atténuations, entre 0 et 10^{-2}.

Contrôleurs de polarisation passifs de la société General Photonics. Ils permettent d'atteindre un point arbitraire de la sphère de Poincaré en écrasant et tordant la fibre.

Source laser Nous avons principalement utilisé une diode laser DFB continue de marque Princeton Lightwave (épuisée), délivrant 200 mW à une longueur d'onde de 1545 nm. Nous l'utilisons à un niveau de sortie de 80 mW, niveau pour lequel elle présente un bruit de phase minimal. Les résultats du chapitre 9 sont partiellement obtenus avec une diode laser DFB pulsée électriquement, de référence Alcatel A1905LMI, et de puissance crête 15 mW.

Contrôleur de polarisation actif DPC55000 de la société Thorlabs. Ce contrôleur muni d'un polarimètre peut stabiliser automatiquement une polarisation fluctuante par une boucle de rétroaction. Nous envisageons l'utilisation de ce dispositif en sortie du canal quantique pour compenser les fluctuations de polarisation dans les fibres optiques télécom installées. Le rouleau de fibre de 25 km que nous utilisons comme canal quantique au laboratoire est assez isolé des perturbations extérieures (perturbations mécaniques et thermiques) pour ne pas nécessiter ce contrôleur de polarisation.

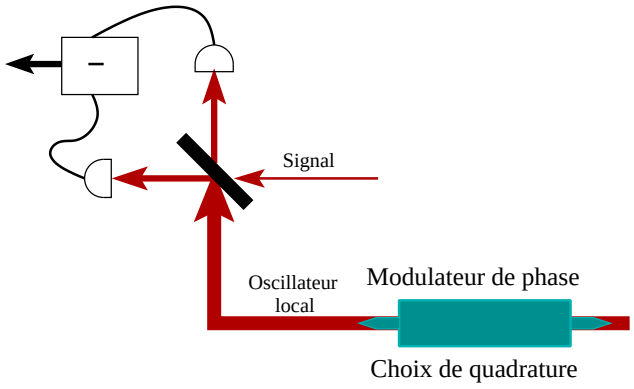


FIG. 6.1: Schéma de principe d'une détection homodyne.

Tous les connecteurs entre les composants sont de type FC/APC. Pour ces connecteurs, l'interface de la connexion est inclinée de 7 ° par rapport au plan transverse de la fibre pour éviter toute réflexion parasite de la lumière dans la fibre.

6.3 Détection homodyne impulsionnelle limitée au bruit de photon

Une détection homodyne est un interféromètre qui permet de mesurer une quadrature du champ électromagnétique. Pour cela, on fait interférer le champ signal E_{signal} avec un champ de référence E_{OL}, appelé oscillateur local. On mesure ensuite l'intensité en sortie de chaque bras de l'interféromètre (figure 6.1). On exprime ces intensités $I_\pm$ en fonction des champs électriques d'entrée :

$$\begin{aligned} E_{\text{OL}} &= \sqrt{I_{OL}} e^{i\phi} e^{i\omega t} \\ E_{\text{signal}} &= (X + iP) e^{i\omega t} = (X_\phi + iP_\phi) e^{i\phi} e^{i\omega t}. \end{aligned}$$

d'où

$$\begin{aligned} E_\pm &= \frac{1}{\sqrt{2}} (\sqrt{I_{OL}} \pm X_\phi \pm iP_\phi) e^{i\phi} e^{i\omega t} \\ \text{puis} \quad I_\pm &= |E_\pm|^2 = \frac{1}{2}((\sqrt{I_{OL}} \pm X_\phi)^2 + P_\phi^2) \\ \text{d'où} \quad I_+ - I_- &= 2\sqrt{I_{OL}} X_\phi \end{aligned}$$

Ainsi, lorsque la détection est bien équilibrée, la différence des intensités en sortie de l'interféromètre est proportionnelle à la valeur X_ϕ de la quadrature mesurée, amplifiée par l'intensité de l'oscillateur local. On sélectionne la quadrature de mesure désirée en faisant varier la phase relative ϕ entre le repère associé aux quadratures (X, P) et l'oscillateur local à l'aide d'un modulateur de phase. Pour éviter toute perte sur la voie signal, ce modulateur est placé sur la voie oscillateur local du dispositif de Bob.

Typiquement, l'oscillateur local contient 10^7 photons par impulsion, alors que le signal contient 10 photons par impulsion. La différence des photo-courants est donc de l'ordre de

5 10^4 électrons pour 100 ns, soit un courant électrique de $0,1$ μA. Pour obtenir un signal observable, nous devons amplifier ce courant en introduisant un bruit électronique assez faible (chapitre 7).

En pratique, la soustraction des intensités n'est pas parfaite car le coupleur responsable de l'interférence a une transmission différente de 1/2 et car les voies de sorties de la détection homodyne peuvent avoir des pertes différentes (notamment, les efficacités des deux photodiodes ne sont pas exactement identiques). De plus, l'oscillateur local peut présenter un bruit autour de sa valeur moyenne. Nous modélisons ces défauts en introduisant un déséquilibre ϵ entre les voies de sortie de la détection homodyne et en exprimant le champ oscillateur local

$$E_{\mathrm{OL}} = \overline{E_{\mathrm{OL}}} + \delta E_{\mathrm{OL}} \tag{6.1}$$

où $\overline{E_{\mathrm{OL}}}$ est un champ réel[3] constant, et δE_{OL} est un bruit complexe. Dans ce modèle, nous reformulons les équations précédentes :

$$E_{\pm} = \sqrt{\frac{1}{2} \pm \epsilon}\left(\overline{E_{\mathrm{OL}}} + \delta E_{\mathrm{OL}} \pm E_{\mathrm{signal}}\right) \tag{6.2}$$

$$I_{\pm} = \left(\frac{1}{2} \pm \epsilon\right)\left[\left(\overline{E_{\mathrm{OL}}} + \mathrm{Re}[\delta E_{\mathrm{OL}}] \pm \mathrm{Re}[E_{\mathrm{signal}}]\right)^2 + \left(\mathrm{Im}[\delta E_{\mathrm{OL}}] \pm \mathrm{Im}[E_{\mathrm{signal}}]\right)^2\right] \tag{6.3}$$

En supposant $|E_{\mathrm{signal}}| \ll \overline{E_{\mathrm{OL}}}$ et $\delta E_{\mathrm{OL}} \ll \overline{E_{\mathrm{OL}}}$, on ne conserve que les termes au moins proportionnels à $\overline{E_{\mathrm{OL}}}$ dans le développement de $I_{\pm}$. On obtient finalement

$$I_+ - I_- = 2\overline{E_{\mathrm{OL}}}\left[\mathrm{Re}[E_{\mathrm{signal}}] + 2\epsilon\left(\overline{E_{\mathrm{OL}}} + \mathrm{Re}[\delta E_{\mathrm{OL}}]\right)\right]. \tag{6.4}$$

Deux termes s'ajoutent à la mesure $\mathrm{Re}[E_{\mathrm{signal}}]$ de la quadrature du signal en phase avec l'oscillateur local dérivée précédemment. D'abord, un terme constant $2\epsilon\overline{E_{\mathrm{OL}}}$ qui vient ajouter un décalage constant dans le signal de sortie de la détection homodyne. Pour une intensité d'oscillateur local $|E_{\mathrm{OL}}|^2 = 10^7$ photons, nous pouvons équilibrer la détection homodyne de façon à observer un décalage de l'ordre de l'écart type du bruit de photon $\sqrt{N_0}$. Nous en déduisons l'estimation $\epsilon = \frac{1}{2\sqrt{10^7}} \simeq 10^{-4}$. Le deuxième terme dû au déséquilibre de la détection homodyne est le bruit $2\epsilon\mathrm{Re}[\delta E_{\mathrm{OL}}]$. Ce bruit n'est autre que le bruit d'amplitude (c'est-à-dire le bruit selon la quadrature parallèle à l'amplitude) de l'oscillateur local, atténué d'un facteur 2ϵ. Dans le chapitre suivant, consacré à l'étude des bruits de notre dispositif de distribution de clé, nous quantifierons les effets respectifs du bruit électronique et du bruit d'amplitude de l'oscillateur local.

L'équilibrage de la détection homodyne est obtenu en courbant les fibres de sortie du coupleur d'interférence avec un rayon de courbure de l'ordre de 2 cm. Nous avons d'abord effectué ce réglage à l'aide d'un système mécanique commercial (société IDIL fibres optiques) monté sur une platine de translation. Toutefois, ce réglage si sensible en espace libre est particulièrement stable pour une détection homodyne fibrée. Nous pouvons alors obtenir un réglage de précision équivalente en maintenant une courbure avec du ruban adhésif. Nous attribuons la stabilité remarquable de l'équilibrage de la détection homodyne à l'insensibilité à la polarisation de notre coupleur fibré monomode : les petites fluctuations de polarisation à l'entrée de la détection

[3] Supposer ce champ réel est équivalent à fixer la phase globale de notre interféromètre.

homodyne ne changent pas le rapport de couplage. Ensuite, les photodiodes serties permettent de garder un couplage constant entre le faisceau optique et la surface de la photodiode. En effet, une des sources d'instabilité d'une détection espace libre est le déplacement du faisceau lumineux sur la photodiode, qui provoque de petites fluctuations d'efficacité.

Nous devons équilibrer les longueurs des deux voies de sortie de la détection homodyne. Si les deux impulsions de temps de montée τ mesurées par chacune des photodiodes sont soustraites avec un décalage δt, on observe deux pics de signe opposé de part et d'autre du signal de détection homodyne. On peut interpréter ces deux pics comme la dérivée du signal optique. On montre simplement que l'intensité δI de ces pics est proportionnelle à l'intensité I de l'impulsion optique et au décalage δt : $\delta I/I = \delta t/\tau$. Pour éviter de saturer notre signal de détection homodyne, nous souhaitons que l'amplitude de ces pics soit au plus de l'ordre de grandeur de ce signal. Nous voulons donc $\delta I \simeq 5\ 10^4$ photons pour $I = 10^7$ photons. Compte tenu du temps de montée de notre impulsion optique $\tau = 2$ ns, nous obtenons $\delta t = 2$ ps, ce qui correspond à une différence de longueur entre les voies de sortie de la détection homodyne de 0,4 mm. Comme nous l'avons vu, cet équilibrage est effectué par la soudure des fibres optiques et l'ajustement de la longueur des broches des photodiodes.

Enfin, la bande passante le l'électronique de la détection homodyne doit être inférieure à l'inverse de la largeur des impulsions. Ainsi, la détection homodyne ne résout pas les impulsions, et fournit un signal proportionnel à l'énergie totale de l'impulsion. Si nous choisissions une bande passante plus large, l'échantillonnage du signal homodyne fournirait une mesure de l'énergie contenue dans une partie seulement de l'impulsion optique. Dans ce cas, Ève pourrait exploiter la partie de l'impulsion invisible à notre système. Nous avons évalué la bande passante de la détection homodyne en mesurant l'amplitude du signal de détection homodyne pour différentes largeurs d'impulsions. Nous constatons que le rapport entre le signal mesuré et l'aire de l'impulsion reste constant pour des impulsions de largeur inférieure à 100 ns. Au-delà de 100 ns, ce rapport décroît.

6.4 Modulation gaussienne avec des modulateurs électro-optiques

Alice module l'amplitude et la phase du signal transmis à travers le canal quantique à l'aide de modulateurs électro-optiques pilotés par une carte d'acquisition. Ces modulateurs sont constitués d'un guide d'onde en niobate de lithium dont l'indice selon un axe cristallin particulier varie significativement avec le champ électrique qui le traverse. Il est ainsi possible d'introduire un déphasage arbitraire dans le guide d'onde avec une tension modérée. Ces modulateurs électro-optiques ont une bande passante de 10 GHz, particulièrement adaptée aux télécommunications optiques. Le temps nécessaire à l'établissement d'un signal constant est donc bien inférieur aux capacités de nos cartes d'acquisition, qui fonctionnent à un taux de répétition de 1 MHz et qui ont un temps de monté de 200 ns (section 10.1).

Un modulateur de phase est simplement un guide d'onde en niobate de lithium sur lequel sont posées des électrodes alimentées par une tension V_ϕ. La tension V_π est la tension que l'on doit appliquer pour obtenir une variation de phase de π ; elle est de l'ordre de 4 V. La phase ϕ appliquée au signal est proportionnelle à la tension de commande du modulateur V_ϕ. La phase

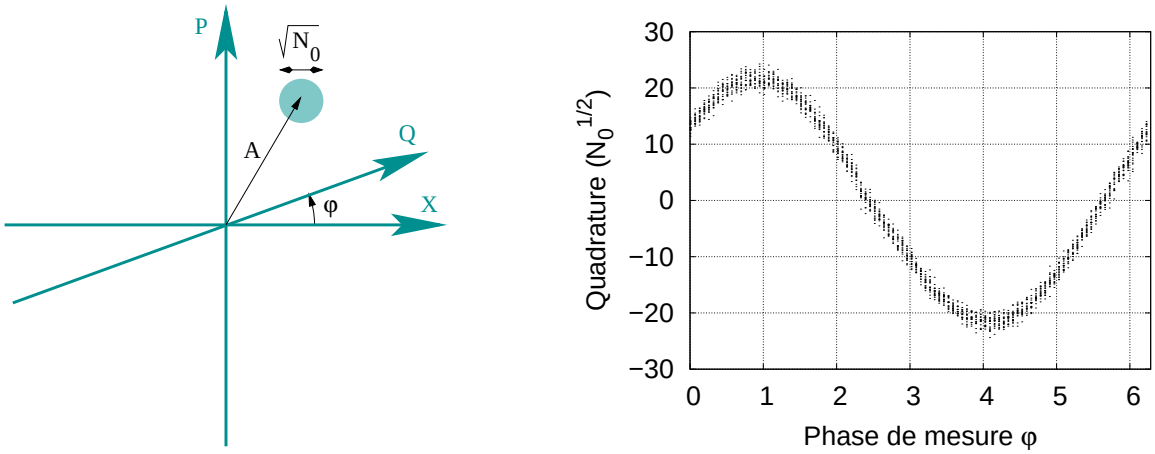


FIG. 6.2: Tomographie d'un état cohérent. En faisant varier la phase de mesure ϕ, on observe la projection de l'état cohérent sur chaque quadrature Q. Dans cette figure, nous reconstituons la distribution de probabilité de la projection de l'état cohérent sur la phase de mesure en effectuant 18 mesures pour chacune des 100 phases choisies.

étant définie à une constante additive près, nous avons choisi la transformation :

$$V_\phi = V_\pi \left(\frac{\phi}{\pi} - 1 \right). \tag{6.5}$$

Un modulateur d'amplitude est un interféromètre de Mach-Zender intégré dans un guide d'onde en niobate de lithium. En appliquant une tension sur l'une des voies de l'interféromètre, on peut changer la différence de phase entre les deux voies, et donc l'intensité lumineuse en sortie. Pour ces modulateurs, la tension V_π est la variation de tension qu'il faut appliquer pour passer d'une frange sombre (extinction du modulateur) à une frange brillante (maximum de transmission). Comme pour le modulateur de phase, cette tension est de l'ordre de 4 V. Pour un modulateur d'amplitude, la tension appliquée au guide d'onde est la somme de deux tensions. La première est la tension de modulation proprement dire, notée V_{RF}, qui permet de sélectionner une transmission. Cependant, à tension V_{RF} constante, la phase relative entre les deux voies de l'interféromètre peut varier, par exemple par dilatation thermique d'une des voies. Ainsi, au cours du temps, une même tension V_{RF} ne correspond pas à une transmission fixe. C'est pourquoi nous pouvons appliquer une deuxième tension V_{biais}, appelée tension de biais, qui est régulièrement ajustée pour stabiliser la relation entre la tension V_{RF} et la transmission du modulateur. En pratique, il est nécessaire d'ajuster la tension de biais sur l'échelle d'une dizaine de minutes.

À l'aide de notre système de détection homodyne, nous mesurons la position de l'état cohérent dans le plan complexe en sortie du modulateur d'amplitude en fonction de la tension $V = V_{RF} + V_{biais}$ appliquée. Cette mesure se heurte à deux difficultés que nous rencontrerons fréquemment. D'abord, à cause du bruit de photon, le signal de détection homodyne ne mesure pas le centre de l'état cohérent, mais un point dans le nuage d'incertitude d'aire N_0 autour de ce centre : alors que le bruit de photon permet de garantir la sécurité du protocole, il rend la caractérisation du dispositif plus difficile. Ensuite, la mesure homodyne ne

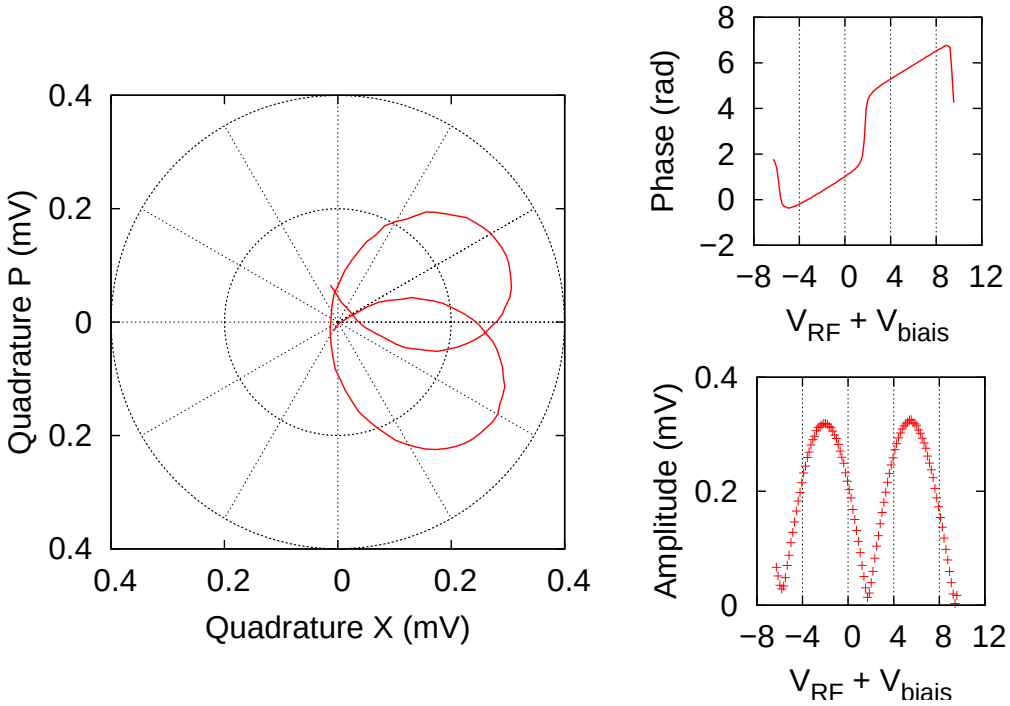


FIG. 6.3: Effet du modulateur d'amplitude sur un état cohérent. La figure de gauche représente le déplacement du centre de l'état cohérent en sortie du modulateur d'amplitude lorsque nous balayons la tension V_{RF} appliquée au modulateur. L'amplitude et la phase sont reportées individuellement sur les figures de droite. Nous constatons que le modulateur d'amplitude introduit un déphasage linéaire $\phi = \alpha V_{RF}$, avec $\alpha = 0,31$ rad/V, que nous devons compenser. De plus, on observe plusieurs cycles transmission maximale – extinction, de qualités d'extinction inégales. Pour notre modulation d'amplitude, nous nous placerons donc sur la dernière demi-arche, entre 5 V et 9 V, qui présente une dynamique de modulation de 20 dB en amplitude, ou 40 dB en intensité. Les courbes représentées sur ces figures sont issues de l'acquisition de 5 000 000 d'impulsions (soit 100 tensions de modulation différentes, chacune demandant l'acquisition de 50 000 impulsions nécessaires à la tomographie de l'état cohérent), acquises en 5s. À cette échelle de temps, la tension de biais du modulateur $V_{bias} = 1,7$ V est considérée constante.

donne accès qu'à la projection de l'état cohérent sur la quadrature de mesure. Une technique de tomographie nous permet de surmonter ces deux problèmes : nous éliminons le bruit de photon en effectuant des mesures sur plusieurs états cohérents identiques et nous répétons ces mesures selon plusieurs quadratures. En pratique, nous faisons 500 mesures sur 100 quadratures différentes. La tomographie d'un état cohérent (figure 6.2) dessine une sinusoïde dont l'amplitude et la phase sont l'amplitude et la phase de notre état cohérent.

Nous répétons la tomographie décrite ci-dessus pour différentes tensions de modulation entre -8 V et +8V ; nous obtenons les courbes de la figure 6.3. La variation de l'amplitude avec la tension nous permet de mesurer $V_\pi = 3,8$ V. Nous constatons que le modulateur d'amplitude introduit une modulation de phase linéaire $\phi = \alpha V_{RF}$ ($\alpha = 0,31$ rad/V) que nous modélisons par une dissymétrie entre les tensions appliquées sur les deux voies de l'interféromètre :

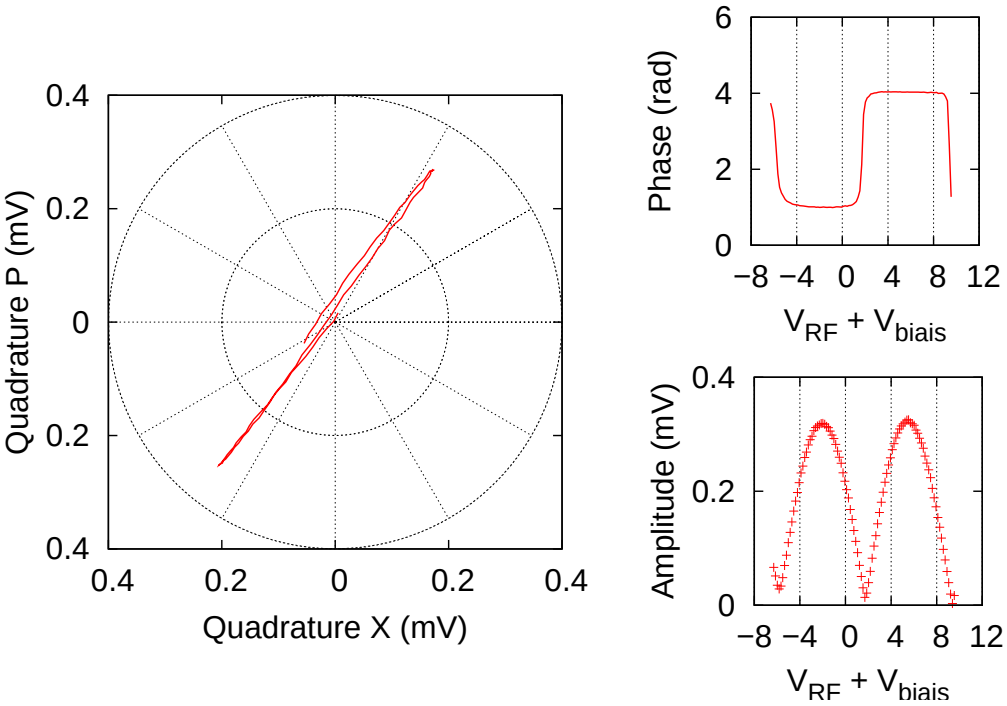
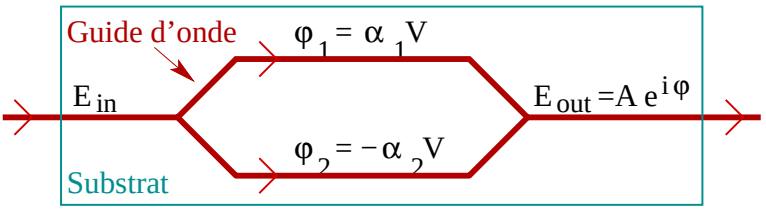


FIG. 6.4: Modulation d'amplitude corrigée. Avec le modulateur de phase, nous introduisons un déphasage qui vient compenser la modulation de phase introduite par le modulateur d'amplitude. Ici, nous retranchons une phase αV_{RF}, avec $\alpha = 0.31$ rad/V, aux mesures représentées figure 6.3.



$$\begin{cases} \phi_1 = \alpha_1 V \\ \phi_2 = -\alpha_2 V \end{cases} \Rightarrow \begin{cases} E_{out} &= \frac{E_{in}}{2}\left(e^{i\phi_1} + e^{i\phi_2}\right) = E_{in} e^{i\frac{\alpha_1-\alpha_2}{2}V} cos\left(\frac{\alpha_1+\alpha_2}{2}V\right) \\ A &= |E_{out}| = |E_{in} cos\left(\frac{\alpha_1+\alpha_2}{2}V\right)| \\ \phi[\pi] &= \frac{\alpha_1-\alpha_2}{2}V \\ I_{out} &= A^2 = \frac{I_{in}}{2} cos\left(\left(\alpha_1+\alpha_2\right)V\right). \end{cases} \tag{6.6}$$

Ces expressions rendent compte des courbes expérimentales, avec $\alpha_1 + \alpha_2 = \pi/V_\pi$ et $\alpha_1 - \alpha_2 = 2\alpha$. Nous calculons $\alpha_1 = 0,73$ rad/V et $\alpha_2 = 0,10$ rad/V. La dissymétrie des électrodes provient de la technologie (appelée "Z-cut") employée pour la construction des modulateurs, qui permet d'avoir une meilleure bande passante et une meilleure extinction.

La transformation reliant les tensions aux bornes des modulateurs de phase et d'amplitude (V_{RF}, V_ϕ) et la position de l'état cohérent (A, ϕ) dans le plan complexe en sortie d'Alice devient finalement :

$$V_{RF} = \frac{V_{\pi,A}}{\pi} \cos^{-1}\left(\pm\left(1 - 2\,A^2\right)\right) \tag{6.7}$$

$$V_\phi = V_{\pi,\phi}\left(\frac{\phi - \alpha V_{RF}}{\pi} - 1\right). \tag{6.8}$$

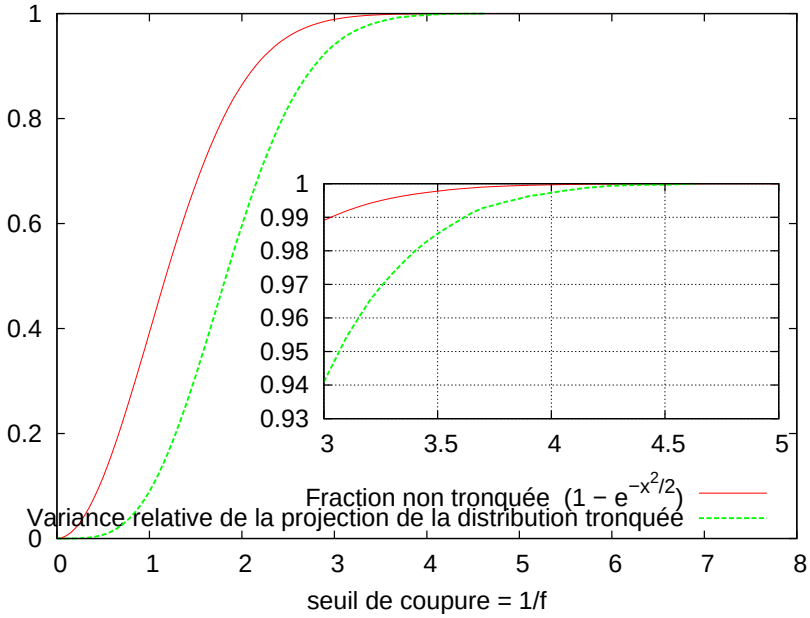


FIG. 6.5: Effet de la troncature d'une modulation gaussienne bidimensionnelle, de variance σ, à un rayon $f\sigma$. La courbe pleine représente la proportion P de données non tronquées. La courbe pointillée montre le rapport théorique entre la variance σ' des données homodynes telles que mesurées par Bob et la variance σ de la distribution originale, non tronquée. Cette dernière courbe est obtenue par une simulation de Monte-Carlo sur 500 000 échantillons. Pour $f = 3$, seul 1% des données est tronqué. Pourtant, une erreur de 6% est commise sur l'évaluation de la variance des données mesurées par Bob. Nous avons effectivement constaté cet écart sur nos données expérimentales, c'est pourquoi nous avons finalement opté pour $f = 4$.

où $V_{\pi,A}$ et $V_{\pi,\phi}$ sont les tensions V_π respectives des modulateurs d'amplitude et de phase. Nous corrigeons la modulation de phase introduite par le modulateur d'amplitude à l'aide du modulateur de phase, comme représenté figure 6.4. Cette transformation fait varier la tension V_{RF} entre 0 et V_π. Nous ajustons ensuite la tension de biais V_{bias}, appliquée à l'aide d'un générateur de tension statique variable, pour que l'extinction du modulateur corresponde à une tension $V_{RF} = 0$ ($\pm \equiv +$) ou $V_{RF} = V_\pi$ ($\pm \equiv -$). Nous avons choisi la deuxième configuration, pour laquelle notre modulateur présente une meilleure extinction. Notons encore une fois que la tension V_ϕ est définie à une constante près.

Enfin, les guides d'onde des modulateurs électro-optiques sont dissymétriques selon les axes de polarisation vertical et horizontal. Par conséquent, la tension V_π diffère selon que la polarisation de la lumière est suivant l'un ou l'autre des deux axes. Nous avons mesuré $V_{\pi,horiz} \simeq 4V_{\pi,vert}$. Si la polarisation de la lumière n'est pas exactement alignée avec l'axe usuel vertical, nous observons une modulation de polarisation à la sortie des modulateurs et une mauvaise extinction des modulateurs d'amplitude. Un tel défaut d'alignement peut être provoqué par des connecteurs (les connecteurs entre fibres à maintien de polarisation respectent l'orientation des fibres avec une précision de 3 °), ou par un mauvais alignement de l'axe de la fibre avec l'axe du guide d'onde. Pour éviter ces problèmes, nous avons utilisé des modulateurs dotés d'un polariseur intégré à l'entrée du guide d'onde, et nous avons placé des polariseurs fibrés entre les modulateurs.

Nous utilisons les relations entre position de l'état cohérent et tensions de commande pour appliquer une modulation gaussienne en amplitude et en phase. Comme une modulation gaussienne a une extension infinie incompatible avec la dynamique finie de la modulation d'amplitude, nous devons tronquer notre modulation à f écarts types σ. Ainsi, l'amplitude maximale que nous pouvons sélectionner avec notre modulateur d'amplitude est $f\sigma$. Pour choisir f, nous calculons la fraction P d'impulsions non tronquées :

$$P = \int_0^{f\sigma} \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}} 2\pi r dr = 1 - e^{-\frac{1}{2f^2}} \tag{6.9}$$

Par exemple, pour $f = 3$, $P = 99,1\%$. Pour que la troncature des données n'affecte pas la distribution des mesures homodynes, nous voulons également que la variance σ' de la distribution des données tronquée projetée sur une quadrature de mesure soit proche de la variance de la distribution des données non tronquée. Nous avons calculé cette variance σ' par une simulation de Monte-Carlo, dont nous reproduisons le résultat figure 6.5. Nous choisissons finalement $f = 4$, valeur pour laquelle $P > 99,96\%$ et $\sigma'/\sigma > 99,7\%$.

Les pertes sur la voie signal permettent l'écart nécessaire entre la puissance de l'oscillateur local et la puissance du signal. D'abord, les coupleurs de séparation et de multiplexage introduisent des pertes totales de 40 dB sur la voie signal. Les pertes des trois modulateurs disposés sur la voie signal introduisent 10 dB supplémentaires. Le rapport $f^2 = 16$ entre la transmission maximale du modulateur d'amplitude et la variance de modulation compte pour 12 dB. Comme nos impulsions comportent initialement environ 10^8 photons par impulsion, nous obtenons une variance de modulation V_A maximale de 50 photons. Finalement, nous pouvons ajuster cette variance à l'aide d'un deuxième modulateur d'amplitude placé sur la voie signal.

Le dispositif que nous avons décrit dans ce chapitre permet de réaliser l'échange quantique : Alice peut envoyer des états cohérents modulés en amplitude et en phase ; Bob, de son côté, peut mesurer une quadrature aléatoire du signal envoyé par Alice. Le chapitre suivant a pour objectif de quantifier l'information secrète échangée, compte tenu des diverses imperfections expérimentales de notre dispositif.

Chapitre 7

Excès de bruit et analyse de sécurité

La transmission quantique produit des chaînes de variables gaussiennes corrélées entre Alice et Bob. Avant d'extraire une clé secrète de ces données gaussiennes, nous devons calculer le taux d'information secrète transmise sur le canal quantique. L'évaluation de ce taux nécessite :

1. le calcul de la corrélation entre les données d'Alice et de Bob, obtenue par des statistiques sur un échantillon de données révélées (voir section 7.1) ;
2. la mesure de la variance du bruit de photon, qui permet de calibrer la variance V_B des données reçues par Bob (voir sections 7.3 et 10.7) ;
3. la mesure de la variance de la modulation V_A issue d'Alice (voir section 10.7).

Ces trois grandeurs sont les moments d'ordre deux de la distribution de probabilité conjointe des données collectées par Alice et Bob. Comme nous l'avons vu au chapitre 4, le taux secret calculé à partir de ces grandeurs est prouvé sûr contre les attaques individuelles ou collectives, gaussiennes ou non gaussiennes.

La corrélation quantifie le rapport signal à bruit de la transmission. Ce rapport signal à bruit est en pratique dégradé par des bruits de diverses sources que nous allons rencontrer dans ce chapitre. Nous détaillerons l'ensemble de ces bruits, afin de comprendre et corriger les imperfections de notre dispositif expérimental. Mais la connaissance de leur nature et de leur poids respectif n'est en aucun cas nécessaire à la détermination du taux secret, seul paramètre pertinent pour caractériser la clé secrète produite. Toutefois, ceux de ces bruits dont la connaissance est précise pourront être retranchés du bruit accessible à l'espion afin de ne pas détériorer le taux secret. C'est le mode réaliste que nous avons déjà rencontré au chapitre 3.

Notre cheminement vers la détermination du taux secret ira des grandeurs les plus immédiates vers celles dont l'évaluation requiert le plus d'éléments. Ainsi, nous introduirons, dans l'ordre, nos trois moments d'ordre deux au fil des sections de ce chapitre. Nous commencerons par le calcul de l'information mutuelle I_{AB}, puis nous détaillerons une à une chacune des sources de bruit de notre expérience. Enfin, le bruit total, somme de tous les bruits expérimentaux, permet de calculer le taux secret ΔI produit par notre transmission quantique.

Les résultats expérimentaux de ce chapitre ont fait l'objet de la publication [50].

7.1 Corrélations et information mutuelle I_{AB}

Le taux d'information mutuelle I_{AB} est le nombre de bits qu'Alice et Bob se sont échangés lors de la transmission quantique. Il se calcule directement à partir des

corrélations. Le coefficient de corrélation ρ quantifie les corrélations entre deux variables aléatoires X_A et X_B de variances respectives V_A et V_B et de valeurs moyennes nulles. Il s'exprime

$$\rho^2 = \frac{\langle X_A X_B \rangle^2}{V_A V_B}. \tag{7.1}$$

S'il s'agit de caractériser notre dispositif, les calculs statistiques sont effectués sur l'ensemble des données transmises par le canal quantique. En revanche, si nous nous plaçons dans une situation de distribution quantique de clé, les calculs sont effectués sur un sous-ensemble des données quantiques, sacrifié à cet effet.

Le coefficient de corrélation est une grandeur sans dimension, qui ne dépend ni de la normalisation ni de la dimension des variables X_A et X_B. Ainsi, aucune calibration des données expérimentales n'est nécessaire pour l'évaluer : on peut utiliser le signal de sortie de la détection homodyne exprimé en mV et les données envoyées par Alice normalisées à une variance unité. Le coefficient de corrélation n'a pas de signification physique directe. On peut cependant en déduire le rapport signal à bruit de la transmission quantique :

$$SNR = \frac{1}{1-\rho^2} - 1 \tag{7.2}$$

Démonstration. Nous notons X_A la valeur de la quadrature choisie par Alice, et X_B le résultat de la mesure de Bob. Ces deux grandeurs sont reliées par :

$$X_B = g(X_A + X_N) \tag{7.3}$$

où $G = g^2$ est la transmission entre Alice et Bob. On rappelle les notations introduites dans le chapitre 2 : la transmission $G < 1$ entre Alice et Bob est le produit de la transmission T du canal quantique et de l'efficacité η de la détection. X_N est un bruit de variance V_N, appelé «bruit ramené à l'entrée», décorrélé de X_A. Ce bruit regroupe toutes les sources de bruits que nous allons rencontrer dans ce chapitre, et que nous avions introduites de façon théorique dans le chapitre 2. Alors,

$$\langle X_A X_B \rangle = g V_A \qquad \text{et} \qquad V_B = G(V_A + V_N) \tag{7.4}$$

$$\text{d'où} \quad \rho^2 = \frac{G V_A^2}{V_A G (V_A + V_N)} = \frac{V_A}{V_A + V_N} \tag{7.5}$$

$$\text{puis} \quad \frac{1}{1-\rho^2} - 1 = \frac{V_A}{V_N} \equiv SNR \tag{7.6}$$

□

L'information mutuelle I_{AB} se déduit simplement du coefficient de corrélation, à l'aide du théorème de Shannon :

$$I_{AB} = \frac{1}{2}\log_2(1 + SNR) = -\frac{1}{2}\log_2\left(1-\rho^2\right). \tag{7.7}$$

Maintenant que nous avons exprimé l'information mutuelle I_{AB}, nous allons détailler l'ensemble des bruits physiques que viennent dégrader le rapport signal à bruit de la transmission quantique.

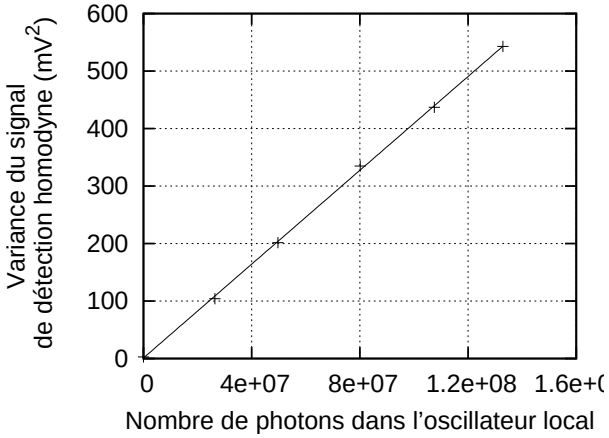


FIG. 7.1: Mesure homodyne d'un mode vide, obtenu en bloquant le signal d'entrée. La courbe représente la variance du signal observé en sortie de la détection homodyne en fonction de la puissance de l'oscillateur local, en nombre de photons par impulsion. On ajuste les données expérimentales par une droite dont l'ordonnée à l'origine est la variance du bruit électronique ($V_{\text{élec}} = 2,9$ mV^2).

7.2 Bruit à la sortie

La deuxième grandeur permettant de quantifier la qualité de la transmission quantique est le bruit GV_N observé par Bob à l'issue de la transmission quantique. Ce bruit, que nous nommons «bruit à la sortie», est une grandeur physique dont l'interprétation est immédiate : il représente la variance de l'écart entre les données reçues par Bob et les données envoyées par Alice (corrigées d'un facteur g). Expérimentalement, on calcule ce bruit à partir du coefficient de corrélation rencontré dans la section précédente, et de la variance V_B des données de Bob :

$$V_B = G(V_A + V_N) \tag{7.8}$$

$$\Rightarrow \quad V_B = GV_N(1 + SNR) \tag{7.9}$$

$$\text{d'où} \quad GV_N = \frac{V_B}{1 + SNR} = V_B\left(1 - \rho^2\right) \tag{7.10}$$

Notons que cette expression ne fait intervenir ni la transmission du canal quantique ni la variance des données d'Alice. C'est une grandeur intrinsèque au signal reçu par Bob. En revanche, le bruit à la sortie n'est pas indépendant de la normalisation des données de Bob : pour exprimer la variance V_B en unités de bruit de photon, nous devons d'abord mesurer la variance du bruit de photon.

7.3 Bruit de photon

La variance du bruit de photon, nécessaire pour calibrer les mesures de Bob, s'obtient en étalonnant la détection homodyne. Pour cela, nous coupons la voie signal de la détection homodyne (c'est-à-dire que nous laissons uniquement entrer le mode vide et ses fluctuations par la voie signal), et nous faisons varier la puissance de l'oscillateur local. La figure 7.1 montre le résultat de cette mesure. Le fait que les données s'ajustent par une droite est

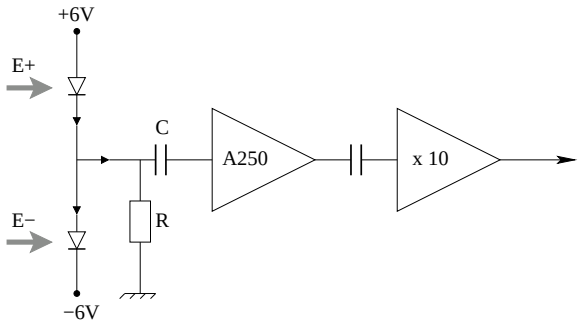


FIG. 7.2: Schéma électronique de la détection homodyne. Un amplificateur de charge amplifie la mesure de la charge accumulée par un condensateur. Cette charge est l'intégrale de la différence des photocourants issus des photodiodes. Une résistance permet de décharger le condensateur ; un deuxième étage d'amplification permet de dériver le signal délivré par l'amplificateur de charge, et applique ensuite un gain de 10.

la signature du bruit de photon. En effet, dans le cas de la mesure d'un mode vide, la variance du signal de sortie de la détection homodyne est proportionnelle à (voir équation 6.4)

$$4I_{\mathrm{OL}}\left(N_0 + 4\epsilon^2\langle\delta E_{\mathrm{OL}}^2\rangle\right), \tag{7.11}$$

avec $\epsilon = 10^{-4}$ est le défaut d'équilibrage de la détection homodyne, et δE_{OL} est un bruit classique sur le faisceau oscillateur local, dont la variance est proportionnelle à I_{OL}. Nous avons vérifié la linéarité de la variance du signal de détection homodyne en fonction de la puissance d'oscillateur local jusqu'à 10^9 photons par impulsion. Pour l'ensemble du travail décrit dans cette thèse[1], nous avons opté pour une puissance d'oscillateur local autour de 6 10^7 photons par impulsion (soit une variance du bruit de photon de 250 mV^2, compte tenu de la calibration tracée figure 7.1). Si nous utilisons le calibre de 500 mV de notre carte d'acquisition, cette intensité de l'oscillateur local nous permet d'observer des signaux d'intensité allant jusqu'à $\frac{(500\ \mathrm{mV})^2}{250\ \mathrm{mV}^2} = 1000$ photons, soit des modulations de variance $\frac{1000\ \mathrm{photons}}{16} \simeq 60$ photons (voir section 6.4).

Maintenant que nous savons que notre détection homodyne est limitée par le bruit de photon, nous pouvons également mesurer la variance du bruit de photon en présence de signal. Pour cela, nous faisons varier l'intensité du signal envoyé dans la détection homodyne. Nous obtenons une droite (que nous verrons à la figure 7.6) dont l'ordonnée à l'origine nous indique le niveau de bruit de photon. La pente de cette droite provient de l'ensemble des bruits autres que le bruit de photon introduits par la voie signal de la détection homodyne. Nous allons maintenant détailler ces bruits.

7.4 Bruit électronique

L'amplificateur en entrée du circuit de détection homodyne génère un bruit électronique. Dans les précédentes versions du dispositif de cryptographie quantique, le courant issu des photodiodes était converti en tension à l'aide d'une résistance de 4,7 kΩ, puis amplifié avec un amplificateur de tension [6]. Dans cette configuration, le bruit de Schottky associé à la

[1]Le système de multiplexage décrit au chapitre 9 impose des contraintes supplémentaires sur la puissance d'oscillateur local disponible. Nous détaillerons les valeurs choisies dans ce chapitre

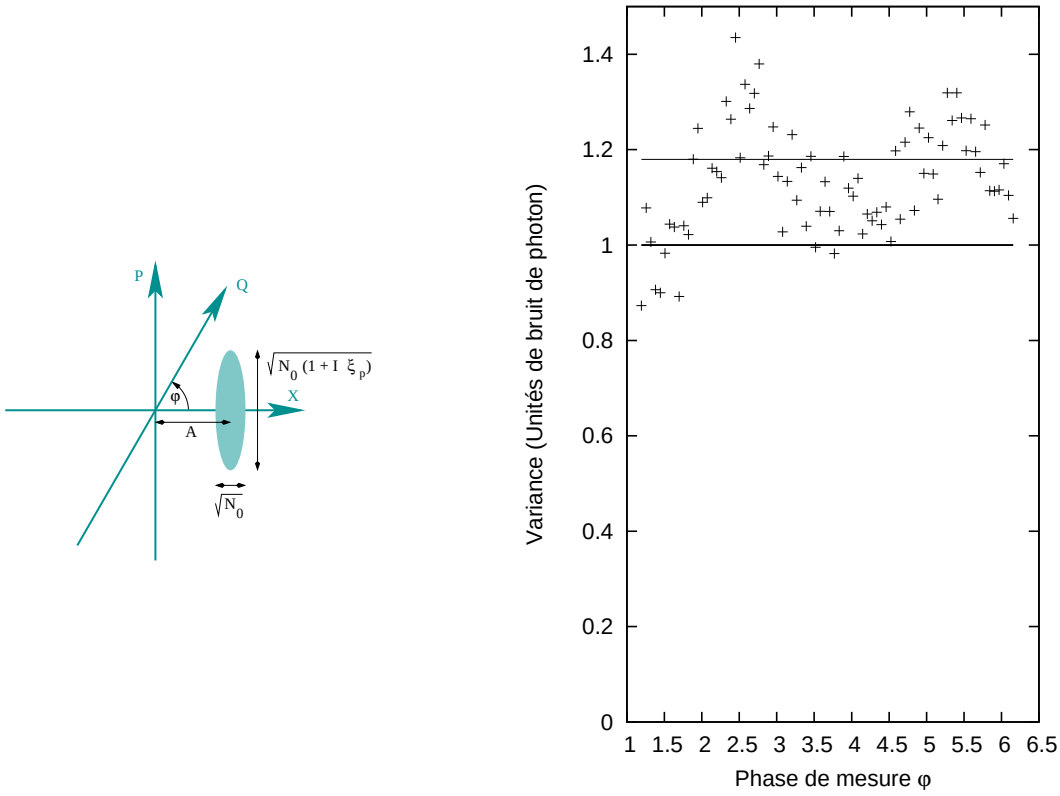


FIG. 7.3: À gauche : représentation du bruit de phase dans l'espace des phases. Selon la quadrature X (choisie arbitrairement), notre signal est limité au bruit de photon de variance N_0. Selon la quadrature P, le bruit de phase se manifeste par une variance supérieure à N_0. À droite : tomographie de l'état représenté. Nous mesurons la variance de l'état selon différentes quadratures Q. La variance mesurée est celle du bruit de photon lorsque l'on mesure X, auquel est ajouté le bruit de phase lorsque l'on mesure P. La variance de chacune des 100 quadratures est calculée à partir de 500 mesures. Le signal acquis présente une intensité de $480N_0$ et un bruit de phase $\xi_p = 3,8\ 10^{-4}N_0$ par photon.

résistance était comparable à la variance du bruit de photon. Pour remédier à ce problème, nous convertissons maintenant le courant issu des photodiodes en tension à l'aide d'un condensateur. Un amplificateur de charge (Amptek A250) amplifie la tension aux bornes du condensateur, puis un second étage d'amplification dérive la tension amplifiée [51] (figure 7.2).

Le bruit électronique a un niveau constant, indépendant du signal. Nous devons donc retrancher sa variance à celle de notre mesure du bruit de photon présentée dans la section précédente. Nous mesurons la variance du bruit électronique à la sortie de la détection homodyne à 2,9 mV^2 (voir ordonnée à l'origine de la droite figure 7.1). L'intensité d'oscillateur local choisie (6 10^7 photons par impulsion) place finalement la variance du bruit électronique à $V_{\text{élec}} = 0,01N_0$.

7.5 Bruit de phase

Les fluctuations de phase dans la diode laser introduisent un bruit de phase. Ce bruit de phase se représente dans le plan complexe par une zone de bruit elliptique allongé selon la quadrature de phase (perpendiculaire à la quadrature d'amplitude), en lieu et place de

la zone de bruit ronde qui représente usuellement le bruit de photon (figure 7.3). On mesure le bruit de phase en faisant varier la quadrature de mesure Q, afin de placer successivement la mesure selon le grand axe de l'ellipse puis le petit axe de l'ellipse. La différence des variances des bruits entre les deux axes donne la variance totale du bruit de phase pour l'état observé, exprimée en unités de bruit de photon. Le bruit de phase est un bruit classique : sa variance dépend linéairement de l'intensité lumineuse du signal. On note alors cette variance $\xi_p I_s$ où I_s est l'intensité du faisceau signal, et ξ_p est le bruit de phase par photon exprimé en unités de bruit de photon par photon. Il dépend de la source laser choisie ainsi que des conditions expérimentales (température, vibrations mécaniques au niveau de la source, retour de lumière dans la diode, différence de marche de l'interféromètre, temps de stabilisation...). Au cours de nos années de thèse, nous avons utilisé deux sources laser, l'une (de marque Princeton Lightwave) utilisée en régime continu, l'autre (Alcatel A 1950 LMI, de puissance nominale 15 mW) en régime impulsionnel. Dans nos conditions expérimentales, ces deux diodes DFB ont un bruit de phase typique de $\xi_p = 2,5\ 10^{-4} N_0$ par photon. À titre de comparaison, un laser TUNIX (ANDO AQ8201) présente un bruit de phase un ordre de grandeur supérieur.

Le bruit de phase est un excès de bruit considéré comme généré par l'espion. En effet, la variation de sa variance au cours de l'échange quantique rend sa calibration délicate. Le retrancher des sources de bruit accessibles à l'espion serait une faille potentielle pour la sécurité de notre transmission.

Pour un signal d'intensité I_S fixée, nous venons de voir que le bruit de phase total mesuré s'exprimait $\xi_p I_S$, selon notre définition du bruit de phase par photon ξ_p. Pour un signal modulé dans le plan complexe avec une modulation gaussienne de variance V_S, on montre que le bruit de phase total $\overline{\xi}$ est le produit du bruit de phase par photon et de la variance de modulation du signal V_S :

$$\overline{\xi} = \xi_p V_S \tag{7.12}$$

Démonstration. Pour montrer ce résultat, on intègre sur la modulation gaussienne la projection du bruit de phase sur la quadrature de mesure.

$$\begin{aligned}
(1+\overline{\xi})N_0 &= \int_0^\infty dr \int_0^{2\pi} r d\theta \frac{1}{2\pi V_S N_0} e^{-\frac{r^2}{2V_S N_0}} \left(N_0 + \xi_p r^2 \sin^2\theta\right) && (7.13)\\
&= N_0 \left[1 + \frac{\pi \xi_p}{N_0} \int_0^\infty dr \frac{r^3}{2\pi V_S N_0} e^{-\frac{r^2}{2V_S N_0}}\right] && (7.14)\\
&= N_0 \left[1 + \frac{\xi_p V_S}{2} \int_0^\infty dy y^3 e^{-\frac{y^2}{2}}\right] && (7.15)\\
&= N_0 \left(1 + \xi_p V_S\right) && (7.16)
\end{aligned}$$

□

Nous mesurons continuellement les bruit de photon et bruit de phase à l'aide d'impulsions test de phase convenue et d'intensité maximale V_M autorisée par notre modulateur d'amplitude. Nous utilisons trois groupes d'impulsions test distincts de phases respectives 0, $\frac{2\pi}{3}$ et $\frac{4\pi}{3}$ (le détail de l'organisation des impulsions test sera décrit au chapitre 10). Le calcul des variances σ_i^2 de chacun permet d'obtenir trois points de la courbe sinusoïdale représentée figure 7.3. À partir ce ces points, il est possible de déduire les niveaux de bruit de photon N_0 (niveau bas de la sinusoïde) et de phase $(1+\xi_p V_M)N_0$ (niveau haut), à l'aide de

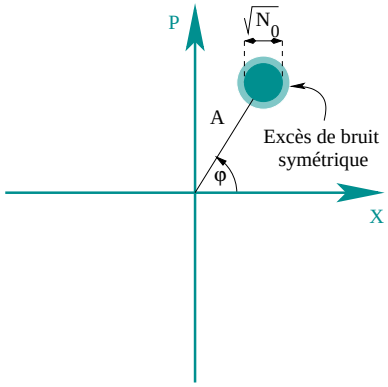


FIG. 7.4: Le bruit symétrique est un excès de bruit indépendant de la phase. Le bruit d'émission spontanée dans un amplificateur optique est un exemple d'un tel bruit. Tout comme le rapport signal à bruit est dégradé par une atténuation, les relations de Heisenberg imposent un excès de bruit minimal sur tout phénomène d'amplification.

quelque ruse trigonométrique, que nous ne détaillerons pas :

$$N_0 = \Sigma - \Delta \qquad \text{et} \qquad (1 + \xi_p V_M) N_0 = \Sigma + \Delta \tag{7.17}$$

$$\text{avec} \qquad \Sigma = \frac{1}{3} \sum_i \sigma_i^2 \qquad \text{et} \qquad \Delta = \sqrt{\frac{2}{3} \sum_i \left(\sigma_i^2 - \Sigma\right)^2} \tag{7.18}$$

Les niveaux de bruit de photon et de bruit de phase représentés figure 7.3 ont été déterminé à l'aide de ces formules. Dans une situation de transmission quantique réelle, l'espion peut manipuler les impulsions test. C'est pourquoi le niveau de bruit de photon mesuré à partir des impulsions test ne saurait se substituer à une mesure directe, issue de l'intensité de l'oscillateur local. Quant au bruit de phase, nous avons remarqué en introduction que sa connaissance était inutile à la détermination du taux secret, seul paramètre pertinent en situation de distribution de clé.

7.6 Bruit d'amplitude

Le bruit d'amplitude est un excès de bruit sur la quadrature d'amplitude perpendiculaire à la quadrature de phase. Tout comme le bruit de phase, le bruit d'amplitude est un bruit classique : sa variance est proportionnelle à l'intensité du signal ; on la note $\xi_a I_S$. Alors que le bruit de phase s'observe à l'aide d'un dispositif interférométrique, le bruit d'amplitude peut être mesuré par une observation directe d'un mode du champ électromagnétique à l'aide d'une photodiode suffisamment amplifiée. Toutefois, nous pouvons également le mesurer à partir de nos données homodynes. Pour cela, nous modulons linéairement l'intensité du faisceau signal, tout en laissant dériver la phase de mesure. Ce faisant, nous obtenons une droite de pente ξ_p quand nous mesurons la quadrature de phase, puis une droite de pente ξ_a quand nous mesurons la quadrature d'amplitude.

Nous avons observé un excès de bruit d'amplitude lors de l'utilisation d'une source pulsée avec un interféromètre assez déséquilibré (déséquilibre supérieur à 10 cm). En équilibrant la longueur des voies signal et oscillateur local, nous arrivons à rendre l'excès de bruit d'amplitude négligeable devant les autres bruits, notamment devant le bruit de phase.

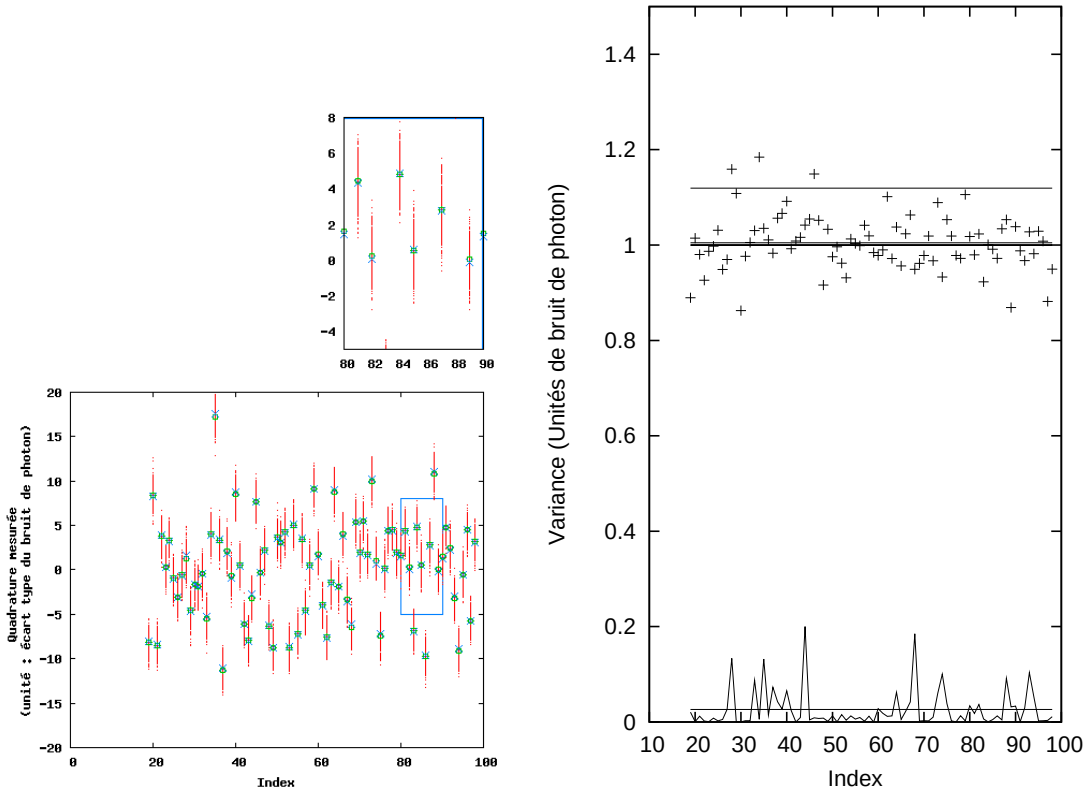


FIG. 7.5: Erreurs de modulation. Pour mesurer les erreurs de modulation, nous envoyons entre Alice et Bob une séquence de 80 impulsions numérotées de 19 à 98 sur les abscisses des deux graphiques, d'amplitude et phase différentes et répétée 500 fois à l'identique. Les mesures de Bob sont représentées sur la figure de gauche par des points, qui, s'ils sont trop rapprochés, forment des lignes. Pour chacune des impulsions de la séquence, nous pouvons donc moyenner le bruit de photon (la moyenne est représentée par un rond) et comparer cette moyenne à la quadrature qu'Alice est censée avoir envoyée (représentée par le symbole $\times$). La figure de droite trace pour chaque impulsion la variance des bruits non systématiques (bruit de photon et bruit de phase, représentés par le symbole $+$) et la compare à l'erreur de modulation (trait continu). La dispersion des variances des bruits non systématiques provient des fluctuations statistiques lors du moyennage sur 500 impulsions. Cette séquence de données présente une modulation de variance $30N_0$ à la réception, un bruit de phase $\xi_p = 2,5\ 10^{-4}N_0$ par photon et des erreurs de modulation de $8,4\ 10^{-4}N_0$ par photon (ligne horizontale en bas de la figure de droite).

Il existe d'autres sources de bruit d'amplitude. Par exemple, un amplificateur optique génère un bruit symétrique, c'est-à-dire un bruit dont la variance ne dépend pas de la quadrature de mesure. Ce bruit est associé au phénomène d'émission spontanée dans le milieu amplificateur. Un tel bruit est représenté sur la figure 7.4. On détecte ce bruit de la même manière que l'on détecte le bruit d'amplitude : la variance du bruit sur le signal varie linéairement avec l'intensité du signal. Pour un bruit symétrique, la pente de cette droite ne dépend pas de la quadrature de mesure.

7.7 Erreurs de modulation

Les modulations d'amplitude et de phase choisies par Alice et Bob sont générées informatiquement, puis appliquées sur le faisceau optique à l'aide de modulateurs électro-optiques pilotés par une carte d'acquisition. Or, la précision de la modulation dépend de multiples calibrations : les valeurs des V_π de chacun des trois modulateurs, ainsi que le facteur de corrélation phase/amplitude (défini au chapitre 6) du modulateur d'amplitude doivent êtres mesurées pour permettre la conversion informatique entre modulation et tension ; la tension de biais appliquée au modulateur d'amplitude doit être ajustée. Nous avons donc au total cinq paramètres ajustables. Ces paramètres dérivent, principalement avec la température. Les tensions de biais restent stables sur l'échelle de quelques dizaines de minutes, les tensions V_π subissent des variations saisonnières.

Les défauts de calibration des paramètres des modulateurs électro-optiques occasionnent un bruit, dit «technique», sur la transmission des données entre Alice et Bob. Comme le bruit de phase, le bruit technique est un bruit classique dont la variance croît linéairement avec l'intensité du signal. Là encore, ce bruit est difficilement quantifiable ; il est donc attribué à l'espion. Pour régler les paramètres des modulateurs, nous mesurons ce bruit en comparant les données reçues par Bob et les données envoyées par Alice. Pour ce faire, nous envoyons successivement 500 fois la même séquence de données. Le moyennage des mesures de quadrature effectuées par Bob permet de s'affranchir des sources d'erreurs non systématiques. Les erreurs systématiques, principalement dues aux défauts de réglage des paramètres, sont quantifiées en comparant la moyenne de chaque élément de la séquence de modulation avec la valeur envoyée par Alice. Une telle séquence est représentée figure 7.5.

L'affichage du bruit technique nous permet de régler les paramètres des modulateurs en minimisant les erreurs de modulation. Compte tenu du nombre de paramètres indépendants, ce réglage est souvent délicat. Les résultats présentés dans ce manuscrit présentent un bruit technique typique de $0,001N_0$ par photon, issu de l'ensemble des modulateurs d'amplitude et de phase chez Alice et du modulateur de phase chez Bob. L'analyse du prélèvement d'une partie du signal envoyé par Alice nous a récemment permis d'améliorer l'aisance du réglage (voir section 10.7 pour plus de détails)

7.8 Bruit ramené à l'entrée et excès de bruit

Le bruit à la sortie GV_N, que nous avons exprimé directement à partir de statistiques sur les données expérimentales en section 7.2, contient tous les bruits que nous avons détaillés jusqu'à présent, c'est-à-dire bruit de photon, bruit électronique, bruit de phase, bruit d'amplitude, erreurs de modulation, et éventuellement d'autres sources de bruit minoritaires que nous n'aurions pas identifiées (par exemple, du bruit provenant d'un espion sur le canal quantique). La figure 7.6 détaille la composition du bruit à la sortie.

Pour en déduire le bruit ramené à l'entrée V_N, que nous avons introduit au chapitre 2, nous devons déterminer la transmission totale G entre Alice et Bob. Pour cela, nous utilisons la

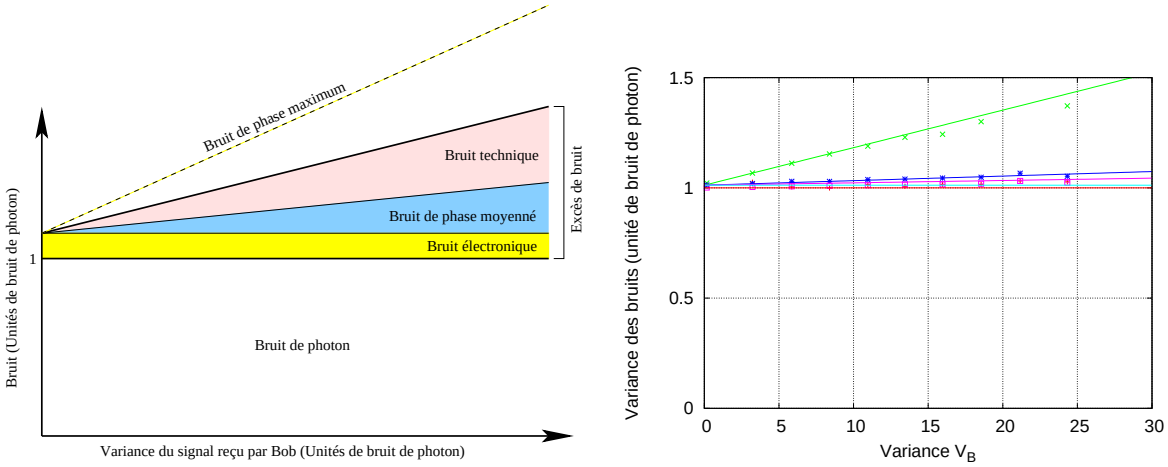


FIG. 7.6: À gauche : schéma représentant l'ensemble des bruits observés par Bob lors d'une mesure homodyne d'un signal envoyé par Alice. L'excès de bruit est la somme du bruit électronique, du bruit de phase moyenné sur la modulation gaussienne et du bruit technique dû aux erreurs de modulations chez Alice et Bob. À droite : mesures expérimentales des bruits pour différentes valeurs de V_B. Ces mesures sont présentées sous la même forme que le schéma associé. On mesure un bruit de phase $\xi_p = 1.0\ 10^{-3} N_0$ par photon, $V_{\text{élec}} = 1,2 N_0$ et un excès de bruit total $\xi = 2,1\ 10^{-3} N_0$ par photon.

variance de modulation d'Alice, le dernier moment d'ordre deux qu'il nous restait à introduire :

$$V_B = G(V_A + V_N) = GV_A\left(1 + \frac{1}{SNR}\right) = \frac{GV_A}{\rho^2} \tag{7.19}$$

$$\text{d'où} \quad G = \rho^2 \frac{V_B}{V_A} \tag{7.20}$$

En situation cryptographique, la détermination de V_A peut se faire en mesurant l'intensité lumineuse sur la voie signal, en sortie d'Alice. Une calibration est nécessaire pour exprimer V_A en unités de bruit de photon.

Pour caractériser notre dispositif, nous pouvons fixer arbitrairement la variance V_A (ce qui revient à définir arbitrairement sur notre montage le point appelé «sortie d'Alice»). Une fois V_A défini, nous faisons varier la transmission T du canal avec un modulateur électro-optique[2]. Cette définition de V_A nous permet de nous affranchir des pertes nominales du modulateur et ainsi d'explorer toute la gamme des transmissions entre $T = 0$ et $T = 1$.

Le bruit *ajouté* ramené à l'entrée χ est le bruit ramené à l'entrée duquel nous soustrayons l'unité de bruit de photon initialement présente sur le signal envoyé par Alice. Il s'exprime donc

$$\chi = V_N - 1 \tag{7.21}$$

[2]Notons que les résultats présentés dans ce chapitre ont été obtenus avant réalisation du multiplexage temporel. C'est pourquoi notre modulateur simulant la transmission du canal a été placé uniquement sur la voie signal. Un véritable canal de transmission a également pour effet de faire décroître la puissance d'oscillateur local, et par là même de faire décroître la variance du bruit de photon telle qu'observée en sortie de la détection homodyne. Cette variation a pour conséquence d'augmenter la variance du bruit électronique relativement au bruit de photon. Cet aspect sera considéré dans le chapitre 9, consacré au multiplexage temporel.

Nature	valeur	contribution à ξ pour $V_A = 12$
bruit électronique	$V_{\text{élec}} = 10^{-2}N_0$, fixe à la sortie	$10^{-2}/GN_0$
bruit de phase	$2,5\ 10^{-4}N_0$ par photon	$3\ 10^{-3}N_0$
bruit d'amplitude	négligeable	négligeable
bruit technique	$10^{-3}N_0$ par photon	$1,2\ 10^{-2}N_0$
	Total	$(1.5 + 1/G) \times 10^{-2}N_0$

TAB. 7.1: Tableau récapitulatif des différentes sources d'excès de bruit.

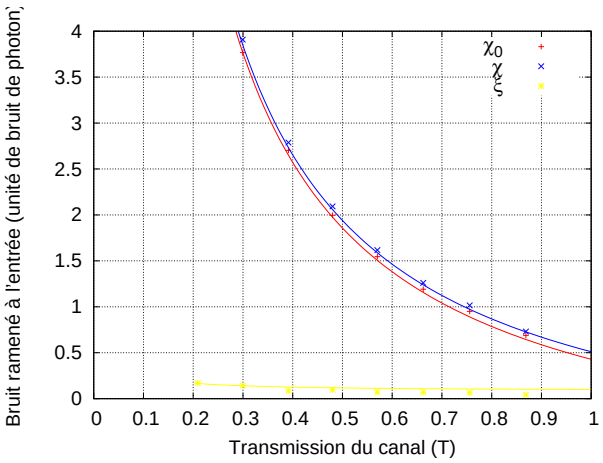


FIG. 7.7: Bruits ramenés à l'entrée correspondant aux bruits mesurés figure 7.6, pour $V_A = 40N_0$ et $\eta = 0,7$.

L'excès de bruit est la somme de l'ensemble des bruits au-delà du niveau du bruit de photon. Il comprend toutes les sources de bruit qui composent le bruit total GV_N, sauf le bruit de photon. Ramené à l'entrée, il s'exprime donc :

$$\xi = \frac{GV_N - 1}{G} = \chi - \chi_0, \quad \text{avec} \quad \chi_0 = \frac{1}{G} - 1 \tag{7.22}$$

Cet excès de bruit ramené à l'entrée quantifie la qualité de notre dispositif. Comme les fibres optiques que nous utilisons n'introduisent pas d'excès de bruit, l'excès de bruit mesuré provient des imperfections expérimentales. Le tableau 7.1 résume les différentes sources de bruit que nous avons vues dans ce chapitre, ainsi que leurs contributions respectives à l'excès de bruit total. L'excès de bruit total est représenté par la figure 7.7.

7.9 Information secrète

Nous avons finalement en notre possession tous les paramètres du canal gaussien reliant Alice et Bob : G et χ, ou de façon équivalente G et ξ. Avec ces deux paramètres, nous pouvons calculer les informations mutuelles I_{AB} et I_{BE} selon les

expressions dérivées au chapitre 2 :

$$I_{AB} = \frac{1}{2}\log_2\left(\frac{V+\chi}{1+\chi}\right) \tag{7.23}$$

$$I_{BE} = \frac{1}{2}\log_2\left[G^2\left(\chi+V\right)\left(\chi+\frac{1}{V}\right)\right] \tag{7.24}$$

avec $V = V_A + 1$. Bien entendu, cette dernière expression de I_{AB} est équivalente à celle que nous avions écrite au début de ce chapitre.

Le mode réaliste permet de différencier les bruits dont l'espion peut profiter, des bruits expérimentaux contrôlés. Pour cela nous distinguons le bruit ajouté par le canal, ramené à l'entrée du canal :

$$\chi_{\text{canal}} = \frac{1}{T} - 1 + \xi \tag{7.25}$$

du bruit ajouté par le détecteur imparfait de Bob, ramené à l'entrée du détecteur :

$$\chi_{\text{hom}} = \frac{1}{\eta} - 1 + \frac{V_{\text{élec}}}{\eta}. \tag{7.26}$$

Ce bruit χ_{hom} est la somme du bruit dû aux pertes de la détection homodyne $\frac{1}{\eta} - 1$ et du bruit électronique $\frac{V_{\text{élec}}}{\eta}$. Le bruit total de la transmission entre Alice et Bob est la combinaison des bruits χ_{canal} et χ_{hom}, ramenés à l'entrée du canal :

$$\chi_{\text{tot}} = \chi_{\text{canal}} + \frac{\chi_{\text{hom}}}{T} = \frac{1}{G} - 1 + \xi + \frac{V_{\text{élec}}}{G}. \tag{7.27}$$

Le mode «réaliste» [4] exclut que l'espion puisse manipuler le bruit du détecteur χ_{hom}. On obtient alors les expressions :

$$I_{AB} = \frac{1}{2}\log_2\left(\frac{V+\chi_{\text{tot}}}{1+\chi_{\text{tot}}}\right) \tag{7.28}$$

$$I_{BE} = \frac{1}{2}\log_2\left[\frac{T^2\left(\chi_{\text{tot}}+V\right)\left(\chi_{\text{canal}}+\frac{1}{V}\right)}{1+T\chi_{\text{hom}}\left(\chi_{\text{canal}}+\frac{1}{V}\right)}\right], \tag{7.29}$$

La détermination expérimentale de ces bruits est la suivante : les moments d'ordre deux de la distribution des données fournissent le bruit total χ_{tot} et la transmission G. La caractérisation de la détection homodyne donne η et $V_{\text{élec}}$ d'où nous calculons χ_{hom}. Enfin, nous déduisons $\chi_{\text{canal}} = \chi_{\text{tot}} - \frac{\chi_{\text{hom}}}{T}$.

À titre d'exemple, une détection homodyne d'efficacité $\eta = 0,7$ et de bruit électronique $V_{\text{élec}} = 0,01N_0$ ajoute un bruit inaccessible à l'espion $\chi_{\text{hom}} = 0,44$.

L'expression de l'entropie de Holevo χ_{BE}, définie au chapitre 4, que l'on doit considérer dans un modèle de securité incluant les attaques collectives, s'étend elle aussi au mode réaliste :

$$\chi_{BE} = G\left(\frac{\lambda_1-1}{2}\right) + G\left(\frac{\lambda_2-1}{2}\right) - G\left(\frac{\lambda_3-1}{2}\right) - G\left(\frac{\lambda_4-1}{2}\right), \tag{7.30}$$

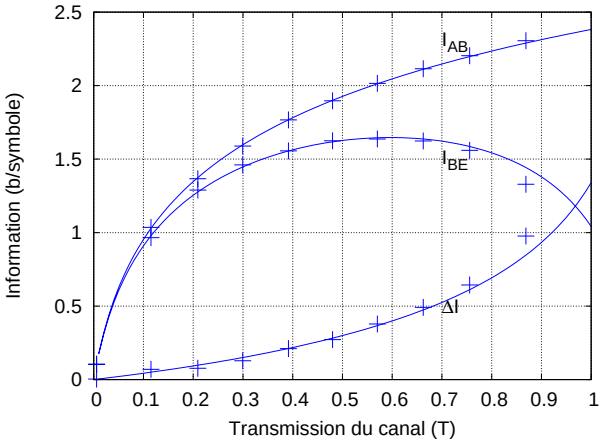


FIG. 7.8: Informations mutuelles I_{AB} et I_{BE} et information secrète ΔI correspondant aux bruits ramenés à l'entrée mesurés figure 7.7.

avec

$$G(x) = (x+1)\log_2(x+1) - x\log_2(x) \tag{7.31}$$

$$\lambda_{1,2}^2 = \frac{C \pm \sqrt{C^2 - 4D}}{2} \tag{7.32}$$

$$\lambda_{3,4}^2 = \frac{A \pm \sqrt{A^2 - 4B}}{2} \tag{7.33}$$

$$C = V^2(1-2T) + 2T + T^2(V + \chi_{\text{canal}})^2 \tag{7.34}$$

$$D = (T(V\chi_{\text{canal}} + 1))^2 \tag{7.35}$$

$$A = \frac{V\sqrt{D} + T(V + \chi_{\text{canal}}) + C\chi_{\text{hom}}}{T(V + \chi_{\text{tot}})} \tag{7.36}$$

$$B = \sqrt{D}\frac{V + \sqrt{D}\chi_{\text{hom}}}{T(V + \chi_{\text{tot}})}. \tag{7.37}$$

La figure 7.8 représente l'information secrète dérivée de nos mesures de bruit pour différentes transmissions du canal quantique. Cette courbe simule donc une attaque de type «lame séparatrice», dans laquelle Ève remplace le canal quantique reliant Alice à Bob par un canal parfait, et prélève une partie du signal correspondant à la transmission du canal réel. Le chapitre suivant est consacré à l'étude d'autres attaques plus réalistes et plus générales.

Chapitre 8

Réalisation expérimentale d'une attaque non gaussienne

L'attaque de type «lame séparatrice» est la plus immédiate contre un système de distribution quantique de clé. Elle consiste à prélever une partie du signal qui sera accordé à l'espion. La transmission de cette lame séparatrice définit l'atténuation du canal, que l'espion aura pris soin de remplacer par un canal parfait. Pour réaliser cette attaque, l'espion doit également disposer d'une mémoire quantique lui permettant de conserver l'état quantique prélevé sur le canal jusqu'au moment où Bob révèle son choix de mesure.

En pratique, une attaque de type «lame séparatrice» est simulée en introduisant des pertes entre Alice et Bob, sans réellement utiliser ou analyser le signal prélevé. La réalisation de ce type d'attaque a été évoquée à la fin du chapitre 7.

Dans ce chapitre, nous proposons la réalisation expérimentale d'une attaque à la fois plus réaliste et plus complète : une attaque de type «interception-réémission», qui permet à l'espion d'introduire un excès de bruit et qui peut être réalisée sans remplacer le canal quantique par un canal parfait. Comme cette attaque n'est pas encore la plus générale permise par le modèle du canal gaussien, nous avons réalisé une attaque «interception-réémission partielle», mélangeant aléatoirement et dans une proportion variable une attaque de type «interception-réémission» et une attaque de type «lame séparatrice».

Ce chapitre a fait l'objet de la publication [52].

8.1 Attaque «interception-réémission»

Une attaque «interception-réémission», représentée figure 8.1, se déroule en deux étapes : l'espion fait une mesure optimale de l'état quantique, puis ré-émet un état quantique fonction de sa mesure. Pour mesurer l'état quantique, Ève sépare en deux parties, à l'aide d'une lame séparatrice, le signal envoyé par Alice. Sur chacune des deux voies de sortie de sa séparatrice, Ève mesure respectivement les quadratures $\hat{X}$ et $\hat{P}$[1] du signal. Ensuite, Ève génère un nouvel état cohérent en utilisant les résultats classiques de ses mesures (X_E, P_E) qu'elle peut multiplier par un gain classique g_c simulant les pertes du canal. Ce nouvel état cohérent, centré sur $(g_c X_E, g_c P_E)$ est envoyé à Bob par un canal quantique de transmission

[1] L'attaque interception-réémission faisant intervenir des mesures, c'est-à-dire des variables classiques, nous prendrons soin, dans ce chapitre, de noter les variables quantiques avec un chapeau $(\hat{\cdot})$, et les variables classiques sans chapeau.

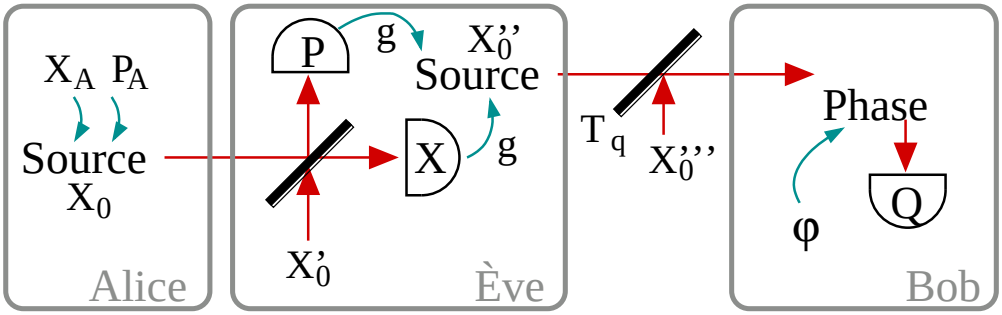


FIG. 8.1: Schéma de principe d'une attaque interception-réémission. Alice et Bob font un échange de clé usuel : Alice module aléatoirement (RNG) l'amplitude et la phase d'impulsions cohérentes, et Bob mesure une quadrature aléatoire Q du signal reçu, sélectionnée par un modulateur de phase. Entre Alice et Bob, Ève intercepte les impulsions, les sépare en deux voies et fait une mesure des quadratures X et P respectivement sur chacune des voies. Ensuite, Ève régénère le signal en envoyant à Bob des états cohérents centrés sur le résultat de ses mesures de quadrature.

T_q. Comme pour une transmission usuelle, Bob mesure enfin une quadrature aléatoire de chaque état cohérent issu du canal quantique.

Les quadratures $(\hat{X}, \hat{P})$ des états quantiques envoyés par Alice s'expriment

$$\hat{X} = X_A + \hat{X}_0 \tag{8.1}$$
$$\hat{P} = P_A + \hat{P}_0 \tag{8.2}$$

où (X_A, P_A) est la valeur classique de la modulation d'Alice de variance $V_A N_0$, et $(\hat{X}_0, \hat{P}_0)$ sont les observables quantiques du bruit de photon associé au signal, de variance N_0. La variance totale du signal en sortie de chez Alice est alors $VN_0 = (V_A + 1)N_0$. Les quadratures $(\hat{X}_E, \hat{P}_E)$ mesurées par Ève à la sortie de la lame séparatrice s'expriment en fonction de (X_A, P_A) :

$$\hat{X}_E = \frac{1}{\sqrt{2}}(X_A + \hat{X}_0 - \hat{X}_0') \tag{8.3}$$
$$\hat{P}_E = \frac{1}{\sqrt{2}}(P_A + \hat{P}_0 + \hat{P}_0'), \tag{8.4}$$

où $(\hat{X}_0', \hat{P}_0')$ sont les quadratures du mode vide introduit par la lame séparatrice. Les résultats des mesures d'Ève (X_E, P_E) ont donc une variance $V_E N_0 = \frac{V+1}{2} N_0$, et un bruit $V_{E|A} = N_0$. L'état reçu par Bob est finalement de la forme :

$$\hat{X}_B = t_q(g_c X_E + \hat{X}_0'') + \sqrt{1 - t_q^2}\hat{X}_0''' \tag{8.5}$$
$$\hat{P}_B = t_q(g_c P_E + \hat{P}_0'') + \sqrt{1 - t_q^2}\hat{P}_0'''. \tag{8.6}$$

Le bruit X_0'' de variance N_0 est le bruit de photon généré par la source d'Ève, et le bruit X_0''' provient des pertes associées au canal de communication quantique reliant Ève et Bob. La variance des quadratures reçues par Bob est donc $V_B N_0 = (G_c T_q V_E + 1)N_0$ avec $G_c = g_c^2$, et le bruit $V_{B|E} = N_0$. Comme seul le produit $G_c T_q$ intervient dans l'expression de V_B, il est équivalent pour Ève d'introduire un gain classique G_c entre l'interception et la réémission puis une transmission sans perte jusqu'à Bob, et/ou de transmettre les états quantiques ré-émis par

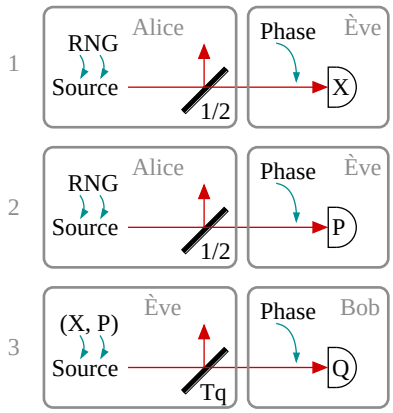


FIG. 8.2: Schéma de mise en œuvre de l'attaque interception-réémission. Afin d'éviter une réplication inutile de notre matériel, nous séparons l'attaque en trois parties : 1 – Ève mesure X. 2 – Ève mesure P sur une deuxième émission identique par Alice. 3 – Ève ré-émet un signal centré sur (X, P) sur un canal de transmission T_q. Bob mesure une quadrature aléatoire Q de ce signal.

un canal quantique de transmission T_q, tant que $G_c T_q$ est constant. Dans les deux cas, Alice et Bob constateront une transmission T sur la ligne qui les relie. Cette propriété rend l'attaque interception réaliste, car Ève ne doit pas remplacer le canal quantique par un canal sans perte. De plus, dans cette attaque, Ève mesure les deux quadratures X et P. Elle n'a donc pas besoin d'attendre la révélation des quadratures choisies par Bob pour réaliser cette mesure : l'attaque interception-réémission ne nécessite pas de mémoire quantique.

Une attaque «interception-réémission» ne permet plus le transfert d'intrication sur le canal quantique. L'équation 8.6 permet d'exprimer la quadrature mesurée par Bob (par exemple $\hat{X}_B$) en fonction de la quadrature envoyée par Alice :

$$\hat{X}_B = \frac{t_q g_c}{\sqrt{2}} \left(X_A + X_0 + X_0' + \frac{\sqrt{2}}{g_c} \hat{X}_0'' + \frac{\sqrt{2}}{g_c t_q} \sqrt{1 - t_q^2} X_0''' \right). \tag{8.7}$$

On en déduit la transmission totale $T = \frac{G_c T_q}{2}$ entre Alice et Bob, et le bruit ajouté ramené à l'entrée $\chi = \chi_0 + 2$ avec $\chi_0 = \frac{1-T}{T}$. Ainsi, l'attaque interception-réémission introduit un excès de bruit ramené à l'entrée (c'est-à-dire un bruit au-dessus du bruit de photon associé aux pertes du canal χ_0) de deux unités de bruit de photon. Ce bruit ajouté correspond à la frontière entre transmission classique et transmission quantique qui délimite les canaux capables de transmettre de l'intrication des canaux purement classiques. L'attaque interception-réémission – bien évidemment classique car interrompue par un canal classique chez Ève – se trouve exactement à la limite entre les domaines classique et quantique.

8.2 Réalisation expérimentale

L'attaque interception-réémission nécessite trois détections homodynes et deux systèmes de modulation. La figure 8.1 montre qu'il faut deux détections homodynes et un système de modulation pour réaliser l'attaque. De plus, nous avons toujours besoin du système de modulation d'Alice et de la détection homodyne de Bob. Afin d'éviter une duplication

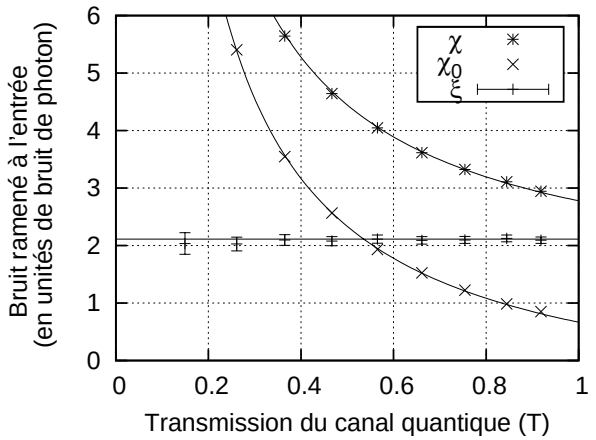


FIG. 8.3: Bruits ramenés à l'entrée entre Alice et Bob pour une attaque de type interception-réémission. Le bruit ajouté total est la somme du bruit ajouté dû aux pertes $\chi_0 = \frac{1}{\eta T} - 1$ et du bruit en excès $\xi = 2+\xi_T$ où $\xi_T \simeq 0,1$ est le bruit technique de l'expérience. On note $\eta = 0,6$ l'efficacité de la détection homodyne.

inutile de notre matériel de cryptographie, mous avons séparé la mise en œuvre de l'attaque «interception-réémission» en trois étapes (figure 8.2) :

1. Dans une première étape, Alice envoie un signal modulé aléatoirement en amplitude et en phase avec une distribution gaussienne dans l'espace des phases. Ève, empruntant la détection homodyne habituellement possédée par Bob, mesure la quadrature X du signal.

2. Alice réitère l'émission de la même séquence aléatoire. Cette fois, Ève utilise Bob pour mesurer P.

3. Munie de ces deux mesures de quadrature, Ève utilise le système de modulation d'Alice pour envoyer le signal à travers un canal atténuant (simulé par un modulateur d'amplitude) vers la détection homodyne, qui est à nouveau attribuée à Bob.

Une boucle de rétroaction logicielle mesurant la phase relative entre le signal et l'oscillateur local (la méthode de cette mesure sera décrite au chapitre 10), et agissant avec un délai de 200 ms, permet à Ève de choisir la quadrature absolue $\hat{X}$ ou $\hat{P}$. Un système de synchronisation entre Alice et Bob (lui aussi décrit chapitre 10) permet d'associer chaque mesure de la quadrature $\hat{X}$ à la mesure de la quadrature $\hat{P}$ correspondante.

Afin de prendre en compte la lame séparatrice qui sépare les deux mesures de quadrature d'Ève, le niveau de sortie V_A du signal chez Alice est défini comme deux fois la variance de modulation mesurée par Ève V_E. Comme nous l'avions vu au chapitre précédent, ce choix consiste simplement à définir le point du montage que nous appelons «sortie d'Alice», et ne change en rien le résultat des mesures physiques. Cette définition permet de simuler un espion possédant une détection homodyne sans perte avec notre détection imparfaite (efficacité $\eta = 0.6$). Nous avons mesuré une variance de modulation de $V_E = 18,29$ pour la quadrature $\hat{X}$ et de $V_E = 18,26$ pour la quadrature $\hat{P}$, d'où une variance de modulation $V_A = 36,6$.

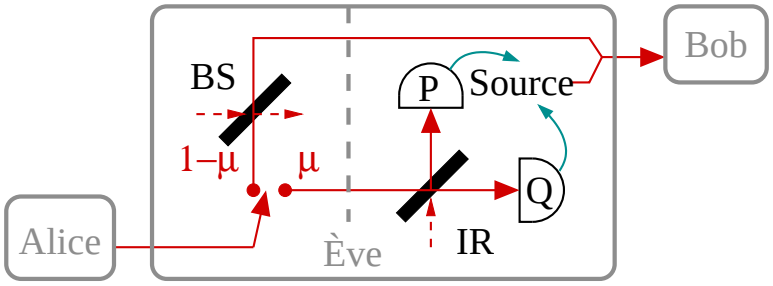


FIG. 8.4: Schéma d'une attaque interception-réémission partielle. Avec une probabilité μ, Ève intercepte le signal envoyé par Alice et ré-émet un signal fonction de sa mesure. Sur les autres impulsions, Ève réalise une attaque lame séparatrice standard.

8.3 Analyse du bruit

L'excès de bruit entre Alice et Bob est mesuré à partir des corrélations entre leurs séries de données. La méthode utilisée pour cette mesure est identique à celle employée au chapitre 7. En faisant varier la transmission du canal entre Ève et Bob, nous traçons les différents bruits ramenés à l'entrée (figure 8.3). Le bruit ajouté total χ est calculé à partir des corrélations entre les données d'Alice et de Bob. La transmission du canal quantique permet de déterminer le bruit ajouté dû aux pertes $\chi_0 = \frac{1-T_q}{T_q}$. L'excès de bruit se déduit de la différence entre ces deux bruits : $\xi = \chi - \chi_0$. Nous mesurons un excès de bruit typique de $2,1N_0$. L'écart à l'excès de bruit théorique de 2 unités de bruit de photon est dû aux imperfections de notre dispositif expérimental :

- bruit de phase du laser ;
- défauts de modulation (intervenant deux fois) ;
- bruit électronique de la détection homodyne (intervenant deux fois) ;
- écart entre la quadrature mesurée et les quadratures absolues $\hat{X}$ et $\hat{P}$, due à une incertitude sur la détermination de la phase relative.

8.4 Attaque interception-réémission partielle

Une attaque interception-réémission partielle permet à l'espion de contrôler les deux paramètres du modèle du canal quantique gaussien. Nous avons vu que l'attaque de type «interception-réémission» permettait d'aller au-delà de la traditionnelle attaque par lame séparatrice en introduisant du bruit en excès au-delà du bruit standard dû aux pertes du canal. Cependant, cette attaque ne permet pas de contrôler la quantité d'excès bruit ; elle reste limitée à une quantité de bruit ajouté importante, ce qui interrompt toute transmission quantique entre Alice et Bob. Nous proposons maintenant la réalisation expérimentale d'une attaque mélangeant aléatoirement les deux types d'attaques. Nous la baptisons «interception-réémission partielle». Cette attaque permet de faire varier les deux paramètres du modèle du canal gaussien. Ainsi, cette attaque se veut la plus générale dans ce modèle.

Pour mettre en œuvre cette attaque, nous introduisons aléatoirement les données envoyées par Alice à la place d'une fraction des données interceptées par Ève. Le déroulement de l'attaque interception-réémission partielle est en tout point similaire au

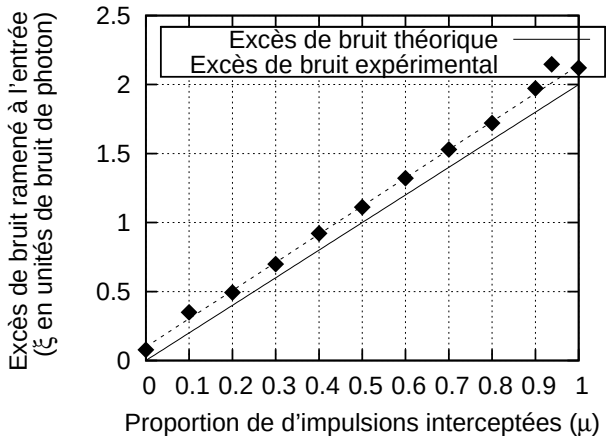


FIG. 8.5: Excès de bruit ramené à l'entrée pour différentes attaques interception-réémission partielles, depuis l'attaque lame séparatrice ($\mu = 0$) jusqu'à l'attaque interception-réémission totale ($\mu = 1$). À cause des imperfections expérimentales, l'excès de bruit mesuré est supérieur à l'excès de bruit théorique $\xi = 2\mu$. La courbe pointillée indique un bruit technique de 0,1 unité de bruit de photon. Notons que, en théorie, le bruit technique de l'attaque lame séparatrice doit être légèrement inférieur au bruit technique de l'attaque interception-réémission, car cette dernière attaque comporte plus d'étapes expérimentales susceptibles d'ajouter du bruit. Cet effet n'est pas visible sur cette figure à cause d'un artefact expérimental : la minimisation du bruit technique demande un réglage minutieux des paramètres des modulateurs. Pour chaque μ, la valeur du bruit technique indique davantage le degré de satisfaction de l'expérimentateur que la valeur ultime que l'expérience permet d'atteindre.

déroulement de l'attaque interception-réémission. Simplement, lors du transfert classique des données mesurées par Ève entre son système de détection et son système d'émission, nous réintroduisons une fraction $1 - \mu$ des données initialement envoyées par Alice. Ces données ne voient donc que le canal quantique de transmission T_q situé entre Alice et Bob, qui est simulé par un modulateur d'intensité : c'est ainsi qu'on simule usuellement une attaque de type lame séparatrice [2]. La figure 8.4 résume notre protocole expérimental.

8.5 Analyse du bruit de l'attaque interception-réémission partielle

Commençons par une analyse théorique du bruit introduit par notre attaque interception-réémission partielle. Chacune des attaques lame séparatrice, interception-réémission, et interception-réémission partielle transforment la quadrature X_A envoyée par Alice en

$$\hat{X}_B = t(X_A + \hat{X}_N) \tag{8.8}$$

L'analyse du bruit repose sur la détermination des moments d'ordre deux de la distribution de probabilité conjointe entre Alice et Bob, que nous résumons dans le tableau suivant, pour

[2]Notons que la partie interception-réémission de notre attaque est physiquement réalisée, alors que la partie lame séparatrice est seulement simulée : comme expliqué dans l'introduction de ce chapitre, une véritable attaque lame séparatrice nécessite une mémoire quantique, et n'est donc pas réalisable avec les technologies d'aujourd'hui.

chacune des attaques :

	lame séparatrice	interception-réémission	interception-réémission partielle
V_A	V_A	V_A	V_A
$\langle AB\rangle^2$	TV_A^2	TV_A^2	TV_A^2
V_B	TV_A+1	$T(V_A+2)+1$	$T(V_A+2\mu)+1$

(8.9)

Les deux premières colonnes reprennent les valeurs déjà rencontrées dans ce manuscrit, la troisième est simplement la somme pondérée des deux premières. Cela se justifie aisément : l'attaque interception-réémission partielle est l'entrelacement des deux phénomènes aléatoires que sont les attaques lame séparatrice de distribution de probabilité $p_{BS}(X_A, X_B)$ et interception-réémission de distribution de probabilité $p_{IR}(X_A, X_B)$. La distribution de probabilité finale est donc $p_{IR\,\text{partielle}} = \mu p_{IR}(X_A, X_B) + (1-\mu)p_{BS}(X_A, X_B)$. La moyenne statistique d'une grandeur quelconque Y selon cette loi de probabilité se calcule donc par $\langle Y\rangle = \mu\langle Y\rangle_{IR} + (1-\mu)\langle Y\rangle_{BS}$.

Munis des moments d'ordre deux, nous pouvons calculer le bruit ramené à l'entrée $V_N = V_A\left(\frac{V_A V_B}{\langle AB\rangle^2} - 1\right)$, le bruit ajouté ramené à l'entrée $\chi = V_N - 1$, et l'excès de bruit $\xi = \chi - \chi_0$ (rappel : $\chi_0 = \frac{1-T}{T}$) :

	lame séparatrice	interception-réémission	interception-réémission partielle
V_N	$\frac{1}{T}$	$\frac{1}{T}+2$	$\frac{1}{T}+2\mu$
χ	$\frac{1}{T}-1$	$\frac{1}{T}+1$	$\frac{1}{T}-1+2\mu$
ξ	2	0	2μ

(8.10)

La théorie prédit donc un excès de bruit $\xi = 2\mu$ variable avec la fraction de données interceptées μ, comme nous l'annoncions dans la section précédente.

Nous pouvons mesurer expérimentalement l'excès de bruit introduit par l'attaque interception-réémission partielle, de la même manière que nous l'avions fait pour l'attaque interception-réémission totale dans la section 8.3. La figure 8.5 trace cet excès de bruit pour différentes valeurs de μ, depuis $\mu = 0$ (attaque lame séparatrice) jusqu'à $\mu = 1$ (attaque interception-réémission totale). Comme pour l'attaque interception-réémission totale, l'excès de bruit est environ $0,01N_0$ au-dessus de l'excès de bruit théorique, à cause des imperfections expérimentales équivalentes à un bruit technique.

8.6 Informations accessibles à l'espion

Nous calculons dans cette section les informations mutuelles entre Ève et Bob pour différents types d'attaques.

Attaque optimale. La première information que l'on peut calculer est l'information qu'aurait obtenue Ève en attaquant de façon optimale le canal quantique reliant Alice et Bob. Pour calculer cette information, Alice et Bob évaluent le gain G et le bruit ajouté χ du canal à partir de la variance et de la corrélation de leurs données. Nous avons établi au chapitre 7 l'expression de cette information :

$$I_{BE}^{opt} = \frac{1}{2}\log_2 \frac{\eta TV + \eta T\xi}{\eta/\left[1 - T + T\xi + \frac{T}{V}\right] + 1 - \eta} \tag{8.11}$$

où η est l'efficacité de détection homodyne. Ève obtient cette information en utilisant une cloneuse intriquante (voir section 2.5).

Comme nous disposons des données interceptées par Ève, nous pouvons évaluer l'information acquise par l'attaque interception-réémission partielle. Elle est la somme pondérée des informations acquises par les deux composantes de l'attaque :

$$I_{BE}^{reel} = \mu I_{BE}^{IR} + (1-\mu) I_{BE}^{BS} \tag{8.12}$$

I_{BE}^{IR} est l'information apportée par la partie interception-réémission de l'attaque. La formule de Shannon permet de la calculer à partir du calcul du coefficient de corrélation ρ entre les données d'Ève et de Bob à partir des données expérimentales, comme nous l'avions fait chapitre 7 pour la calcul de I_{AB} :

$$I_{BE}^{IR} = -\frac{1}{2}\log_2\left(1-\rho^2\right). \tag{8.13}$$

I_{BE}^{BS} est l'information acquise par une attaque lame séparatrice usuelle. Comme ce type d'attaque ne fournit pas les données acquises par l'espion, nous reprenons le calcul de l'information associée à une attaque de type lame séparatrice que nous avions réalisé en section 2.5. Pour cela, nous calculons les moments d'ordre deux entre Bob et Ève dans le cas où Ève et Bob sont tous deux soumis à un bruit technique ξ :

$$\begin{aligned}
V_B &= G\eta\left(V_A + \xi + \frac{1}{G\eta}\right) && (8.14)\\
V_E &= (1-G)\left(V_A + \xi + \frac{1}{1-G}\right) && (8.15)\\
\langle Q_B Q_E\rangle &= \sqrt{G(1-G)}(V_A+\xi) && (8.16)\\
I_{BE}^{BS} &= -\frac{1}{2}\log_2\left(1 - \frac{\langle Q_B Q_E\rangle^2}{V_B V_E}\right) && (8.17)
\end{aligned}$$

Notons que même si ce calcul est théorique, il prend en compte les caractéristiques mesurées de l'expérience. De plus, on peut voir l'information extraite par l'espion au cours d'une transmission T comme l'information extraite par Bob au cours d'une transmission $1-T$. Ainsi, les paramètres expérimentaux que nous utilisons sont ceux effectivement mesurés dans une situation identique à celle de l'espion au cours de notre expérience interception-réémission.

L'information I_{BE}^{reel} est inférieure à I_{BE}^{opt} pour deux raisons. Premièrement, bien que l'attaque lame séparatrice soit optimale pour $\xi = \mu = 0$, l'attaque interception-réémission n'est pas l'attaque optimale pour $\xi = 2$. Donc sauf pour $\mu = 0$, $I_{BE}^{reel}(\mu) < I_{BE}^{opt}(\mu)$. Deuxièmement, nos attaques lame séparatrice et interception-réémission ne sont pas parfaitement réalisées. Les mesures d'Ève sont entachées d'excès de bruit de nature technique. La figure 8.6 illustre ce dernier défaut : les courbes I_{BE}^{reel} et I_{BE}^{opt} sont légèrement décalées pour $\mu = 0$. Toutefois, l'écart est d'autant plus faible que T est petit, car l'excès de bruit est diminué de la transmission du canal.

Comme I_{BE}^{BS} est l'information qu'obtient l'espion par une attaque lame séparatrice, c'est-à-dire une attaque qui exploite uniquement les pertes du canal, la différence entre I_{BE}^{reel} et I_{BE}^{BS} représente la quantité d'information que l'excès de bruit apporte à l'espion. Notre attaque, bien que sous-optimale, permet clairement d'exploiter l'excès de bruit.

La figure 8.6 trace ces informations en fonction de la fraction μ des données interceptées et ré-émises, pour plusieurs transmissions T du canal reliant Alice à Bob.

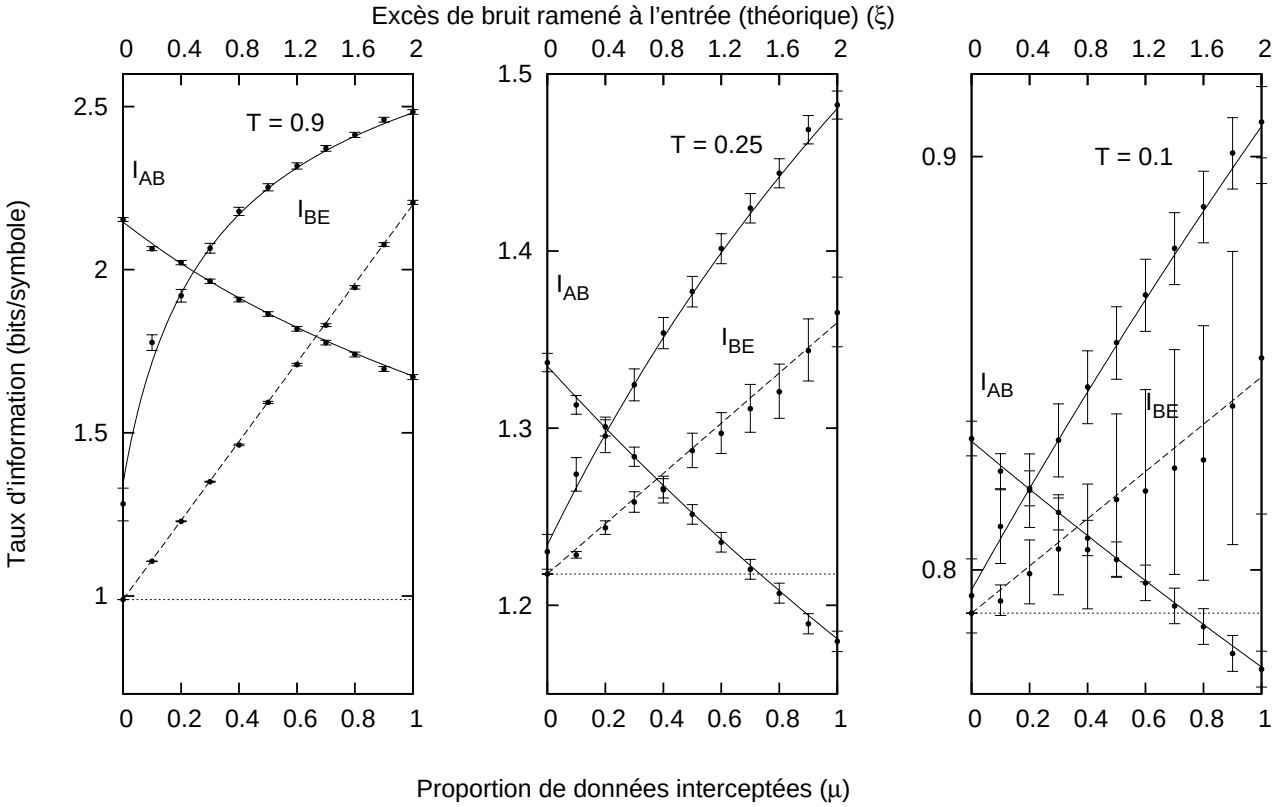


FIG. 8.6: Quantité d'information acquise par l'espion lors d'une attaque interception-réémission partielle, pour différentes valeurs de la transmission du canal. Ces figures tracent l'information maximale qu'aurait pu obtenir Ève en utilisant une attaque optimale (trait plein), l'information qu'elle obtient avec une simple attaque lame séparatrice n'utilisant pas l'excès de bruit (points), et enfin l'information qu'elle obtient avec une attaque interception-réémission partielle (tirets). Même si cette dernière attaque n'est pas optimale, elle permet à l'espion de tirer profit de l'excès de bruit présent sur la ligne. Dans une situation de distribution quantique de clé, Alice et Bob doivent estimer les informations mutuelles I_{AB} et I_{BE} à partir d'un échantillon de données. La taille finie de cet échantillon introduit des fluctuations statistiques dont l'écart type est représenté par des barres d'erreur (pour une taille arbitraire de 5000 échantillons). Plus la transmission est faible, plus les incertitudes sont grandes. Dans ce cas, Alice et Bob doivent prévoir une marge de sécurité suffisante, ou accroître la taille de leur échantillon. Nos données expérimentales montrent qu'Alice et Bob gardent un avantage, même pour de faibles transmissions, tant que l'excès de bruit est modéré (environ inférieur à $0,2N_0$).

8.7 Attaque non gaussienne

L'attaque interception-réémission partielle permet non seulement de réaliser l'ensemble des attaques possibles dans le modèle du canal gaussien, mais elle est aussi une attaque non gaussienne En effet, la distribution des quadratures Q_B mesurées par Bob s'exprime

$$p(Q_B = q) = \mu \frac{1}{\sqrt{2\pi\sigma_{IR}^2}} e^{-\frac{q^2}{2\sigma_{IR}^2}} + (1-\mu) \frac{1}{\sqrt{2\pi\sigma_{BS}^2}} e^{-\frac{q^2}{2\sigma_{BS}^2}} \tag{8.18}$$

avec $\sigma_{IR}^2 = T_q\,(V + \chi_0 + 2)$ et $\sigma_{BS}^2 = T_q\,(V + \chi_0)$, où $\chi_0 = \frac{1-T_q}{T_q}$.

La sous-optimalité de notre attaque est une illustration des preuves de sécurité face à des attaques non gaussiennes : l'information obtenue par une attaque non gaussienne est toujours inférieure à l'information obtenue par l'attaque gaussienne optimale :

$$I_{BE}^{reel} < I_{BE}^{opt}. \tag{8.19}$$

Nous n'avons bien sûr pas exploré l'ensemble des attaques non gaussiennes possibles, donc notre argument est une simple vérification plutôt qu'une véritable preuve.

Afin d'illustrer de façon plus convaincante le théorème d'extrémalité des attaques gaussiennes énoncé au chapitre 4, considérons l'attaque suivante : Ève traite séparément les impulsions interceptées des impulsions qui ont été atténuées, et fait une attaque optimale sur chacun des groupes. Cette attaque présente la même distribution des données vues par Bob (équation 8.18), mais elle est évidemment meilleure que la nôtre. L'information acquise par Ève lors de cette attaque est

$$I_{BE}^{ng} = \mu I_{BE}^{opt}(\mu = 1) + (1-\mu) I_{BE}^{opt}(\mu = 0). \tag{8.20}$$

C'est la droite rejoignant les points extrêmes de I_{BE}^{opt}. Pour $T = 0.1$ on constate que cette information est très proche de l'information acquise par l'attaque optimale (la différence est inférieure à 0,5%), mais lui reste toujours inférieure. Cette attaque illustre donc encore une fois l'optimalité des attaques gaussiennes : l'information qu'elle apporte à l'espion se rapproche de la meilleure attaque, sans pour autant la dépasser.

Nous pouvons également calculer l'information mutuelle I_{AB}^{ng} en utilisant la formule de Shannon

$$I_{AB}^{ng} = S(A) + S(B) - S(A, B), \tag{8.21}$$

où l'entropie S est calculée numériquement à partir de la distribution des données 8.18, d'après l'expression

$$S(X) = \int dx\, p(X = x) \log_2(p(X = x)). \tag{8.22}$$

I_{AB}^{ng} est l'information maximale qu'Alice et Bob pourraient extraire de leurs données non gaussiennes. Cette information est supérieure à l'information qu'Alice et Bob extraient dans le modèle du canal gaussien :

$$I_{AB}^{ng} > I_{AB}^{gauss}. \tag{8.23}$$

Le sens de cette inégalité est conforme aux preuves de sécurité du protocole face aux attaques non gaussiennes, et inversé par rapport à l'inégalité pour I_{BE} (équation 8.20) : le modèle du canal gaussien, qui calcule l'information mutuelle uniquement à partir de mesures de variances,

ne surestime pas l'information contenue dans les corrélations entre les données d'Alice et de Bob. On remarque que l'écart entre I_{AB}^{ng} et I_{AB}^{gauss} est négligeable (moins de 10^{-5} pour toute valeur de T) : Alice et Bob gagneraient peu d'information s'ils utilisaient la distribution non gaussienne exacte des données, au lieu d'une estimation dans le modèle du canal gaussien.

En conclusion, l'attaque interception-réémission partielle que nous avons réalisée permet de vérifier la robustesse de notre système de distribution quantique de clé en couvrant une large gamme d'attaques. En effet, cette attaque nous permet de simuler l'ensemble des canaux quantiques gaussiens. Notamment, nous avons montré qu'Ève peut tirer parti de l'excès de bruit pour acquérir de l'information. D'autre part, cette attaque est un attaque non gaussienne. Nous avons vérifié qu'elle fournit moins d'information à l'espion que l'attaque gaussienne optimale.

Chapitre 9

Multiplexage temporel

Notre système de distribution quantique de clé avec des états cohérents est un interféromètre délocalisé entre Alice et Bob. Il nécessite donc la transmission sur le canal quantique d'un signal et d'une référence de phase. Une première idée serait de transmettre les deux faisceaux dans deux fibres séparées. Cependant, les fluctuations entre les deux fibres (polarisation, phase, temps) détruiraient la cohérence entre le signal et l'oscillateur local. De plus, il est raisonnable de penser qu'un canal mis à disposition d'un système de cryptographie quantique ne comportera qu'une seule fibre (c'est le cas notamment du canal qui sera fourni par Siemens dans le cadre du projet SECOQC). Nous devons donc mettre en place un système de multiplexage qui permet de transporter le signal et la référence de phase dans une même fibre optique.

Plusieurs contraintes nous sont imposées dans la réalisation de ce système de multiplexage. Tout d'abord, le système de démultiplexage, chez Bob, doit introduire le moins de pertes possible sur la voie signal : toute dégradation du rapport signal à bruit entraîne une perte de taux secret. Ensuite, le système doit être stable et, si possible, simple à mettre en œuvre.

On peut envisager deux systèmes de multiplexage. Premièrement, un multiplexage fréquentiel dans lequel le signal et la référence de phase sont séparés fréquentiellement lors du passage dans la fibre optique. Cette méthode a l'avantage d'être couramment employée dans les télécommunications classiques. La bande C[1] est divisée en canaux de 0,5 nm de large et séparés de 3 nm. Il existe des filtres fibrés introduisant moins de 0,5 dB de pertes conçus pour séparer ces canaux. Pour décaler les fréquences des deux faisceaux, nous pouvons utiliser des modulateurs électro-optiques en niobate de lithium. Ces modulateurs ont une bande passante de quelques dizaines de gigahertz ; ils peuvent donc introduire un décalage fréquentiel de plusieurs dixièmes de nanomètres. Un inconvénient majeur de ce système est sa complexité : il faut disposer de deux modulateurs électro-optiques et de deux sources de modulation indépendantes (l'une chez Alice et l'autre chez Bob, sans possibilité de transmettre de signal électrique de l'une à l'autre) de fréquence supérieure à 10 GHz et synchronisées avec une précision meilleure que le hertz, ordre de grandeur de la dérive de phase entre signal et oscillateur local que notre système d'acquisition est capable de compenser.

Nous avons opté pour un système de multiplexage temporel, qui est à la fois plus simple conceptuellement et dans sa mise en œuvre. Ce système de multiplexage, qui

[1] Plage de fréquences entre 1530 nm et 1570 nm, centrée sur le minimum d'absorption des fibres optiques en silice.

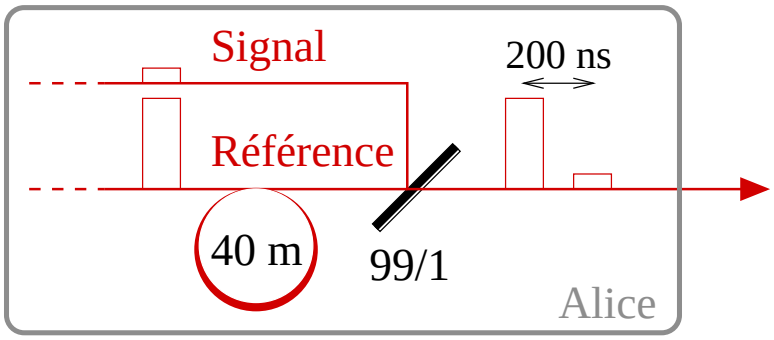


FIG. 9.1: Schéma de principe du multiplexage temporel. La référence de phase est décalée de 200 ns par rapport au signal. Les deux faisceaux sont injectés dans le canal quantique par un coupleur 99/1

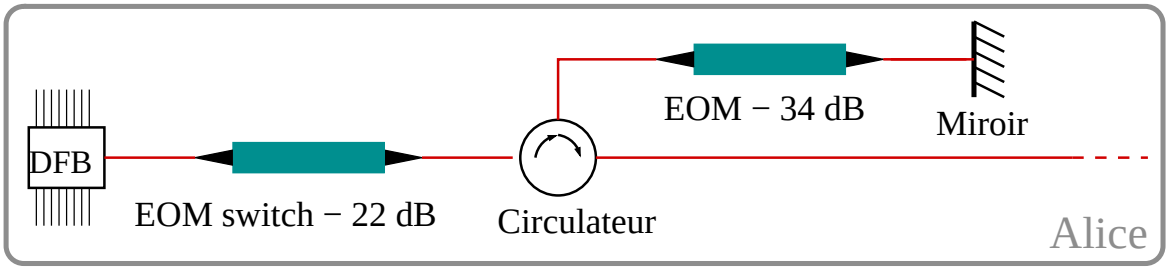


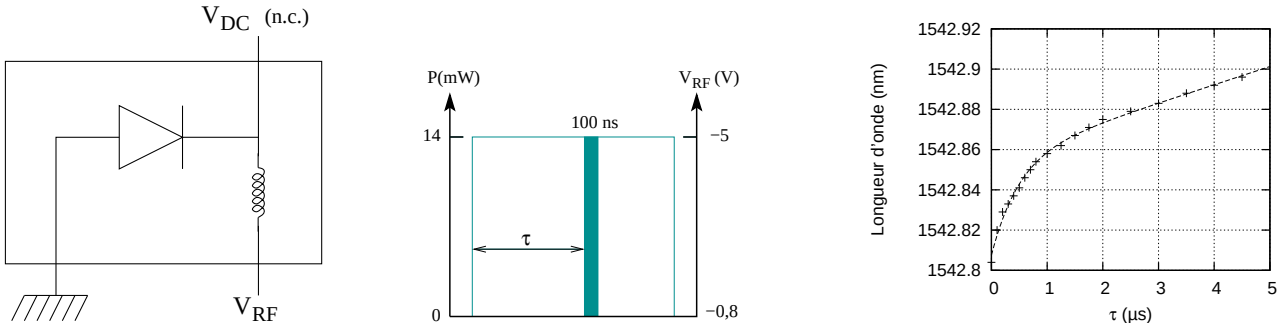
FIG. 9.2: Préparation des impulsions de grande profondeur de modulation. Un switch électro-optique d'une extinction de 22 dB est suivi d'un double passage dans un modulateur de 34 dB d'extinction, soit un total de 90 dB

fonctionne avec une détection homodyne limitée au bruit de photon, a fait l'objet du dépôt de brevet [53].

9.1 Multiplexage temporel avec une diode laser continue

Notre système de multiplexage temporel est représenté figure 9.1. Chez Alice, l'oscillateur local est retardé par rapport au signal par une ligne à retard optique de 40 mètres (rouleau de fibre monomode), occasionnant un délai de 200 nanosecondes, c'est-à-dire deux fois plus long que la largeur des impulsions. Le signal, puis l'oscillateur local, sont ensuite couplés dans le même canal de transmission à l'aide d'un coupleur 99/1. Le choix de ce rapport de couplage permet d'atténuer suffisamment le signal jusqu'à la variance de modulation V_A, tout en laissant la puissance de l'oscillateur local presque constante. Nous verrons à la section 9.3 que notre système de démultiplexage impose que le signal traverse le canal quantique avant l'oscillateur local.

Nous découpons les impulsions avec un modulateur d'amplitude qui a une extinction de 34 dB. L'oscillateur local comporte typiquement 10^8 photons par impulsion de 100 ns, laissant ainsi un fond inférieur à 10^5 photons par 100 ns. Ce fond est donc très grand devant un signal dont la variance de modulation n'est que d'une dizaine de photons par impulsion.



(a) Schéma de câblage. L'entrée RF de la diode laser est directement reliée à un générateur d'impulsion, tandis que la voie DC est non connectée.

(b) Tension d'alimentation et puissance lumineuse crête. Un modulateur d'amplitude placé en sortie de la diode permet de sélectionner une impulsion de largeur 100 ns, à un instant τ après le début de l'impulsion.

(c) Variation de la longueur d'onde après établissement du courant dans la diode ($\tau = 0$). La longueur d'onde centrale de l'impulsion de 100 ns est mesurée avec un analyseur de spectre optique. Les données expérimentales sont ajustées de façon *ad hoc* par la fonction $\lambda(\tau) = -0,048e^{-2,02\tau} + 0,0090\tau + 1542,86$, puis $f(\tau) = \frac{hc}{\lambda(\tau)}$.

FIG. 9.3: Fonctionnement de la diode laser pulsée AL1905LMI

Pour ne pas saturer le signal par le fond de l'oscillateur local, nous devons préparer des impulsions avec une profondeur de modulation supérieure à 80 dB. Nous avons atteint une telle extinction par trois passages successifs dans des modulateurs électro-optiques (figure 9.2). Un premier passage dans un switch électro-optique permet une atténuation de 22 dB. Ensuite, l'association d'un circulateur optique et d'un miroir permet le double passage dans un modulateur d'extinction de 34 dB, ce qui permet une extinction totale de 90 dB. Dans cette configuration, nous avons mesuré un fond inférieur à 2 photons par 100 ns, qui a pour seul effet d'introduire un décalage constant sur le signal de détection homodyne. Cette méthode fonctionne, mais présente plusieurs désavantages. Tout d'abord, cet enchaînement de modulateurs nécessite une synchronisation complexe des impulsions électriques qui les alimentent, ainsi que deux contrôles de tension de biais. Ensuite, chacun des passages dans un modulateur électro-optique et le passage dans le circulateur induisent des pertes d'insertion de 3 dB, soit des pertes totales de 12 dB. Dans ces conditions le niveau d'oscillateur local conduit à une variance du bruit de photon seulement 2,5 fois supérieure au bruit électronique, ce qui dégrade le taux secret (voir section 9.4).

9.2 Multiplexage temporel avec une diode pulsée

Pour remédier au problème du fond de l'oscillateur local introduit par la génération d'impulsions à partir d'une diode continue, nous avons utilisé une diode Alcatel AL1905LMI pulsée électriquement. En effet, une diode pulsée électriquement a une extinction parfaite puisqu'entre deux impulsions la tension de commande de la diode est en dessous du seuil laser et seule une faible lumière de fluorescence incohérente, donc invisible pour notre interféromètre, est émise. L'utilisation d'une diode pulsée permet donc *a priori* d'éliminer tout fond sur l'oscillateur local qui se superposerait au signal lors du multiplexage. Nous verrons dans cette section que l'utilisation d'une diode pulsée n'est pas aussi immédiate qu'il n'y paraît.

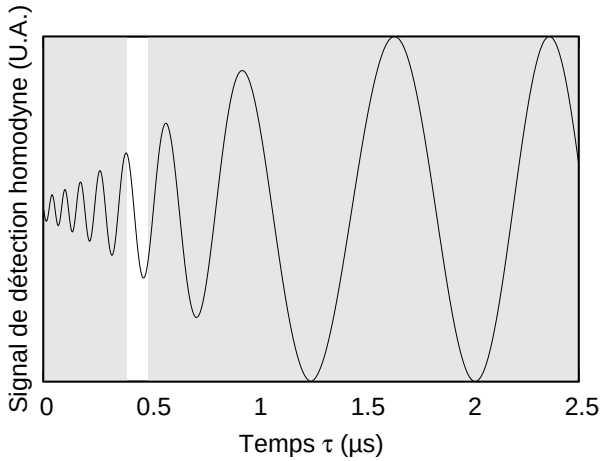


FIG. 9.4: Modèle du signal issu de la détection homodyne obtenu en utilisant notre diode pulsée. L'origine de l'axe temporel correspond au début de l'impulsion lumineuse. Nous observons en sortie de la détection homodyne une tranche de largeur 100 ns de ce signal (partie non grisée). Ce signal est obtenu en faisant interférer deux ondes de fréquences respectives $f(\tau)$ et $f(\tau + \Delta T)$, où $\Delta T = 0{,}75$ ns correspond à une différence de marche de 15 cm. L'électronique de la détection coupe les fréquences supérieures à $f_c = 10$ MHz (fonction de transfert T constante pour $f < f_c$, puis décroissance de 20 dB par décade

En effet, un régime transitoire, de plusieurs centaines de nanosecondes après l'établissement du courant dans la diode, pendant lequel les propriétés spectrales de la diode sont mauvaises, nous oblige à générer de larges impulsions électriques dans lesquelles nous sélectionnons une impulsion de 100 ns avec un modulateur d'amplitude. Nous rencontrons alors de nouveau notre problème initial : le signal se propage dans le canal 200 ns avant l'oscillateur local ; il est donc superposé au fond du régime transitoire de l'oscillateur local. Finalement, nous verrons qu'en ajustant astucieusement le délai entre le début de l'impulsion électrique et le début de l'impulsion optique sélectionnée par le modulateur, nous pouvons rendre le fond de l'oscillateur local assez incohérent avec le signal pour qu'il ne perturbe pas la mesure interférométrique.

Le schéma de câblage de la diode, ainsi que sa tension de modulation sont représentées figure 9.3(a). La diode est alimentée par un générateur de fonctions arbitraires dont la tension de sortie est convertie en courant par la résistance interne à la diode (50 Ω). Le courant électrique dans la diode varie entre 15 mA (juste en dessous du seuil d'émission laser) et 100 mA, ce qui correspond à une puissance crête de 16 mW.

Pour caractériser cette diode, nous la modulons avec des impulsions électriques de largeur 3 μs à un taux de répétition de 200 kHz dans lesquelles nous sélectionnons une impulsion de largeur 100 ns, à un instant τ après le début de l'impulsion électrique (figure 9.3(b)). Cette sélection nous permet d'éliminer le régime transitoire après l'établissement du courant dans la diode et nous caractérisons ce régime en faisant varier τ. La durée entre la fin de l'impulsion de 100 ns et la fin de l'impulsion électrique n'a pas d'influence sur le signal observé.

La fréquence centrale du mode issu de la diode, notée $f(\tau)$, varie au cours de l'impulsion, comme représenté figure 9.3(c). Cette variation de fréquence, conjuguée avec une différence de marche non nulle Δx entre les voies signal et oscillateur local, altère le signal de sortie de la détection homodyne. Nous modélisons cet effet en faisant interférer deux ondes

de fréquences respectives $f(\tau)$ et $f(\tau + \Delta t)$, avec $\Delta t = \frac{n\Delta x}{c}$. Le signal est ensuite filtré par la fonction de transfert T de l'électronique de la détection homodyne, de fréquence de coupure f_c = 10 MHz :

$$s(\tau) = \sin\left(2\pi f(\tau + \Delta t)(\tau + \Delta t) - 2\pi f(\tau)\tau + \phi\right) T\left(f(\tau + \Delta t) - f(\tau)\right), \tag{9.1}$$

où ϕ est la phase relative entre le signal et l'oscillateur local, dérivant lentement. Ce signal, tracé figure 9.4 pour une différence de marche de 15 cm, rend compte des observations expérimentales. On constate des oscillations dont la fréquence décroît avec τ, et dont l'amplitude croît avec τ. L'amplitude de modulation mesurée par la détection homodyne est proportionnelle à cette amplitude : quand celle-ci n'est pas maximale, le rapport signal à bruit de notre transmission est dégradé, et le taux secret diminue.

À cause de la différence de marche entre le signal et l'oscillateur local, nous observons le bruit de phase dû à la dérive de la phase de la diode. Pour une différence de marche de 15 cm, nous observons un bruit de phase supérieur à 0,2 N_0 par photon (défini chapitre 7). Pour ces raisons, nous avons équilibré les voies de l'interféromètre à moins d'1 cm. Pour ce faire, nous avons placé le début de l'impulsion de 100 ns à $\tau < 100$ ns et modifié la différence de marche, afin de maximiser la période des oscillations observées sur le signal de détection homodyne.

Finalement, la diode pulsée nous permet de réaliser notre schéma de multiplexage. Pour cela, nous générons une impulsion électrique de largeur 900 μs dans laquelle nous sélectionnons une impulsion de largeur 100 ns, à un instant $\tau = 800$ ns après le début de l'impulsion électrique. Dans ces conditions, l'impulsion optique est assez cohérente pour présenter de bonnes caractéristiques de bruit compte tenu de l'équilibrage de notre interféromètre. D'un autre côté, nous nous plaçons suffisamment près du régime transitoire de la diode pour que la longueur d'onde du signal soit différente de la longueur d'onde du fond de l'oscillateur local qui s'y superpose. Quantitativement, le signal et le fond de l'oscillateur local sont générés à 200 ns d'intervalle (c'est-à-dire le temps de propagation dans notre ligne à retard). Ce décalage correspond à une différence de longueur d'onde d'environ 0,005 nm, différence qui provoque un battement trop rapide pour être observé avec la détection homodyne. Le schéma de multiplexage avec une diode pulsée permet donc d'éliminer totalement la perturbation du signal par le fond de l'oscillateur local.

Toutefois, cette configuration est de manipulation délicate. En effet, nous minimisons le fond de l'oscillateur local en ajustant la tension de biais du modulateur d'amplitude de dynamique 34 dB avec lequel nous découpons les impulsions optiques. Or cette tension de biais dérive sur une échelle de temps de quelques dizaines de minutes. Supposons que, à la suite de cette dérive, l'intensité du fond soit à 1% du niveau de l'oscillateur local. Dans ce cas, ce fond joue lui-même le rôle d'oscillateur local et mesure le bruit de photon d'un mode signal vide. Ce bruit, dont la variance est de 1% du bruit de photon observé avec le véritable oscillateur local, se superpose au signal utile et devient du même ordre de grandeur que les autres bruits du système.

Pour obtenir une expérience stable sur une durée plus longue, nous envisageons la réalisation d'un système de démultiplexage, décrit dans la section précédente, qui nous permettra d'intervertir le signal et l'oscillateur local dans le canal de communication. Ainsi, plus aucun fond ne se superposera au signal.

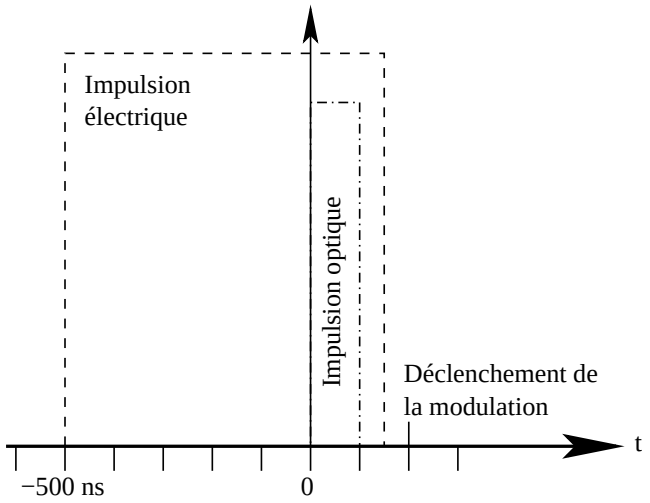


FIG. 9.5: Signaux de déclenchement. L'origine des temps est fixée au début de la génération de l'impulsion optique de 100 ns. La génération de l'impulsion électrique est avancée (500 ns sur notre schéma) pour éviter le régime transitoire de la diode laser pulsée. La modulation du signal est rafraîchie 100 ns après la fin de la génération de l'impulsion optique, délai supérieur au temps de propagation de l'impulsion jusqu'aux modulateurs d'amplitude et de phase. Cette nouvelle tension de modulation arrive en régime stationnaire 200 ns plus tard, produisant ainsi une modulation stable pour l'impulsion suivante.

Le multiplexage avec une diode pulsée nécessite la synchronisation de plusieurs signaux électriques : l'impulsion électrique qui pilote la diode laser, l'impulsion électrique de commande du modulateur d'amplitude, et le signal de déclenchement de la carte d'acquisition responsable de la modulation. Ces signaux sont représentés sur la figure 9.5.

9.3 Démultiplexage

Nous avons multiplexé le signal et l'oscillateur local dans le canal de communication quantique. Nous devons maintenant les séparer au niveau de Bob, en introduisant le moins de pertes possible sur le signal. Pour ce faire, le moyen le plus immédiat consiste à utiliser un switch électro-optique capable d'aiguiller chacune des impulsions avec un transitoire de quelques nanosecondes. Mais, ces composants introduisant 3 dB de pertes, il est préférable d'utiliser un coupleur passif dirigeant par exemple 90% du signal et 10% de l'oscillateur local dans leurs voies respectives. Nous avons donc opté pour cette dernière solution, plus simple à mettre en œuvre. Nous déterminerons le rapport de couplage optimal dans la section suivante.

Le démultiplexage avec un coupleur introduit des impulsions non désirées, représentées figure 9.6. 10 % du signal arrivent en premier sur la détection homodyne (à droite sur la figure). Ce signal n'est pas visible sur le signal de détection homodyne car il n'interfère avec aucun oscillateur local. Ensuite arrive le signal utile, que l'on désire observer. Enfin, les 90 % de l'oscillateur local se trouvant sur la voie signal interfèrent avec un mode vide. Avec une détection homodyne parfaite, nous observerions le bruit de photon du mode vide. Mais revenons sur la méthode employée pour équilibrer la détection homodyne : l'oscillateur local interfère avec le signal sur un coupleur légèrement déséquilibré (de rapport de couplage $0,5 \pm \epsilon$).

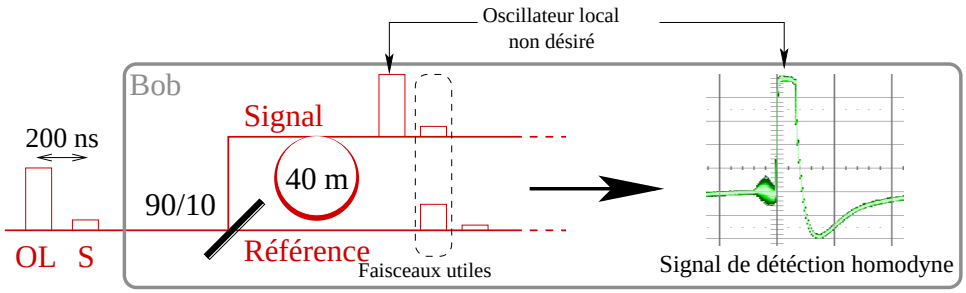


FIG. 9.6: Un coupleur, de rapport 90/10 est utilisé pour démultiplexer l'oscillateur local et le signal. L'utilisation d'un coupleur ajoute aux impulsions utiles (au centre) des impulsions non désirées (à gauche et à droite).

Les intensités dans chacune des voies de sortie de l'interféromètre sont alors $I_\pm = (0,5 \pm \epsilon) I_{OL}$, et, pour équilibrer la détection homodyne, nous introduisons des pertes $1 - \frac{0,5-\epsilon}{0,5+\epsilon} \sim 4\epsilon$ sur la voie $+$. Maintenant, si nous envoyons l'oscillateur local par la voie signal, nous obtenons des intensités $I_\pm = (0,5 \mp \epsilon) I_{OL}$ qui ne sont plus équilibrées par les pertes sur la voie $+$, et qui font saturer l'électronique de la détection homodyne. Finalement, comme la détection homodyne est équilibrée pour le faisceau oscillateur local utile, elle ne l'est pas pour l'oscillateur local non désiré entrant par la voie signal, qui vient saturer la détection homodyne.

Ce problème d'oscillateur local déséquilibré a deux solutions. Si la saturation de la détection homodyne est temporaire et disparaît avant l'arrivée de l'impulsion suivante, nous pouvons tout simplement ignorer cette saturation. La deuxième solution consiste à utiliser pour l'interférence homodyne un coupleur variable, dont le rapport de couplage est ajustable par une vis micrométrique. Ainsi, nous pouvons obtenir un couplage exactement égal à 1/2, et la détection homodyne devient équilibrée vis-à-vis des deux voies d'entrée, ce qui élimine la saturation due à l'oscillateur local non désiré.

La réponse de la détection homodyne nous a permis d'opter pour la première solution, plus immédiate. Le signal issu de la détection homodyne est représenté figure 9.7. Toutefois, cette solution impose que l'oscillateur local non désiré arrive sur la détection homodyne *après* le signal utile. Le signal doit donc traverser le canal quantique avec l'oscillateur local. Comme nous l'avons remarqué à la fin de la section précédente, nous envisageons de réaliser la deuxième solution, plus élégante, dans laquelle la détection homodyne sera équilibrée pour ses deux voies d'entrée.

9.4 Choix du coupleur de démultiplexage

Il nous reste maintenant à déterminer le rapport de couplage que nous allons utiliser pour séparer le signal et l'oscillateur local issus du canal quantique.

Le choix du rapport de couplage du coupleur de démultiplexage résulte d'un compromis. Si le rapport de couplage tend vers 0, la totalité du signal et de l'oscillateur local se trouvent sur la voie oscillateur local, et plus aucun signal n'est détecté. Au contraire, si le rapport de couplage tend vers 1, il n'y a plus de perte sur la voie signal, mais l'oscillateur local interférant avec le signal devient petit, et la variance du bruit de photon en sortie de la

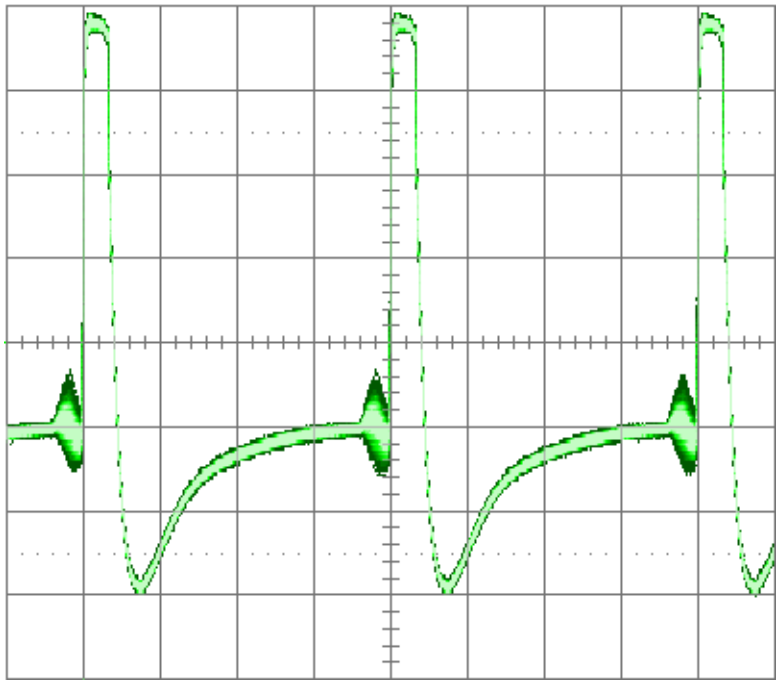


FIG. 9.7: Capture d'écran du signal de sortie de la détection homodyne (échelle : 1 V par division, 500 ns par division). Les impulsions lumineuses ont une largeur de 100 ns et un taux de répétition de 500 kHz. Le signal est modulé aléatoirement en amplitude et en phase avec une distribution gaussienne dans l'espace des phases. Nous avons sélectionné le mode persistance de l'oscilloscope afin de pouvoir observer cette modulation (partie large de la courbe). Après le signal utile, une impulsion oscillateur local non désirée vient saturer la détection homodyne. Celle-ci revient à l'équilibre avant l'arrivée de l'impulsion signal suivante.

détection homodyne devient petite devant le bruit électronique. Il existe donc un rapport de couplage optimal, équilibrant au mieux les pertes sur le signal et la variance du bruit de photon.

La figure 9.8 illustre ce compromis. Elle trace le taux secret $\Delta I = I_{AB} - I_{BE}$ selon les expressions 7.28 en fonction du rapport entre bruit électronique et bruit de photon, pour plusieurs valeurs des pertes totales sur le signal introduites chez Bob (courbes décroissantes). En faisant varier le rapport de couplage T_d du coupleur de démultiplexage, on se déplace sur les courbes grasses, en bas avec une diode continue, en haut avec une diode pulsée. Dans tous les cas, un rapport de couplage de $T_d = 90\%$ est assez proche de l'optimal. Quand le rapport de couplage augmente, ou quand la puissance nominale de l'oscillateur local augmente, les courbes grasses se rapprochent de leur asymptote (courbe décroissante pour $\eta = 0,6$). La figure 9.8 montre que le niveau de puissance atteint avec une diode pulsée est satisfaisant, et un gain conséquent en puissance ne produirait qu'un gain marginal en taux secret.

9.5 Multiplexage temporel dans une fibre de 25 km

Nous avons testé notre système de multiplexage sur un rouleau de fibre de longueur 25 km, en utilisant notre diode pulsée comme source laser. L'utilisation d'une véritable fibre optique suscite les remarques suivantes :

- Aucune fluctuation de polarisation sur une période de plusieurs heures n'est observée :

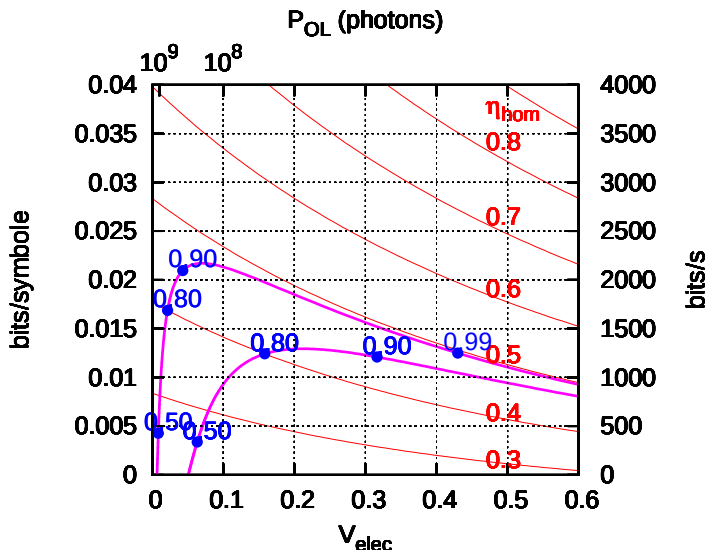


FIG. 9.8: Taux secret en fonction des imperfections de la détection homodyne. Pour des paramètres du canal T et ξ fixés, on calcule le taux secret $\Delta I(V_{\text{elec}}, \eta)$, où η est l'efficacité homodyne et V_{elec} la variance du bruit électronique exprimée en unités de bruit de photon. Pour tracer ce taux secret, nous choisissons de porter le bruit électronique V_{elec} en abscisse, et de tracer $\Delta I(V_{\text{elec}}, \eta)$ pour différentes valeurs de η (courbes fines décroissantes, portant mention de l'efficacité η). Dans une situation de démultiplexage, nous avons vu que les paramètres V_{elec} et η sont liés, et dépendent du rapport du coupleur de démultiplexage T_d. Les courbes grasses tracent la fonction paramétrique $(V_{\text{elec}}(T_d), \Delta I(V_{\text{elec}}(T_d), \eta(T_d))$, en mentionnant quelques valeurs de T_d. L'axe des ordonnées est gradué en bits par symbole (à gauche) et à, titre indicatif, en bits par seconde (à droite), en considérant une vitesse de réconciliation de 100 000 symboles par seconde. On constate une valeur optimale du rapport de couplage T_d résultant d'un compromis entre puissance du signal et puissance de l'oscillateur local. La courbe grasse du bas correspond au multiplexage avec une diode continue ($V_{\text{elec}} = 0.04 N_0$ pour $T_d = 0$), celle du haut au multiplexage avec une diode pulsée ($V_{\text{elec}} = 0,0043 N_0$ pour $T_d = 0$). Ces courbes ont été tracées avec les paramètres suivants : $V_A = 12$, $\xi = 0,025$, $T = 0,31$, $\beta = 0,87$ (efficacité de réconciliation). On estime la transmission nominale de la détection homodyne (*i.e.* hormis le coupleur de démultiplexage : $T_d = 1$) $\eta_{\text{nom}} = 0,5$, comme l'indique l'asymptote des courbes grasses.

le contrôle actif de la polarisation dans la fibre optique n'est pas nécessaire. Cependant, nous prévoyons un contrôleur de polarisation actif pour l'utilisation future d'une fibre optique installée.

- Aucune dérive sensible de la phase relative entre l'oscillateur local et signal n'est observée au cours de la propagation dans la fibre de transmission. Cela valide notre système de multiplexage.
- En revanche, la propagation dans la fibre introduit une gigue d'environ 50 ns sur l'instant d'arrivée des impulsions. Nous devons donc utiliser le signal optique comme source de déclenchement électronique chez Bob. Ce point sera précisé dans le chapitre suivant.

Notre transmission a les caractéristiques suivantes :

- Taux de répétition : 500 kHz.
- Taux de répétition moyen des impulsions utiles (hors impulsions test et impulsions de calibration) : 350 kHz.

- Atténuation du canal : 5 dB ($T = 0,31$).
- Variance de modulation : $V_A = 12$.
- Excès de bruit ramené à l'entrée : $\xi = 0,025N_0$.
- Transmission du coupleur de démultiplexage : $T_d = 0,9$.
- Variance du bruit électronique : $0,043N_0$ pour $T_d = 0,9$.
- Transmission nominale (*i.e.* tenant compte des pertes des composants, de l'inefficacité des photodiodes, hors coupleur de démultiplexage) : $\eta_{\text{nom}} = 0,5$.
- Pertes totales chez Bob : $\eta = \eta_{\text{nom}} T_d = 0,45$.

Avec ces paramètres, on calcule les taux secrets à l'aide des formules indiquées à la fin du chapitre 7 :

Nature du taux	taux (bits/symbole)	taux (bits/seconde)
Taux brut	0,11	39
Taux brut (Holevo)	0,094	33
Taux distillable	0,021	7,2
Taux final	0,021	**2,1 kb/s**

Le taux brut est le taux secret contenu dans nos données expérimentales. Le taux distillable tient compte de l'efficacité limitée des algorithmes d'extraction de l'information mutuelle I_{AB} contenue dans ces données (actuellement $\beta = 0,87$). Le taux final tient compte de la vitesse limitée des algorithmes de réconciliation (actuellement de 100 000 symboles par seconde). La troisième partie de ce manuscrit est consacrée à l'étude du processus d'extraction de l'information secrète à partir de nos données expérimentales.

Chapitre 10

Intégration et prototypage

Notre expérience de cryptographie a été construite pour être intégrée au réseau quantique du projet européen SECOQC. À cette fin, un haut degré d'automatisation et d'intégration est nécessaire. Cette automatisation, qui permet de gérer simplement et rapidement une grande quantité de données et d'expériences, nous a par ailleurs apporté de l'aisance pour la caractérisation du bruit et elle s'est avérée indispensable à l'étude d'attaques de type interception-réémission.

Cette intégration a trois aspects. Premièrement, l'utilisation avancée de cartes d'acquisition – convertisseurs analogique-numérique – permet des acquisition et modulation en continu. Ensuite, un logiciel d'exploitation peut traiter les données en temps réel. Enfin, les composants matériels de l'expérience doivent être empaquetés sous forme de rack 19 pouces en vue de l'intégration au projet SECOQC, prévue à Vienne en mars 2008.

10.1 Pilotage des cartes d'acquisition

Notre expérience exige l'utilisation de cartes d'acquisition dotées de nombreuses caractéristiques. Nous avons choisi les cartes National Instruments de la série S 611X (6110, 6111 et 6115)

Résolution A/D Les cartes sélectionnées ont une résolution de 12 bits, en entrée comme en sortie. Comme nous l'avons vu chapitre 7, notre signal de détection homodyne est acquis sur une plage de tension entre -500 mV et +500 mV. Le pas de l'acquisition est donc de 0,24 mV, c'est-à-dire bien inférieur au bruit électronique de l'amplificateur de la détection homodyne, d'écart type 2,9 mV. Un raisonnement similaire s'applique à la résolution de la modulation. L'effet de la numérisation a été étudié en détail dans la thèse de Frédéric Grosshans [4].

Nombre d'entrées et sorties Alice doit utiliser deux sorties pour générer une modulation en amplitude et en phase. Bob utilise une entrée pour acquérir le signal électrique issu de la détection homodyne. De plus, nous verrons en section 10.7 qu'ils ont chacun besoin d'une entrée supplémentaire pour calibrer la transmission quantique. Nous envisageons également deux canaux d'acquisition pour la génération de nombres aléatoires chez Alice (section 10.8). Nos cartes d'acquisition possèdent 4 entrées et 2 sorties ; elles couvrent donc tous ces besoins.

Taux de répétition Le taux de répétition maximum de la carte d'acquisition fixe le cadencement global de l'expérience. Ce taux est de l'ordre du Mégahertz pour les raisons suivantes :

- il existe peu de cartes qui rassemblent toutes les caractéristiques nécessaires à l'expérience et dont la cadence dépasse quelques Mégahertz ;
- une acquisition continue demande une intervention logicielle régulière, or les cadences de calcul logiciel ne peuvent raisonnablement pas dépasser ce taux. Notamment, un ordinateur standard ne peut générer des nombres aléatoires et transformer les quadratures choisies en tensions EOM à un taux plus élevé ;
- un taux de 1MHz permet une analyse expérimentale confortable ;
- le taux de clé final est limité par la vitesse de l'algorithme de réconciliation (voir chapitre 12), actuellement limitée à 100 kHz. L'excédent d'impulsions optiques non traitées permet la caractérisation du canal.

Cependant, un taux d'acquisition supérieur est envisageable. La partie optique de l'expérience peut atteindre le gigahertz, l'électronique de la détection homodyne peut atteindre 100 MHz [27]. La conception d'une électronique analogique rapide dédiée reste cependant nécessaire pour l'exploitation des données.

Temps de montée et bande passante La modulation aléatoire du signal implique de fortes variations de la tension de sortie de la carte d'une impulsion à l'autre. C'est pourquoi un court temps de montée est nécessaire. Nos cartes d'acquisition atteignent leur tension cible à 99% en 200 ns. L'instant d'arrivée de l'impulsion signal dans les modulateurs doit être ajusté pour éviter ce régime transitoire.

Pour l'acquisition du signal de détection homodyne, la faible largeur des impulsions (environ 200 ns à la sortie du circuit amplificateur de la détection homodyne) par rapport au taux maximal d'acquisition nécessite une large bande passante d'acquisition. Nos cartes ont une bande passante de seulement 600 kHz. On arrive tout de même à observer le signal, tout en introduisant un bruit négligeable devant le bruit électronique de la détection homodyne.

Acquisitions et modulations simultanées Alice doit moduler l'amplitude et la phase du signal, et Bob doit moduler la phase de l'oscillateur local en même temps qu'il acquiert le signal de détection homodyne. Pour cela, nos cartes d'acquisition ont 4 entrées et 2 sorties qui opèrent de façon simultanée : la carte dispose d'un signal d'horloge qui déclenche toutes les acquisitions et modulations en même temps.

De plus, nos cartes d'acquisition peuvent utiliser le signal de modulation (Analog Output) comme signal de déclenchement pour l'acquisition (Analog Input). Cela permet de synchroniser les processus de modulation et d'acquisition.

Acquisitions et modulations continues Nos cartes d'acquisition permettent l'acquisition de blocs de données en continu, sans temps de latence entre deux blocs. De plus, les données peuvent être transférées entre la carte et le logiciel d'exploitation au cours de l'acquisition. Cette fonctionnalité cruciale n'est pas fournie par les cartes d'acquisition plus rapides (100 MHz). Enfin, les fonctions logicielles fournies par le constructeur pour piloter les cartes (API), possèdent une option qui prévoit l'émission d'un signal au logiciel d'exploitation si ce dernier ne parvient pas à envoyer assez rapidement de nouveaux blocs de données à moduler. Cette fonctionnalité permet d'éviter une ré-émission successive des mêmes données, qui serait fatale pour le processus de cryptographie quantique.

Résolution A/D	12 bits
Plage d'acquisition	± 500 mV, réglable
Taux de répétition	5 à 10 MHz (à diviser par le nombre de canaux)
Temps de montée de la modulation	$<$ 200 ns
Bande passante en entrée	600 kHz
Nombre d'entrées/sorties	4/2

TAB. 10.1: Tableau récapitulatif des caractéristiques des cartes d'acquisition National Instruments PCI 611X.

Déclenchement externe Nous utilisons deux systèmes d'acquisition indépendants pour Alice et Bob. Pour cela, au moins une des cartes d'acquisition doit pouvoir utiliser un signal d'horloge externe. En pratique, nous utilisons le signal TTL issu du générateur d'impulsions utilisé pour la création des impulsions optiques comme horloge maîtresse pour la carte d'acquisition d'Alice. L'observation d'un signal optique permet de synchroniser la carte de Bob (voir section 10.7).

Compteur Les cartes d'acquisition ont une fonction de compteur, qui numérote les impulsions en entrée et sortie depuis le début de l'acquisition. Ce compteur permet de positionner les blocs acquis et émis les uns par rapport aux autres. Ainsi, il est possible de synchroniser Alice et Bob.

10.2 Protocole de communication quantique

La transmission de données entre Alice et Bob nécessite l'établissement d'un protocole de communication. Ce protocole définit la nature et l'enchaînement des données envoyées sur le canal quantique, et doit contenir toute l'information nécessaire à la transmission. Notamment, le protocole doit pouvoir fournir des éléments de synchronisation entre Alice et Bob, doit permettre la détermination de la phase relative entre l'oscillateur local et le signal, et doit éventuellement fournir des indications sur les caractéristiques du canal.

Contrairement à un protocole de communication classique, notre protocole quantique est soumis à des contraintes spéciales :

- Le protocole doit être robuste face au bruit quantique inséparable du signal.
- Le protocole doit intégrer le fait que Bob ne peut mesurer qu'une seule quadrature, ne disposant ainsi que d'une information partielle sur l'information envoyée par Alice, y compris pour des impulsions sondes.
- Le protocole pourra s'adapter en temps réel aux modifications imposées au canal : dérive du gain, du bruit et de la phase.
- Le protocole doit être robuste face à une dérive de la valeur moyenne du signal, due à la dérive lente de l'équilibrage de la détection homodyne.
- Le protocole doit prendre en compte l'absence de signaux de synchronisation externes, hormis l'oscillateur local reçu par Bob.
- Ce protocole doivent pouvoir être réalisés à partir des informations fournies par les cartes d'acquisition utilisées, détaillées en section 10.1.

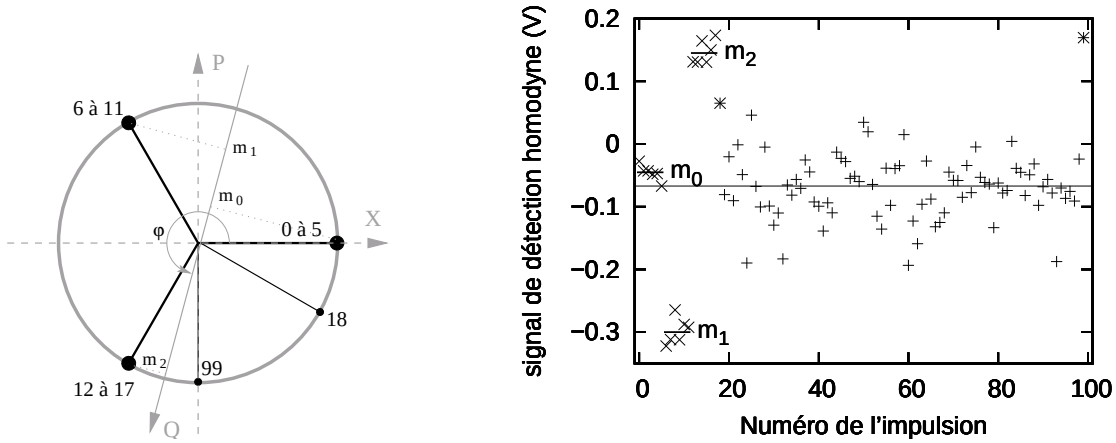


FIG. 10.1: Structure d'un datagramme. La figure de gauche montre la répartition des impulsions test dans le plan complexe. Lorsque Bob sélectionne une phase nulle avec son modulateur de phase, sa quadrature de mesure Q est tout de même déphasée d'un phase ϕ par rapport à la quadrature X qu'Alice sélectionne par phase nulle avec son modulateur de phase. Cette phase (que nous appelons «phase relative») est due à la différence de marche nominale entre les voies signal et oscillateur local. Elle dérive lentement dans le temps, décrivant 2π en une dizaine de secondes quand l'interféromètre est stabilisé. La figure de droite montre la structure temporelle d'un datagramme telle que mesurée par Bob selon la quadrature Q. Les 80 impulsions portant l'information quantique sont représentées par le symbole $+$. Les trois groupes de 6 impulsions test sont représentés par le symbole $\times$. Enfin, le symbole $*$ désigne les deux impulsions test frontières. On note m_i les niveaux moyens de chaque groupe d'impulsion test. Ils permettent de calculer l'équilibrage de la détection homodyne (ligne horizontale) et la phase relative ϕ.

Le signal quantique est entrecoupé d'impulsions test, d'amplitude et de phase déterminées. En référence aux notions employées dans les protocoles réseau standards, nous appelons *datagramme* un ensemble élémentaire d'impulsions regroupant des impulsions test et des impulsions modulées portant l'information quantique. La transmission est une suite de datagrammes adjacents ; aucun signal auxiliaire n'indique le début de la séquence ou ne délimite les datagrammes. En conséquence, l'enchaînement des datagrammes doit être régulier : le récepteur pourra présupposer sa régularité. Nous avons choisi des datagrammes de longueur $r = 100$ impulsions[1], réparties comme suit (figure 10.1) :

- le datagramme commence par trois séquences de six impulsions test d'amplitude maximale (modulateur d'amplitude réglé sur sa transmission maximale), et de phases respectives 0, $2\pi/3$ et $4\pi/3$. Ces impulsions sont numérotées de 0 à 17. Ces valeurs de phases permettent de sonder les caractéristiques de la transmission.
- l'impulsion suivante (numéro 18) a une amplitude maximale et une phase de $11\pi/6 = 4\pi/3 + \pi/2$
- la dernière impulsion du datagramme (numéro 99) a une amplitude maximale et une phase de $3\pi/2 = -\pi/2$. Les phases des impulsions 18 et 99 permettent de délimiter une frontière entre les 18 autres impulsions test et les impulsions utiles. Cette démarcation sera utile à l'algorithme de synchronisation décrit plus loin dans cette section.

[1]L'ensemble des tailles utilisées dans le protocole sont des paramètres librement ajustables du logiciel d'exploitation.

- les 80 autres impulsions du datagramme (numéros 19 à 98) sont les impulsions utiles et reçoivent une modulation appropriée : aléatoire et gaussienne dans le cas d'une distribution quantique de clé, ou régulière pour la caractérisation du dispositif (voir chapitre 7).

Les impulsions test permettent de caractériser le signal reçu par Bob, c'est-à-dire les paramètres du canal. Nous avions déjà vu au chapitre 7 comment calculer les niveaux de bruit de photon et de bruit de phase à l'aide de calculs de variance sur les 18 premières impulsions test. Maintenant, Bob calcule les niveaux moyens m_i de chaque groupe de 6 impulsions test, comme représenté figure 10.1, par moyennage sur 500 datagrammes consécutifs. Ces niveaux permettent de calculer :

- le niveau moyen de la détection homodyne

$$M = \frac{1}{3}\sum_i m_i \tag{10.1}$$

- l'amplitude des impulsions test, qui est l'amplitude maximale du signal autorisée par le mosulateur d'amplitude d'Alice. On peut en déduire la variance de modulation $V_A = \left(\frac{A}{f}\right)^2$ où f est défini au chapitre précédent.

$$A = \frac{2}{3}\sum_i (m_i - M)^2 \tag{10.2}$$

- la phase relative ϕ entre le signal et l'oscillateur local

$$\cos(\phi) = \frac{3}{A}(2m_0 - m_1 - m_2) \tag{10.3}$$

$$\sin(\phi) = \frac{\sqrt{3}}{A}(m_2 - m_1) \tag{10.4}$$

$$\text{puis} \quad \phi\,[\pi] = \tan^{-1}\frac{\sin(\phi)}{\cos(\phi)} \tag{10.5}$$

Notons que la valeur de la variance de modulation ainsi calculée est très sensible à un réglage défectueux des tensions de biais des modulateurs (car le cas échéant, l'amplitude des impulsions test n'est plus maximale). En revanche, les mesures de la moyenne de la détection homodyne, ainsi que de la phase relative, qui ne dépendent que de la position relative des impulsions test, sont très robustes.

Les informations fournies par les impulsions test ne peuvent être utilisées qu'à des fins de caractérisation. En situation cryptographique, seules les informations issues des impulsions utiles peuvent être considérées, car l'espion peut manipuler les impulsions test. Nous utilisons tout de même la mesure de la phase relative pour connaître la quadrature mesurée par Bob dans le référentiel défini par Alice. Cette utilisation n'est pas dangereuse, car toute manipulation de cette phase relative par l'espion est équivalente à une manipulation sur la phase des impulsions signal.

Les impulsions test permettent à Bob de localiser la position des datagrammes dans la séquence de données mesurées. La distribution particulière des impulsions test est

reconnaissable parmi le datagramme. Nous avons développé un algorithme empirique capable de détecter cette distribution, malgré le bruit de photon, et pour une modulation d'impulsions utiles arbitraire, même régulière.

Cet algorithme repose sur l'observation que les 18 premières impulsions du datagramme sont composées de 3 paliers bien distincts de 6 impulsions de valeurs de quadrature mesurée comparables (au bruit de photon près). Sur $r = 100$ points consécutifs de quadrature $Q_i, \quad i \in [0; r[$, Bob calcule la somme diff_i des différences entre points successifs de chaque palier présumé :

$$\begin{aligned} \text{diff}_i &= |Q_{i+1} - Q_i| + |Q_{i+2} - Q_{i+1}| + \ldots + |Q_{i+5} - Q_{i+4}| \\ &+ |Q_{i+7} - Q_{i+6}| + \ldots + |Q_{i+11} - Q_{i+10}| \\ &+ |Q_{i+13} - Q_{i+12}| + \ldots + |Q_{i+17} - Q_{i+16}| \end{aligned} \tag{10.6}$$

Cette somme est moyennée sur 500 datagrammes consécutifs (un bloc, voir section 10.3) pour éliminer le bruit de photon. Quand i coïncide avec le début du datagramme (dit *point d'entrée*, et noté j), on s'attend à une faible valeur de diff_i, car Q est constant (au bruit de photon près, comme illustré figure 10.1) sur chaque groupe de 6 impulsions test. En revanche, si i est entre $j+1$ et $j+20$, le passage d'un groupe d'impulsions test à l'autre conduira à une grande valeur de diff_i. Toutefois, chercher le minimum de diff_i ne suffit pas car, pour de faibles modulations ou des modulations régulières, les impulsions modulées peuvent être assez semblables pour être confondues avec un groupe de 6 impulsions test. Le point d'entrée j des datagrammes est donc caractérisé par une valeur diff_j petite et des valeurs diff_{j+1} et diff_{j-1} grandes. Compte tenu de ces remarques, nous déterminons le point d'entrée en détectant un grand écart entre diff_i et diff_{i+1} :

$$\max_{i \in [0;r[} \quad |\text{diff}_i - \text{diff}_{i-1}| + |\text{diff}_{i+1} - \text{diff}_i| \tag{10.7}$$

Les impulsions 18 et 99 permettent d'assurer un différentiel important entre les paliers extrêmes et les données utiles. En plus de deux années d'utilisation, cet algorithme *ad hoc* n'a jamais défailli.

10.3 Découpage en blocs

Une transmission de clé est divisée en entités indépendantes ou blocs. Chaque bloc est indépendant des autres blocs, et contient une fraction élémentaire de la clé. On considère que les paramètres du canal sont constants à l'échelle du bloc. La taille d'un bloc est de 50 000 impulsions (acquises en 50 ms au taux de 1 MHz), soit 500 datagrammes, comportant 10 000 impulsions test et 40 000 impulsions modulées. Plusieurs contraintes ont conditionné notre choix :

- le temps d'acquisition d'un bloc doit être petit devant les échelles de variation des caractéristiques du canal quantique. Ces caractéristiques sont la dérive de la phase relative entre l'oscillateur local et le signal (dérive typique de 1 rad/s), la transmission et le bruit ajouté du canal (dérives de l'ordre de la seconde) ;
- la taille d'un bloc de données doit être adaptée à la taille des blocs utilisés par l'algorithme de réconciliation. Nous verrons que cet algorithme nécessite une entrée de 200 000 impulsions modulées, soit 5 blocs ;
- la gestion de chaque bloc (écriture de la modulation et sauvegarde des données mesurées, voir section 10.5) nécessite une intervention logicielle, soumise à la latence du système

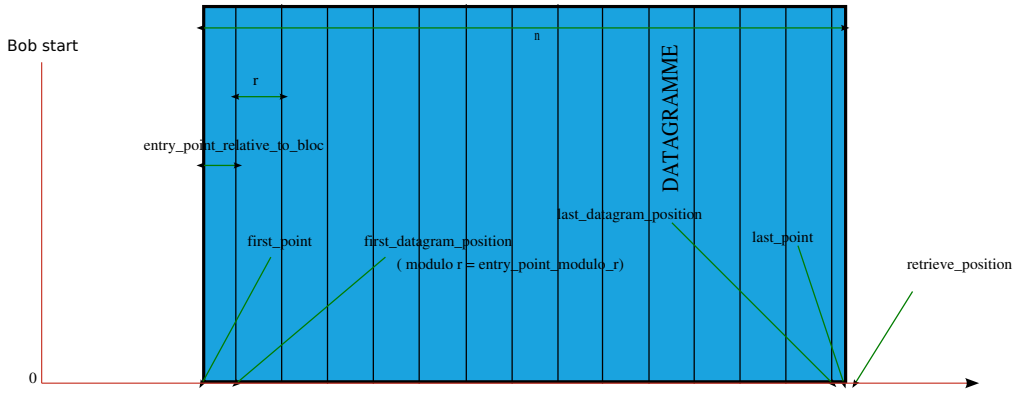


FIG. 10.2: Décomposition d'un bloc de données. Un bloc est composé de 500 datagrammes comprenant 20 impulsions test et 80 impulsions utiles portant l'information quantique. Cette figure illustre l'ensemble des paramètres associés à chaque bloc qui permettent de localiser les datagrammes à l'intérieur du bloc, ainsi que de localiser le bloc parmi la séquence de modulation.

d'exploitation de l'ordinateur de pilotage. Cette latence est due à l'attribution du processeur aux autres processus tournant sur la machine (réseau, interface graphique...). Elle est typiquement de 10 ms, et une marge significative doit être prévue car cette latence n'est pas bornée supérieurement[2]. En pratique, la latence des systèmes d'exploitation courants, sur un ordinateur moderne, dépasse rarement 50 ms ;

- la taille de la mémoire nécessaire à la représentation du bloc doit être raisonnable. Dans notre cas, l'amplitude et la phase de chaque impulsion doivent être mémorisées. Le coût du stockage d'un nombre réel étant de 4 octets (32 bits), la mémorisation d'un bloc nécessite 400 kilo-octets.

Un bloc est accompagné d'un ensemble de caractéristiques qui permettent sa localisation au sein de la séquence d'acquisition. Ces caractéristiques sont représentées figure 10.2. Elles permettent notamment de localiser les datagrammes dans le bloc et de localiser le bloc par rapport à la première impulsion.

10.4 Synchronisation entre Alice et Bob

Les structures de datagrammes (section 10.2) et de blocs (section 10.3) répondent au problème de la synchronisation entre Alice et Bob. En effet, Alice et Bob doivent pouvoir faire correspondre leurs données respectives, sans autre signal que le signal issu du canal quantique. Dans notre protocole de synchronisation, Bob débute sa période de mesure avant qu'Alice ne commence sa modulation et va tenter de détecter le début de la modulation d'Alice selon la procédure présentée dans cette section. Tout comme la détection des impulsions test, la synchronisation doit fonctionner malgré la présence du bruit de photon sur la mesure de Bob.

Alice fait précéder sa modulation d'un bloc vide, composé uniquement d'impulsions test. Ce bloc initial permet de marquer le début de la modulation. L'enchaînement des blocs est représenté figure 10.3 Les blocs sont indexés séquentiellement (Reconciliation Bloc Number)

[2]L'utilisation d'un système temps réel permettrait de borner la latence du système. Pour ces systèmes, on est sûr qu'un processus aura la main assez fréquemment.

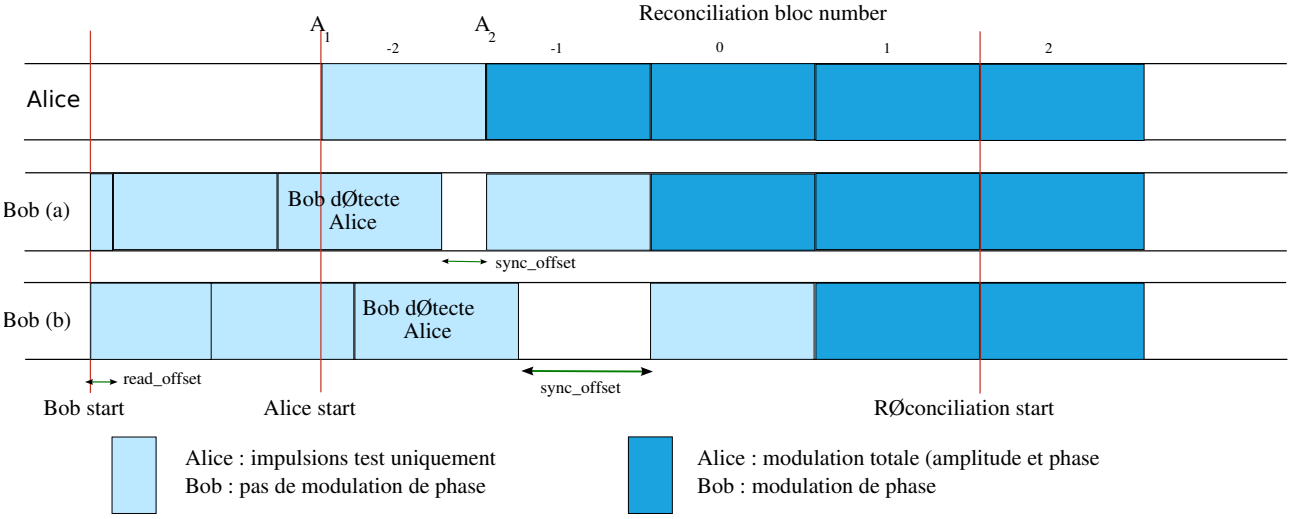


FIG. 10.3: Synchronisation des blocs de données. Cette figure montre la succession des blocs de données au début de la modulation d'Alice. Bob tente de déterminer le début de la modulation d'Alice en détectant le point A_1 (Bob (a)). S'il échoue, il tente ensuite de détecter le point A_2 (Bob (b)).

afin d'être identifiés lors de la réconciliation. Le bloc numéro 1 est le premier bloc correctement modulé ; c'est le premier bloc qui pourra être réconcilié.

À l'aide des impulsions test, Bob détecte le début de la modulation d'Alice. Pour ce faire, nous avons développé un algorithme empirique, qui traite successivement chaque bloc de la façon suivante, jusqu'à ce que Bob détecte Alice :

1. Bob détecte le point d'entrée des datagrammes, selon l'algorithme présenté section 10.2. Bien entendu, si Alice n'a pas commencé sa modulation, la valeur du point d'entrée n'est pas pertinente. En revanche, si le bloc acquis par Bob contient le début des impulsions test (point A_1 sur la figure 10.3), deux situations peuvent se présenter, représentées respectivement par «Bob a» et «Bob b» sur la figure 10.3 :
   (a) si le point A_1 est suffisamment proche du début du bloc de Bob, ce bloc contient assez de datagrammes pour que l'algorithme de détection du point d'entrée des datagrammes fonctionne correctement. Dans ce cas, la procédure de synchronisation continue au point 2.
   (b) si le point A_1 est trop proche de la fin du bloc de Bob, il n'y a pas assez d'impulsions test dans le bloc et l'algorithme de détection des datagrammes échoue. Dans ce cas Bob, acquiert un bloc supplémentaire, contenant le point A_2. Cette fois, le bloc contient 500 datagrammes, et l'algorithme de détection des datagrammes fonctionne. La procédure de synchronisation continue au point 3.
2. Bob parcourt individuellement les datagrammes du bloc de façon séquentielle et calcule la variance des impulsions test de chaque datagramme. Le point A_1 est détecté quand Bob observe un saut significatif de cette variance d'un datagramme à l'autre[3].

[3]Le seuil de ce saut est ajustable. Nous avons choisi une valeur de 3.0. Si ce seuil est trop haut, Bob ne pourra pas détecter les signaux de faible amplitude ; s'il est trop bas, Bob pourra détecter de façon erronée la modulation.

3. De même que pour le cas précédent, Bob parcourt séquentiellement les datagrammes, mais calcule cette fois la variance des impulsions utiles. Le point A_2 est détecté quand Bob observe un saut significatif de cette variance d'un datagramme à l'autre.
4. Les deux procédures précédentes sont appliquées sur chaque bloc jusqu'à ce que l'une d'entre elles aboutisse. Notons que si la première méthode échoue sur le bloc i à cause de l'échec de la détection du point d'entrée, la deuxième réussira sur le bloc $i+1$. Une fois la modulation d'Alice détectée, Bob décale son acquisition pour faire coïncider ses blocs avec ceux d'Alice.

Empiriquement, cet algorithme fonctionne sans erreur si la variance de la modulation du signal reçu par Bob dépasse quelques photons (1 à 2). En dessous de ce seuil, la détection est aléatoire.

Après détection de la modulation d'Alice, Bob commence la modulation de phase de l'oscillateur local. Bob ne doit pas moduler la phase de sa mesure pendant les impulsions test. C'est pourquoi il doit attendre le début de la modulation d'Alice pour entamer sa propre phase de modulation. De plus, Bob en profite pour décaler le début de ses blocs pour les faire coïncider avec les blocs d'Alice (figure 10.3). Cette procédure apporte plus de confort au moment de la réconciliation.

10.5 Gestion des blocs de données

Maintenant que nous disposons d'un protocole de communication quantique, il nous reste à le mettre en œuvre. Pour cela, nous devons définir une procédure de génération de la modulation, de mémoire des blocs de données et de communication avec la carte d'acquisition.

Un module logiciel permet de remplir un bloc avec une modulation arbitraire. Nous avons programmé des modulations constantes, linéaires, aléatoires mais répétitives d'un datagramme à l'autre ou totalement aléatoires gaussiennes dans le plan complexe. Une boucle d'asservissement logicielle permet éventuellement de retrancher à la phase choisie par Bob la phase relative entre le signal et l'oscillateur local mesurée lors des précédents blocs. Cette opération permet de mesurer une quadrature absolue, comme X ou P. Nous avons utilisé cette boucle d'asservissement pour la mesure de deux quadratures orthogonales lors de notre réalisation de l'attaque interception-réémission.
Un second module logiciel permet de faire la conversion entre la modulation choisie et la tension à appliquer aux modulateurs électro-optiques selon les relations établies au chapitre 6.

La tension à appliquer aux modulateurs est ainsi calculée puis envoyée à la carte d'acquisition grâce aux pilotes National Instruments NIDAQmx. La continuité de la modulation est soumise à la capacité de l'ordinateur à enchaîner le processus assez rapidement et sans latence notable : la génération informatique des tensions des modulateurs doit être inférieure au temps d'émission d'un bloc, soit 50 ms pour un taux de 1 MHz. Pour des modulations aléatoires générées par l'ordinateur, nous avons constaté des taux de répétition jusqu'à 1 MHz.

Un module logiciel appelé «Réservoir» ou "Pool" s'occupe de gérer les blocs de données. Il est responsable de la création, de la destruction des blocs ; il fournit des pointeurs

vers les blocs courant, précédent et suivant. Il mémorise les blocs au fur et à mesure qu'ils ont été acquis ou émis et vide périodiquement cette mémoire pour ne pas saturer l'ordinateur hôte[4].

Le réservoir est également responsable de gérer le remplissage des blocs par le module "Modulation" et la conversion des quadratures en tension pour les EOM (voir section 6.4). Après acquisition, il déclenche les algorithmes de détection de datagrammes.

10.6 Programmation et dépendances logicielles

Le logiciel de pilotage est programmé en C++ standard. Il dépend des bibliothèques de pilotage des cartes d'acquisition. Pour vérifier la procédure de traitement des données, nous avons programmé une fausse bibliothèque de pilotage des cartes, qui acquiert un signal fictif de caractéristiques connues (amplitude, phase relative, point d'entrée, excès de bruit...).

Divers logiciels sont utilisés pour compiler et gérer les sources du programme. C'est l'environnement de programmation :

- Le logiciel *Subversion* [54] est un système de versionnement des sources. Il permet de garder trace de l'état du programme à tout instant de son développement.
- Le logiciel *Doxygen* [55] permet de générer le manuel d'utilisation du programme à partir de commentaires insérés dans le code source.
- Le compilateur *GCC* [56] est utilisé pour compiler les sources.
- Nous utilisons *CMake* [57] comme environnement de construction. Il gère l'ensemble du processus de compilation.

10.7 Calibration de la transmission

Nous avons vu au chapitre 7 que le calcul du taux secret I_{BE} nécessitait deux calibrations : la variance de modulation en sortie d'Alice et la variance du bruit de photon (proportionnelle à la puissance d'oscillateur local) au niveau de la détection homodyne. Cette section est dédiée à ces deux mesures.

À l'aide d'un coupleur, nous prélevons une partie du signal d'Alice après modulation pour mesurer son intensité. Cette intensité est mesurée pour chaque impulsion individuelle. Le signal circulant dans le canal de communication a une intensité d'une dizaine de photons par impulsion. Avant multiplexage et atténuation, nous mesurons donc environ 10^4 photons par impulsion. Ce signal est du même ordre que la différence des photo-courants d'une détection homodyne. C'est pourquoi nous amplifions le signal mesuré avec un circuit électronique de détection homodyne (amplificateur de charge et amplificateur secondaire) dépourvu de sa deuxième photodiode. La sortie de la chaîne d'amplification est acquise par la carte d'acquisition d'Alice. Nous pouvons alors tracer la courbe reliant la tension envoyée au modulateur d'amplitude et l'intensité mesurée, reproduite figure 10.4. Cette courbe est la caractéristique transmission – tension du modulateur. On ajuste cette caractéristique par une sinusoïde $a\cos(\omega V+\phi)+b$, dont l'amplitude a est proportionnelle à la variance de modulation. Le coefficient de proportionnalité est obtenu par calibration. Outre l'amplitude de la modulation, cet ajustement met également à disposition la période de la sinusoïde qui mesure la

[4]Nous ne gardons en mémoire que les 20 derniers blocs.

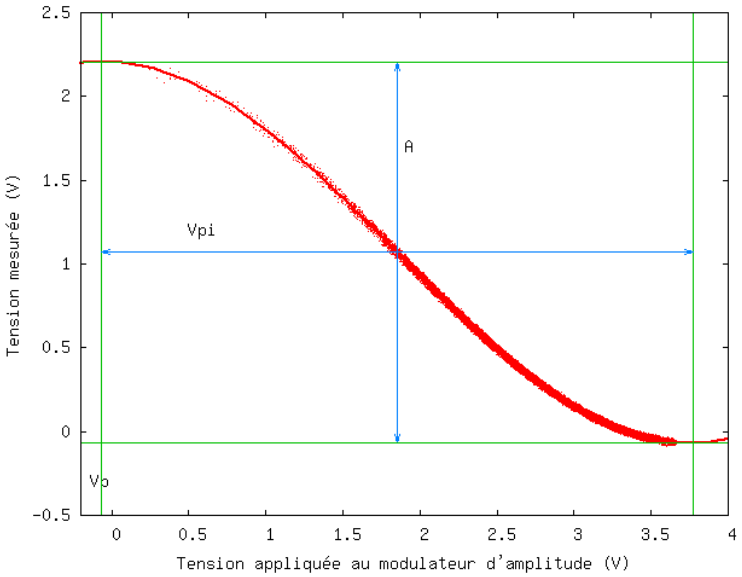


FIG. 10.4: Intensité du signal modulé issu d'un signal de photodiode amplifié, en fonction de la tension envoyée au modulateur d'amplitude. Ces données sont ajustées par une fonction sinusoïdale dont l'amplitude A est proportionnelle à la variance de modulation V_A. De plus, cette fonction fournit les caractéristiques V_π et V_b du modulateur d'amplitude. On observe sur cette figure le bruit électronique de l'amplificateur de charge.

caractéristique $V_\pi = \pi/\omega$ du modulateur, ainsi que la phase ϕ qui mesure la tension de biais $V_b = \phi/\omega$. À l'aide de ces deux valeurs, il est prévu de réaliser une rétroaction logicielle capable de compenser leurs fluctuations.

Nous mesurons l'intensité de l'oscillateur local en observant le signal électrique aux bornes d'une des photodiodes de la détection homodyne. Ce signal est prélevé avant soustraction des photo-courants, il est donc proportionnel à la puissance instantanée de l'oscillateur local. Il nous permet d'abord de régénérer un signal de synchronisation électrique pour le système de Bob. En effet, le seul lien physique entre Alice et Bob est la fibre optique de transmission : Alice ne peut pas envoyer de signal de synchronisation électrique à Bob. Ensuite, nous faisons une acquisition du signal prélevé pour chaque impulsion oscillateur local. Nous calculons alors la valeur moyenne de ces acquisitions. Cette valeur moyenne est une mesure de l'intensité de l'oscillateur local, qui nous permet de calibrer la variance du bruit de photon, selon l'étalonnage décrit en section 7.3. Enfin, nous calculons la dispersion des mesures d'intensité de l'oscillateur local. Si cette dispersion est supérieure au bruit électronique de notre acquisition, alors il est probable qu'Ève tente de manipuler l'intensité de l'oscillateur local afin de camoufler une attaque sur le signal, comme expliqué en section 4.7.

10.8 Génération de nombres aléatoires

La génération de nombres aléatoires est un aspect important de la sécurité de notre dispositif. Pour chaque impulsion, Alice doit disposer de deux nombres aléatoires réels (un pour chaque quadrature). Bob, de son coté, choisit sa quadrature de mesure soit à l'aide d'un bit aléatoire

(choix parmi deux quadratures orthogonales), soit à l'aide d'un nombre aléatoire réel (choix de quadrature continu). Pour un taux de répétition de 1 MHz, du fait de la résolution de 12 bits de la carte d'acquisition, nous devons être capables de générer 24 Mbits aléatoires par seconde. Cette section fait un rapide tour d'horizon des différents générateurs de nombre aléatoires disponibles.

L'estimation de la qualité d'une séquence aléatoire n'est possible qu'avec une séquence infinie. Pour toute situation pratique, les avis divergent sur la façon de qualifier la qualité d'une séquence aléatoire. Le test le plus courant consiste à recenser parmi la séquence aléatoire le nombre de suites consécutives de bits identiques. Il existe diverses méthodes pour améliorer l'aléa d'une séquence. Notamment, une séquence biaisée, c'est-à-dire avec plus de 1 que de 0, peut être équilibrée en associant chaque groupe de deux bits selon la transformation suivante

Entrée		Sortie
00	→	rien
01	→	0
10	→	1
11	→	rien

Générateurs pseudo-aléatoires

Les générateurs pseudo-aléatoires proposent des nombres en apparence aléatoires, mais qui sont en réalité déterministes. Beaucoup de ces générateurs utilisent des algorithmes mathématiques de congruence de la forme :

$$x_{n+1} = ax_n + b \quad [r] \tag{10.8}$$

Le choix des paramètres de la congruence est crucial pour la qualité de l'aléa produit, notamment concernant la période de répétition du générateur. Les générateurs standards proposés par les bibliothèques de programmation sont souvent de mauvaise qualité[5]. Le référence [58] propose des paramètres a et b acceptables. Notons que les polynômes irréductibles que nous rencontrerons dans le chapitre 14 sont à la base de générateurs pseudo-aléatoires de très grande période.

Toutes les expériences décrites dans ce manuscrit ont été réalisées avec un générateur pseudo-aléatoire. Il nous a permis de générer des modulations reproductibles, utiles à des fins de caractérisation, par exemple pour que Bob puisse aisément reconstruire la modulation envoyée par Alice. Notre générateur a été programmé par Gilles Van Assche, et se fonde sur la fonction de hachage SHA ("Secure Hash Algorithm"). Bien entendu, il est hors de question d'utiliser ce générateur hors du laboratoire : il serait vain de vouloir dépasser la sécurité proposée par l'algorithme de cryptage SHA avec un dispositif quantique dont l'aléa est justement généré par SHA.

[5]Le générateur de la bibliothèque C standard a une période de 32 768. Il ne convient à aucun usage scientifique, que ce soit pour des simulations de Monte-Carlo ou pour une transmission quantique.

Générateurs entropiques Certains générateurs de nombres aléatoires utilisent une source chaotique. Par exemple, certains utilisent l'ensemble des valeurs des registres d'un ordinateur depuis son premier allumage, les paquets TCP/IP arrivant sur le port ethernet, les frappes du clavier. Ces sources chaotiques doivent ensuite être traitées pour éliminer tout caractère non aléatoires (biais, ...). Du fait de ce traitement, il faut se méfier de la qualité de ces générateurs.

Générateurs physiques Des générateurs de nombres aléatoires produisent de l'aléa à partir de bruits physiques, par exemple le bruit électronique dans des diodes Zener. Pour éliminer les perturbations extérieures, on utilise la différence de tension entre deux diodes mitoyennes. Beaucoup d'ordinateurs de bureau sont actuellement dotés d'un tel générateur. Sous le système d'exploitation linux, ces générateurs sont accessibles via le périphérique /dev/hwrandom. Ces générateurs peuvent produire des séquences aléatoires de bonnes qualité, mais à un taux insuffisant, de l'ordre de 10 kb/s.

Générateurs quantiques Aux grands maux, les grands remèdes. Un système de distribution quantique de clé se doit de disposer d'un aléa quantique. La société suisse idQuantique propose à la vente des cartes PCI capables de générer un aléa quantique jusqu'à 16 Mb/s. Ces générateurs reposent sur l'incertitude sur la voie de sortie d'un photon arrivant sur une lame séparatrice. Pour faire fonctionner notre système de distribution quantique de clé au taux de 1 MHz, il nous faudrait acquérir une carte pour Bob, et deux cartes pour Alice.

Le bruit de photon mesuré par une détection homodyne est une source aléatoire quantique gaussienne. Au lieu d'utiliser un générateur de nombres binaires transformés en modulation gaussienne par une fonction mathématique, il est plus intéressant de disposer directement de nombres quantiques gaussiens. Pour cela, nous envisageons l'utilisation d'une détection homodyne mesurant le bruit quantique d'un mode vide. Le dispositif est simple : une source laser est injectée dans un coupleur 50/50. Les intensités des deux voies de sortie du coupleur sont mesurées par des photodiodes raccordées à un circuit de détection homodyne. Nous pouvons ensuite profiter des voies d'entrée libres de notre carte d'acquisition pour acquérir ce signal et disposer de nombres aléatoires gaussien pour notre logiciel de modulation. Moyennant une renormalisation, ces nombres peuvent être directement utilisés pour choisir les quadratures X et P envoyées par Alice.

10.9 Intégration en rack 19 pouces

Le projet SECOQC vise à construire un réseau quantique composé de 6 liens de natures différentes, entre quatre nœuds distants. Ce réseau, dont la topologie est schématisée figure 10.5, sera implanté à Vienne en mars 2008. L'un de ces liens sera assuré, nous l'espérons, par notre système de distribution quantique de clé avec des états cohérents. À cette fin, notre système doit être intégré dans un boîtier au format 19 pouces. Nous avons d'ores et déjà regroupé l'ensemble de l'optique de Bob dans un tel boîtier, afin de vérifier la stabilité de notre interféromètre dans ces conditions. Un autre compartiment sera dédié à l'électronique d'acquisition et contiendra un ordinateur capable de traiter nos données et de fournir une clé secrète aux organes de plus haut niveau du réseau. L'établissement d'une clé secrète à partir de nos mesures quantiques fait l'objet de la troisième partie de ce manuscrit.

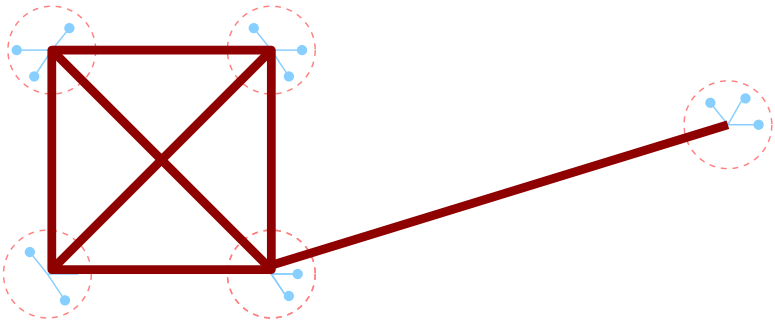


FIG. 10.5: Topologie du réseau quantique prévu par le projet SECOQC. 6 liens principaux ("Quantum Backbone") relient 4 nœuds répartis dans la ville de Vienne, à des distances variant entre 12 km et 35 km. Un lien externe de 80 km est également prévu. Éventuellement, d'autres liaisons fonctionnant à plus faible distance ("Quantum Access Network") peuvent se connecter au réseau principal. Des protocoles réseau sont en cours d'élaboration pour gérer les clés produites par chaque lien.

Troisième partie

Distillation d'une clé secrète

Chapitre 11

Introduction

À l'issue de l'échange quantique, Alice et Bob possèdent chacun une chaîne de variables aléatoires réelles, distribuées avec une probabilité gaussienne. Pour Bob, cette chaîne est directement l'ensemble de ses mesures homodynes, normalisées en unités de bruit de photon. Alice, de son côté, possède les deux quadratures X_A et P_A de chaque état cohérent envoyé. Alors Bob révèle publiquement son choix de quadrature (X ou P) pour chaque mesure. Ainsi Alice forme sa chaîne de variables réelles en ne conservant que la quadrature qui a été mesurée par Bob. L'autre quadrature reste inutilisée. Nous avons vu au cours des chapitres précédents que ces deux chaînes présentent les propriétés suivantes :

1. Les chaînes d'Alice et de Bob sont corrélées. Le théorème de Shannon nous apprend qu'il est possible d'en extraire au plus I_{AB} bits communs : I_{AB} est l'information mutuelle contenue dans les corrélations.
2. Les relations de Heisenberg sur les variances conditionnelles dans le cas de la réconciliation inverse, écrites en section 2.4, nous apprennent que l'espion connaît au plus I_{BE} bits parmi les I_{AB} bits.

Suite à ces deux remarques, l'obtention d'une clé secrète se déroule en deux étapes classiques (figure 11.1) de traitement des données gaussiennes :

1. *La réconciliation* tente de générer I_{AB} bits identiques pour Alice et Bob à partir des données gaussiennes corrélées. Cette étape fait face à deux problèmes : d'abord, il faut définir un système de discrétisation permettant de transformer nos variables continues en variables discrètes, ou mots binaires. Ensuite, il faut corriger les erreurs introduites par le bruit du canal quantique sur ces variables discrètes. Pour ce faire, Alice et Bob utilisent des codes correcteurs d'erreurs, et disposent d'un canal classique public authentifié [59] par lequel ils peuvent s'échanger I_{rev} bits d'information. Comme le canal classique n'est pas sécurisé, Ève acquiert aussi cette information de I_{rev} bits, dits «bits révélés». En pratique, les algorithmes de réconciliation échouent à extraire toute l'information disponible dans les corrélations, et n'exploitent finalement qu'une fraction β de l'information I_{AB} initialement disponible. On appelle β l'efficacité de la réconciliation. On note b le nombre de bits produits par la réconciliation.
2. L'*amplification de confidentialité* [60] permet à Alice et Bob d'obtenir $\Delta I = \beta I_{AB} - I_{BE}$ bit totalement inconnus de l'espion à partir des b bits sans erreur produits par la réconciliation. On appelle parfois cette étape la *distillation* d'une clé secrète car elle permet d'obtenir un nombre ΔI de bits parfaitement secrets à partir d'un nombre b plus

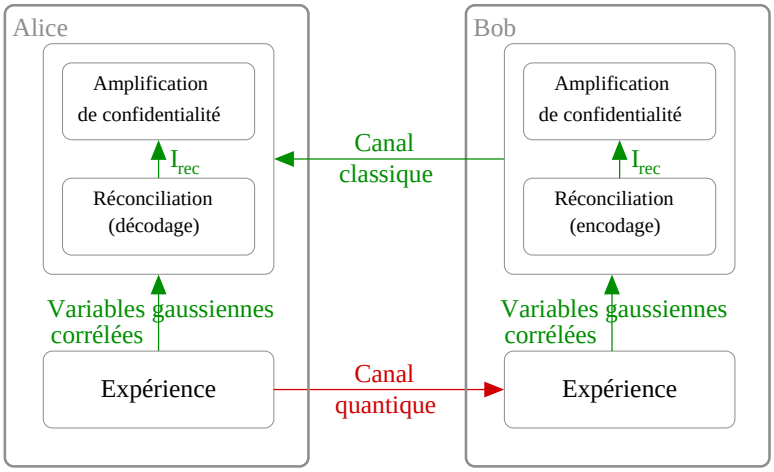


FIG. 11.1: Étapes du processus de génération d'une clé secrète à partir des données expérimentales.

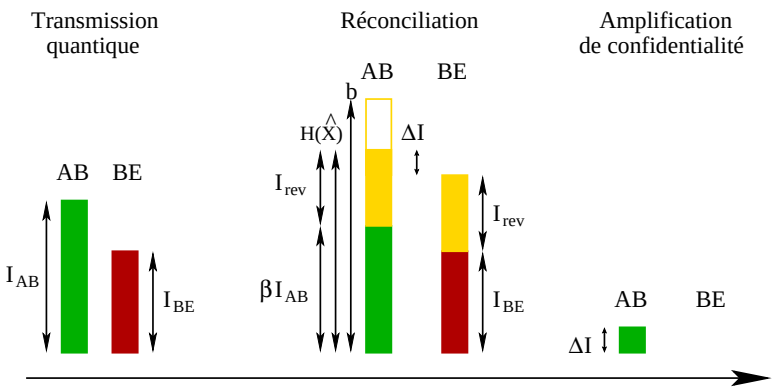


FIG. 11.2: Histogrammes représentant les informations échangées au cours du processus de réconciliation et de distillation.

grand de bits partiellement connus de l'espion. Cette étape est réalisée par l'application d'une fonction de hachage.

On peut établir un bilan de l'information échangée entre Alice et Bob. D'abord, βI_{AB} bits sont extraits de la transmission quantique. Ensuite, I_{rev} bits sont échangés par le canal classique. Ces deux échanges d'information permettent à Alice et Bob de partager un ensemble de mots binaires, qu'on peut voir comme des réalisations d'une variable aléatoire discrète $\hat{X}$[1]. Nous avons vu que le nombre de bits nécessaires pour spécifier une variable aléatoire $\hat{X}$ est son entropie $H(\hat{X})$. Ainsi, l'information échangée est égale à l'entropie $H(\hat{X})$. En pratique, le processus de discrétisation introduit une redondance dans la variable discrétisée, et le nombre de bits b produit par la réconciliation est supérieur à l'entropie $H(\hat{X})$. Finalement, notre bilan informationnel s'écrit :

$$b > H(\hat{X}) = \beta I_{AB} + I_{rev} \tag{11.1}$$

Les informations mises en jeu dans les processus de réconciliation et de distillation sont symbolisées par la figure 11.2.

[1]Dans cette partie du manuscrit, un chapeau $(\hat{\cdot})$ désigne une variable discrète

Le processus d'obtention d'une clé secrète est soumis à plusieurs contraintes. D'abord, les usages du canal classique doivent être modérés. On ne peut envisager des taux supérieurs à ceux autorisés par les liaisons standards (typiquement le Megabit par seconde). Ensuite, et c'est là-dessus que nous allons concentrer nos efforts, la génération de la clé doit être rapide. À ce jour, le temps nécessaire pour réconcilier une clé est le facteur limitant pour le taux secret : le taux avec lequel nous réconcilions les blocs de données est environ un ordre de grandeur inférieur au taux de répétition de l'expérience.

Chapitre 12

Réconciliation et codes correcteurs d'erreurs

À l'issue de l'échange quantique, Alice et Bob possèdent deux chaînes de variables gaussiennes corrélées. Ces corrélations ne sont pas parfaites : le signal envoyé par Alice est entaché du bruit de photon, et ces corrélations sont encore dégradées par le bruit ajouté introduit par l'espion (ou le canal de communication quantique). Afin d'obtenir une chaîne de bits sans erreur, il est nécessaire d'établir un protocole de «réconciliation» visant à discrétiser nos variables gaussiennes, et à en éliminer les erreurs.

Pour arriver à leur fin, Alice et Bob disposent, en parallèle du canal quantique, d'un canal classique sans erreur (par exemple un lien ethernet). Ce canal permet d'échanger de l'information nécessaire à la correction d'erreurs. Cet échange est doublement contraint :

- Le canal classique n'est pas sécurisé. Toute information y transitant est considérée comme parfaitement connue de l'espion.
- Le taux de clé secrète $\Delta I = \beta I_{AB} - I_{BE}$ dérivé chapitre 2 a été calculé dans le cadre de la réconciliation inverse, pour laquelle la chaîne de Bob sert de référence à la clé. C'est pourquoi tout échange d'information sur le canal classique doit se faire de façon *unidirectionnelle*, de Bob vers Alice, pour éviter de construire une clé qui proviendrait de l'information envoyée par Alice plutôt que de l'information reçue par Bob[1].

Dans le processus de réconciliation, nous utiliserons deux types de codes correcteurs d'erreurs unidirectionnels couramment employés dans les télécommunications pour leurs bonnes efficacités :

- les codes LDPC ("low density parity check"), qui consistent à faire des calculs de parités sur les bits erronés que l'on doit corriger ;
- et dans une moindre mesure les turbo codes, qui atteignent de bonnes performances en utilisant plusieurs codes en parallèle (voir annexe A).

Ce chapitre introduit les concepts que l'on rencontre dans la manipulation de ces codes, et montre comment on les adapte à la réconciliation de variables gaussiennes corrélées. Ensuite, nous aborderons brièvement le principe des algorithmes de décodage. Enfin, nous traiterons du décodage multi-niveaux, qui nous permet d'extraire l'information de nos variables continues.

[1]Les premières réalisations de la réconciliation de variables continues [14] utilisaient des protocoles bidirectionnels. Des aménagements spécifiques [7] devaient alors être considérés dans le décompte de l'information accessible à l'espion, au détriment de l'efficacité du processus.

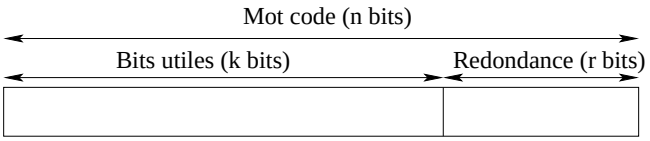


FIG. 12.1: Représentation d'un mot code de n bits contenant k bits d'information et r bits de redondance. Cette représentation est symbolique : la plupart du temps, les k bits utiles ne se retrouvent pas explicitement à l'intérieur du mot code. Toutefois, on appelle «code systématique» un code pour lequel on retrouve les k bits utiles dans le mot code. Ce sont ces codes systématiques que nous pourrons utiliser dans notre situation de réconciliation.

12.1 Codes correcteurs d'erreurs

Un code correcteur d'erreurs est un algorithme permettant une transmission sur un canal avec un taux aussi proche que possible de la limite de Shannon définie à la section 1.3.

Pour transmettre sans erreur k bits sur un canal bruité, il faut introduire une redondance de r bits, soit un total de n bits envoyés ($n = k + r$, figure 12.1). On parle alors d'un code (n, k) de taux $R = \frac{k}{n}$. La redondance r doit être choisie pour que le taux du code soit égal à la capacité du canal. On remarque que le taux du code est le nombre de bits utiles échangés par bit transmis dans le canal de communication : un code correcteur d'erreurs parfait peut corriger les erreurs avec une taux égal à la capacité du canal. En pratique, les codes n'atteignent pas la limite de Shannon et on doit choisir un taux inférieur à la capacité.

12.2 Exemples de codes correcteurs d'erreurs

L'encodage est la transformation du mot de k bits à transmettre en un mot code de n bits. On distingue deux grandes familles de codes :

1. Les codes par blocs traitent le flux de données entrant par blocs indépendants de k bits. Une fois le bloc encodé, l'encodeur traite le bloc suivant. On appelle alors «code» l'ensemble des 2^k mots code de n bits valides. Si ce code est un espace vectoriel (*i.e.* la somme[2] de deux mots code est un mot code), on parle de «code linéaire».

2. La seconde classe regroupe les codes dont l'encodeur est doté d'une mémoire. L'encodage se fait de façon continue, la sortie de l'encodeur dépend du bit entrant, mais aussi de l'état interne de l'encodeur. On appelle ces codes de façon générique «codes convolutifs», puisque la sortie de l'encodeur dépend de la valeur du bit d'entrée et de celle de ses prédécesseurs. Nous verrons un exemple de code convolutif dans l'annexe A.

Le codage par répétition est le plus simple des codes par blocs. Il consiste à répéter n fois chaque bit. Ce code est linéaire ; sa performance est très limitée.

Exemple. Redondance ternaire : $k = 1, n = 3, r = 2, R = 1/3$. Le code est l'ensemble $\{000, 111\}$. Il est capable de détecter et de corriger une erreur par bloc.

[2]Pour les nombres binaires, on appelle somme l'opération logique «ou exclusif», et produit l'opération logique «et».

Codes "parity checks"

Les codes "parity check", dont font partie les codes LDPC, sont des codes par blocs linéaires faisant intervenir des calculs de parité. Il en existe deux utilisations. La première consiste à calculer r parités sur les k bits à transmettre, puis à envoyer l'ensemble des $n = r+k$ bits par le canal. L'encodage est trivial, puisqu'il s'agit uniquement de simples calculs de parité. Le code par redondance en est un exemple. Pour les canaux peu bruités, un encodage simple consiste à calculer le bit de parité d'un bloc de taille idoine ($r = 1, n = k+1, R = \frac{k}{k+1}$). Tous ces codes sont systématiques, puisque les k bits d'origine se retrouvent inchangés dans le mot code. Il est pratique de représenter un ensemble de calculs de parité sous forme d'une matrice binaire H possédant k colonnes et r lignes, chaque ligne i étant remplie de "1" aux index des bits intervenant dans le calcul de parité i. Ainsi, le calcul des r parités se fait simplement en multipliant la matrice H par le vecteur d'entrée. Dans cette multiplication matricielle, l'addition et la multiplication s'entendent modulo 2. Le taux R_1 associé à cette matrice H est donc

$$R_1 = \frac{k}{n} = \frac{\text{nombre de colonnes de } H}{\text{nombre de colonnes} + \text{nombre de lignes}}. \tag{12.1}$$

La seconde utilisation des codes "parity check" consiste à définir les mots code N comme les mots de n bits vérifiant r contraintes de parité fixées (par exemple 0) ; elle est couramment représentée par l'équation matricielle :

$$H \cdot N = 0 \tag{12.2}$$

où H est une matrice binaire qui possède n colonnes, et r lignes indépendantes. Alors que l'encodage de la première catégorie de codes consistait à calculer des parités sur des bits utiles, les parités sont ici fixées à 0, et ne sont plus transmises sur le canal. L'encodage n'est pas immédiat, car l'équation matricielle ne fait qu'indiquer les mots code valides, sans explicitement fournir la transcription entre les k bits à encoder et le mot code correspondant. Pour procéder à l'encodage, il est nécessaire de définir une matrice G associant chaque entrée K de k bits à un mot code N valide de n bits, c'est-à-dire vérifiant les contraintes de parité imposées par H, selon l'opération $N = G \cdot K$. Il existe une méthode explicite pour calculer G à partir de H, que nous ne détaillerons pas ici. Il est cependant important de remarquer que, pour les codes dont les matrices H et G sont trop grandes pour être explicitées (voir les codes LDPC, section 12.5), il existe des méthodes algébriques d'encodage, de complexité modérée, qui ne nécessitent pas l'explicitation de G [61]. Cette deuxième utilisation des codes "parity check" n'est pas systématique car on ne retrouve pas, en général, les k bits d'origine dans le mot code correspondant. Pour une matrice H donnée, le taux associé à cette matrice est maintenant :

$$R_2 = \frac{k}{n} = \frac{\text{nombre de colonnes} - \text{nombre de lignes}}{\text{nombre de colonnes}}. \tag{12.3}$$

Malgré la simplicité de son encodage, le premier type de codes est peu utilisé en pratique, car les algorithmes de décodage usuels imposent que les parités soient transmises sur un canal sans erreur bien souvent non disponible. On utilise donc couramment le deuxième type dont les parités sont fixées, malgré la complexité de son encodage. Cependant, la réconciliation utilisée en cryptographie quantique autorise la transmission sans erreur de bits de parité sur un canal classique. Nous profiterons donc de cette première utilisation systématique, et de son encodage simple.

Taille du code			
Taille d'un mot code n	400380	40000	40000
Redondance r	380	6320	28512
Bits utiles $k = n - r$	400000	33680	11488
Taux du code $R = k/n$	0.99905	0.842	0.287
Capacité de correction			
Erreurs corrigibles e	20	400	2000
Taux d'erreur corrigible $p = e/n$	$5 \cdot 10^{-5}$	0.01	0.05
Limite de Shannon $C = 1 - h(p)$	0.9992	0.919	0.714
Efficacité R/C	0.9998	0.916	0.402

TAB. 12.1: Performance des codes BCH pour un canal binaire symétrique sur lequel un mot code subit des erreurs avec une probabilité p. Le théorème de Shannon nous apprend qu'un code optimal capable de corriger ces erreurs est de taux $C = 1-h(p)$, avec $h(x) = -x\log_2(x)-(1-x)\log_2(1-x)$. L'efficacité d'un code correcteur de taux R capable de corriger ces erreurs est le rapport R/C. Les codes BCH affichent une bonne efficacité pour les très faibles taux d'erreur. Par exemple, ils peuvent corriger de façon déterministe 20 erreurs parmi $k = 400000$ bits utiles, à une distance négligeable de la limite de Shannon. Pour ces taux d'erreur, les algorithmes d'encodage et de décodage durent une fraction de seconde sur un ordinateur de bureau. En revanche, dès que le taux d'erreur devient de l'ordre de 0.01, l'efficacité commence à se dégrader.

Terminons notre étude élémentaire des codes "parity check" en calculant quelques mots code du célèbre code inventé par Hamming en 1948. Il est défini par la matrice H :

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{12.4}$$

1. Dans la première utilisation, systématique (peu usuelle), un mot d'entrée K a $k = 7$ bits, par exemple $K = 1101101$. On calcule les $r = 3$ parités associées : $P = H \cdot K = 101$. On obtient alors un mot code de $n = 10$ bits : 1101101 101. Le code est de taux $R_1 = 7/10$.
2. Dans la deuxième utilisation, non systématique (usuelle), le mot $N = 1101001$ de $n = 7$ bits est un mot code car $H \cdot N = 0$. Le code est de taux $R_2 = 4/7$.

Codes BCH

Les codes BCH, nommés d'après les initiales de leurs inventeurs Bose, Ray-Chaudhuri et Hocquenghem, sont des codes correcteurs binaires d'une complexité faible utilisés pour le décodage déterministe d'un petit nombre d'erreurs. Ils reposent sur la notion de corps de Galois, que nous découvrirons au chapitre 14. Le décodage LDPC utilisé par le processus de réconciliation n'est pas déterministe : il laisse quelques erreurs, typiquement une dizaine sur 400 000 bits décodés, que nous éliminons grâce à un code BCH. Pour ces faibles taux d'erreur, les codes BCH sont très performants[3] (voir tableau 12.1), ils ne dégradent donc

[3]Dans la littérature traitant des codes correcteurs d'erreurs, les performances d'un code se mesurent par l'écart en dB entre la caractéristique du code et la limite de Shannon sur un graphe représentant le taux d'erreur final en fonction du rapport signal à bruit. Ici, nous préférons parler en termes d'erreurs corrigibles par le code, notion plus accessible au profane, et directement applicable à notre utilisation des codes BCH.

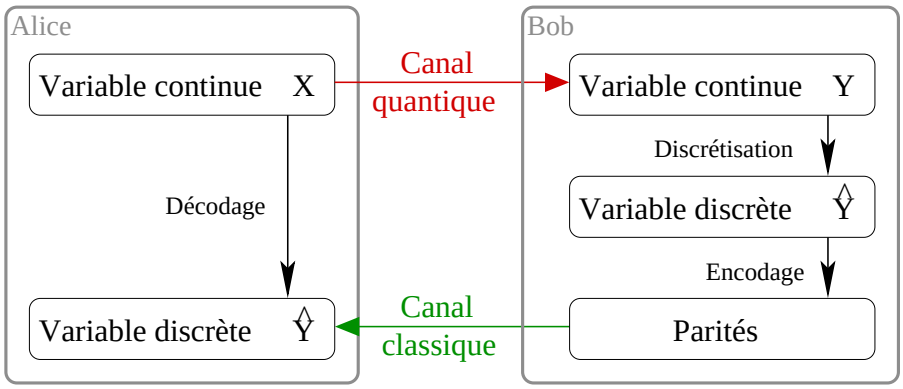


FIG. 12.2: Schéma résumant les étapes de la réconciliation de variables gaussiennes corrélées X et Y. D'abord, Bob discrétise ses variables réelles en nombres binaires notés $\hat{Y}$. Ensuite, Bob calcule les parités associées à sa chaîne de bits $\hat{Y}$ (processus d'encodage). Ces parités sont transmises à Alice par le canal classique. À l'aide de sa chaîne de variables réelles X et des parités, Alice est capable de restituer la chaîne $\hat{Y}$ de Bob (processus de décodage).

pas sensiblement l'efficacité de la réconciliation.

12.3 Réconciliation et "side information"

Après ces remarques générales sur les codes correcteurs d'erreurs, nous allons explorer dans cette section la configuration spécifique de la réconciliation de variables gaussiennes corrélées. Nous avons vu en introduction que la réconciliation se divisait en plusieurs étapes :

1. Alice et Bob possèdent chacun une chaîne de variables continues notées X pour Alice et Y pour Bob. Bob transforme chaque valeur réelle en un nombre binaire $\hat{Y}$, composé d'un ou plusieurs bits. C'est l'étape de discrétisation des données.
2. Bob encode l'ensemble des bits obtenus avec un code systématique, qui génère un nombre r de bits de parité. Comme la valeur de la discrétisation est imposée par la mesure du signal issu du canal quantique, il est nécessaire que le code soit systématique pour que l'ensemble des bits discrétisés et des bits de parité forment un mot code valide. Ainsi, nous pourrons par exemple utiliser les codes "parity check" dans leur première utilisation systématique ou les codes convolutifs systématiques pour réconcilier nos données.
3. Bob envoie à Alice ces r bits de parité par le canal classique sans erreur.
4. À l'aide des bits de parité reçus et des valeurs X de ses données, Alice retrouve la chaîne discrétisée $\hat{Y}$. Alice et Bob partagent maintenant deux chaînes de bits identiques : c'est la fin de l'étape de réconciliation. Notons qu'au cours du processus de réconciliation, Alice ne discrétise pas sa variables continue X, mais cherche la valeur la plus probable de $\hat{Y}$ connaissant sa variable continue X.

Nous résumons le processus de réconciliation par un diagramme représenté figure 12.2. Une représentation abstraite équivalente est introduite dans [8]. Elle prend pour origine les variables discrètes $\hat{X}$ obtenues par discrétisation des variables réelles d'Alice et les variables $\hat{Y}$, reliées par un canal classique sans perte transmettant les bits de parité, et un canal bruité, composition des opérations de discrétisation et de transmission par le canal quantique. Cette représentation

est avantageuse du point de vue théorique, car elle transpose notre problème dans le cadre standard des codes correcteurs d'erreurs. Elle permet de dériver rigoureusement les relations que nous établirons dans cette section, et de trouver des optimisations des codes correcteurs d'erreurs (voir chapitre 13).

La dissymétrie entre l'information transmise par le canal quantique *bruité* et les bits de parité transmis par le canal classique *sans erreur* fait de la réconciliation une configuration de décodage particulière : on parle de décodage avec "side information". Dans cette configuration, le décodeur tente de retrouver le mot code encodé grâce aux bits de parité et grâce à une source d'information auxiliaire (le canal quantique), appelée "side information" [62]. Dans cette configuration, on peut définir le taux[4]

$$R_s = 1 - \frac{r}{k}, \tag{12.5}$$

où k est le nombre de bits utiles à l'entrée de l'encodage, et r est le nombre de bits de parité produits par l'encodage. Si aucune parité n'est nécessaire, alors $R_s = 1$; à l'opposé, si on envoie autant de parités que de bits d'information (auquel cas il ne reste plus d'information nette transmise sur le canal bruité), $R_s = 0$. On montre simplement que les taux R et R_s d'un code sont reliés par[5] :

$$R_s = 2 - \frac{1}{R}. \tag{12.6}$$

Avec la notation que nous avons introduite, nous pouvons reformuler simplement l'expression de l'efficacité de la réconciliation β rencontrée dans l'introduction (équation 11.1). Notons N le nombre de bits issus de la discrétisation d'une variable continue (ou symbole), c'est-à-dire le nombre de bits par symbole du canal quantique en entrée du code correcteur, et P le nombre de parités générées par symbole. Alors le nombre de bits par symbole révélés sur le canal classique est $I_{rev} = P = N(1 - R_s)$. On peut donc exprimer l'efficacité de la réconciliation :

$$\beta = \frac{H(\hat{X}) - N(1 - R_s)}{I_{AB}}, \tag{12.7}$$

où toutes les grandeurs s'expriment en bits par symbole. Notons qu'à discrétisation fixée, le calcul de l'efficacité ne fait intervenir que le taux du code utilisé. La maximisation de l'efficacité consiste donc en la recherche d'un code de taux le plus élevé possible capable de décoder les données binarisées d'Alice.

Maintenant que nous avons décrit le schéma général de la réconciliation, nous allons explorer quelques aspects du décodage des codes LDPC. Ensuite, nous appliquerons ces concepts à la réconciliation.

[4]Notation personnelle.

[5]Nous utiliserons des codes de type "parity check" (LDPC) dans leur utilisation systématique. Nous considérerons donc le taux $R_s = 2 - \frac{1}{R_1}$ associé à l'utilisation systématique d'une matrice H. Fortuitement, ce taux R_s n'est autre que le taux R_2 associé à l'utilisation non systématique habituelle de cette même matrice H. Ainsi, quand nous parlerons d'un code LDPC de taux R_s, ce taux coïncidera avec le taux usuel du code. Il n'en va pas de même pour les codes convolutifs.

12.4 Décodage mou

Revenons un instant à une situation de codage traditionnelle. Après encodage, un mot code est traduit en symboles physiques par une étape de modulation. Par exemple, on peut attribuer le niveau -1 d'un signal au bit 0, et le niveau $+1$ au bit 1. Cette modulation est une modulation binaire. Nous aborderons dans la section 12.6 des modulations plus complexes qui nous amèneront à la modulation propre à nos protocoles quantiques.

Le canal de communication introduit ensuite une perturbation sur le signal modulé. Par exemple, le canal symétrique binaire ("Binary Symmetric Channel", ou BSC) intervertit les symboles associés aux bits 0 et 1 avec une probabilité e. Un code correcteur adapté à ce type de canal, ou «code correcteur binaire» aura donc pour but de corriger ces erreurs. Ce type de canal est rencontré dans les dispositifs de distribution quantique de clé avec des photons uniques, dans lesquels le résultat de la mesure de Bob est binaire. Les protocoles de correction d'erreurs interactifs Cascade [63] ou Winnow [64] sont adaptés à une telle situation. Le décodage du signal issu d'un canal binaire est appelé **décodage dur**, en ce sens qu'il prend en entrée des bits erronés qu'il doit convertir en bits sans erreur.

Dans les télécommunications classiques, le canal binaire symétrique est peu rencontré en pratique. En effet, la transmission de l'information se fait souvent grâce à des variables continues (modulation d'amplitude ou de fréquence d'une onde électromagnétique), et la dégradation du signal au cours de la transmission est donc elle aussi continue, même si l'information initialement introduite dans le canal est sous forme d'une modulation binaire. On rencontre souvent le canal AWGN (Additive White Gaussian Noise), qui ajoute un bruit blanc gaussien au signal d'entrée. Le canal AWGN est l'appellation «télécom» de notre modèle du canal gaussien, que nous avons rencontré au chapitre 2 (à ceci près que le canal AWGN classique n'est pas contraint par des relations de Heisenberg !)

Le *décodage mou* a pour objectif de tirer parti du *caractère continu du bruit ajouté* sur le signal pour faciliter le décodage. En effet, il serait peu efficace d'utiliser un décodage dur sur un canal continu en binarisant les symboles mesurés à la sortie du canal, par exemple en ne gardant que le signe de la mesure. On perdrait en effet l'information portée par l'amplitude du signal. Par exemple, si, pour une modulation binaire en entrée (symboles ± 1), nous mesurons en sortie un signal de $1,5$, il y a de fortes chances pour que le bit transmis soit 1. En revanche, si le signal mesuré est $0,3$, alors nous sommes plus indécis sur la valeur du bit d'origine.

Pour formaliser ce concept, on définit le rapport de vraisemblance ("Log-Likelyhood Ratio", ou LLR) comme le logarithme du rapport entre la probabilité p pour que le bit encodé x_i soit 1, et la probabilité $1-p$ pour qu'il soit 0 :

$$LLR_i = \ln\left(\frac{P(x_i=1)}{P(x_i=0)}\right) \tag{12.8}$$

Le signe du LLR indique la valeur la plus probable, et sa valeur absolue indique la «confiance» que l'on a dans la valeur du bit. Si le LLR est très négatif, on est sûr que le bit est 0 ; s'il est très positif, on est sûr que le bit est 1. S'il est proche de 0, on accorde peu de confiance à la valeur du bit obtenu. Inversement, $P(x_i=1) = \frac{e^{LLR_i}}{1+e^{LLR_i}}$ se déduit du rapport de vraisemblance.

Un décodeur mou possède deux entrées (figure 12.3). La première est destinée aux rapports

FIG. 12.3: On peut représenter un décodeur mou comme une boîte noire caractérisée par ses entrées et sorties. En entrée, on trouve les informations intrinsèque (apportée par la mesure du canal) et les parités (la redondance introduite à l'encodage). En sortie, on obtient l'information *a posteriori*, c'est-à-dire une connaissance améliorée de la chaîne de bits encodée, dont l'apport original du décodage est l'information extrinsèque.

de vraisemblance intrinsèques, qui s'expriment :

$$LLR_i^{\text{intrinsèque}} = \ln\left(\frac{P(y_i|x_i=1)}{P(y_i|x_i=0)}\right). \tag{12.9}$$

ou y_i est la valeur (réelle) du i^{e} symbole mesuré à l'issue du canal de communication. Ces rapports contiennent l'information provenant uniquement du canal de communication, sans prendre en compte l'information apportée par les contraintes imposées par l'encodage. Ils indiquent si le symbole reçu y_i permet davantage de penser que le bit envoyé x_i est 1 ou 0. On calcule simplement les rapports de vraisemblance intrinsèques pour un canal binaire symétrique (BSC) de taux d'erreur e :

$$LLR_i^{\text{intrinsèque}} = \ln\left(\frac{1-e}{e}\right) y_i \quad \text{avec } y_i = \pm 1, \tag{12.10}$$

ou pour une modulation binaire ± 1 transmise par un canal AWGN caractérisé par un bruit de variance σ^2 :

$$LLR_i^{\text{intrinsèque}} = \frac{2}{\sigma^2} y_i \quad \text{avec } y_i \in \mathbb{R}. \tag{12.11}$$

La deuxième entrée est consacrée aux parités, c'est-à-dire à la redondance d'information. Dans les configurations usuelles de décodage, ces parités sont transmises par le canal bruité, au même titre que les bits utiles. La plupart du temps, cette entrée est donc confondue avec l'entrée intrinsèque. Cependant, pour notre situation de réconciliation dans laquelle les parités ne subissent pas d'altération, il est pratique de les distinguer.

On trouve en sortie du décodeur le rapport de vraisemblance *a posteriori*, c'est-à-dire l'estimation du bit x_i à la fin du décodage :

$$LLR_i^{a\ posteriori} = \ln\left(\frac{P(x_i=1|\{y_1,\ldots,y_n\})}{P(x_i=0|\{y_1,\ldots,y_n\})}\right). \tag{12.12}$$

Cette estimation provient du symbole y_i, ainsi que des contraintes du code liant le symbole y_i aux autres symboles. Pour différencier ces deux sources, on définit le rapport de vraisemblance extrinsèque comme l'apport original du décodage : il représente l'information apportée uniquement par les contraintes du code. Il s'exprime :

$$LLR_i^{\text{extrinsèque}} = LLR_i^{a\ posteriori} - LLR_i^{\text{intrinsèque}} \tag{12.13}$$

En utilisant la formule de Bayes et en supposant que le canal n'introduit pas de dépendance entre les symboles $\{y_1, \ldots, y_n\}$, on peut montrer que les LLR extrinsèques s'expriment :

$$LLR_i^{\text{extrinsèque}} = \ln\left(\frac{P(x_i = 1|\{y_k\}_{k\neq i})}{P(x_i = 0|\{y_k\}_{k\neq i})}\right) \tag{12.14}$$

On reconnaît dans cette formule que les LLR extrinsèques expriment la connaissance sur le bit x_i apportée par les symboles autres que y_i, c'est-à-dire à dire la connaissance apportée par les contraintes du codes liant le symbole y_i aux autres symboles transmis. Notons qu'avant décodage, les LLR extrinsèques sont nuls, et donc les expressions des LLR intrinsèques 12.9 et *a posteriori* 12.12 sont égales. En d'autres termes, avant décodage, notre connaissance des bits envoyés provient uniquement des symboles reçus du canal. Pour les étapes suivantes, le décodage apporte de l'information et les LLR *a posteriori* sont donnés par 12.12.

Les décodeurs mous les plus courants sont l'algorithme BCJR pour les codes convolutifs (utilisés pour les turbo codes, voir annexe A) et le décodage par propagation de croyance ("Belief Propagation") des codes LDPC, qui sont tous les deux des codes à base de graphes.

Notre connaissance de la chaîne de bits envoyée est donc donnée par l'ensemble des valeurs des LLR pour chaque symbole reçu. On peut représenter cette connaissance sous forme d'un histogramme représentant la distribution des LLR. Si, à des fins de caractérisation, nous révélons la chaîne de bits envoyée $\{x_i\}$, alors nous pouvons tracer deux distributions distinctes : la distribution f_+ des LLR associée aux bits $x_i = 1$ et la distribution f_- des LLR associée aux bits $x_i = s0$. Par exemple, pour une modulation binaire transmise par un canal AWGN, ces distributions sont des gaussiennes de variance $4/\sigma^2$ respectivement centrées sur $\pm 2/\sigma^2$ (cf. équation 12.11). Plus les distributions f_+ et f_- sont disjointes et plus elles sont éloignées de l'origine, plus nous avons de connaissance sur la chaîne de bits envoyée.

Pour quantifier simplement cette connaissance, nous voudrions pouvoir calculer une grandeur réelle représentative de cette distribution. Pour cela, nous pouvons calculer le taux d'erreur. Puisque la valeur d'un bit la plus probable est le signe du LLR associé à ce bit, le taux d'erreur est simplement la fraction des distributions f_+ (resp. f_-) qui se trouve du côté négatif (resp. positif). Le taux d'erreur nous permet donc de quantifier à quel point les distributions f_+ et f_- sont disjointes, mais ne dit rien en général sur leur valeur moyenne.

À partir des distributions des LLR, nous définissions l'information, représentative de la connaissance apportée par les LLR [65] :

$$I = \frac{1}{2}\sum_{b=\pm 1}\int f_b(l)\log_2\frac{2f_b(l)}{f_+(l)+f_-(l)}dl, \tag{12.15}$$

Si les distributions f_+ et f_- sont bien distinctes, l'information sera importante. Au contraire, si f_+ et f_- sont toutes deux regroupées autour de 0, l'information sera faible. On associe une information à chaque type de LLR, définissant ainsi les informations intrinsèque, extrinsèque, *a priori*...

Bien entendu, l'information ainsi définie ne suffit pas à déterminer de façon unique la configuration de décodage. Elle ne rend pas compte de la distribution exacte des LLR. Ainsi, on peut trouver plusieurs distributions de LLR apportant la même information, mais dont l'issue du décodage (information extrinsèque) diffère. Cependant, pour des canaux simples (BSC ou

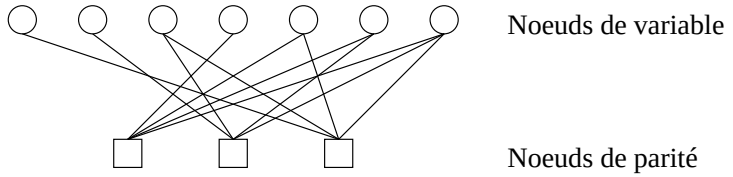


FIG. 12.4: Graphe bipartite représentant la matrice du code linéaire de Hamming $(7, 4)$ (section 12.2). La représentation sous forme de graphe est nécessaire pour les codes LDPC car leur matrice H est trop grande pour qu'on puisse l'écrire.

AWGN), la distribution des LLR est caractérisée par un seul paramètre, équivalent à l'information.[6]

L'information extrinsèque issue d'un décodeur permet d'évaluer la qualité du décodage : le taux d'erreur final est d'autant plus faible que l'information extrinsèque s'approche de 1, et devient satisfaisant pour des valeurs d'information extrinsèque typiquement supérieures à 0,999. À taux donné, un bon code est un code apportant beaucoup d'information extrinsèque.

12.5 Décodage des codes LDPC

Gallager a publié en 1962 [66] un algorithme de décodage des codes linéaires "parity check" dans le cas où la matrice H est peu dense. Ce sont les codes "Low Density Parity Check" ou LDPC [67]. Les codes LDPC utilisés pour la réconciliation ont une matrice H de 200 000 colonnes, et leur densité en 1 est de l'ordre de 10^{-4}. Dans ces conditions, La matrice H est trop grande pour être explicitée ; le code LDPC est alors représenté par un graphe constitué d'un nœud de variable associé à chaque bit à décoder, et d'un nœud de parité associé à chaque calcul de parité ("parity check"). Les nœuds de variable impliqués dans un calcul de parité sont reliés au nœud de parité correspondant par un segment (figure 12.4).

Cet algorithme de décodage des codes LDPC est qualifié de "Message Passing" (MP), ou "Belief Propagation" (BP). Il repose sur l'expression du rapport de vraisemblance du bit de parité b_p issu d'un calcul de parité[7] $b_p = \prod_{i=1}^{m} x_i$ sur un ensemble de m bits $\{x_1, \ldots, x_m\}$ en fonction des rapports des vraisemblance $LLR(x_i)$ des bits x_i :

$$LLR(b_p = \prod_{i=1}^{m} x_i) = -\prod_{i=1}^{m} \text{sign}(-LLR(x_i))\, \phi\left(\sum_{i=1}^{m} \phi(|LLR(x_i)|)\right) \tag{12.16}$$

$$\text{avec} \quad \phi(x) = -\ln(\tanh(x/2)) \tag{12.17}$$

$$\text{et} \quad LLR(b_p) = \ln\left(\frac{p(b_p = 1 \ \textit{i.e.} \text{ parité impaire})}{p(b_p = 0 \ \textit{i.e.} \text{ parité paire})}\right). \tag{12.18}$$

L'algorithme BP consiste à mettre à jour itérativement les LLR des bits à décoder en propageant ces LLR le long du graphe représentant les contraintes de parité du code LDPC. À l'itération n, le décodeur prend comme entrée les valeurs des LLR *a posteriori* fournis par l'itération $n-1$. Ces LLR représentent notre meilleure estimation de chaque bit à décoder,

[6] On calcule simplement : $I_{\text{BSC}}^{intrinsèque} = 1 - h(e)$, où $h(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ est l'entropie de Shannon binaire, d'où l'appellation "information"

[7] Rappelons que pour des bits, le produit n'est autre que l'opération logique «et».

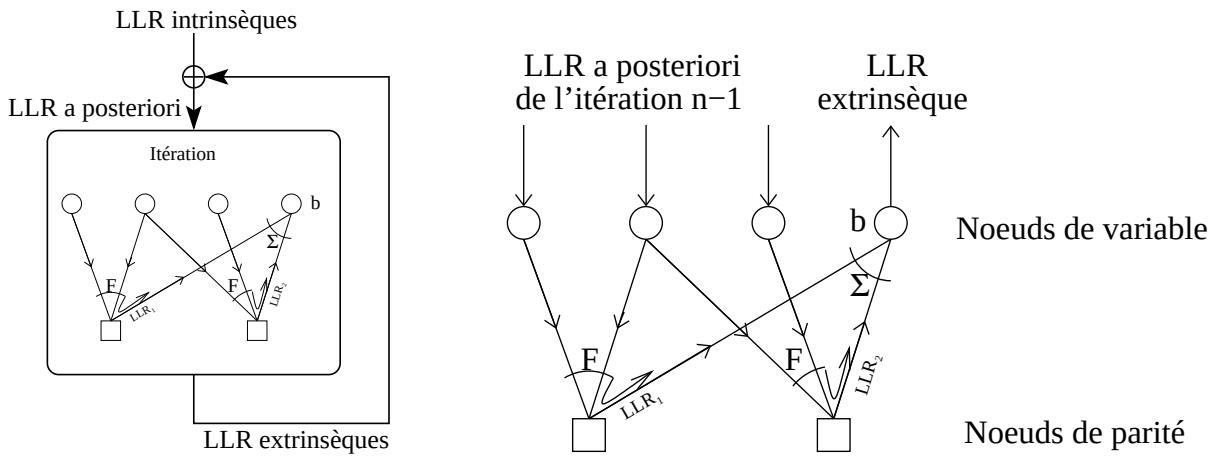


FIG. 12.5: À gauche : flux des LLR au cours du décodage BP itératif. Dans cette figure, nous considérons le schéma usuel dans lequel toutes les parités doivent être paires (utilisation non systématique des codes : $H \cdot x = 0$). Nous montrerons par la suite comment considérer l'entrée «parités» de la figure 12.3, propre à notre situation de réconciliation. À droite : étape élémentaire de l'itération n. Chaque nœud de variable i est associé à un symbole issu du canal, et donc à un bit à décoder. Il est initialisé par le LLR *a posteriori* i de l'itération précédente. Une itération du décodage consiste successivement à mettre à jour le LLR de chaque bit à décoder. Ce schéma représente la mise à jour du nœud de variable le plus à droite (noté b). Cette mise à jour comporte deux étapes : dans la première, on calcule (opération F, écrite à l'équation 12.16) le rapport de vraisemblance LLR_j de chacun des j calculs de parité dans lesquels intervient le bit b, en excluant le bit b. Si le rapport de vraisemblance LLR_j est élevé, cela signifie que le calcul de parité est très probablement impair, et donc que le bit b est très probablement 1, pour rétablir une parité paire (contrainte imposée par le code). Dans une deuxième étape, la somme (Σ) de tous les LLR_j fournit le LLR extrinsèque du bit b qui représente la croyance apportée par le décodage. Ce processus est appliqué à chaque nœud de variable.

à la fin de l'itération $n-1$. En utilisant les contraintes de parité imposées par le code, le décodeur fournit de l'information supplémentaire sur la valeur des bits à décoder, sous forme de LLR extrinsèques. Les LLR *a posteriori* de l'itération n, somme des LLR extrinsèques et intrinsèques, sont ensuite utilisés comme entrée de l'itération $n+1$. On arrête le décodage quand l'apport (LLR extrinsèques) de la dernière itération devient négligeable. Nous avons vu qu'avant décodage l'expression des LLR *a posteriori* (équation 12.12), qui quantifient notre meilleure estimation du bit x_i, coïncide avec l'expression des LLR intrinsèques (équation 12.9). Les LLR intrinsèques, calculés à partir des valeurs mesurées en sortie du canal, sont donc utilisés pour initialiser la première itération du décodage. Ce décodage est résumé par la figure 12.5.

Les codes LDPC sont des codes pouvant atteindre des performances arbitrairement proches de la limite de Shannon, pour une complexité modérée. Par manque de moyens de simulation au moment de leur découverte, ces codes ne sont utilisés que depuis la fin des années 1990.

Décodage LDPC et codes systématiques

Nous devons adapter les décodeurs mous que nous avons présentés à la réconciliation, c'est-à-dire à une situation de décodage systématique dans laquelle les bits de parité sont transmis dans un canal sans perte. Le décodage "BP" des codes LDPC s'étend simplement en ajoutant un nœud de variable dont le bit est 1 avec une probabilité de 1 (*i.e.*

dont le LLR intrinsèque est $+\infty$), connecté à chaque noeud de parité dont la parité doit être impaire. Ce nœud de variable supplémentaire permet de transformer tous les "parity checks" impairs en des "parity checks" pairs. Comme $\phi(+\infty) = 0$, on constate que cette opération est équivalente à changer le signe de la fonction F (équation 12.16) pour les "parity checks" impairs.

12.6 Modulation codée

Nous avons introduit dans la section 12.4 la notion de rapport de vraisemblance, qui nous a permis d'exploiter le caractère continu du bruit ajouté par les canaux usuels. Cependant, nous avons jusqu'alors uniquement considéré des modulations binaires. Si ces modulations binaires permettent des développements théoriques simples, elles sont peu efficaces en pratique, car elles ne tirent pas parti du caractère continu des signaux utilisés dans les télécommunications.

La modulation codée ("Coded Modulation") permet une meilleure utilisation de la bande passante en encodant plusieurs bits par symbole transmis par le canal. Pour transmettre N bits par symbole, on définit 2^N points dans l'espace des phases du signal, dits «constellation». La modulation ASK (Amplitude Shift Keying) place ces points régulièrement espacés sur une ligne droite ; la modulation PSK (Phase Shift Keying) les place en cercle, à amplitude constante ; enfin, la modulation QAM (Quadrature Amplitude Modulation) forme un réseau carré selon les deux quadratures du signal. On étiquette chacun des 2^N points définis dans l'espace des phases par un nombre de N bits.

La procédure de décodage doit tirer parti des liens entre les N bits encodés dans chaque symbole, grâce à l'information extrinsèque. Le canal de communication AWGN ajoute un bruit gaussien à la modulation codée. Si la modulation est 2-ASK (modulation binaire) ou 4-PSK (4 points répartis en cercle), les bits transmis dans un même symbole restent découplés (la distance de Hamming entre les bits est identique à la distance euclidienne dans l'espace des phases). Mais dans tous les cas plus complexes, le décodage d'un des N bits conditionne le décodage des autres bits contenus dans le symbole. C'est pourquoi beaucoup de stratégies de décodage de modulation codée font appel à un processus *itératif*. Pour chaque symbole transmis par le canal de communication, ce décodage permet d'affiner l'estimation d'un bit $B \in [1; N]$ donnée par la valeur du symbole mesuré à la sortie du canal (c'est-à-dire l'information intrinsèque) grâce à l'apport du décodage des autres bits $B' \neq B$ (c'est-à-dire l'information extrinsèque). Nous verrons que le décodage multi-niveaux que nous utilisons pour la réconciliation suit ce principe : le décodage d'un niveau est facilité par le résultat du décodage des autres niveaux.

Il existe plusieurs types d'assemblages de codes correcteurs d'erreurs utilisées dans une situation de "coded modulation". Citons les techniques de Bit Interleaved Coded Modulation (BICM), de Trellis Coded Modulation (TCM), de Turbo TCM, de Parallel Concatenated TCM ou de MultiLevel Coding (MLC) [68]. Cette dernière technique sera utilisée dans notre algorithme de réconciliation.

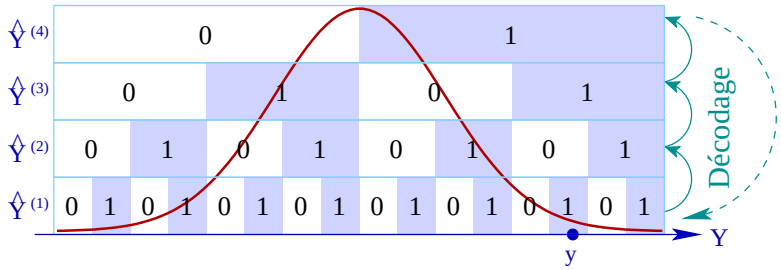


FIG. 12.6: Schématisation du décodage multi-niveaux sur $N = 4$ niveaux. La variable continue Y issue du canal quantique est discrétisée en N bits. Le schéma montre l'exemple d'une valeur y discrétisée en $\hat{y} = 1101$. Les erreurs sur l'ensemble des bits de niveau b sont corrigées par un code correcteur d'erreurs mou. Le décodage est itératif : on décode successivement chaque ensemble des bits de niveau B, du niveau $B = 1$ (bit de poids faible) au niveau $B = N$ (bit de poids fort). Ensuite, le décodage revient sur le niveau $B = 1\ldots$ jusqu'à ce que le décodage aboutisse.

12.7 Décodage multi-niveaux et réconciliation

Dans [8], Matthieu Bloch a adapté le décodage multi-niveaux au problème de la réconciliation. Cette méthode est conceptuellement très proche de la réconciliation par tranches initialement développé par Gilles Van Assche. Elle apporte à la réconciliation par tranches les concepts d'*information molle* et de *décodage itératif*.

La réconciliation commence par une binarisation de l'axe réel en 2^N intervalles adaptés à la distribution gaussienne des données continues mesurées par Bob. L'optimisation de la position de ces intervalles sera abordée en section 13.3. Cette discrétisation permet la transformation de chaque variable continue X issue du canal quantique en un nombre $\hat{X}$ de N bits. Chacun des bits de la variable $\hat{X}$ est noté $\hat{X}^{(B)}$, du bit le moins significatif ($B = 1$) au bit le plus significatif ($B = N$). Le décodage fait intervenir un ensemble de k variables continues (typiquement, k = 200 000), et fournit donc $b = Nk$ bits. Cet ensemble de N bits par symbole issus de la discrétisation des variables continues nous place dans une situation de modulation codée. Nous pouvons donc appliquer les méthodes de décodage multi-niveaux, dont le principe est le suivant : à l'aide de N codes correcteurs d'erreurs mous, on décode successivement chacun des ensembles de k bits de position B, appelés niveaux, ou tranches. Le décode du niveau B se fait à l'aide de la valeur des symboles continus qui ont transité dans le canal qantique (entrée intrinsèque du décodeur mou), des parités envoyées par le canal classique, et du résultat du décodage des autres tranches. Après le décodage du dernier niveau, on revient sur le décodage des premiers niveaux (processus itératif). Ce décodage est symbolisé par la figure 12.6. Comme nous l'avons remarqué dans la section précédente, ce décodage itératif permet de tirer parti du fait que les valeurs des bits de chaque tranche ne sont pas indépendantes. Nous verrons comment utiliser quantitativement cette dépendance.

Pour chaque niveau, le taux du code utilisé dépend du taux d'erreur. Les tranches rassemblant les bits les moins significatifs ont beaucoup d'erreurs, car le bruit introduit par la transmission quantique peut facilement changer un bit 0 en un bit 1. Ces tranches nécessitent donc un code générant beaucoup de bits de parité, c'est-à-dire un code de faible taux. Au contraire, les tranches rassemblant les bits les plus significatifs sont moins bruitées, et nécessitent des codes de taux proche de 1.

Quantitativement, on exprime le taux optimal du code à utiliser pour décoder la tranche B en fonction des entropies $H(\hat{X}^{(B)})$ et $H(\hat{X}^{(B)}|X)$. Ces entropies se calculent à partir des distributions de probabilité des variables aléatoires X et $\hat{X}^{(B)}$, en utilisant les équations 1.2 et 1.6. Pour un décodage depuis les bits les moins significatifs vers les bits les plus significatifs, on montre [68, 8] que le taux optimal est donné par l'expression :

$$R_{s\ \mathrm{opt}}^{(B)} = 1 - \Big(H(\hat{X}^{(B+1)}|X) - H(\hat{X}^{(B)}|X)\Big) + \Big(H(\hat{X}^{(B+1)}) - H(\hat{X}^{(B)})\Big). \tag{12.19}$$

Ces taux ne dépendent que du rapport signal à bruit de la transmission, et de la position des intervalles de binarisation. Bien entendu, les codes que nous utilisons n'atteignent pas la limite de Shannon, et nous devons utiliser des codes de taux $R_s^{(B)} < R_{s\ \mathrm{opt}}^{(B)}$. Nous aborderons au chapitre suivant les méthodes d'optimisation nous permettant de choisir les intervalles de binarisation et les taux effectifs des codes. Munis de ces taux, nous pouvons expliciter l'efficacité de réconciliation que nous avions écrite en section 12.3 :

$$\beta = \frac{H(\hat{X}) - N + \sum_{B=1}^{N} R_s^{(B)}}{I_{AB}}. \tag{12.20}$$

En pratique, nous utilisons 4 tranches, soit 16 intervalles de binarisation. Pour les rapports signal à bruit utilisés, les taux des deux premières tranches sont typiquement inférieurs à 1%. Ces tranches peuvent donc être décodées en les révélant totalement sur le canal classique (décodage de taux 0), sans sacrifier significativement l'efficacité β. Les deux autres tranches sont décodées à l'aide de codes LDPC. Finalement, la réconciliation produit 2 bits (c'est-à-dire 1 bit par tranche non révélée) par symbole issu du canal classique, c'est-à-dire $b = 400\,000$ bits par bloc de 200 000 symboles.

L'encodage de chaque niveau B par Bob, c'est-à-dire le calcul des parités qui circulent sur le canal classique, utilise directement les valeurs discrétisées $\hat{Y}^{(B)}$. Dans l'exemple des codes LDPC, on calcule les parités en multipliant la matrice H du code LDPC choisi, par l'ensemble des valeurs $\hat{Y}^{(B)}$. L'encodage est donc une opération peu coûteuse en termes de temps de calcul.

Les décodeurs mous ont deux entrées : l'entrée de parité et l'entrée des LLR intrinsèques. La première entrée est directement alimentée par les parités issues du canal classique. Nous avons vu dans la section précédente que l'estimation des LLR intrinsèques dépendait de l'information fournie par la variable continue X, mais aussi de l'information apportée par le décodage des autres tranches, car les valeurs des bits de chaque tranche ne sont pas indépendantes. Cette information se trouve dans les LLR extrinsèques issus des décodages des autres tranches. Pour les tranches qui ne sont pas encore décodées, on fixe les LLR extrinsèques à 0. Quantitativement, on écrit le LLR intrinsèque de la tranche B du symbole i [68, 8] :

$$LLR_i^{\mathrm{intrinsèque}(B)} = \log \frac{\sum_{\hat{X} t.q. \hat{X}^{(B)}=1} p(X_i, \hat{X}) \prod_{B' \neq B} e^{\hat{X}^{(B')} LLR_i^{\mathrm{extrinsèque}(B')}}}{\sum_{\hat{X} t.q. \hat{X}^{(B)}=0} p(X_i, \hat{X}) \prod_{B' \neq B} e^{\hat{X}^{(B')} LLR_i^{\mathrm{extrinsèque}(B')}}}, \tag{12.21}$$

où $p(X_i, \hat{X})$ est la probabilité conjointe pour que la variable continue soit X_i et que la variable binaire à décoder soit $\hat{X}$. Si les LLR extrinsèques sont nuls (pas d'apport provenant du décodage

des autres niveaux), on remarque que cette expression coïncide avec l'expression 12.9. Le LLR intrinsèque exprime donc s'il est plus probable que le bit à décoder soit 1 ou 0, connaissant le symbole continu X_i. Quand les LLR intrinsèques sont non nuls, cette probabilité est pondérée par l'information, sous forme de probabilité, apportée par le décodage des autres niveaux.

Cette expression permet de boucler les sorties extrinsèques des décodeurs mous des tranches $B' \neq B$ sur l'entrée intrinsèque du décodeur de la tranche B. Le décodage multi-niveaux étant itératif, chaque nouvelle itération tire parti du décodage, de plus en plus fin, des autres tranches.

À titre d'illustration, nous représentons figure 12.7 l'évolution de la distribution des LLR intrinsèques au cours du décodage itératif.

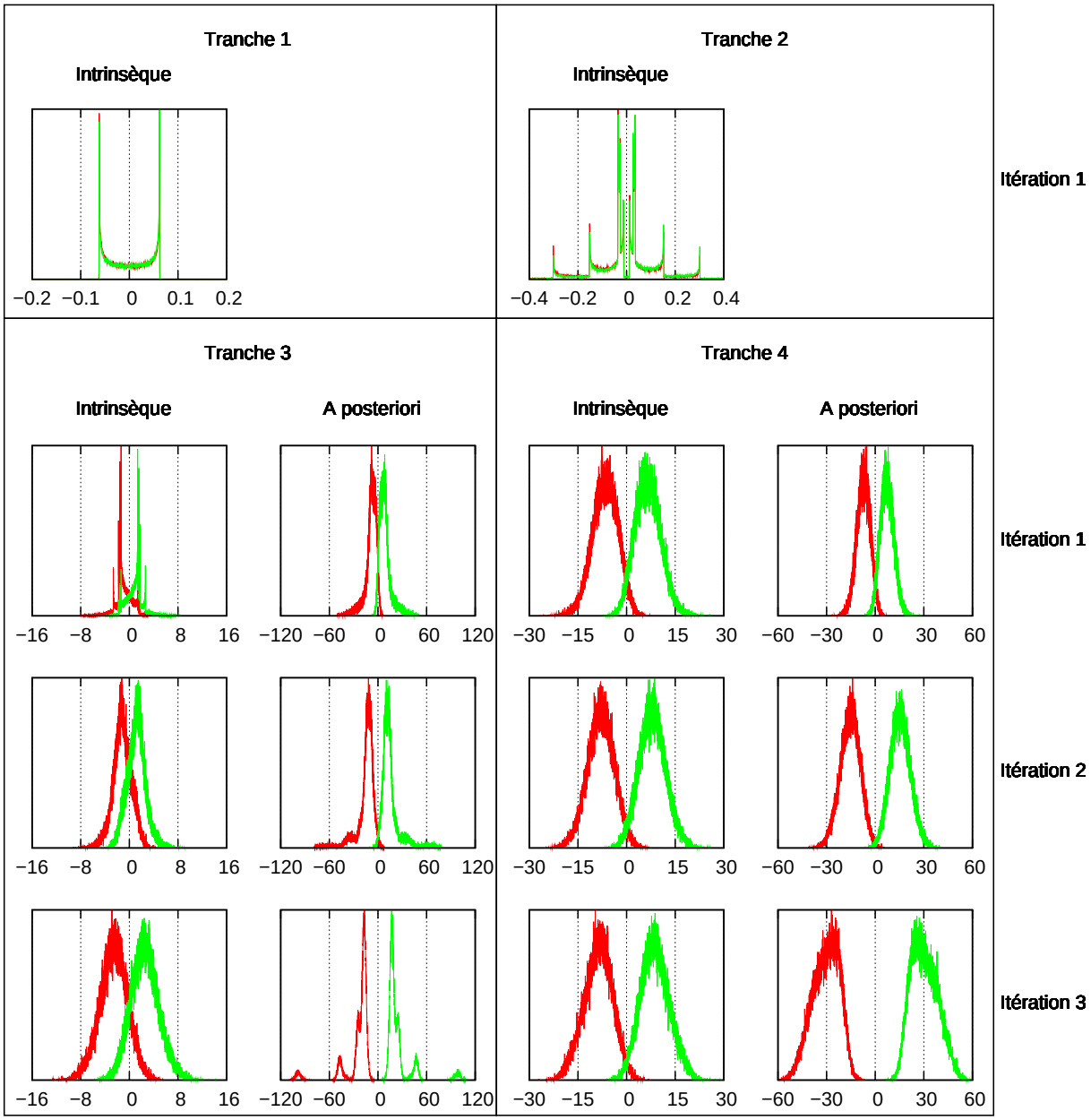


FIG. 12.7: Histogrammes représentant la distribution des LLR intrinsèques et *a posteriori* au cours du décodage itératif, pour 3 itérations sur 4 tranches. On distingue les LLR associés aux bits d'origine 0 (courbes foncées, notées LLR_0) et aux bits d'origine 1 (courbes claires, notées LLR_1). Comme le signe des LLR indique notre estimation du bit à décoder, le taux d'erreur est la fraction de la distribution LLR_1 (resp. LLR_0) qui se trouve du côté négatif (resp. positif). Ainsi, des distribution LLR_0 et LLR_1 confondues et proches de 0 indiquent la situation initiale incertaine. Au contraire, des distribution LLR_0 et LLR_1 clairement disjointes et éloignées de 0 indiquent le succès du décodage. On constate que les distributions des LLR intrinsèques des tranches 1 et 2 sont presque confondues et très proches de 0 : ces tranches contiennent donc peu d'information et sont révélées lors de la première itération. Ensuite le décodage itère entre les tranches 3 et 4. Pour chaque décodage, on trace les distributions des LLR intrinsèques (en entrée du décodeur) et *a posteriori* (issus de décodage). On constate que le décodage a bien l'effet escompté : les distributions LLR_0 et LLR_1 *a posteriori* sont plus disjointes et éloignées de 0 que les LLR intrinsèques. De plus, au cours des itérations, les distribution des LLR *a posteriori* d'une tranche donnée s'éloignent de plus en plus l'une de l'autre. En effet, les LLR intrinsèques offrent une meilleure situation initiale au décodeur d'une tranche donnée : le décodage d'une tranche est facilité par le décodage des autres tranches. Les taux des codes des tranches 3 et 4 sont respectivement $0,23$ et $0,86$.

Chapitre 13

Implémentation et optimisation de la réconciliation molle et du décodage LDPC

Les codes LDPC sont les briques de base du processus de réconciliation. Ils permettent, comme nous l'avons montré au chapitre précédent, d'extraire l'information contenue dans les corrélations de nos variables gaussiennes. Afin d'obtenir un taux de clé secrète maximal, nous devons optimiser les deux paramètres essentiels de la réconciliation : l'efficacité et la vitesse de décodage. Souvent, un compromis est nécessaire : perdre un peu d'efficacité permet de décoder plus rapidement.

Dans ce chapitre, nous passerons en revue les optimisations numériques, algorithmiques, informatiques et matérielles qui nous ont permis d'augmenter le taux de clé secrète. Nous avons concentré nos efforts d'optimisation pour un canal quantique long de 25 km, c'est-à-dire une transmission de 25% (pertes de 6 dB). En effet, le projet SECOQC a posé comme critère commun une transmission de 1 kb/s sur 25 km. Les résultats présentés dans ce chapitre sont issus d'un travail commun avec Matthieu Bloch.

13.1 Compromis entre vitesse et efficacité

Le taux secret obtenu à la fin du processus de réconciliation s'écrit

$$K = v\left(\beta I_{AB} - I_{BE}\right) \tag{13.1}$$

où

- K est le taux de bits secrets, en bits par seconde.
- v est la vitesse de décodage, en symboles par seconde.
- I_{AB} et I_{BE} sont les informations mutuelles, en bits par symbole, calculées à partir des paramètres du canal, en réconciliation inverse.
- β est l'efficacité de réconciliation. C'est le rapport entre le nombre de bits effectivement extraits des symboles issus du canal quantique par la réconciliation et la limite de Shannon I_{AB}.

Pour augmenter le taux secret, il faut augmenter à la fois la vitesse v et l'efficacité β. Certaines des optimisations présentées dans ce chapitre portent sur la vitesse, d'autres sur l'efficacité. D'autres influent sur les deux paramètres à la fois, et seront sujettes à un compromis.

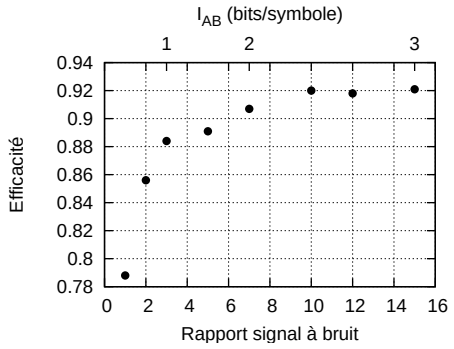


FIG. 13.1: Efficacité de la réconciliation en fonction du rapport signal à bruit, ou de l'information mutuelle I_{AB}. L'efficacité croît avec la rapport signal à bruit. Ces efficacités sont indicatives : elles peuvent être améliorées en sacrifiant la vitesse de réconciliation.

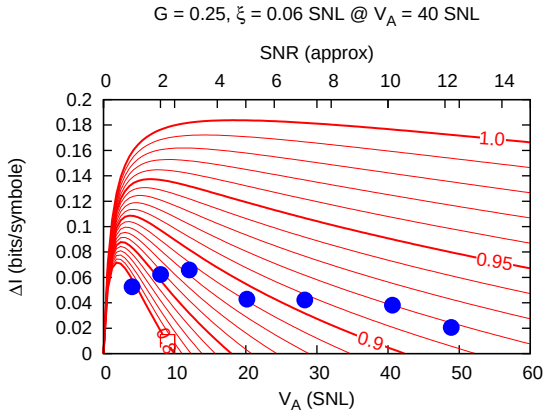


FIG. 13.2: Optimisation du rapport signal à bruit. On fixe le gain $G = 0,25$ (transmission définie par le projet SECOQC) et l'excès de bruit $\xi = 0,0015V_A$. Le rapport signal à bruit est déterminé par la variance de modulation V_A (abscisse), seul paramètre restant libre. Les courbes représentent le taux secret $\Delta I = \beta I_{AB} - I_{BE}$ pour différentes efficacités β, indiquées le long de la courbe (courbes de même efficacité). Les points indiquent le taux secret obtenu en fonction de V_A, en utilisant les efficacités indiquées par la figure 13.1. Le taux secret est optimal pour $SNR = 3$, ce qui correspond à une variance de modulation $V_A = 12$.

13.2 Quel rapport signal à bruit choisir ?

Le processus de réconciliation a pour but d'extraire $I_{AB} = \frac{1}{2}\log_2(1 + SNR)$ bits sans erreur à partir de variables gaussiennes corrélées de coefficient de corrélation $\rho^2 = \frac{SNR}{1+SNR}$. L'efficacité de ce processus, une fois optimisé, ne dépend que de la valeur du rapport signal à bruit. De plus, l'efficacité croît avec le rapport signal à bruit, car moins les données sont bruitées, plus elles sont faciles à corriger. La figure 13.1 trace l'efficacité de réconciliation en fonction du rapport signal à bruit.

Pour une réconciliation d'efficacité limitée, il existe une valeur du rapport signal à bruit qui maximise le taux secret $\Delta I = \beta I_{AB} - I_{BE}$. Cet optimum, tout comme I_{BE}, ne dépend plus seulement du rapport signal à bruit, mais aussi des deux paramètres du canal (le gain G et l'excès de bruit ξ) et de la variance de modulation V_A. Le gain étant déterminé par la

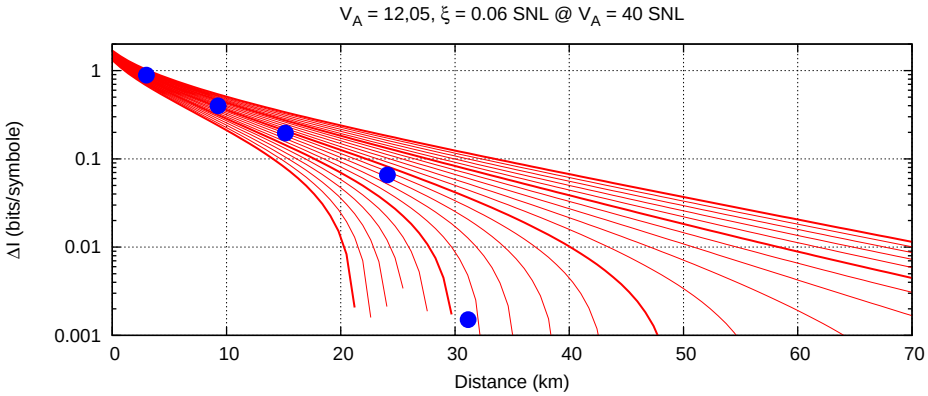


FIG. 13.3: Portée du protocole de distribution quantique de clé. À V_A fixé, on peut faire varier le rapport signal à bruit en faisant varier la transmission du canal quantique, ou en d'autres termes la distance d entre Alice et Bob ($d = -\frac{10}{\alpha}\log_{10}(G)$, avec $\alpha = 0,25$ db/km).

fibre de transmission, et l'excès de bruit par les imperfections de notre dispositif, on fait varier le rapport signal à bruit en faisant varier V_A, à gain et excès de bruit fixés. L'optimisation du rapport signal à bruit est donc équivalente à l'optimisation de la variance de modulation. La figure 13.2 trace le taux secret pour différentes variances de modulation. On trouve un optimum autour de $V_A = 12$, qui correspond à un rapport signal à bruit de 3. Cette valeur convient bien aux codes correcteurs d'erreurs.

L'efficacité limitée de la réconciliation limite la portée du protocole de distribution quantique de clé. Pour faire varier le rapport signal à bruit, on peut également faire varier la transmission du canal quantique, ou, en d'autres termes, la distance entre Alice et Bob (figure 13.3). L'inefficacité se traduit d'abord par une perte du taux secret, typiquement d'un facteur deux. Mais surtout, elle introduit un seuil au-delà duquel plus aucune clé, aussi petite soit elle, ne peut être transmise. Cet effet se retrouve dans tous les protocoles de cryptographie quantique : le taux de clé décroît exponentiellement avec la distance (à cause de l'atténuation exponentielle de la fibre de transmission), jusqu'à arriver à un seuil où toute transmission est impossible. Pour notre protocole, ce seuil survient à 30 km environ, en l'état actuel des algorithmes de réconciliation. Il existe deux différences fondamentales à cet égard entre notre système à variables continues et les protocoles à photons uniques. La première est la portée limitée de notre système : les systèmes à photons uniques les plus performants à ce jour atteignent des distances de l'ordre de 100 km. Notre système se situe donc dans un régime de courtes à moyennes distances, avec un taux important. La deuxième différence est la nature de la limitation : la portée des systèmes à photons uniques est déterminée par le bruit d'obscurité des compteurs de photons, qui est une limitation matérielle difficile à dépasser. Pour notre système, la limitation est algorithmique : les corrélations que nous obtenons expérimentalement permettent de transmettre des clés à plus longue distance[1], et seule notre capacité intellectuelle à extraire des bits secrets de ces corrélations limite la portée. Les progrès actuels sur les codes correcteurs d'erreurs, motivés par l'industrie des télécommunications, laissent espérer des

[1]Le protocole à variables continues permet une distribution de clé pour toute transmission, avec les valeurs d'excès de bruit que nous mesurons. Seule la prise en compte de marges de sécurité sur le taux secret peut introduire un effet de seuillage.

améliorations possibles dans ce domaine.

13.3 Paramètres de la discrétisation

L'efficacité du processus de réconciliation est le produit de deux efficacités :

$$\beta = \beta_{\text{codes}}\beta_{\text{tranches}} \tag{13.2}$$

La première est l'efficacité du décodage proprement dit : les codes (en l'occurrence les codes LDPC) que nous utilisons n'atteignent pas la limite de Shannon. La deuxième est l'efficacité de la discrétisation : même si nous utilisions des codes qui atteignent la limite de Shannon, nous ne pourrions extraire tous les bits disponibles dans les corrélation gaussiennes, car de l'information continue s'est perdue lors de la répartition des données dans chaque intervalle de discrétisation. Pour obtenir une discrétisation parfaite, il faudrait une discrétisation infiniment petite. Néanmoins, nous arrivons à obtenir de bonnes efficacités avec un nombre modéré de tranches.

Pour calculer l'efficacité de la binarisation, nous utilisons la formule 12.20 : $\beta = \frac{H(\hat{X})-I_{\text{Révélé}}}{I_{AB}}$ que nous avons établie au chapitre précédent, dans laquelle le nombre de bits révélés $I_{\text{Révélé}}$ est la somme des bits révélés lors du décodage $I^{(b)}_{\text{Révélé}}$ de chacune des tranches b. Pour simuler un décodage parfait, nous considérons que le code révèle le minimum de bits autorisé par le théorème de Shannon : $I^{(b)}_{\text{Révélé}} = 1 - R^{(b)}_{s\ \text{opt}}$, où $R^{(b)}_{s\ \text{opt}}$ est le taux optimal de la tranche b (équation 12.19). Ce faisant, nous isolons l'inefficacité de la discrétisation de l'inefficacité des codes. On écrit donc :

$$\beta_{\text{tranches}} = \frac{H(\hat{X}) - N + \sum_{B=1}^{N} R^{(B)}_{s\ \text{opt}}}{I_{AB}} \quad \text{puis} \quad \beta_{\text{codes}} = \frac{\beta}{\beta_{\text{tranches}}}. \tag{13.3}$$

Le processus d'optimisation de l'efficacité de binarisation est numérique : nous décidons d'un nombre de tranches n, nous divisons l'axe réel en 2^n intervalles, et nous calculons β_{tranches}. En testant un grand nombre de positions des intervalles, nous espérons obtenir la meilleure efficacité possible.

Nous avons exploré ce problème d'optimisation à $2^n - 1$ paramètres (c'est-à-dire 15 paramètres pour 4 tranches) avec diverses distributions d'intervalles, et un algorithme de recherche d'optimum par recuit simulé (ou "annealing"), en complément de l'algorithme d'optimisation "Powell" utilisé par Gilles Van Assche. Nous faisons les remarques empiriques suivantes :

- La division intuitive en intervalles d'équiprobabilité de la distribution gaussienne donne une mauvaise efficacité (typiquement 20% en dessous de l'optimum).
- En prenant une distribution d'intervalle aléatoire assez régulière, on approche aisément de la meilleure efficacité. Autrement dit, il est possible d'optimiser la disposition des intervalles de binarisation en faisant varier un nombre restreint des $2^n - 1$ paramètres.

Au regard de ces remarques, nous avons opté pour des intervalles de même largeur (voir figure 13.4). Ce problème d'optimisation à une variable présente l'avantage d'être simple à mettre en œuvre et atteint l'optimum trouvé par les autres méthodes. Cette simplicité est particulièrement appréciable car nous devons optimiser la position des tranches pour chaque rapport signal à bruit. De plus, la position des intervalles optimisée par cette méthode a la propriété de varier régulièrement quand ont fait varier le rapport signal à bruit. Le résultat de l'optimisation est représenté figure 13.5.

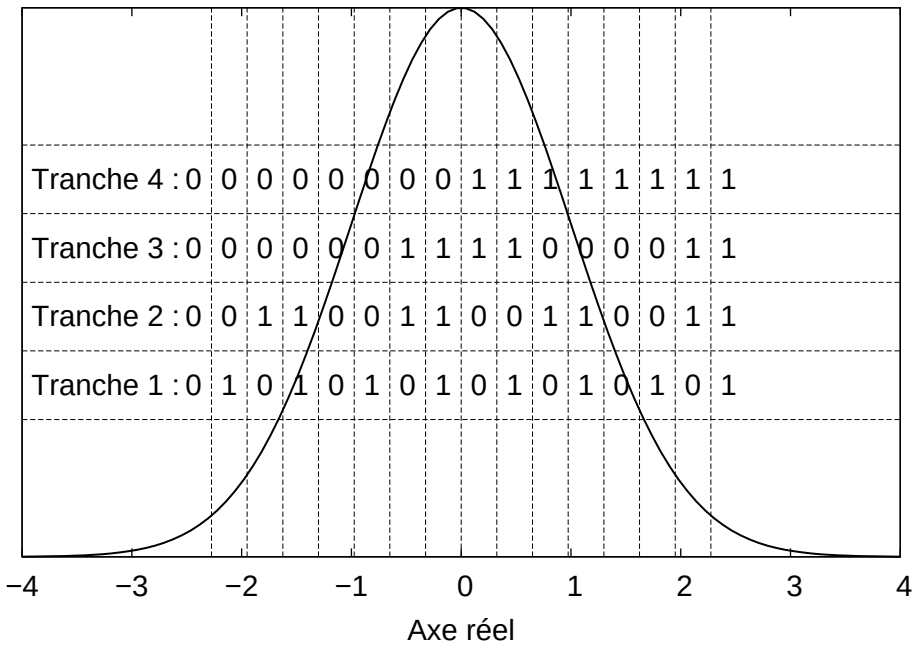


FIG. 13.4: Nous choisissons de diviser l'axe réel en intervalles de binarisation de largeurs identiques. L'unité de longueur est l'écart type de la distribution gaussienne des données réelles. Pour un rapport signal à bruit de 3, la largeur optimale des intervalles de binarisation est 0,312. Le premier intervalle s'étend depuis $-\infty$ et le dernier s'étend jusqu'à $+\infty$

Au premier ordre, l'optimisation de la binarisation agit sur l'efficacité plus que sur la vitesse de réconciliation. Cependant, l'efficacité de binarisation varie peu lorsque l'ont fait varier la largeur de l'intervalle de binarisation. Par exemple, pour un rapport signal à bruit de 3, une largeur de 0,312 produit l'efficacité maximale de 0,977 ; mais cette dernière reste supérieure à 0,97 pour des largeurs variant entre 0,25 et 0,45. Or, dans cette gamme de largeurs, les taux des codes peuvent varier considérablement. Dans notre exemple, le taux de la troisième tranche varie entre 0,2 et 0,6 : la densité d'un code LDPC, et donc sa vitesse de décodage peuvent varier en conséquence. La recherche de taux optimaux fait l'objet de la section suivante.

13.4 Taux des codes LDPC

Maintenant que nous avons discrétisé notre axe réel, nous devons choisir des codes LDPC. Plusieurs effets sont à prendre en compte.

Nous ne disposons de codes LDPC que pour certains taux discrets. Richardson [69] a montré un résultat intéressant : les performances d'un code LDPC ne dépendent que de la distribution de ses connectivités, c'est-à-dire de la distribution des densités des lignes ou des colonnes de la matrice de "parity check". Pour caractériser un code, ils nous suffit donc de connaître, pour tout m, le nombre de nœuds de variable impliqués dans m calculs de parité. Cette distribution résulte d'une optimisation numérique ardue qui a été réalisée par une équipe du LTHC à Lausanne [70] pour un canal de type AWGN. Ils proposent ainsi des codes pour

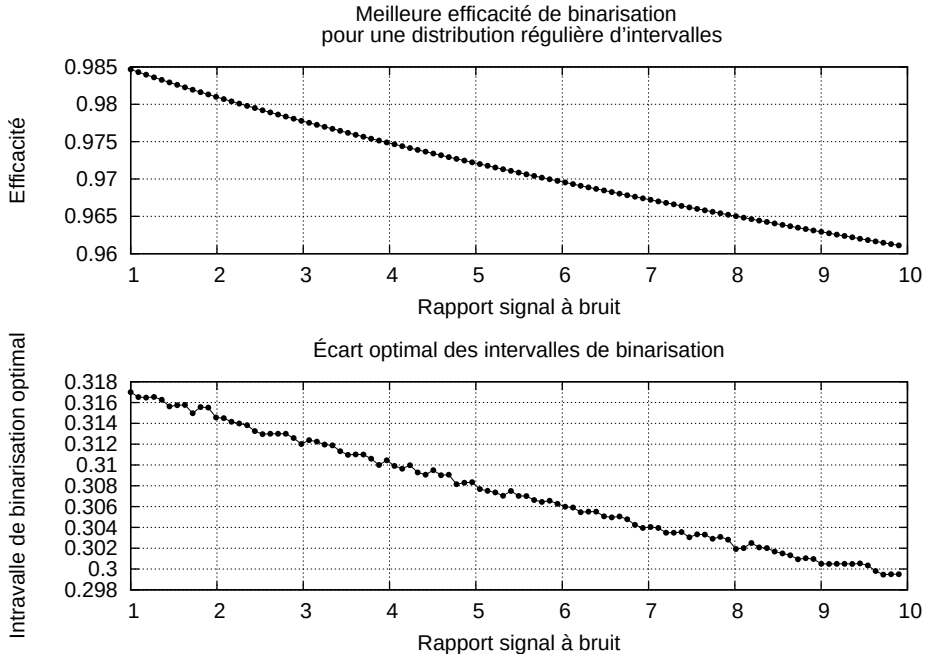


FIG. 13.5: Efficacité de discrétisation. Pour chaque rapport signal à bruit, nous faisons varier la position des intervalles de binarisation afin d'optimiser l'efficacité de binarisation β_{tranches}. Nous avons choisi des intervalles de même largeur, et 4 tranches de binarisation (soit 16 intervalles séparés par 15 frontières). La figure du haut montre la meilleure efficacité trouvée, celle du bas la largeur optimale des intervalles de binarisation permettant d'obtenir cette efficacité, exprimée en écarts types de la distribution gaussienne des données binarisées. Pour les grands rapports signal à bruit, l'efficacité diminue : quand la variance du bruit est petite, il faut envisager une discrétisation plus fine de l'axe réel. Comme nous travaillons autour d'un rapport signal à bruit de 3, cet effet n'est pas gênant.

divers taux discrets avec deux chiffres significatifs, de 0,05 à 0,99. En utilisant ces densités et un programme écrit par Matthieu Bloch, nous avons généré un grand nombre de graphes de divers taux et de diverses connectivités. Nous avons sélectionné les codes offrant une plus grande vitesse de décodage.

Nous devons choisir l'écart entre les taux des codes utilisés et les taux idéaux exprimés par la formule 12.19, c'est-à-dire l'efficacité de chaque code LDPC. Si nous choisissons un écart faible, l'efficacité de réconciliation sera bonne, mais la quantité d'information extrinsèque collectée à chaque étape du décodage multi-niveaux sera faible. Ainsi, le nombre d'itérations du décodage multi-niveaux nécessaires augmentera, réduisant ainsi la vitesse de réconciliation. Au contraire, si l'écart est important, le décodage multi-niveaux convergera rapidement et la vitesse sera importante, mais l'efficacité sera mauvaise. Là encore, un compromis est à trouver. Nous avons choisi un écart de 0,04 entre le taux idéal de la troisième tranche et le taux de son code, et un écart de 0,025 pour la quatrième tranche. Le code disponible ayant le taux le plus proche du taux désiré est sélectionné.

Une fois l'efficacité des codes choisis, nous devons sélectionner un graphe parmi ceux disponibles. On constate que les graphes des codes LDPC les plus efficaces comportent

plus de liens entre les nœuds de variable et les nœuds de parité [70]. Or, le temps nécessaire au décodage est proportionnel au nombre de liens du graphe. À taux donnée, le choix d'un code LDPC résulte donc encore d'un compromis entre vitesse et efficacité de décodage. En pratique, on teste plusieurs codes jusqu'à maximisation du taux secret. Bien entendu, cette dernière optimisation est couplée à la précédente : plus nous choisissons un taux éloigné du taux idéal, plus nous pouvons nous permettre d'utiliser un code de moins bonne qualité.

Nous avons vu que la binarisation de nos variables continues nous laissait une grande marge de manœuvre quant au choix du taux du code de la troisième tranche. Cette tranche est en effet la plus intéressante, car les deux premières tranches sont entièrement révélées, et la dernière tranche contient peu d'erreurs. Pour un rapport signal à bruit de 3, nous avons choisi un intervalle de binarisation de 0,358. Pour cet intervalle, nous choisissons un code de taux 0,41 pour la 3e tranche, et un code de taux 0,95 pour la 4e tranche[2]. Ces choix offrent un meilleur compromis entre vitesse et efficacité : l'efficacité de binarisation $\beta_{\text{tranches}} = 0,976$, l'efficacité des codes est $\beta_{\text{codes}} = 0,903$ pour une efficacité totale $\beta = \beta_{\text{tranches}}\beta_{\text{codes}} = 0,883$. Enfin, nous avons optimisé ces paramètres pour tout rapport signal à bruit, comme représenté figure 13.6.

13.5 Décodage des codes LDPC : l'algorithme Message Passing et ses variantes

Le décodage LDPC se fonde sur la propagation de la confiance accordée à chacun des bits à décoder, entre les nœuds de variable et les nœuds de parité. Nous avons recontré le schéma de principe à le section 12.5. À chaque nœud de variable est associé initialement le LLR intrinsèque fourni par le canal. Un LLR est également associé à chaque connexion. L'algorithme Message Passing standard met à jour les LLR associés aux connexions à partir des LLR associés aux nœuds de variable, en utilisant la relation 12.16 reliant les LLR d'un ensemble de variables et le LLR d'une vérification de parité sur ces variables. Inversement, les LLR associés aux nœuds de variable sont ensuite mis à jour à partir des LLR associés aux connexions.

Il existe plusieurs représentations équivalentes de cet algorithme. Nous privilégions une version compacte décrite dans [71] plutôt qu'un version complète décrite, par exemple, dans [72]. La première version utilisant moins de structures de données, elle requiert moins d'accès mémoire, et s'exécute environ deux fois plus rapidement que la première.

Un autre algorithme de Message Passing permet un décodage plus rapide. Cet algorithme, décrit dans [73] intercale les deux étapes du décodage décrites plus haut dans une même étape. Ainsi, la première étape bénéficie, au cours de son exécution, d'une partie des résultats de la deuxième étape. Cet algorithme se décline en trois versions, dites RMP, CMP et RCMP, selon qu'elles fondent leur point de vue sur les nœuds de variable, de parités, ou sur le graphe dans sa globalité. Nous avons opté pour la version RMP, qui permet d'atteindre des taux d'erreur comparables à ceux obtenus avec l'algorithme standard, en exactement deux fois moins

[2]Pour le premier code, nous choisissons une connectivité entre 7 et 8 pour les nœuds de parité, et entre 2 et 41 pour les nœuds de variable. Pour le deuxième code, ces connectivités sont respectivement entre 59 et 60, et entre 2 et 8

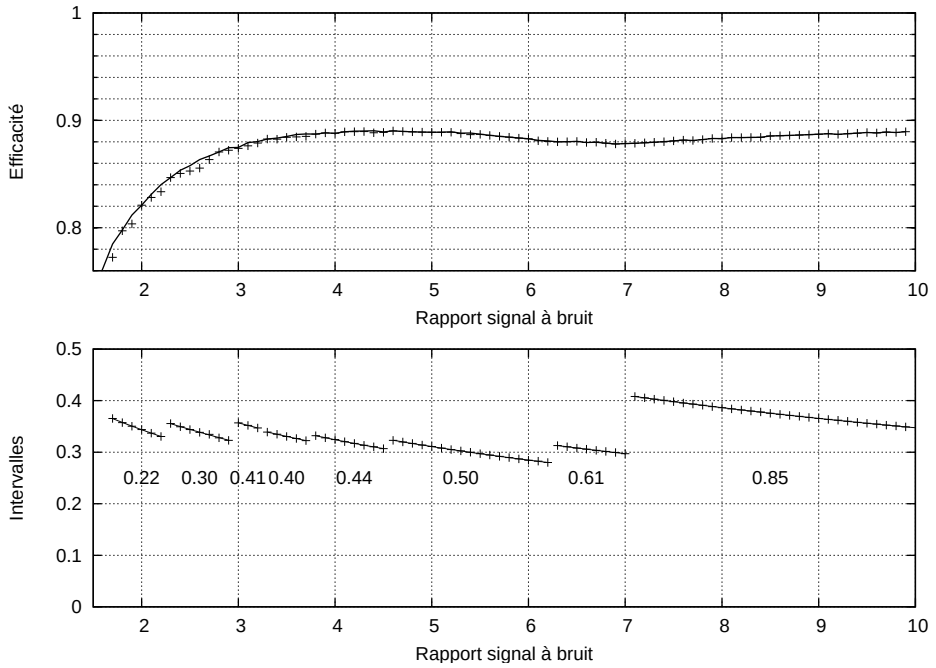


FIG. 13.6: Efficacité totale du processus de réconciliation. La courbe solide du graphique supérieur représente l'efficacité maximale compte tenu de l'efficacité de réconciliation, de l'écart choisi entre le taux de Shannon et le taux réel des codes LDPC et des codes disponibles. Pour chaque rapport signal à bruit, nous choisissons une disposition des tranches (graphique inférieur) qui approche l'efficacité maximale (symboles + sur le graphique supérieur) et qui donne des taux optimaux. L'ensemble des rapports signal à bruit peut se diviser en intervalles sur lesquels la largeur de binarisation varie continûment (ajustement exponentiel), avec un taux fixe pour la troisième tranche (indiqué pour chaque intervalle sur le graphique inférieur)

d'itérations. Cependant, l'algorithme décrit dans [73] n'est pas sous sa forme compacte, ce qui lui fait perdre l'avantage du nombre d'itérations. Nous avons donc modifié cet algorithme pour l'écrire sous sa forme compacte, et réduire ainsi le temps nécessaire à chaque itération. Finalement, l'algorithme RPM dans sa version compacte permet d'atteindre une vitesse exactement deux fois plus grande que l'algorithme compact standard.

13.6 Réduction du temps de calcul

Nous quittons le domaine de l'optimisation numérique et de l'algorithmique pour celui de la programmation, qui traite de l'implémentation logicielle performante des concepts rencontrés dans ce chapitre. Nous détaillons différentes méthodes pour optimiser la vitesse de traitement d'un algorithme donné (le décodage de type "Message Passing") par un processeur d'ordinateur. Cette section vise donc à augmenter la vitesse du décodage, sans faire varier son efficacité.

La caractéristique la plus spécifique d'un code LDPC est la représentation de sa matrice de "parity check" dans la mémoire de l'ordinateur. En effet, les nombreux "parity checks" font l'efficacité remarquable du décodage LDPC. Il est bien entendu hors de question d'enregistrer chacun des éléments de la matrice, car cette matrice est trop importante.

Au contraire, nous tirons parti de la faible densité de cette matrice en ne mémorisant que les emplacements des éléments 1 de cette matrice, ce qui donne une représentation sous forme de graphe que nous avons déjà rencontrée. Le problème de la représentation en mémoire de cet arbre répond aux multiples façons de considérer cet arbre : doit-on mémoriser à quels nœuds de variable chaque nœud de parité est connecté plutôt que l'inverse, ou doit-on plutôt considérer chaque connexion et mémoriser les deux nœuds qu'elle relie ? Enfin, comment doit-on mémoriser les différents LLR mis en jeu dans le décodage LDPC (voir section précédente) ? Cette question est d'importance car chaque étape élémentaire du décodage LDPC demande d'aller chercher en mémoire un ou plusieurs LLR. En fonctions des machines, ces accès mémoire peuvent consommer plusieurs cycles pendant lesquels le processeur est inactif. Enfin, certaines machines possèdent une mémoire cache à accès rapide, mais souvent en trop petite quantité pour enregistrer notre arbre entier.

Nous avons empiriquement opté pour la représentation suivante :

- Un tableau regroupe les LLR associés à chaque nœud de variable.
- Pour chaque nœud de parité, on alloue une structure de données contenant un tableau regroupant les LLR associés à chaque connexion dans laquelle le nœud de parité est impliqué, et un tableau de pointeurs[3] vers les éléments du tableau des LLR des nœuds de variable avec lesquels est associé le nœud de parité.
- Un tableau regroupant les indices (positions) des nœuds de variable connectés au nœud de parité. Cette information n'est pas nécessaire au décodage, mais peut être utile, par exemple pour vérifier la validité d'un mot code.

Chaque étape de l'algorithme est optimisée pour éviter les calculs redondants et choisir les opérations élémentaires les plus rapides. Cette optimisation est réalisée à l'aide d'un logiciel de profilage[4], capable d'annoter chaque ligne de code par le nombre d'appels et le temps total d'exécution. Nous avons ainsi pu détecter certaines lignes de code impliquées dans une structure de boucle, et dont le résultat reste constant pour toute itération de la boucle. Ces lignes de code redondantes sont alors factorisées, c'est-à-dire sorties de la boucle. Compte tenu de la taille de notre arbre, et par conséquent du nombre d'itérations important de chaque boucle, la factorisation de code est une optimisation très fructueuse. Elle nous a permis de gagner plusieurs ordres de grandeur sur le temps d'exécution, aussi bien de l'algorithme LDPC lui même, que sur le calcul d'intégrales gaussiennes préalable au décodage multi-niveaux, ou sur le transfert des LLR entre les niveaux. Parfois, il est nécessaire d'ordonner de façon astucieuse plusieurs boucles imbriquées afin d'exhiber la factorisation.

Au cœur de l'algorithme "Message Passing", une poignée de lignes de code réalisant le calcul de l'équation 12.16 se trouvent au centre de nombreuses boucles imbriquées. L'effort d'optimisation doit donc principalement se porter sur ces lignes[5]. Pour ces lignes de code, nous choisissons donc les opérations les plus rapides : instructions conditionnelles plutôt que multiplication par -1...

[3]Un pointeur est une adresse mémoire, qui nous permet de désigner une variable déjà définie. Ce sont ces pointeurs qui portent l'information de la structure du graphe.

[4]Valgrind, disponible à l'adresse `http://www.valgrind.org`

[5]Cet effet général est toujours surprenant, même pour l'initié : gagner 90% au prix de larges efforts sur une fraction ne représentant que 10% du tout, n'aboutit qu'à une optimisation de quelques pour cents. Cet effet est davantage marqué en algorithmique où quelques lignes de code consomment la plupart du temps d'exécution.

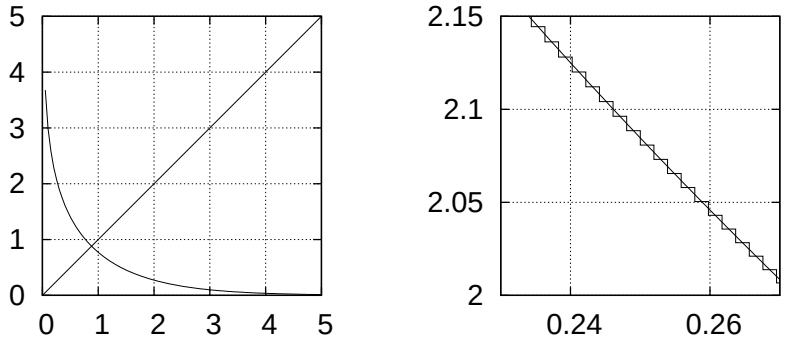


FIG. 13.7: Fonction $\phi(x) = -\ln(\tanh(x/2))$. Cette fonction est appelée au centre de toutes les boucles du décodage LDPC. C'est pour cela que nous devons choisir une méthode d'évaluation optimale pour cette fonction. Notons que cette fonction est son propre inverse : $\phi(\phi(x)) = x$. La figure de droite montre l'approximation que nous avons utilisé pour calculer rapidement la fonction ϕ.

L'algorithme "Message Passing" demande de nombreux appels à la fonction $\phi(x) = -\ln(\tanh(x/2))$, représentée figure 13.7. Nous disposons de plusieurs méthodes pour calculer cette fonction :

- utiliser les fonctions mathématiques standard ln et tanh. Cette méthode n'est pas optimale car elle nécessite deux appels à des fonctions non linéaires programmées avec une précision de 16 ou 32 bits, de $[0;\infty[$ vers $[0;\infty[$.
- approximer la fonction ϕ par une fonction linéaire par morceaux. Nous avons découpé l'intervalle $[0;10]$ contenant les valeurs significatives de ϕ en huit intervalles, sur lesquels nous définissons la meilleure approximation linéaire de ϕ. La détermination de l'intervalle à partir de la valeur d'abscisse se fait par dichotomie. Cette méthode a le désavantage d'approximer parfois grossièrement ϕ, et de parcourir plusieurs instructions conditionnelles pour trouver l'intervalle dans lequel se situe une valeur x donnée.
- calculer à l'avance un tableau de valeurs pour ϕ, dont l'index de l'entrée est une fonction linéaire de l'abcisse x. Ce tableau définit une fonction en forme d'escalier approximant ϕ. Nous avons empiriquement choisi de coder l'intervalle $[0 : 8]$ sur 12 bits, en calculant un tableau de 4 096 entrées. Cette méthode a le désavantage de nécessiter un grand espace mémoire, et les accès associés, pour stocker les valeurs pré-calculées. Aussi, l'intervalle de binarisation est de largeur constante, alors que la fonction ϕ varie rapidement pour des valeurs proches de 0.

À titre de comparaison, la première méthode calcule 10^6 valeurs de ϕ en 4 secondes (sur un ordinateur de type Pentium 4, 2,4 GHz), alors que les deux dernières méthodes mettent environ $0,4$ secondes. Cependant, la mauvaise qualité de l'approximation fournie par la deuxième méthode dégrade l'efficacité de l'algorithme de décodage. Nous avons dont finalement choisi la dernière méthode car, à vitesse égale, elle offre une approximation plus exacte de ϕ. La précision de cette approcimation est illustrée par la figure 13.7.

Les objets manipulés par l'algorithme "Message Passing" – les LLR – sont des grandeurs réelles. Nous pouvons améliorer la vitesse de décodage en utilisant une représentation à virgule fixe des nombres réels. Les architectures de processeurs rencontrées dans les stations de travail utilisent une représentation des nombres réels à virgule

flottante. Ainsi, une architecture 32 bits utilise 23 bits pour représenter les chiffres significatifs du nombre réel (la mantisse), 8 bits pour représenter l'ordre de grandeur du chiffre le plus significatif, et 1 bit pour le signe du nombre (standard IEEE 754). On peut ainsi représenter l'infiniment petit jusqu'à 10^{-127} et l'infiniment grand jusqu'à 10^{127}. Mais les processeurs manipulent en réalité des nombres entiers, et une conversion est nécessaire pour réaliser les opérations de base sur ces nombres réels. Pour améliorer la vitesse de décodage, nous avons choisi d'encoder un réel x par la partie entière de $x \times 2^{16}$. Cette représentation à virgule fixe utilise 15 bits pour représenter la partie entière de x, 16 bits pour sa partie décimale et 1 bit pour le signe. Bien entendu, nous perdons ce faisant notre capacité à représenter les nombres très petits ou très grands. Mais cet effet ne se fait pas sentir, car la perte de précision est inférieure à celle engendrée par la discrétisation de la fonction ϕ. De plus, l'algorithme de décodage ne nécessite aucune modification formelle, car l'addition de deux réels se fait simplement en additionnant leur représentation à virgule fixe. De même, la valeur absolue ou la fonction «signe» restent inchangées. Seule une adaptation de la tabulation de la fonction ϕ est nécessaire. Cette représentation à virgule fixe nous fait gagner environ un facteur 2 sur la vitesse de décodage.

Le compilateur est souvent plus adroit qu'un être humain pour optimiser un programme. Le compilateur choisit comment est allouée la mémoire, quelles variables sont gardées dans les précieux registres du processeur, peut lui même factoriser du code, rassembler plusieurs instructions en une seule, etc. Pour utiliser ces ressources, nous avons sélectionné le degré d'optimisation maximal du compilateur gcc. Au prix de quelques frustrations : certaines optimisations trouvées dans le code avaient déjà été vues par le compilateur. Notons également que le compilateur permet de produire du code optimisé pour un modèle de processeur donné, tirant parti des jeux d'instructions de haut niveau que certains des processeurs modernes fournissent. Ainsi, la vitesse de décodage, mais aussi chacune des optimisations citées ci-dessus, peuvent varier d'un processeur à l'autre, de façon souvent décorrélée de la vitesse d'horloge du processeur.

Avant d'aborder une autre famille de codes correcteurs d'erreurs – les turbo codes –, nous résumons par le tableau 13.1 les caractéristiques optimales de la réconciliation avec des codes LDPC auxquelles nous avons abouti. Le décodage de 200 000 symboles dure 2 s sur un ordinateur de type Pentium D820, soit un taux de 100 000 kHz[6]

13.7 Turbo codes

Nous avons utilisé deux familles de codes correcteurs d'erreurs mous reconnus pour leurs performances de décodage : les codes LDPC, dont nous avons traité jusque-là, et les turbo codes, auxquels nous consacrerons l'annexe A.

L'algorithme de réconciliation multi-niveaux est un assemblage de codes correcteurs d'erreurs mous indépendant du type du code utilisé. En particulier, nous pouvons utiliser comme brique de base les codes LDPC ou les turbo codes sans modifier la réconciliation multi-niveaux. Nous avons ainsi utilisé l'algorithme de décodage de turbo codes BCJR décrit en annexe A, avec différents codes convolutifs. Notre motivation pour étudier les turbo

[6]Nous avons pris ce taux comme référence pour la rédaction de ce manuscrit. Toutefois, ce taux croît avec la puissance des ordinateurs. De récents tests sur un ordinateur de type Pentium Core 2 Duo Extreme 6600 font état d'un taux de réconciliation de 170 000 KHz.

Nombre de tranches	4
Intervalle de binarisation	0,358
Taux des codes	
tranche 1	optimal : 0,001 ; choisi : 0
tranche 2	optimal : 0.012 ; choisi : 0
tranche 3	optimal : 0,457 ; choisi : 0,41
tranche 4	optimal : 0,981 ; choisi : 0,95
Itération MLC 1, tranche 3	
taux d'erreur initial	0,159
taux d'erreur final	0,0067
nombre d'itérations LDPC	29
Itération MLC 2, tranche 4	
taux d'erreur initial	0,0084
taux d'erreur final	0
nombre d'itérations LDPC	< 10
Itération MLC 1, tranche 3	
taux d'erreur initial	0.068
taux d'erreur final	0
nombre d'itérations LDPC	< 10

TAB. 13.1: Caractéristiques de la réconciliation avec des codes LDPC, et performances de chaque itération du décodage multi-niveaux (MLC) pour un rapport signal à bruit de 3. Le nombre d'itérations LDPC est obtenu en utilisant l'algorithme RMP. Pour l'algorithme SMP standard, il faut doubler ce nombre d'itérations.

codes est la disponibilité de processeurs numériques dédiés au décodage turbo, couramment utilisés dans l'électronique grand public. Malheureusement, les turbo codes que nous avons utilisés se sont avérés trop peu efficaces pour obtenir un débit non nul sur une transmission de 25 km. La figure 13.8 compare l'information extrinsèque produite par les codes LDPC à celle produite par les turbo codes. Nous voyons qu'à chaque itération entre différents niveaux, le turbo code fournit moins d'information extrinsèque, et donc facilite moins le décodage des niveaux suivants. Notons de plus que la vitesse de notre algorithme BCJR tournant sur un processeur d'ordinateur de bureau est deux ordres de grandeur en dessous de notre algorithme de décodage LDPC optimisé. Enfin, les processeurs dédiés intègrent le plus souvent un algorithme de décodage simplifié (l'algorithme de Viterbi), sacrifiant un peu d'efficacité au profit de la vitesse de décodage.

13.8 Perspectives d'amélioration

Plusieurs optimisations sont encore envisageables pour améliorer davantage le taux secret.

Les binarisations multidimensionnelles sont couramment employées pour améliorer le découpage en tranches. Cette réconciliation multidimensionnelle consiste à se placer dans un espace de grande dimension n (typiquement, n peut aller jusqu'à 30), et à définir un ensemble de 2^k points répartis judicieusement dans cet espace[7]. Les données continues prove-

[7] On parle de «constellation» dans le jargon de la théorie du codage.

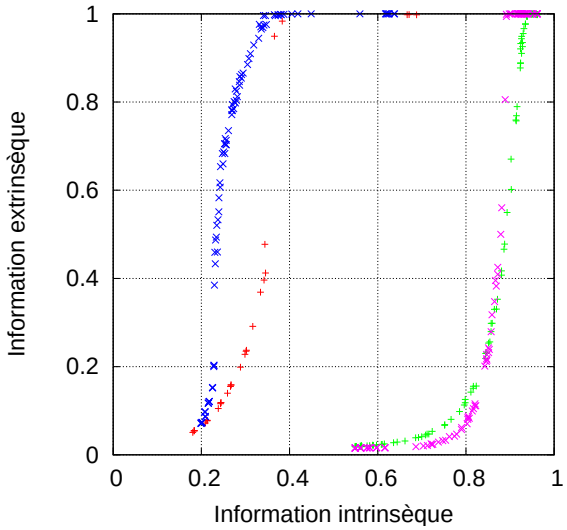


FIG. 13.8: Carte EXIT comparant les performances des codes LDPC (symboles ×) et les turbo codes (symboles +). Une carte EXIT représente l'information extrinsèque apportée par le décodeur en fonction de l'information intrinsèque qu'on lui fournit. Un décodeur est d'autant meilleur qu'il produit plus d'information extrinsèque. On fait varier l'information intrinsèque en modifiant le rapport signal à bruit autour de sa valeur nominale (3). Les courbes de gauche représentent le décodage de la tranche 3 d'un décodage multi-niveaux, avec des codes de taux 0,23. Les courbes de droite représentent le décodage de la tranche 4, avec des codes de taux 0,86. Alors que pour cette dernière tranche, les performances des codes LDPC et des turbo codes sont comparables, on remarque que, pour la tranche 3, les turbo codes fournissent moins d'information extrinsèque que les codes LDPC. Cette information extrinsèque n'est pas suffisante pour amorcer le décodage itératif multi-niveaux. Le code convolutif utilisé dans les turbo codes est représenté par le polynôme (15,13) (voir annexe A). Nous avons constaté un résultat similaire pour les turbo codes (7,5) (23,35) (37,21) (255,223) et duo binary (23,35,20,32) [74].

nant du canal quantique sont regroupées en vecteurs de n composantes, et chaque vecteur est binarisé par le point le plus proche de ce vecteur dans l'espace à n dimensions. Chaque vecteur fournit ainsi k bits, définissant k tranches, que nous pouvons décoder comme nous le faisons dans le cas unidimensionnel. En général, à haute dimension, le maillage optimal n'est pas régulier : à première vue, la constellation semble être composée de points aléatoirement répartis ; dans cette configuration, il est difficile et coûteux de trouver le point le plus proche de notre vecteur de données. Une constellation idoine doit donc être trouvée pour optimiser l'efficacité de réconciliation sans sacrifier la vitesse.

Deux effets peuvent être recherchés. Tout d'abord, une bonne discrétisation peut, comme nous l'avons vu dans ce chapitre, conduire à une bonne efficacité de réconciliation β_{tranches}, actuellement limitée à 95%. Ensuite, on peut envisager de garder une efficacité constante, pour profiter d'une réconciliation plus rapide. En effet, notre maillage unidimensionnel est équivalent, en termes de vitesse et d'efficacité, à un maillage «hyper-cubique» à n dimensions. Dans ce maillage hyper-cubique, beaucoup d'intervalles de binarisation, notamment dans les intervalles éloignés de l'origine, ont une probabilité très faible d'accueillir des données continues. Un maillage multidimensionnel de type polaire permettrait d'éliminer ces intervalles de binarisation inutilisés, améliorant ainsi la vitesse de décodage.

Nous pouvons trouver une structure de codes LDPC adaptée à la distribution des symboles issus de la discrétisation de nos données continues. Nous évoqué une propriété intéressante des codes LDPC : l'efficacité de décodage ne dépend que de la connectivité du graphe représentant le code. La connectivité optimale dépend de la distribution des LLR intrinsèques fournis par le canal de communication. Nous avons utilisé des connectivités optimisées pour des distributions de LLR gaussiennes, comme celles rencontrées sur les canaux AWGN avec une modulation binaire[8]. Si notre distribution est proche d'une gaussienne lors des dernières étapes de décodage, elle s'en écarte significativement au début du décodage, où l'efficacité est critique pour la réussite du décodage itératif (voir figure 12.7). L'utilisation de codes optimisés permettrait donc d'augmenter l'efficacité de décodage, actuellement limité à 93%.

La programmation de processeurs numériques spécialisés permettrait d'améliorer la vitesse de réconciliation. Les processeurs DSP[9] sont optimisés pour le traitement numérique du signal. Ils sont notamment couramment utilisés pour le décodage de codes correcteurs d'erreurs mous dans les appareils électroniques grand publique. Nous avons testé l'un de ces processeurs : le DPS TMS320C6416 cadencé à 1 GHz de Texas Instruments, monté sur une carte de test. Comme ce processeur se programme directement en C, nous pouvons utiliser notre algorithme de décodage LDPC existant, moyennant quelques modifications mineures. Malheureusement, nous avons obtenu des vitesses de décodage un ordre de grandeur en dessous de celles obtenues avec un ordinateur de bureau standard. En effet, les DSP sont particulièrement adaptés aux algorithmes demandant beaucoup de calculs pour peu d'accès mémoire, comme le décodage de codes moins complexes et donc moins efficaces (codes LDPC réguliers, mais surtout turbo codes) que les codes que nous utilisons. Les performances de ces processeurs pour le décodage des turbo codes avait d'ailleurs motivé notre intérêt pour ces codes. Ces propriétés répondent en effet aux motivations industrielles, qui favorisent les décodages peu efficaces, mais rapides et peu gourmands en puissance électrique. Toutefois, plusieurs projets récents explorent l'implémentation des codes LDPC irréguliers sur DSP ou FPGA [75].

Nous envisageons également l'utilisation de cartes graphiques. Ces cartes, usuellement dédiées au rendu graphique de jeux vidéo, sont de puissants calculateurs parallèles à virgule flottante. Nous avons adapté le décodage des codes LDPC à ces processeurs graphiques, et des résultats préliminaires font état d'une vitesse de décodage identique à celle obtenue avec un puissant ordianteur, pour un encombrement réduit.

L'utilisation de modulations non gaussiennes peut augmenter le taux secret final. Les preuves de sécurité présentées au chapitre 4 ne nécessitent en rien l'utilisation de modulations gaussiennes. Nous avons pourtant utilisé ces modulations, car elles permettent d'exprimer analytiquement le taux secret en fonction des paramètres du canal et car elles sont optimales pour une efficacité de réconciliation $\beta = 1$ [36]. Cependant, pour une réconciliation imparfaite ($\beta < 1$), nous pouvons trouver des modulations non gaussiennes qui améliorent le taux secret. Considérons les deux modèles de sécurité suivants :

Attaques individuelles Les preuves entropiques de la sécurité du protocole de distribution quantique de clé utilisant la réconciliation inverse [36] donnent une expression du taux

[8]Rappelons qu'un canal AWGN est un canal ajoutant un bruit gaussien. Une modulation binaire encode les bits 0 et 1 sur deux symboles -1 ou 1 introduits en entrée du canal.

[9]Digital Signal Processor.

secret indépendante de la modulation utilisée, que nous avions écrite au chapitre 4 (nous reprenons ici les notations de ce chapitre) :

$$\Delta I \geq 2H_0 - (1-\beta)H(X_B) - \beta H(X_B|X_A) - H(P_B|P'_A). \tag{13.4}$$

Pour $\beta = 1$, cette expression est optimale quand la modulation est gaussienne. Pour $\beta < 1$ dépendant de la modulation, un algorithme d'optimisation numérique pourrait permettre de trouver une modulation non gaussienne fournissant un taux secret supérieur.

Attaques collectives À ce jour, les preuves de sécurité prenant en compte les attaques collectives ne fournissent pas d'expression explicite pour le taux secret quand la modulation n'est pas gaussienne. Cependant, le taux secret est borné par

$$\Delta I = \beta I_{AB} - \chi_{BE}^{gauss}, \tag{13.5}$$

où χ_{BE}^{gauss} est l'entropie de Holevo gaussienne entre Bob et Ève, qu'Alice et Bob déterminent en calculant les moments d'ordre deux de la distribution de leurs données (voir chapitre 4). Pour des paramètres du canal constants, c'est-à-dire à moment d'ordre deux constants, χ_{BE}^{gauss} est constant, et ne dépend pas de la forme de la distribution des données. Si $\beta \leq 1$ est également constant, la modulation gaussienne maximise ΔI, car elle maximise l'information mutuelle I_{AB} à variance constante. Mais β peut dépendre du type de modulation utilisée : certaines modulations non gaussiennes peuvent avoir une efficacité ou une vitesse supérieures. Là encore, une optimisation numérique est nécessaire pour déterminer la modulation optimale. Cependant, si une amélioration est possible, elle sera limitée : avec notre modulation gaussienne, I_{AB} est optimale, et β vaut environ 87%. Il faut donc trouver une modulation dont l'information mutuelle s'écarte de moins de 15% d'une modulation gaussienne, tout en présentant une efficacité proche de 100%.

Chapitre 14

Amplification de confidentialité

À l'issue de la réconciliation, Alice et Bob partagent un série de bits sans erreur, mais partiellement connue de l'espion. Par exemple, un décodage de 200 000 impulsions sur 4 tranches, dont deux sont totalement révélées produit 400 000 bits contenant environ 11 000 bits secrets pour une transmission sur 25 km. L'amplification de confidentialité a pour but de générer le nombre désiré de bits secrets à partir des bits issus du décodage.

Dans ce chapitre, nous reprendrons simplement les résultats obtenus par Gilles Van Assche [7], et nous montrerons comment les étendre à des blocs de données de taille arbitraire, notamment pour des entrées de 400 000 bits. Enfin, nous montrerons comment combiner plusieurs fonctions de hachage afin d'améliorer la vitesse de l'algorithme d'amplification.

14.1 Familles de fonctions de hachage comme amplificateurs de confidentialité

Dans ce chapitre, nous utiliserons les notations suivantes :

- la chaîne de bits à amplifier est notée B. Sa longueur est notée b. Pour notre algorithme de réconciliation, $b = 400\ 000$.
- la chaîne de bits issue de l'amplification (c'est-à dire la clé secrète) est notée K, et sa longueur est k. En général, on choisit $k = \Delta I - s$, où ΔI est le nombre de bits secrets contenus dans la chaîne à amplifier, et s est une marge de sécurité (voir section 14.6).

Une fonction de hachage est une fonction de l'ensemble des chaînes de b bits vers l'ensemble des chaînes de k bits. [1] En d'autres termes, on peut voir le hachage de b bits (c'est-à-dire l'image d'une chaîne de b bits par une fonction de hachage) comme un échantillon représentatif des b bits. Si la fonction de hachage est correctement choisie, deux chaînes distinctes auront un hachage différent avec une bonne probabilité. On rencontre ces fonctions de hachage dans la vie courante pour garantir l'intégrité des données, par exemple dans les systèmes de vérification de CD, ou dans la clé RIB.

L'amplification de confidentialité nécessite de définir au préalable un ensemble de fonctions de hachage appelé «famille». Pour amplifier leur chaîne B en une chaîne K, Alice et Bob appliquent à B une fonction de hachage choisie aléatoirement parmi cette famille, et différente

[1] En toute rigueur, on peut définir des fonctions de hachage dans un ensemble quelconque. Nous nous limiterons ici aux chaînes de bits.

pour chaque chaîne à amplifier. Il est prouvé que ce processus conduit inconditionnellement à une chaîne secrète K, à condition que la famille de fonctions de hachage soit universelle (ou proche de l'universalité, voir section 14.6), selon la définition suivante : une famille de fonctions de hachage est universelle si, pour toutes chaînes d'entrée différentes B_1 et B_2, la probabilité pour que $f(B_1) = f(B_2)$ est au plus $1/2^b$. Dans cette définition, le terme probabilité s'entend «quand f parcourt la famille de fonctions de hachage» [60]. Une fonction est dite ϵ-universelle quand cette probabilité est au plus $\epsilon/2^b$, avec[2] $\epsilon \geqslant 1$.

Une famille de fonctions de hachage universelle doit vérifier plusieurs contraintes pour être utilisable en pratique dans une situation d'amplification de confidentialité [7] :

- La famille de fonctions de hachage doit être conçue pour accepter des entrées de b bits et des sorties de k bits telles que définies par notre protocole de réconciliation. Notamment, la taille de sortie k doit être variable, car elle dépend du taux secret calculé à partir des paramètres du canal quantique.
- L'évaluation de la fonction de hachage doit être rapide, pour ne pas dégrader la vitesse d'établissement de la clé secrète.
- Alice et Bob doivent s'accorder via le canal classique sur le choix d'une fonction de hachage parmi cette famille. La famille doit donc être assez petite pour que la désignation d'une fonction requière un nombre raisonnable de bits.

14.2 Une petite excursion dans le monde des corps finis

En cryptographie classique, beaucoup de résultats reposent sur la notion de corps finis. C'est le cas des codes correcteurs d'erreurs BCH que nous avons déjà rencontrés sans détailler leur fonctionnement (section 12.1). Un corps est un ensemble mathématique dans lequel on définit deux lois, traditionnellement appelées «addition» et «multiplication». Ces lois sont associatives et possèdent un élément neutre (0 pour l'addition et 1 pour la multiplication) ; la multiplication est distributive pour l'addition. Tout élément possède un inverse pour l'addition et pour la multiplication (sauf 0) : cette propriété, qui est à l'origine de l'utilité pratique des corps, permet de définir la soustraction et la division. Un corps est entièrement défini par ses tables d'addition et de multiplication, indépendamment de sa représentation, c'est-à-dire de la façon dont on nomme ses éléments. Deux corps sont dits «isomorphes» quand ils ont des tables identiques.

Un corps fini est un corps comportant un nombre fini d'éléments. On peut montrer deux résultats importants sur les corps finis :

- Les corps finis ont p^n éléments, où p est un nombre premier et n un nombre entier. Ainsi, il existe des corps finis comportant $3 = 3^1$ ou $9 = 3^2$, mais il n'existe pas de corps de $6 = 3 \times 2$ éléments.
- Tous les corps finis de même cardinal sont isomorphes. Autrement dit, pour un nombre donné d'éléments, il suffit de connaître un corps pour connaître tous les autres. Le corps fini comportant p^n éléments est noté $GF(p^n)$.

[2] Il existe plusieurs conventions pour définir ϵ. Avec notre définition, $\epsilon = 1$ indique l'universalité de la famille.

Il existe une représentation simple de chaque corps fini sous forme de polynômes. Explorons tout d'abord le cas où $n = 1$: le corps $GF(p)$ n'est autre que $\mathbb{Z}/p\mathbb{Z}$, l'ensemble des nombres entiers de 0 à $p-1$, dans lequel l'addition et la multiplication s'effectuent modulo p. Cette représentation de $GF(p)$ permet de construire une représentation de $GF(p^n)$: la représentation polynomiale de $GF(p^n)$ est l'ensemble des polynômes de degré strictement inférieur à n, à coefficients dans $GF(p)$. Chacun des n coefficients du polynôme peut prendre p valeurs, soit un total de p^n éléments : le compte est bon. Pour compléter notre représentation, il faut maintenant définir les deux lois du corps :

- l'addition est l'addition usuelle des polynômes : les coefficients de chacun des polynômes s'ajoutent terme à terme dans $GF(p)$, c'est-à-dire modulo p.
- la multiplication, quant à elle, pose problème, car le produit de deux polynômes de degré $n-1$ est un polynôme de degré $2n-2 > n-1$. La multiplication usuelle des polynômes ne convient donc pas comme loi de multiplication dans $GF(p^n)$. Cette difficulté est résolue de la manière suivante : la multiplication dans la représentation polynomiale de $GF(p^n)$ est la multiplication usuelle des polynômes, suivie d'une réduction du résultat modulo f, où f est un polynôme irréductible[3] de degré n à coefficients dans $GF(p)$. Cette réduction permet d'éliminer les termes de degré n ou plus apparus au cours de la multiplication polynomiale.

Nous nous intéresserons plus particulièrement aux corps de type $GF(2^n)$ dont la représentation polynomiale est l'ensemble des polynômes de degré inférieur à n, à coefficients binaires (ou bits). On représente également $GF(2^n)$ par l'ensemble des nombres de n bits écrits sous forme binaire, chaque bit représentant un coefficient du polynôme. Dans cette dernière représentation, l'addition est l'opérateur logique «ou exclusif» bit à bit. La multiplication n'a pas de représentation simple en termes d'opérateurs logiques. On privilégie donc la représentation polynomiale quand il s'agit de multiplier des éléments de $GF(2^n)$.

Terminons par un exemple illustrant l'addition et la multiplication dans $GF(2^5)$, avec pour polynôme irréductible $f(X) = X^5 + X^2 + 1$:

$$\begin{aligned}
a(X) &= X^4 + X^3 + X^2 + 1 \in GF(2^5) && (14.1)\\
b(X) &= X^3 + X + 1 \in GF(2^5) && (14.2)\\
a(X) + b(X) &= X^4 + X^2 + X && (14.3)\\
a(X) \times b(X) &= X^7 + X^6 + X^3 + X^2 + X + 1 \quad [f(X)] && (14.4)\\
&= (X^2 + X)f(X) + X^4 + 1 \quad [f(X)] && (14.5)\\
&= X^4 + 1 && (14.6)
\end{aligned}$$

14.3 Des exemples simples de familles de fonctions de hachage

La famille de fonctions de hachage la plus immédiate est l'ensemble des fonctions des chaînes de b bits (soit $GF(2^b)$ puisque nous sommes maintenant savants !) vers les chaînes de k bits ($GF(2^k)$). Malheureusement, comme elle contient 2^{k2^b} éléments, son utilisation demanderait de transmettre $k2^b$ bits[4] entre Alice et Bob avant chaque amplification, pour définir la fonction de

[3]Un polynôme irréductible est un polynôme qui ne peut pas s'écrire comme le produit de deux polynômes. Le caractère irréductible de f est nécessaire pour que la multiplication dans $GF(p^n)$ soit inversible.

[4]Soit environ $2^{400\,013}$ bits si $b = 400\,000$ et $k = 11000$.

l	s
127	1 (ou 7, 15, 30 et 63)
521	32 (ou 48, 158 et 168)
607	105 (ou 147 et 273)
1279	216 (ou 418)
2281	715 (ou 915 et 1029)
3217	67 (ou 576)
4423	271 (ou 369, 370, 649, 1393, 1419 et 2098)
9689	84 (ou 471, 1836, 2444 et 4187)
19937	881 (ou 7083 et 9842)
23209	1530 (ou 6619 et 9739)
44497	8575 (ou 21034)
110503	25230 (ou 53719)
132049	7000 (ou 33912, 41469, 52549 et 54454)
756839	279695 (ou 215747 et 267428)
859433	288477 (ou 170340)
3021377	361604 (ou 1010202)
6972593	3037958

TAB. 14.1: Liste de trinômes irréductibles à coefficients dans $GF(2)$ de la forme $X^l + X^s + 1$. Ces polynômes permettent de définir une représentation polynomiale de $GF(2^l)$. La recherche de polynômes irréductibles de grand degré est une tâche difficile. Brent *et al.* [76] ont conçu un algorithme permettant de tester l'irréductibilité des trinômes dont le degré l est tel que $2^l - 1$ est premier (exposant de Mersenne). Nous reproduisons ici l'ensemble des trinômes irréductibles pour ces valeurs particulières de l.

hachage utilisée parmi cette famille. Nous devons donc trouver des familles moins nombreuses...

Une famille de fonctions de hachage conceptuellement très simple est la multiplication du vecteur de bits d'entrée par une matrice binaire arbitraire :

$$\mathcal{H}_1 = \{h_M(B) = M \cdot B, M \text{ matrice binaire de taille } k \times b\} \tag{14.7}$$

Cette multiplication est une multiplication binaire terme à terme, comme celle que nous avons vue pour les "parity checks". Ainsi, l'ensemble des "parity check" de taille k sur un ensemble de b bits est une famille de fonctions de hachage universelle [60]. Cette famille permet de choisir des tailles d'entrée et de sortie arbitraires. Bien que séduisante au premier abord, elle possède deux inconvénients majeurs :

- le nombre de bits à transmettre entre Alice et Bob pour caractériser un élément de la famille est égal au nombre d'éléments de la matrice. Pour $b = 400\,000$ et $k = 11\,000$, cela représente une transmission de 500 Moctets.
- le temps de calcul d'une multiplication matricielle est quadratique en la taille du vecteur d'entrée. Ainsi, cette opération se prête mal à l'amplification d'un grand nombre de bits.

La multiplication par un vecteur arbitraire de $GF(2^l)$ est une famille de fonctions de hachage universelle. Sa taille d'entrée est de l bits. Pour ajuster la taille de la clé produite,

on prélève k bits quelconques[5] des l bits du résultat de la multiplication :

$$\mathcal{H}_2 = \left\{ h_c(x) = cx|_{k \text{ premiers bits}}, c \in GF(2^l) \right\}. \tag{14.8}$$

La représentation binaire de $GF(2^l)$ nous permet d'utiliser directement les b bits issus de la réconciliation, car un nombre binaire de b bits est un élément de $GF(2^b)$. Mais pour définir la multiplication dans $GF(2^l)$, nous devons mettre la main sur un polynôme irréductible de degré l. Or, aucun polynôme irréductible de degré 400 000 n'est connu à ce jour... Cependant, Brent *et al.* [76] ont trouvé quelques trinômes irréductibles de degré élevé (voir tableau 14.1). Il nous suffit donc d'utiliser un de ces polynômes tel que $l > b$ pour définir le groupe multiplicatif de $GF(2^l)$. L'entrée de la multiplication est dans ce cas les b bits à amplifier concaténés avec $l - b$ bits non significatifs[6]. Les fonctions de hachage employées pour la démonstration de cryptographie quantique présentée dans [14] utilisaient un trinôme de degré $l = 110503$. Le nouveau protocole de réconciliation présenté chapitre 12 produisant des séquences de $b = 400000$ bits à amplifier, nous utilisons un trinôme de degré $l = 756839$.

Cette famille de fonctions de hachage fait un usage modéré du canal de communication reliant Alice à Bob, puisqu'un élément de la famille est caractérisé par un élément de $GF(2^l)$ qui nécessite la transmission de l bits. Cependant, l'algorithme "Shift and add"[7] traditionnellement utilisé pour réaliser la multiplication dans $GF(2^l)$ est caractérisé par un nombre d'opérations qui varie quadratiquement avec l. Nous emploierons la section suivante à la description d'un algorithme permettant la multiplication rapide dans $GF(2^l)$.

14.4 Multiplication rapide dans $GF(2^l)$ utilisant la transformée de Fourier discrète

Une méthode numérique, décrite dans cette section, permet de multiplier rapidement des polynômes de $GF(p)[X]/(X^L - 1)$, où p est un nombre premier et L un nombre entier quelconques. La notation $GF(p)[X]/(X^L - 1)$ indique qu'après la multiplication polynômiale usuelle, les coefficients du polynôme produit sont réduits modulo p, et les termes de degré supérieur ou égal à L sont ramenés à des termes de degré inférieur à L. Cette multiplication peut être utilisée pour multiplier rapidement deux polynômes de $GF(2^l)$. En effet, on vérifie simplement que le produit de deux polynômes de degré inférieur à l à coefficients binaires produit un polynôme de degré inférieur à $2l$ et à coefficients inférieurs à l. Donc si nous choisissons $p > l$ et $L > 2l$, la multiplication dans $GF(2^l)$ et la multiplication dans $GF(p)[X]/(X^L - 1)$ sont strictement identiques, car les réductions modulo p et modulo L n'ont jamais lieu. Nous allons maintenant décrire la multiplication rapide dans $GF(p)[X]/(X^L - 1)$.

On peut définir dans le corps $GF(p)$ des opérations similaires à celles rencontrées dans le corps des nombres complexes. Notamment, on peut définir les L racines Lièmes de l'unité

[5]La position des bits prélevés n'importe pas. Dans notre implémentation, nous gardons les k premiers bits du produit.

[6]Dans notre implémentation, ces bits sont fixés à 0. Ces bits ne jouent aucun rôle dans l'amplification, car étant connus de tout le monde, ils ne changent par la différence entre I_{AB} et I_{BE}. Ces bits ajoutés sont équivalents aux bits classiques ayant transité par le canal classique.

[7]L'algorithme "Shift and add" intercale la multiplication et la réduction. Il calcule de façon récursive le produit de deux polynômes $A(X)$ et $B(X)$, en utilisant l'écriture $A(X) \times B(X) = a_0 B + X(a_1 B + X(\cdots X(a_{n-1} B + X(a_n B)) \cdots))$, en retranchant, si nécessaire, $f(X)$ à chaque étape.

$\{\omega^i, i = 1 \dots L-1\}$. Muni de ces racines de l'unité, on peut définir, par analogie avec la transformée de Fourier discrète, la transformée de Fourier discrète dans $GF(p)$ ou NTT ("Number Theoretic Transform") d'un vecteur V de L éléments de $GF(p)$:

$$\mathcal{F}(V)_i = \sum_{j=1}^{L-1} v_j \omega^{ij} \quad \text{et son inverse} \quad \mathcal{F}^{-1}(V)_i = \frac{1}{N} \sum_{j=1}^{L-1} v_j \left(\omega^{-1}\right)^{ij} \tag{14.9}$$

De façon similaire à l'algorithme de transformée de Fourier rapide, il existe un algorithme permettant de réaliser une NTT en un nombre d'opérations proportionnel à $L\log(L)$**,** pour les cas particuliers où L s'écrit $L = 2^n$ et p s'écrit $p = \nu L + 1$. Le tableau 14.2 recense plusieurs de ces configurations pour lesquelles la NTT rapide est possible. De plus, la NTT préserve le comportement de la transformée de Fourier vis-à-vis de la convolution : $\mathcal{F}(U * V) = \mathcal{F}(U) \cdot \mathcal{F}(V)$, où le produit $\cdot$ s'entend terme à terme[8]. Ces deux propriétés rendent possible la multiplication rapide de polynômes de $GF(p)[X]/(X^L - 1)$: on montre simplement que cette multiplication n'est autre qu'une convolution entre les coefficients respectifs des deux polynômes. Le résultat de la multiplication s'obtient donc en trois étapes : on calcule la NTT des L coefficients pour chacun des deux polynômes à multiplier ; on multiplie terme à terme les deux résultats ; enfin, on calcule la NTT inverse de ce produit.

Résumons les opérations que nous devons effectuer pour amplifier une chaîne de b bits en une chaîne de k bits en utilisant la multiplication dans $GF(2^l)$ accélérée par la NTT :

- Nous avons $b = 400\,000$ bits à amplifier.
- Nous les complétons par des 0 jusqu'à obtenir $l = 756\,839$ bits à amplifier (l'«entrée»).
- Alice et Bob s'échangent un nombre de l bits aléatoires (le «défi») pour désigner la fonction de hachage qu'il vont appliquer.
- Nous calculons les NTT de l'entrée et du défi, avec comme paramètres $L = 2^{21} = 2\,097\,152 > 2l$ et $p = 23\,068\,673 > l$.
- Nous multiplions terme à terme les résultats des deux NTT. Nous calculons la NTT inverse de ce produit.
- Nous réduisons chacun des coefficients obtenus modulo 2. Nous réduisons les l bits ainsi générés par le polynôme irréductible $X^{756839} + X^{279695} + 1$.
- Enfin, nous prélevons les k premiers bits du résultat. Ces k bits constituent une clé secrète.

Nous avons programmé l'algorithme ci-dessus, fournissant des familles de fonctions de hachage ayant des entrées et sorties de tailles arbitraires. Les données collectées dans les tableaux 14.1 et 14.2 permettent à notre programme d'amplifier jusqu'à 6 972 593 bits.

14.5 Famille de fonctions de hachage utilisant directement la NTT

La famille de fonctions de hachage que nous avons décrite dans la section précédente ne semble pas optimale. En effet, elle oblige à amplifier $l - b$ bits non nécessaires, et demande des opérations de NTT sur L nombres de $\log_2(p)$ bits. Pour $L = 2^{21}$ et $p = 23\,068\,673$, on doit donc manipuler une quantité de données de 50 Mégabits, de deux ordres de grandeur supérieure au

[8] On rappelle que U, V, $\mathcal{F}(U)$ et $\mathcal{F}(U)$ sont des vecteurs d'éléments de $GF(p)$.

n	L	ν	p	g	ω	ω^{-1}	$L\lfloor\log_2(p)\rfloor$
0	1	1	2	1	1	1	1
1	2	1	3	2	2	2	2
2	4	1	5	2	2	3	8
3	8	2	17	3	9	2	32
4	16	1	17	3	3	6	64
5	32	3	97	5	28	52	192
6	64	3	193	5	125	105	448
7	128	2	257	3	9	200	1024
8	256	1	257	3	3	86	2048
9	512	15	7681	17	7146	7480	6144
10	1024	12	12289	11	10302	8974	13312
11	2048	6	12289	11	1945	4050	26624
12	4096	3	12289	11	1331	7968	53248
13	8192	5	40961	3	243	15845	122880
14	16384	2065*	33832961	3	21257378	2581307	409600
14	16384	4	65537	3	81	8091	262144
15	32768	2	65537	3	9	7282	524288
16	65536	1	65537	3	3	21846	1048576
17	131072	6	786433	10	213567	430889	2490368
18	262144	3	786433	10	1000	710149	4980736
19	524288	11	5767169	3	177147	5087924	11534336
20	1048576	7	7340033	3	2187	4665133	23068672
21	2097152	11	23068673	3	177147	17187657	50331648
22	4194304	25	104857601	3	39193363	96987805	109051904
23	8388608	20	167772161	3	131341181	16470339	226492416
24	16777216	10	167772161	3	59049	149602455	452984832
25	33554432	5	167772161	3	243	114609789	905969664
26	67108864	7	469762049	3	2187	410692747	1879048192

TAB. 14.2: Liste de paramètres pour lesquels il est possible d'employer l'algorithme de NTT rapide. La NTT transforme un vecteur de L valeurs de $GF(p)$. La NTT rapide nécessite que L soit une puissance de 2 ($L = 2^n$), et que p s'écrive $p = \nu L+1$. Pour chaque valeur de n, nous avons consigné dans ce tableau la plus petite valeur de ν possible. Le générateur g est le plus petit entier tel que $g^i \neq 1, \quad \forall i < p-1$. Il permet de déterminer la première Lième racine de l'unité $\omega = g^\nu$, d'inverse ω^{-1}. À n donné, nous avons intérêt à prendre p (et donc ν) le plus petit possible. Toutefois, pour la famille de fonctions de hachage présentée section 14.5, la grandeur pertinente est $L\lfloor\log_2(p)\rfloor$ (dernière colonne). Dans ce cas, il peut être avantageux d'utiliser une valeur de ν supérieure à sa valeur maximale (entrées marquées d'une étoile *).

nombre initial de 400 000 bits. De plus, les premières étapes demandent le calcul de NTT d'un vecteur aléatoire : il est conceptuellement inutile de «mélanger» de l'aléa.

Nous avons programmé une nouvelle famille de fonctions de hachage, introduite par Gilles Van Assche dans [7], se fondant sur la NTT, constituant fondamental de la famille de hachage $\mathcal{H}_2$:

$$\mathcal{H}_3 = \left\{ h_v(x) = \mathcal{F}^{-1}(x \cdot v)|_{k \text{ premiers bits}}, v \text{ vecteur de } L \text{ éléments non nuls de } GF(p) \right\}. \quad (14.10)$$

Notons que l'entrée de ces fonctions de hachage n'est plus une série de bits, mais L nombres entiers compris entre 0 et $p-1$. Une première étape dans l'utilisation de cette famille consiste à transformer notre chaîne de b bits. Pour ce faire, nous créons des entiers en regroupant nos bits par paquets de $\lfloor \log_2(p) \rfloor$. Comme précédemment, nous complétons la chaîne d'entrée par des 0, si nécessaire. En sortie, nous extrayons $\lfloor \log_2(p) \rfloor$ bits par entier jusqu'à obtention de nos k bits secrets. $\log_2(p)$ n'étant pas entier, nous perdons une partie de la capacité d'amplification de l'algorithme, perte largement compensée par la simplicité de la méthode.

Pour spécifier un élément parmi cette famille, Alice et Bob doivent s'échanger $L\lceil \log_2(p) \rceil$ bits sur le canal classique. Cette famille de fonctions de hachage n'est pas universelle, mais seulement $\left(\frac{p}{p-1}\right)^k \sim 1 + \frac{k}{p}$ universelle [7] : c'est un désavantage par rapport à la famille précédente. Nous verrons dans le section suivante comment prendre en compte l'universalité d'une famille de fonctions de hachage.

Pour utiliser cette famille, nous devons déterminer les valeurs de p et L, liées par la relation $L\lfloor \log_2(p) \rfloor > b$. Nous avons tout intérêt à choisir L le plus petit possible (car le temps nécessaire à la NTT croît avec L), et p grand (car la famille se rapproche de l'universalité quand p augmente). Nous avons choisi les paramètres $L = 2^{14}$ et $p = 33\,832\,961$, soit $L\lfloor \log_2(p) \rfloor = 409\,600$. En choisissant $L = 2^{13}$, nous aurions dû utiliser un nombre premier p de 48 bits, incompatible avec notre implémentation de la NTT sur 32 bits. L'utilisation d'un processeur 64 bits permettrait d'atteindre ce domaine, mais pour un gain marginal, car le temps nécessaire à la NTT est déjà négligeable devant le temps nécessaire à la transmission et à l'initialisation des bits. En encodant plus astucieusement nos bits, nous pourrions amplifier jusqu'à $L \log_2(p) = 409\,705$: la quantité de bits que nous amplifions est très proche du nombre de bits que manipule l'algorithme, comparativement à la famille précédente. Avec ces paramètres, nous devons transmettre $L\lceil \log_2(p) \rceil = 425\,984$ bits sur le canal classique ; notre famille est $1 + \frac{k}{p} = 1,00033$-universelle.

14.6 Universalité et paramètre de sécurité

L'amplification de confidentialité avec une famille de fonctions de hachage universelle est inconditionnellement sûre, en vertu du théorème [60] :

$$H(K|h, B_E) \geq k - \frac{2^{-s}}{\ln(2)}, \quad (14.11)$$

où $H(K|h, B_E)$ est l'entropie de la clé sachant la fonction de hachage choisie par Alice et Bob et la chaîne obtenue par Ève à l'issue de la réconciliation, et $s = \Delta I - k$ est la différence entre le nombre de bits secrets ΔI[9] contenus dans nos données réconciliées et la taille k de la clé finale.

[9] En toute généralité, ce théorème n'est pas valable pour l'entropie de Shannon, mais pour l'entropie de Rényi d'ordre 2 définie par $H_2(X) = -\log_2\left(\sum_{x \in X} p^2(X = x)\right)$. Cependant, on montre que ces deux entropies sont identiques pour des blocs de grande taille [77, 78].

Famille	Temps	Universalité
Multiplication dans $GF(2^l)$ avec l'algorithme "Shift and Add"	14 000 s $\simeq$ 4 h	1
Multiplication dans $GF(2^l)$ avec la NTT	13,7 s	1
Fonctions de hachage utilisant directement la NTT	0,08 s	$1,00033$
Composition des deux familles précédentes	0,27 s	$1 + 10^{-2690}$

TAB. 14.3: Comparaison des vitesses des différentes fonctions de hachage présentées dans ce chapitre. Nous amplifions 400 000 bits en 11 000 bits secrets en utilisant un ordinateur de type Pentium IV, cadencé à 2,66 GHz. Les deux premières entrées concernent la famille $\mathcal{H}_2$, avec la multiplication quadratique usuelle dans $GF(2^l)$, et avec la multiplication via la NTT. Bien entendu, ces deux procédures fournissent la même clé à l'issue de l'amplification de confidentialité. La troisième entrée mesure la vitesse des fonctions de hachage de la famille $\mathcal{H}_3$ introduite section 14.5. Cette famille permet de gagner plusieurs ordres de grandeur sur le temps d'exécution, au prix de la perte de l'universalité de la famille de fonctions. Il est à noter que dans cette dernière mesure, les temps d'initialisation et de conversion de bits en valeurs entières est prépondérant devant le temps d'exécution de la NTT. La composition des familles $\mathcal{H}_2$ et $\mathcal{H}_3$ permet de se rapprocher suffisamment de l'universalité, pour une faible perte en termes de vitesse. Pour les trois premiers cas, la vitesse d'amplification ne dépend que de la taille d'entrée. Pour le dernier cas, le temps de calcul est une fonction bilinéaire des tailles d'entrée et de sortie.

s est appelé «paramètre de sécurité». Ce théorème affirme que la fraction de la clé finale connue par l'espion est arbitrairement petite. Les s bits sacrifiés quantifient l'information accessible à l'espion, fixée par l'utilisateur final de la clé.

Ce théorème s'étend aux familles de fonctions de hachage non-universelles [7] :

$$H(K|h, B_E) \geq k - \log_2(\epsilon) - \frac{2^{-s-\log_2(\epsilon)}}{\ln(2)}. \tag{14.12}$$

Considérons le cas où la famille est proche de l'universalité (*i.e.* $\epsilon - 1 \ll 1$). L'équation précédente devient

$$H(K|h, B_E) \geq k - \frac{2^{-s} + \epsilon - 1}{\ln(2)}. \tag{14.13}$$

On remarque que la non-universalité d'une famille de fonctions de hachage vient contrer l'effet du paramètre de sécurité. Quantitativement, la non-universalité limite les valeurs possibles du paramètre de sécurité à $s < -\log_2(\epsilon - 1)$. Pour la famille non-universelle introduite dans la section précédente, on calcule $s < 11$, alors que les paramètres de sécurité couramment choisis atteignent 128 ou 256.

Il est possible d'améliorer l'universalité d'une famille en composant plusieurs fonctions de hachage, sans pour autant dégrader significativement la vitesse d'amplification. La composition de deux fonctions de hachage consiste à amplifier une première fois les b bits issus du décodage en i bits intermédiaires avec une première famille $\mathcal{H}_a$ ϵ_a - universelle, puis à amplifier ces i bits en k bits avec une famille $\mathcal{H}_b$ ϵ_b - universelle. On montre simplement [79] que l'universalité de la famille de fonctions de hachage composée est $\epsilon = \epsilon_a 2^{k-i} + \epsilon_b$. En choisissant une famille $\mathcal{H}_b$ universelle ($\epsilon_b = 1$) et $\epsilon_a \simeq 1$, tout paramètre de sécurité s peut être atteint en choisissant $i > k + s$.

Illustrons cette propriété en composant les familles de fonctions de hachage rencontrées dans ce chapitre. Partant de 400 000 bits initiaux, nous appliquons notre fonction de hachage $\mathcal{H}_3$ non-universelle mais rapide pour obtenir 19 937 bits. Ensuite nous appliquons sur ces bits la fonction de hachage universelle mais lente $\mathcal{H}_2$ pour obtenir 11 000 bits secrets. La taille d'entrée de cette fonction (19 937 bits) est choisie d'après les polynômes irréductibles connus. Finalement, l'universalité de la fonction composée (notée $\mathcal{H}_4$) est $\epsilon = 1 + 2^{-8937}$, ce qui autorise tous les paramètres de sécurité raisonnables (jusqu'à 8937). Notre généralisation des familles $\mathcal{H}_2$ à toutes tailles d'entrée et de sortie nous a permis d'implémenter la famille $\mathcal{H}_4$.

Le tableau 14.3 compare les vitesses respectives des différentes familles de fonctions de hachage que nous avons rencontrées.

Chapitre 15

Distillation d'une clé à travers un réseau classique

Maintenant que nous connaissons de façon formelle les algorithmes nécessaires à l'obtention d'une clé secrète, nous devons tenir compte des contraintes liées à leur implémentation. La difficulté principale est de pouvoir faire communiquer, de façon concertée, deux programmes distants – l'un chez Alice, l'autre chez Bob. La structure générale de cette réconciliation distribuée a été initiée par Gilles Van Assche et Cécile Neu. Nous l'avons étendue, achevée, et adaptée aux nouvelles méthodes de réconciliation ainsi qu'au canal classique fourni par les développeurs réseau du projet SECOQC.

15.1 Contraintes liées à l'utilisation d'un réseau

Lorsqu'Alice et Bob passent d'une réconciliation locale (c'est-à-dire dans laquelle le même programme s'occupe de l'encodage et du décodage), à une réconciliation distribuée entre deux programmes, voire deux machines, distincts, ils doivent établir un canal de communication, le canal classique. Ce changement introduit plusieurs contraintes de débit, de latence et de synchronisation.

Le nombre d'utilisations du canal classique, qui fixe le débit, doit être maîtrisé. Alors qu'en local Alice et Bob peuvent accéder aux ressources du partenaire autant de fois et quand ils le désirent, le programme de réconciliation distribuée doit s'assurer qu'une ressource a été transférée avant d'être utilisée, et doit la sauvegarder s'il doit en faire usage ultérieurement. Les principales sources de consommation du canal classique sont regroupées table 15.1.

La latence du réseau restreint le nombre d'échanges réalisables. Dans un protocole dans lequel Alice et Bob s'échangent une grande quantité de petits messages, la latence du réseau, c'est-à-dire le temps mort associé à une communication élémentaire, devient cruciale. C'est le cas du protocole de réconciliation cascade. Le caractère nécessairement unidirectionnel de la réconciliation inverse lève cette barrière : les échanges de parité doivent s'opérer de Bob vers Alice, sans que Bob ne puisse utiliser de l'information envoyée par Alice dans son calcul. Ainsi Bob peut envoyer toutes ses parités d'une seule traite, rendant ainsi la contrainte de latence négligeable devant celle du débit. Notons tout de même que le protocole de réconciliation nécessite également l'envoi de plusieurs petits messages préalables, détaillés section 15.3. Une

Nature	Usage	Direction
Révélation des bases	1	Bob → Alice
Échantillons révélés	16α	Indifférent
Parités	≤ 4	Bob → Alice

TAB. 15.1: Usages du canal classique, en bits par symbole réconcilié, ainsi que le sens de communication dans le cas de la réconciliation inverse. α est la fraction d'échantillons révélés et chaque échantillon est transmis sur 16 bits. Nous omettons certains usages peu consommateurs en débit, comme les en-tête de messages, ou les usages non extensifs, comme la synchronisation et la transmission de paramètres.

trop grande latence du réseau pourrait ralentir l'échange de ces messages et donc nuire à la transmission. Dans ce cas, il est reste possible de compenser la latence par des blocs de taille plus grande, car le nombre de ces petits messages ne croît pas avec la taille d'un bloc de données.

Quand deux programmes communiquent entre eux, un protocole doit être défini pour décider de l'enchaînement des opérations. Un programme unique se déroule de façon séquentielle, l'ordonnancement des instructions découle directement de l'ordre des lignes de code du programme. Cet ordre de fait n'existe plus pour deux programmes concomitants ; ils doivent se mettre d'accord sur qui fait quelle action et quand. C'est le problème de la synchronisation. Dans la vie courante, des règles tacites permettent de réguler les échanges verbaux entre personnes. Deux ordinateurs dialoguant à travers un canal classique doivent obéir à des règles strictes et bien définies. En cas d'erreur, il n'existe pas de voie de rattrapage...

15.2 Paradigmes de programmation appliqués à la réconciliation

Cette section regroupe quelques notions de programmation utilisées pour l'écriture du logiciel de réconciliation. Quelques-unes de ces notions sont peu usuelles, mais sont essentielles au bon fonctionnement de la réconciliation et à la résolution des contraintes introduites par l'utilisation d'un canal de communication. Cette section permet d'attirer le lecteur du code sur les points clé de la structure du programme.

Abstraction de classes

L'abstraction de classes est une caractéristique essentielle de la programmation orientée objet. Elle permet de regrouper plusieurs objets dont les interfaces ou les caractéristiques sont similaires. Par exemple, tous les objets du programme de réconciliation qui nécessitent un dialogue à travers le canal classique (gestion des blocs à réconcilier, formatage des blocs ou algorithme d'encodage-décodage) ont en commun la nécessité de disposer de fonctions capables d'envoyer ou de recevoir des données sur ce canal. Comme il serait inefficace de coder ces mêmes fonctions pour chacun des objets, nous définissons un objet *abstrait* (appelé *classe* dans les langages C++ ou java), implémentant ces fonctions communes, et dont *dérivent* les objets qu'on désire regrouper. On dit que les fonctions de la classe abstraite sont *héritées* par les classes *dérivées*. Dans l'exemple de l'utilisation du canal classique, la classe de base se nomme Application .

Cette structure de classes héritées présente de multiples avantages. Comme nous l'avons vu, elle permet de *factoriser* du code (c'est-à-dire de l'écrire une fois pour toutes, en vue de diverses utilisations), mais aussi de présenter une interface unifiée d'un ensemble de classes avec le reste du programme. Ainsi, la classe qui gère le canal classique (appelée Node) sait qu'elle aura affaire à des classes de type Application , sans faire de suppositions sur sa fonction particulière. À l'inverse, la classe Node est elle-même une classe abstraite dont dérivent les classes qui implémentent les diverses variétés de canaux classiques disponibles (voir section 15.4). L'abstraction de classes permet donc de générer du code plus modulaire, dont les effets de dépendances internes (et donc les bogues) sont limités.

Singletons

Un singleton est une classe unique, universelle à l'ensemble du programme. En programmation orientée objet, une classe définit un type de variable, au même titre que les types *entier* ou *flottant*. De cette façon, on peut déclarer plusieurs objets, c'est-à-dire plusieurs instanciations de cette classe. Par exemple, deux objets de type "décodeur LDPC" et deux objets de type "révélation totale" seront nécessaires pour le décodage du code multi-niveaux (section 12.7). Au contraire, certaines classes ont pour vocation de n'être instanciées qu'une seule fois. Par exemple, un seul objet de type Node est nécessaire au bon fonctionnement du programme. Ces classes uniques sont appelées *singletons*. Elles ont l'avantage de pouvoir être utilisées depuis tout endroit du programme, alors que les objets usuels (nos décodeurs) ne peuvent être utilisés que par leur propriétaire (ici l'objet "décodeur multi-niveaux").

Boucle principale

La boucle principale permet à un programme de patienter en attendant un événement. Un programme classique s'exécute de façon séquentielle, de la première à la dernière instruction. Après cette dernière, le programme est définitivement interrompu. Notre programme de réconciliation se comporte différemment, en ce qu'il est obligé d'attendre la réponse de son partenaire (un événement réseau) après avoir émis une requête. La partie du code responsable du fait d'attendre en écoutant un événement s'appelle la *boucle principale* ; son implémentation est de la responsabilité du nœud (classe Node). Un nœud implémentant le canal classique avec les fonctions standards de programmation client-serveur bas niveau pourra pour ce faire utiliser la fonction réseau read (celle-ci ne répond qu'à réception d'un message). Un nœud utilisant des bibliothèques de programmation de plus haut niveau pourra utiliser la boucle principale fournie. C'est le cas du nœud développé par Cécile Neu, ou des bibliothèques fournies par le projet SECOQC.

Du point de vue de l'application, l'exécution du code se termine quand le programme entre dans se boucle principale, en ce sens qu'il n'y a plus d'instruction à exécuter. L'application sera "réveillée" par l'exécution de sa fonction responsable de recevoir les messages.

Applications

Les applications sont les utilisateurs du canal classique. Nous les avons déjà rencontrées au cours des exemples des paragraphes précédents. À première vue, une seule application, responsable de l'ensemble du processus de réconciliation semblerait nécessaire. Cependant, il est intéressant de diviser ce processus en plusieurs entités logiques. Ce partage permet d'obtenir un programme plus modulaire, et rend le code plus lisible. De plus, la modularisation permet

de dédoubler certaines parties du processus de réconciliation, par exemple pour tirer parti d'architectures matérielles multiprocesseurs. Chacune des applications a donc une tâche définie, et a la possibilité de lancer d'autres applications, appelées applications filles.

Les applications ont en commun les fonctions suivantes, définies dans la classe abstraite Application :

- start() : Cette fonction est accompagnée d'un état, incrémenté à chaque grande réalisation de l'application. Elle est appelée pour démarrer l'application, ainsi qu'en interne à chaque étape de l'application, selon un protocole défini (section 15.3).
- send(message) : Cette fonction envoie un message sur le canal classique.
- recv(message) : Cette fonction est appelée par le nœud lorsqu'un message à destination de l'application est arrivé. Un identifiant unique, associé à chaque application et indiqué dans l'en-tête de chaque message, permet au nœud de diriger le message vers la bonne application.
- notify() : Cette fonction est appelée par chaque application fille pour indiquer que cette dernière a terminé sa tâche.

Maître-esclave

L'attribution des rôles maître-esclave (ou client-serveur) entre les deux partenaires permet d'ordonner la communication. De façon arbitraire et interchangeable, l'un des partenaires sera promu maître (ou encore client) et surnommé "Claude" conformément à la terminologie adoptée dans [7] ; l'autre, Dominique, sera l'esclave ou serveur. Le maître est seul autorisé à initier les échanges. Conformément à un protocole préalablement établi (les protocoles de communication sont détaillés section 15.3), l'esclave devra fournir une réponse au maître.

Immédiatement après avoir émis une requête (resp. une réponse), le maître (resp. l'esclave) attend l'échange suivant (en entrant dans sa boucle principale). Dans ce cas, les deux partenaires n'exécutent jamais d'instructions simultanément. On parle d'exécution *asynchrone*. Parfois, quand des tâches indépendantes et longues doivent être réalisées par chacun des partenaires (par exemple le mélange aléatoire des symboles), le maître ou l'esclave choisissent de continuer leur exécution après émission d'un message. Les deux programmes sont exécutés de façon simultanée, dite *synchrone*. Dans ce cas, seul le maître est autorisé à rétablir le dialogue.

Le respect scrupuleux de ces règles hiérarchiques assure le bon déroulement du protocole de réconciliation. L'exécution synchrone de deux programmes est un aspect très délicat de la programmation, et exige de la rigueur. En cas de dysfonctionnement, l'erreur sera en général très difficile à déceler.

Multi-threading

Le multi-threading est une technique permettant d'avoir plusieurs points d'exécution courants, ou fils, dans un même programme. Bien plus que l'exécution parallèle de deux programmes séparés communicants, le multi-threading requiert une discipline intransigeante pour qu'une fonction ne soit pas exécutée en même temps par les différents threads, ou qu'une variable ne soit pas modifiée en même temps par différents threads.

Le code pilotant l'expérience et le code gérant la réconciliation sont regroupés dans un même programme, parcouru par deux threads. L'avantage de cette méthode est que les deux threads partagent le même espace mémoire (puisqu'il s'agit d'un seul et même programme...), ce qui évite le recours à de coûteux et contraignants protocoles de communication entre processus.

La bibliothèque Q3P (Quantum Point to Point Protocol), qui implémente le canal classique fourni par le projet SECOQC, fonctionne elle aussi avec deux threads, l'un fourni au protocole de réconciliation, l'autre à la gestion interne de la bibliothèque.

15.3 Protocoles de communication, ou comment éviter la cacophonie

Dans cette section, nous passerons en revue l'ensemble des applications qui composent la réconciliation distribuée, ainsi que les protocoles de communication associés.

Les protocoles sont représentés par des diagrammes de séquence suivant la convention de représentation des diagrammes UML[1]. Les règles de ces diagrammes sont les suivantes :

- Le déroulement chronologique des événements s'opère de haut en bas, en suivant une *ligne de vie* (trait pointillé). Dans notre cas, deux lignes de vie sont nécessaires, une pour le maître, l'autre pour l'esclave.
- Lorsqu'une application est active, un rectangle recouvre la ligne de vie.
- À coté de chaque zone d'activité sont indiquées les fonctions utilisées. Leur nom est assez détaillé pour faire office de description.
- state++ indique l'incrémentation de l'état associé à la fonction start() de l'application (section 15.2).
- Une flèche reliant les deux lignes de vie indique le transfert d'un message sur le canal classique. Le contenu du message (identifiant et information échangée) est indiqué au-dessus de cette flèche. Une flèche pleine représente un message synchrone, une flèche creuse un message asynchrone.
- Les applications filles sont représentées par un rectangle arrondi. Elles sont appelées par le maître, et notifient le maître et l'esclave.
- Des parties du protocole sont conditionnées par l'identité du maître (Alice ou Bob).

Il est à noter que la structure du protocole permet que le même code soit utilisé pour le maître et l'esclave. Cette propriété permet de mettre un commun une importante similitude entre les actions que doivent réaliser le maître et l'esclave.

CSKDP

L'application CSKDP (Continuous Secret Key Distillation Protocol, selon [7]) est l'application principale du processus de réconciliation. Elle est démarrée par le nœud après établissement du canal classique. Elle est responsable :

- de la gestion du pilotage de l'expérience. CSKDP s'occupe de démarrer le pilotage, vérifier son état, et le redémarrer après une interruption.
- de la négociation des blocs à réconcilier et du transfert de ces blocs depuis le logiciel de pilotage.
- de démarrer l'application BSKDP après l'établissement du choix d'un bloc à réconcilier.

L'application CSKDP est un singleton. Son arrêt signifie la fin du processus de réconciliation. Le diagramme de séquence UML de cette application est représenté figure 15.1.

[1]Unified Modeling Language : ensemble de règles standardisées pour la représentation de diagrammes. Ces règles sont une sorte de grammaire pour la conception de diagrammes. Elles confèrent un "sens" aux diagrammes, au-delà de leur sens purement visuel. Un diagramme suivant les règle UML pourra être interprété par une machine, par exemple pour générer du code source.

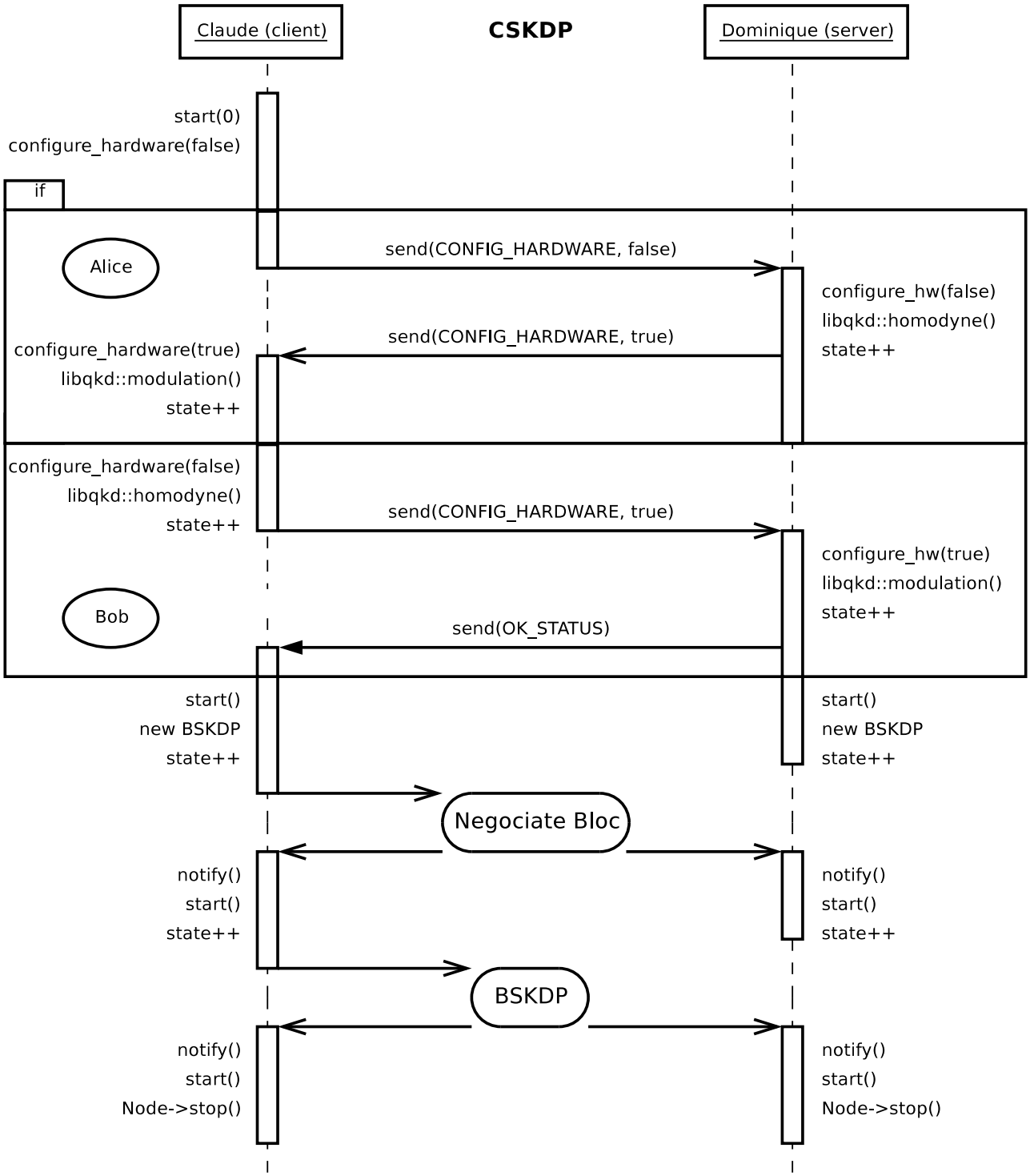


FIG. 15.1: Diagramme de séquence UML pour le protocole de communication CSKDP.

BSKDP

L'application BSDKP (Block Secret Key Distillation Protocol) est responsable de la réconciliation d'un bloc fourni par CSKDP. Elle assure :

- la révélation des phases choisies par Bob (tamisage, ou *sifting*).
- le mélange aléatoire des données.
- le choix d'échantillons révélés et le calcul des paramètres de transmission.
- la négociation d'une chaîne de bits qui identifie une fonction de hachage pour l'amplification de confidentialité.
- l'exécution de l'application de réconciliation à proprement parler.
- l'exécution de l'amplification de confidentialité.
- l'enregistrement de la clé secrète obtenue auprès du nœud.

Plusieurs instances de l'application BSKDP peuvent éventuellement coexister, en fonction des caractéristiques du matériel de décodage (processeurs multiples). Le diagramme de séquence UML de cette application est représenté figure 15.2.

SliceAlgo

L'application SliceAlgo implémente l'algorithme de réconciliation par tranches, utilisant divers protocoles de correction d'erreur binaires interchangeables (Cascade, Winnow, Turbo codes). Elle repose sur les structures développées par Gilles Van Assche implémentant cet algorithme. Cette application est fournie à titre historique, car la réconciliation molle (SoftAlgo) fournit de meilleurs résultats en termes d'efficacité.

SoftAlgo

L'application SoftAlgo implémente l'algorithme de réconciliation multi-niveaux (voir section 12.7). Grâce à sa similitude avec l'algorithme de réconciliation par tranches, elle reprend une grande partie des structures utilisées par SliceAlgo, notamment en ce qui concerne la représentation des tranches, leur création, les calculs d'entropie, d'informations, de bits secrets... À ces structures sont ajoutés les algorithmes de décodage mou (LDPC ou turbo codes) décrits au chapitre 13. La réconciliation molle suit le déroulement suivant :

- initialisation des tranches et des codes correcteurs, en fonction des paramètres de transmission.
- encodage des codes mous (LDPC), ainsi que des codes BCH capables de corriger les dernières erreurs laissées par le décodage mou.
- multiplexage et envoi des parités générées par l'ensemble des codes.
- décodage (correction d'erreurs).
- notification de l'application parent (BSKDP).

Il existe une application SoftAlgo pour chaque application BSKDP. Éventuellement, les applications SoftAlgo peuvent être réutilisées d'une application BSKDP à l'autre, pour éviter l'initialisation des codes pour chaque bloc à réconcilier. Le diagramme de séquence UML de cette application est représenté figure 15.3.

Negociator

Le négociateur permet à Alice et Bob de s'entendre sur un paramètre de configuration, à l'issue de multiples transactions. Son déroulement est le suivant :

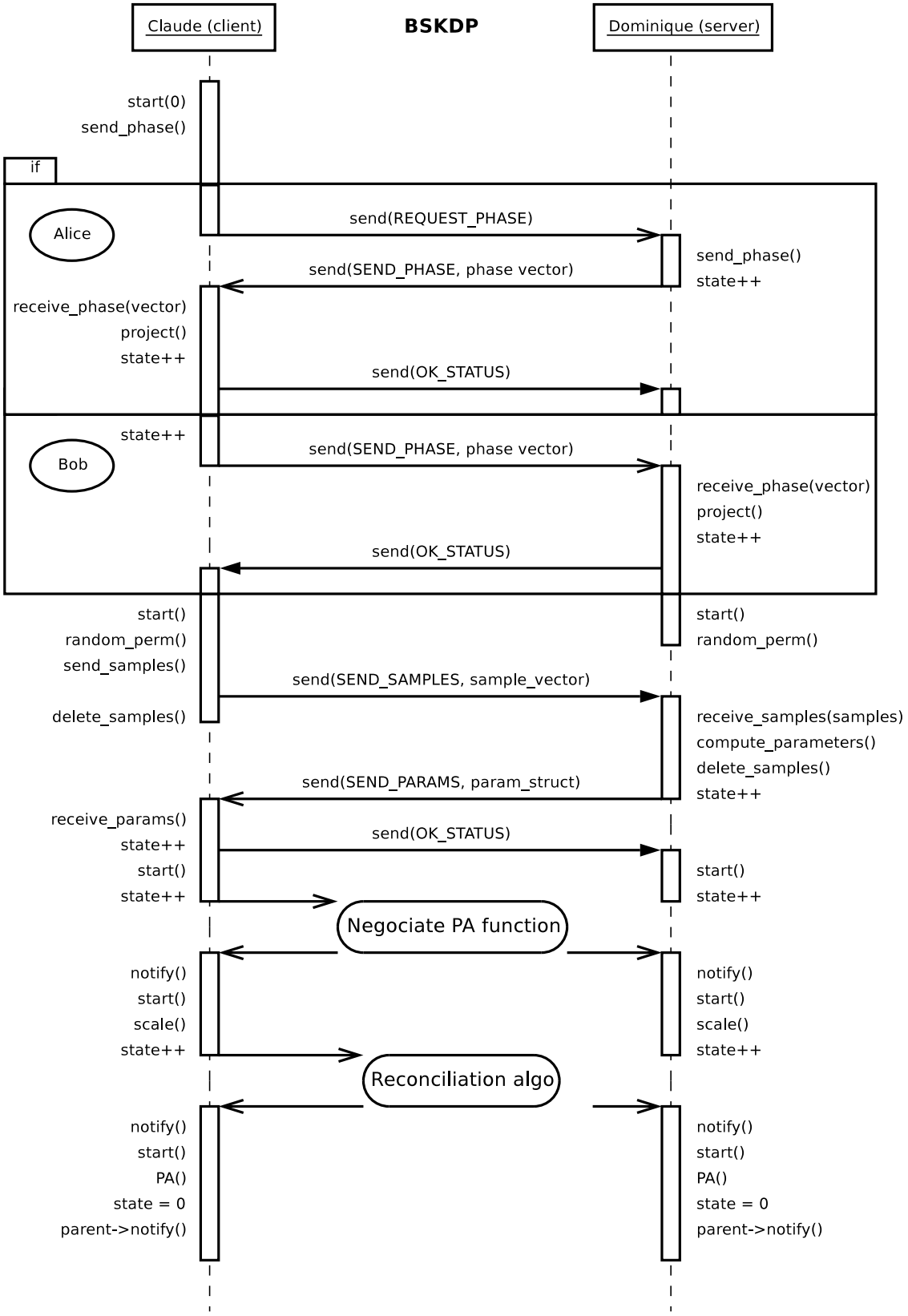


FIG. 15.2: Diagramme de séquence UML pour le protocole de communication BSKDP.

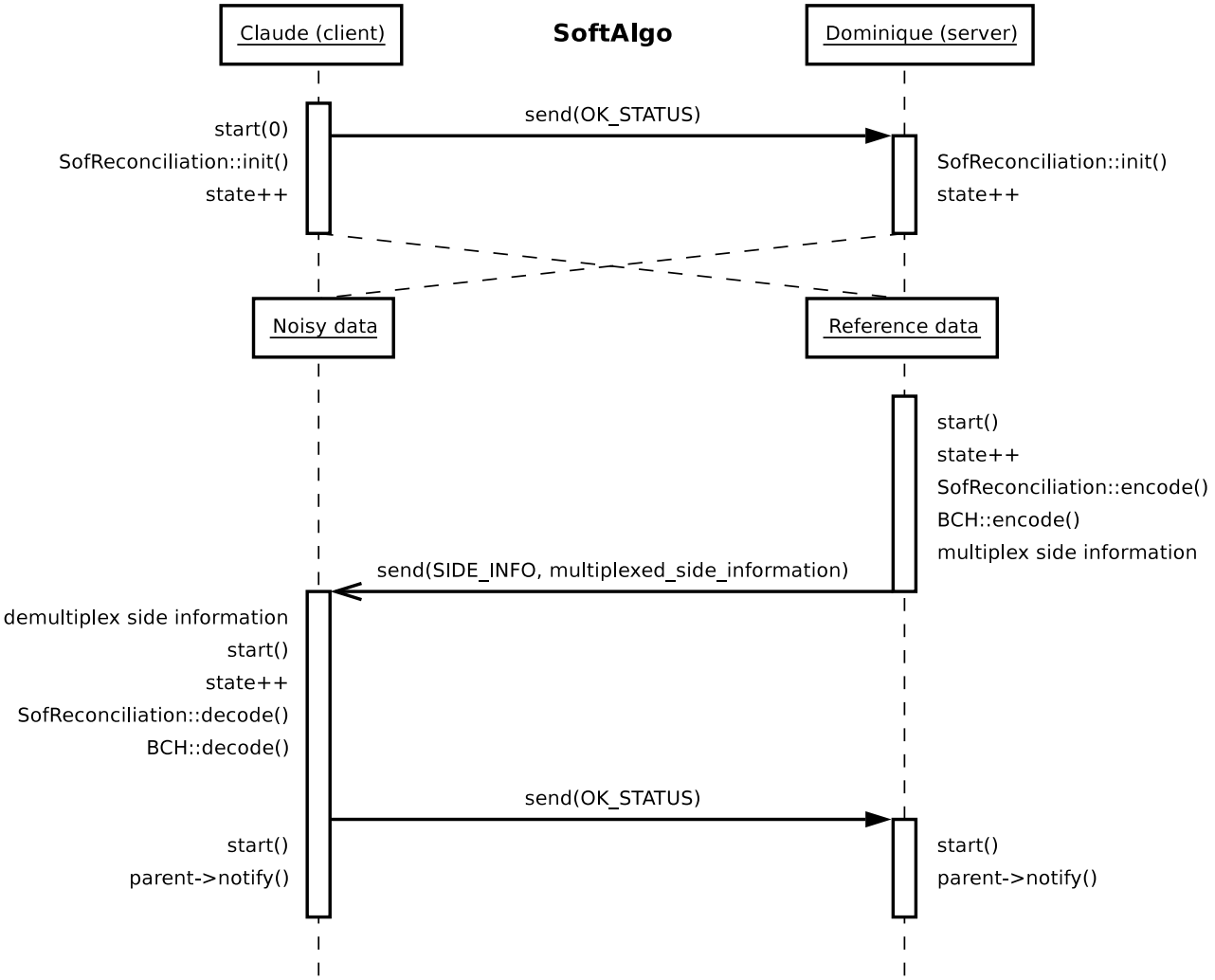


FIG. 15.3: Diagramme de séquence UML pour le protocole de communication SoftAlgo. Suivre les diagonales pointillées suivant que Claude ou Dominique sert de référence pour la clé.

- Claude propose une valeur du paramètre.
- Dominique accepte ou refuse cette valeur. Dans ce dernier cas, il a la possibilité de proposer une alternative.
- En fonction de la réponse de Dominique, Claude entérine la valeur du paramètre, ou relance le processus de négociation.

Le négociateur est utilisé pour la décision du bloc à réconcilier et de la clé utilisée par l'amplification de confidentialité. Il utilise le canal classique pour transmettre les messages de négociation (négociation propriétaire), ou bien, pour le canal classique fourni par le projet SECOQC, les fonctions de négociation fournies par le nœud.

KeyBroker

Dernière partie de notre assemblage, le "KeyBroker", engrange les clés secrètes. Si nous faisons fonctionner notre système de réconciliation dans le cadre de l'architecture fournie par le réseau SECOQC, le "KeyBroker" transfère les clés vers le réservoir de clés du réseau

quantique.

L'ensemble des modules fonctionnels décrits dans ce chapitre, ainsi que leurs interactions, sont représentés figure 15.4 sous forme d'un diagramme UML.

15.4 Implémentation d'un canal classique

Le canal classique permet une communication entre Alice et Bob. Nous avons rencontré plusieurs usages du canal classique : transmission d'informations de configuration, de messages indiquant l'avancement du protocole de réconciliation, envoi des phases de mesures de Bob, des syndromes des codes correcteurs d'erreurs. Nous avons appelé «nœud» l'entité responsable du canal classique : envoi et réception des données, implémentation de la boucle principale ordonnant le protocole de communication. Il nous reste maintenant à spécifier comment nous réalisons en pratique le canal classique.

À l'aide de l'abstraction de classe, nous pouvons définir une interface de nœud abstraite laissant libre l'implémentation du canal classique. Ensuite, nous pouvons dériver de cette classe abstraite plusieurs classes, chacune implémentant un canal classique de nature différente. Ainsi, intégrer notre système dans un environnement donné, avec son propre canal classique, nécessite uniquement la programmation d'une classe dérivée. Nous avons implémenté trois types de canaux classiques correspondant chacun à un environnement donné.

Un canal classique direct non authentifié permet une utilisation souple du programme de réconciliation, pour les phases de test. Ce canal est une liaison TCP/IP directe entre les ordinateurs d'Alice et de Bob, à travers un lien ethernet. Il est dépourvu de toute authentification, et ne peut donc servir à une situation de distribution quantique de clé. Cependant, il permet de séparer l'implémentation de la réconciliation de celle du canal classique, deux problèmes distincts que notre logiciel de gestion tente de résoudre.

Nous disposons d'un canal point à point direct authentifié, programmé par Cécile Neu. L'authentification de ce canal utilise la famille de fonctions de hachage $\mathcal{H}_4$ introduite au chapitre 14 comme la composition des familles $\mathcal{H}_2$ et $\mathcal{H}_3$.

Un canal classique utilise les fonctions fournies par la bibliothèque Q3P, créée dans le cadre du projet SECOQC. Ce canal classique est commun à tous les systèmes qui formeront le réseau quantique prévu par SECOQC. Il fournit des fonctions «envoi» et «réception» qui cachent l'authentification, gérée à un plus haut niveau par le réseau.

15.5 Interfaçage avec le logiciel de pilotage

Enfin, le logiciel de réconciliation est responsable de la gestion de l'expérience. Il allume la transmission quand il a besoin de blocs, vérifie sont état de fonctionnement, et l'arrête lorsqu'il dispose d'assez de blocs à réconcilier. Ces diverses opérations sont fournies par le logiciel de pilotage de l'expérience décrit au chapitre 10 sous forme d'une bibliothèque dynamique. Le pilotage de l'expérience et la réconciliation des blocs acquis tournent en parallèle, sous forme de fils d'exécution.

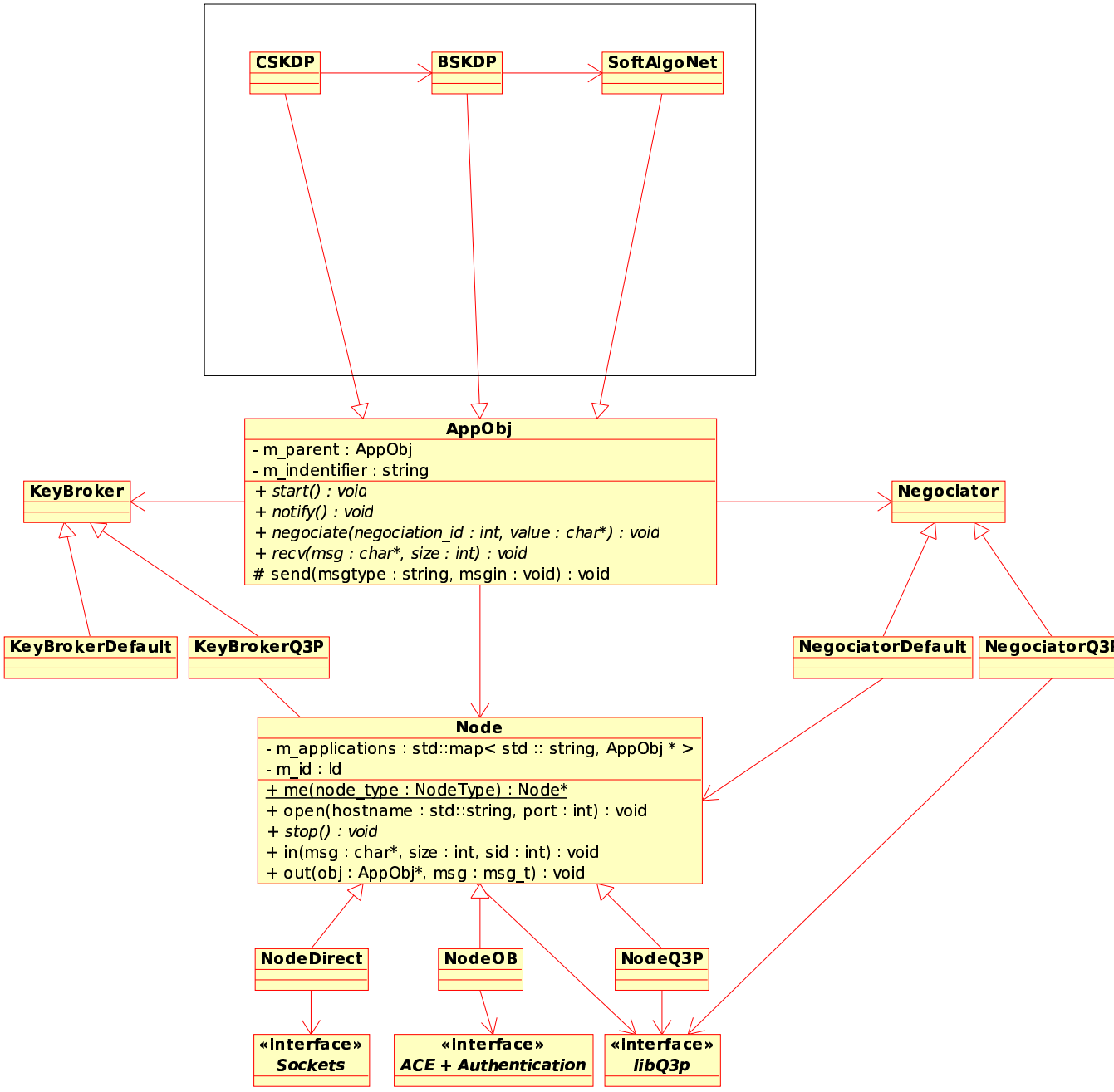


Diagramme : « diagramme de classes » page 1

FIG. 15.4: Diagramme de classe UML pour l'ensemble du programme de réconciliation distribuée. Ce diagramme montre les relations entre les différents objets constitutifs du programme, et indique quelques unes des fonctions et variables principales caractéristiques de chaque objet.

Conclusion

Au cours de cette thèse, nous avons réalisé un dispositif de distribution quantique de clé qui utilise des états cohérents pour transmettre de l'information quantique entre Alice et Bob. Cette réalisation expérimentale est entièrement constituée de fibres optiques et de composants standards des technologies télécom. Notre système est composé d'un système de modulation et d'une détection homodyne limitée au bruit de photon qui fonctionnent à des taux de répétition allant jusqu'à 1 MHz. Nous avons évalué l'information secrète que ce dispositif est capable de transmettre en fonction de la mesure du bruit sur la transmission. Notamment, pour de faibles pertes en ligne, l'information secrète est échangée à un taux supérieur à 1 Mbit/s.

Nous avons réalisé expérimentalement des attaques quantiques. Elles sont assez générales pour explorer l'ensemble des paramètres du modèle du canal gaussien que nous avons utilisé pour établir la sécurité du système. Ces attaques confirment notre analyse du bruit, et donc le taux secret mesuré.

L'adaptation et l'optimisation de méthodes de correction d'erreur et d'amplification de confidentialité conduisent à l'obtention d'une clé secrète distillée à partir de nos données expérimentales. Ces algorithmes font partie d'un logiciel de gestion du canal classique reliant Alice à Bob, qui permet de réconcilier une clé secrète sur deux machines distantes, de façon synchrone avec l'expérience.

Finalement, nous avons pu transmettre une clé secrète sur 25 km. Le taux secret final de cette transmission est de 2 kb par seconde. À l'heure actuelle, ce taux est limité par les performances des algorithmes de réconciliation.

Nous prévoyons la poursuite du travail présenté dans ce manuscrit par l'automatisation totale du dispositif, aboutissant à un fonctionnement ne nécessitant pas d'intervention humaine. D'autre part, nous travaillons à l'amélioration de l'efficacité et de la vitesse des codes correcteurs d'erreurs afin d'améliorer le taux secret final. Deux thèses ont respectivement débuté sur ces deux sujets.

Annexe A

Turbo codes

Nous avons décrit au chapitre 12 le décodage "Belief Propagation" des codes LDPC. Pour la réconciliation, nous avons également utilisé des "turbo codes" qui sont un assemblage de codes convolutifs. Toutefois, dans notre application, les performances des turbo codes se sont avérées moins bonnes que celles des codes LDPC. Cette annexe est consacrée à la description des codes convolutifs et des turbo codes.

A.1 Codes convolutifs

Les codes convolutifs sont caractérisés par un encodeur possédant des registres de mémoire à décalage. Ils sont les briques fondamentales des turbo codes. Le processus d'encodage, représenté par un diagramme (voir figure A.1), est une succession d'étapes. Chaque étape est ponctuée par l'arrivée d'un bit à encoder, par le calcul des bits de sortie de l'encodeur (dits «bits de parité») en fonction du bit entrant et des états des registres, puis par le décalage des états de chaque registre vers le suivant. Pour cette dernière opération, le contenu du dernier registre est effacé, et le contenu du premier registre est calculé à partir du bit d'entrée et des états des registres.

Les turbo codes (voir section A.3) sont un assemblage de codes convolutifs systématiques. Pour ces codes, un des bits de sortie est directement le bit d'entrée, les autres bits de sortie étant les bits de parité à proprement parler.

Les bits de sortie ne dépendant que du bit d'entrée et de l'état des K registres, la chaîne encodée est souvent représentée par une trajectoire dans un graphe, appelé treillis, représentant les 2^K états des registres pour chacun des k bits à encoder. Le code est l'ensemble des trajectoires possibles dans ce treillis, et le décodage consiste à retrouver la trajectoire d'origine à partir des symboles obtenus en sortie du canal. Cette représentation simplifie l'implémentation des algorithmes d'encodage et de décodage des codes convolutifs, car toute la structure du code est décrite par l'ensemble des transitions possibles entre les 2^K états de l'encodeur.

Le code représenté figure A.1 a un taux fixe $R = \frac{1}{2}$, car il possède deux sorties pour une entrée. Pour faire varier le taux du code, on peut utiliser des codes un peu plus complexes, faisant intervenir deux bits en entrée pour chaque bit de parité (suivant les configurations, on parle de codes "duo-binary" [74], ou de codes "trellis coded modulation" [80]). On peut également ajuster le taux du code en éliminant certains bits de parité à la sortie de l'encodeur (on parle de "puncturing").

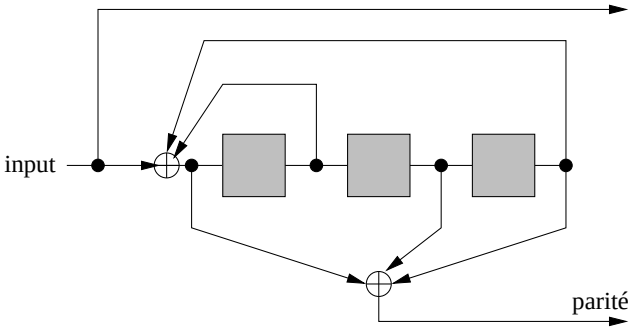


FIG. A.1: Diagramme représentant l'encodage d'un code convolutif. Les boîtes sont des registres à décalage, les $\oplus$ symbolisent l'addition modulo 2, et les flèches le parcours des bits lors d'une étape d'encodage. Ce code est dit «récursif», car les valeurs issues des registres mémoire sont réutilisées en entrée du schéma. Il est systématique, car le bit d'entrée se retrouve inchangé à la sortie. Le deuxième bit de sortie est appelé «bit de parité». On condense un tel schéma en deux polynômes dont les coefficients binaires indiquent la position des nœuds de feed-back et de calcul de parité, ici $(1 + X + X^3, 1 + X^2 + X^3)$ ou de façon encore plus compacte par l'écriture octale du nombre binaire indiquant la position des coefficients non nuls du polynôme, ici $(15, 13)$.

A.2 Décodage mou des codes convolutifs

Les codes convolutifs sont décodés par l'algorithme BCJR [81] [1], de la classe des algorithmes "maximum *a posteriori* probability" (MAP) [83]. Cet algorithme trouve de façon déterministe la chaîne de bits encodée la plus probable, compte tenu de l'observation des bits de sortie du canal et de l'encodeur utilisé. Il s'apparente aux algorithmes de type "maximum likelihood". La performance des codes convolutifs repose donc entièrement sur la capacité de l'encodeur à produire des mots code bien distinguables. Les codes convolutifs ont en général des performances éloignées de la limite de Shannon. Nous verrons dans la section suivante que l'utilisation parallèle de plusieurs codes convolutifs permet d'obtenir des codes performants : les turbo codes.

Pour utiliser le décodage BCJR dans notre situation de réconciliation de variables gaussiennes corrélées, nous devons spécifier comment considérer les parités transmises par le canal classique sans erreur. Usuellement, et algorithme prend pour entrée de façon indifférenciée les LLR intrinsèques des bits issus du canal et les LLR des parités, elles aussi bruitées. Pour traduire le fait que nous connaissons parfaitement les parités, nous pouvons leur attribuer un LLR infini (ou tout du moins très grand devant 1), et utiliser l'entrée de parité standard du décodeur BCJR. Nous avons toutefois simplifié l'algorithme BCJR dans le cas d'une transmission parfaite des parités, afin d'éviter tout problème de débordement de variables dû à des LLR infinis. Cet algorithme modifié est présenté en annexe B.

Dans le cadre du décodage LDPC, nous avions introduit la notion de rapport de vraisemblance. Les rapports de vraisemblance quantifient notre estimation des bit à décoder. Les LLR *a posteriori* indiquent notre connaissance de la chaîne à décoder après décodage. Nous avions distingué deux sources constituant ces LLR *a posteriori* : les LLR intrinsèques, qui représentent l'apport des valeurs des bits issus du canal à cette connaissance, et les LLR extrinsèques, qui représentent l'apport des contraintes apportées par la structure du code. Nous introdui-

[1] On utilise aussi l'algorithme sub-optimal de Viterbi (SOVA) [82].

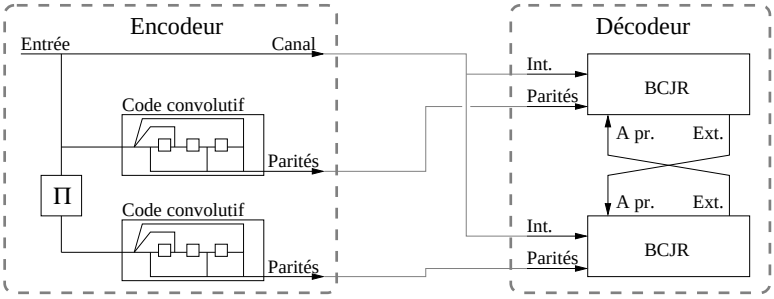


FIG. A.2: Concaténation parallèle de codes convolutifs (turbo codes). La chaîne de bits est encodée dans un ordre différent (interleaver Π) par deux codes convolutifs identiques. Le décodage est un processus itératif, dans lequel l'information extrinsèque d'un décodeur sert d'information *a priori* pour l'autre décodeur (principe du «turbo»).

sons maintenant une troisième composante : les LLR *a priori*. Ces LLR *a priori* indiquent la connaissance apportée par une expérience antérieure au décodage, par exemple un précédent décodage que l'on peut rencontrer dans une situation de décodage itératif faisant intervenir plusieurs codes. Pour les situations de décodages que nous avions considérées précédemment, un bit à décoder était décodé par un et un seul code ; ainsi, l'*a priori* était nul. Nous écrivons finalement le rapport de vraisemblance après décodage, associé au bit à décoder x_i :

$$LLR_i^{a\ posteriori} = LLR_i^{\text{intrinsèque}} + LLR_i^{\text{extrinsèque}} + LLR_i^{a\ priori}. \tag{A.1}$$

Comme nous le verrons dans l'annexe B, le décodeur BCJR comporte naturellement une entrée destinée à un éventuel *a priori*. Il est ainsi possible de considérer l'assemblage, ou concaténation, de plusieurs codes, en bouclant la sortie extrinsèque d'un décodeur à l'entrée *a priori* d'un autre décodeur.

A.3 Concaténation de codes

La modularité offerte par les décodeurs mous permet la concaténation de codes, c'est-à-dire la mise en série ou en parallèle de codes de nature ou de taux différents, afin d'obtenir un code plus performant. Un encodage en série consiste à encoder les bits d'information à l'aide d'un premier encodeur, puis à encoder la sortie de cet encodeur avec un deuxième encodeur. Un encodage en parallèle, plus couramment rencontré qu'un encodage en série, consiste à encoder de façon indépendante des chaînes de bits. Bien évidemment, l'opération ne présente aucun intérêt si les chaînes (et donc les encodages) sont totalement indépendantes. Ainsi, on encode en parallèle des chaînes de bits liées, voire même identiques.

La concaténation la plus simple est la concaténation parallèle de codes convolutifs (PCCC en anglais), dite plus simplement «turbo code» [84]. Elle utilise deux codes convolutifs récursifs systématiques identiques mis en parallèle, la chaîne d'entrée du deuxième encodeur est une version «mélangée» de la chaîne d'entrée du premier encodeur.

Les codes convolutifs sont de mauvais codes, au sens où ils n'arrivent pas à atteindre des taux d'erreur suffisamment faibles ; leur décodage apporte néanmoins une quantité non négligeable d'information extrinsèque inexploitée. Le décodage turbo tire parti de l'information extrinsèque

apportée par l'un des décodeur pour affiner l'estimation *a priori* du deuxième encodeur (d'où le terme «turbo»). Le décodage est itératif : on décode successivement chacun des codes jusqu'à convergence (figure A.2).

Les turbo codes sont comparables aux LDPC en termes de performance, et surpassent de loin les autres codes connus au moment de leur invention par Berroux [84] en 1993.

Annexe B

Décodage BCJR et "side-information"

Cette annexe détaille les adaptations possibles de l'algorithme BCJR de décodage des codes convolutifs et des turbo codes pour notre situation de réconciliation. En effet, la réconciliation permet la transmission sans erreur des bits de parité à travers le canal classique. Cette caractéristique permet de simplifier l'algorithme de décodage BCJR. De plus, dans la littérature, certaines étapes de l'algorithme sont souvent appliquées à des canaux particuliers, tels que les canaux BSC ou AWGN. Nous généraliserons ces étapes à des canaux arbitraires.

Le lecteur de cette annexe profitera de la lecture du chapitre 12 sur les codes correcteurs d'erreurs, ainsi que d'une introduction à l'algorithme BCJR, par exemple [85].

Le décodage consiste à déterminer le LLR *a posteriori* pour chaque bit d'entrée x_i :

$$LLR_i^{a\ posteriori} = \ln\left(\frac{P(x_i = 1|y)}{P(x_i = 0|y)}\right) \tag{B.1}$$

où x_i est le bit transmis, et y la sortie du canal. On peut introduire dans cette équation la représentation sous forme de treillis des codes convolutifs :

$$LLR_i^{a\ posteriori} = \ln\left(\frac{\sum_{S^+} P(s_{i-1} = s', s_i = s, y)}{\sum_{S^-} P(s_{i-1} = s', s_i = s, y)}\right). \tag{B.2}$$

Dans cette équation, s_i représente l'état du treillis (c'est-à-dire l'état des registres internes du code convolutif) à l'instant i, et $S^{\pm}$ est l'ensemble des transitions $s' \rightarrow s$ du treillis occasionnées par un bit d'entrée $x_i = \pm 1$. $P(s_{i-1} = s', s_i = s, y)$ est la probabilité pour que le treillis soit dans l'état s' à l'instant $i-1$, dans l'état s à l'instant i, et que les n sorties du canal soient $y = (y_{1\ldots n})$. On décompose cette probabilité en trois facteurs :

$$\begin{aligned} P(s_{i-1} = s', s_i = s, y) &= p(s_{i-1} = s', y_{1\ldots i-1})p(s_i = s, y_i|s_{i-1} = s')p(y_{i+1\ldots n}|s_i = s) && \text{(B.3)} \\ &\equiv \alpha_{i-1}(s')\gamma_i(s', s)\beta_i(s) && \text{(B.4)} \end{aligned}$$

Les facteurs α et β se calculent de manière récursive à partir de γ, qui reste à calculer :

$$\gamma_i(s', s) = p(y_i^p|x_i, s, s')p(s|s')p(y_i^c|x_i) \tag{B.5}$$

où x_i est la valeur du bit d'entrée de l'encodeur associé à la transition $s' \rightarrow s$, y_i^c est la i^{eme} sortie du canal de communication, et y_i^p la i^{eme} parité reçue.

Il est important de noter que tout facteur qui ne dépend pas de s, s' ou x_i pourra se simplifier dans l'expression finale de $LLR_i^{a\ posteriori}$. Inutile donc de les prendre en compte.

Comme les additions sont meilleur marché que les multiplications, on préfère généralement calculer γ dans le domaine logarithmique, comme somme de trois termes. On voit avec délectation apparaître respectivement dans ces trois termes les trois entrées de notre module de décodage mou :

- Le premier terme $p(y_i^p|x_i, s, s')$ indique si la parité reçue est compatible avec la parité de la transition $s' \to s$. Comme les parités sont transmises sur un canal classique parfait, cette probabilité vaut 0 ou 1. Si la parité a été effacée pour ajuster le taux du code (puncturing), la probabilité vaut 1/2 indépendamment de s et s', donc elle n'est pas à calculer. En conséquence, ce terme permet de simplifier l'algorithme de décodage, car il se résume à ne calculer que les γ correspondant à une transition de parité acceptable. Notons que l'utilisation d'un décodeur usuel, bien que plus complexe pour notre tâche, reste possible.
- Le second terme $p(s|s')$, est la probabilité de l'occurrence de la transition $s \to s'$. Elle découle directement de notre *a priori* sur le bit à décoder, seul paramètre duquel dépend cette transition. On a $\ln(P(s|s')) = \delta(x_i = +1)LLR^{a\ priori} - T_1$ où δ est le symbole de Kronecker, et T_1 un terme qui s'élimine.
- Enfin, le troisième terme $p(y_i^c|x_i)$ caractérise la perturbation introduite par le canal de communication. Elle s'exprime grâce aux LLR intrinsèques :

$$\ln(p(y_i^c|x_i)) = \frac{1}{2}LLR^{\text{intrinsèque}}x_i + T_2 \quad \text{où } T_2 \text{ s'élimine} \tag{B.6}$$

On trouve dans la littérature une démonstration *ad hoc* de cette expression pour les canaux AWGN ou BSC. Nous en proposons une démonstration générale :

$$p(y_i) = p(y_i|x_i = +1)p(x_i = +1) + p(y_i|x_i = -1)p(x_i = -1) \tag{B.7}$$

$$\text{or} \quad p(x_i = +1) = p(x_i = -1) = \frac{1}{2} \tag{B.8}$$

$$\text{car les deux symboles sont équiprobables en entrée du canal} \tag{B.9}$$

$$\text{et} \quad p(y_i|x_i = \pm 1) = p(y_i|x_i = \mp 1)\, e^{\pm LLR^{\text{intrinsèque}}} \tag{B.10}$$

$$\text{d'où} \quad p(y_i|x_i = \pm 1) = \frac{2p(y_i)}{1 + e^{\mp LLR^{\text{intrinsèque}}}} \tag{B.11}$$

$$\Rightarrow \quad p(y_i|x_i) = \frac{2p(y_i)}{1 + e^{-x_i LLR^{\text{intrinsèque}}}} \tag{B.12}$$

$$\Rightarrow \quad \ln(p(y_i|x_i)) = \frac{1}{2}LLR^{\text{intrinsèque}}x_i + T_2 \tag{B.13}$$

$$\text{où} \quad T_2 = \ln\left(\frac{p(y_i)}{\cosh(\frac{1}{2}LLR^{\text{intrinsèque}})}\right) \quad \text{s'élimine.} \tag{B.14}$$

Notons enfin que les LLR *a posteriori* s'obtiennent à partir des expressions de $\ln(\gamma)$ par une relation de la forme $\ln\left(\sum_j \exp(X_j)\right)$. Cette formule est un obstacle pour les implémentations de l'algorithme MAP, car elle est sensible à la numérisation des variables réelles, et nécessite la coûteuse évaluation de fonctions non linéaires. Il existe deux solutions [83] :

- $\ln\left(\sum_j \exp(X_j)\right) \simeq \max_i(X_i)$. Cette approximation brise l'optimalité de l'algorithme de décodage (car elle est équivalente à ne plus parcourir l'ensemble du treillis), mais reste meilleure que l'algorithme SOVA. Elle a de plus le mérite de la simplicité arithmétique. Cette version de l'algorithme MAP est baptisée max-log-MAP.

- $\ln\left(\sum_j \exp(X_j)\right)$ peut se calculer de manière exacte par récursion de l'expression

$$\ln\left(\exp(X_1) + \exp(X_2)\right) = \max(X_1, X_2) + \ln\left(1 + \exp(-|X_2 - X_1|)\right). \tag{B.15}$$

  Cet algorithme, nommé simplement log-MAP, est optimal, et d'une complexité légèrement supérieure à max-log-MAP. Il est peu sensible aux effets de numérisation.

En conclusion, nous avons montré que l'algorithme modifié pour la réconciliation était réductible à l'algorithme usuel, en utilisant les LLR intrinsèques comme sortie du canal, et $\pm\infty$ comme bits de parité.

Bibliographie

[1] Mark Hillery. Quantum cryptography with squeezed states. *Physical Review A*, 61 :022309, 2000.

[2] T. C. Ralph. Continuous variable quantum cryptography. *Physical Review A*, 61 :010303, 2000.

[3] N. J. Cerf, M. Lévy, and G. Van Assche. Quantum distribution of Gaussian keys using squeezed states. *Physical Review A*, 63 :052311, 2001.

[4] Frédéric Grosshans. *Thèse de doctorat. Communication et cryptographie quantiques avec des variables continues.* 2002.

[5] Frédéric Grosshans and Philippe Grangier. Continuous Variable Quantum cryptography Using Coherent States. *Physical Review Letters*, 88(5), 2002.

[6] Jérôme Wenger. *Thèse de doctorat. Dispositifs impulsionnels pour la communication quantique à variables continues.* 2004.

[7] Gilles Van Assche. *Quantum Cryptography and Secret-Key Distillation.* Cambridge University Press, 2006.

[8] Matthieu Bloch, Andrew Thangaraj, and Steven W. McLaughlin. Efficient Reconciliation of Correlated Continuous Random Variables using LDPC Codes. *E-print* `cs.IT/0509041`, 2005.

[9] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press.

[10] Claude Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4), 656–715.

[11] Charles Bennett and Gilles Brassard. Quantum cryptography : Public key distribution and coin tossing. In *IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[12] I. Csiszar and J. Korner. Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 24 :339–348, 1978.

[13] N. J. Cerf, A. Ipe, and X. Rottenberg. Cloning of Continuous Quantum Variables. *Physical Review Letters*, 85 :001754, 2000.

[14] Frédéric Grosshans, Gilles Van Assche, Jérôme Wenger, Rosa Brouri, Nicolas Cerf, and Philippe Grangier. Quantum key distribution using gaussian-modulated coherent states. *Nature*, 421 :238–241, 16 January 2003.

[15] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423, 623–656, 1948.

[16] David J.C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[17] Hans-A. Bachor and Timothy C. Ralph. *A Guide to Experiments in Quantum Optics.* Wiley-VCH, 2004.

[18] Frédéric Grosshans, Nicolas J. Cerf, Jérôme Wenger, Rosa Tualle-Brouri, and Philippe Grangier. Virtual entanglement and reconciliation protocols for quantum cryptography with continuous variable. *Quantum Information and Computation*, 3, No Special 2003.

[19] A. Einstein, B. Podolsky, and N. Rosen. Can quantum mechanical description of reality be considered complete ? *Physical Review*, 47 :777, 1935.

[20] Jaromír Fiurásek. Optical Implementation of Continuous-Variable Quantum Cloning Machines. *Physical Review Letters*, 86 :004942, 2001.

[21] Ryo Namiki, Masato Koashi, and Nobuyuki Imoto. Cloning and optimal Gaussian individual attacks for a continuous-variable quantum key distribution using coherent states and reverse reconciliation. *Physical Review A*, 73 :032302, 2006.

[22] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. Quantum cryptography. *Review of Modern Physics*, 75 :145, 2002.

[23] Alexios Beveratos, Rosa Brouri, Thierry Gacoin, André Villing, Jean-Philippe Poizat, and Philippe Grangier. Single Photon Quantum Cryptography. *Physical Review Letters*, 89 :187901, 2002.

[24] Norbert Lütkenhaus and Mika Jahma. Quantum key distribution with realistic states : photon-number statistics in the photon-number splitting attack. *New Journal of Physics*, 4 :44, 2002.

[25] Won-Young Hwang. Quantum Key Distribution with High Loss : Toward Global Secure Communication. *Physical Review Letters*, 91 :057901, 2003.

[26] Yi Zhao, Bing Qi, Xiongfeng Ma, Hoi-Kwong Lo, and Li Qian. Experimental Quantum Key Distribution with Decoy States. *Physical Review Letters*, 96 :070502, 2006.

[27] Alessandro Zavatta, Silvia Viciani, and Marco Bellini. Tomographic reconstruction of the single-photon Fock state by high-frequency homodyne detection. *Physical Review A*, 70 :053821, 2004.

[28] Daniel Collins, Nicolas Gisin, and Hugues De Riedmatten. Quantum relays for long distance quantum cryptography. *Journal of Modern Physics*, 52(5) :735, 2005.

[29] Frédéric Grosshans and Philippe Grangier. Quantum cloning and teleportation criteria for continuous quantum variables. *Physical Review A*, 64 :010301, 2001.

[30] Samuel L. Braunstein, Nicolas J. Cerf, Sofyan Iblisdir, Peter van Loock, and Serge Massar4. Optimal Cloning of Coherent States with a Linear Amplifier and Beam Splitters. *Physical Review Letters*, 86 :004938, 2001.

[31] Christian Weedbrook, Andrew M. Lance, Warwick P. Bowen, Thomas Symul, Timothy C. Ralph, and Ping Koy Lam. Quantum Cryptography Without Switching. *Physical Review Letters*, 93 :170504, 2004.

[32] Christian Weedbrook, Andrew M. Lance, Warwick P. Bowen, Thomas Symul, Timothy C. Ralph, and Ping Koy Lam1. Coherent-state quantum key distribution without random basis switching. *Physical Review A*, 73 :022316, 2006.

[33] Ulrik L. Andersen, Vincent Josse, and Gerd Leuchs. Unconditional Quantum Cloning of Coherent States with Linear Optics. *Physical Review Letters*, 94 :240503, 2005.

[34] Andrew M. Lance, Thomas Symul, Vikram Sharma, Christian Weedbrook, Timothy C. Ralph, and Ping Koy Lam. No-Switching Quantum Key Distribution Using Broadband Modulated Coherent Light. *Physical Review Letters*, 95 :180503, 2005.

[35] Renato Renner. Security of Quantum Key Distribution. *E-print* `quant-ph/0512258`, 2005.

[36] Frédéric Grosshans and Nicolas J. Cerf. Continuous-Variable Quantum Cryptography is Secure against Non-Gaussian Attacks. *Physical Review Letters*, 92(4), 2004.

[37] William Beckner. Inequalities in Fourier Analysis. *Annals of Mathematics*, 102(1) :159–182, 1975.

[38] Frédéric Grosshans. Collective Attacks and Unconditional Security in Continuous Variable Quantum Key Distribution. *Physical Review Letters*, 94 :020504, 2005.

[39] Miguel Navascu'es and Antonio Ac'in. Security Bounds for Continuous Variables Quantum Key Distribution. *Physical Review Letters*, 94 :020505, 2005.

[40] A. S. Holevo. The capacity of the quantum channel with general signal states. *IEEE Transactions on Information Theory*, 44(1) :269–273, 1998.

[41] Miguel Navascues, Frederic Grosshans, and Antonio Acin. Optimality of Gaussian Attacks in Continuous Variable Quantum Cryptography. *E-print* `quant-ph/0608034`, 2006.

[42] Raul Garcia-Patron and Nicolas J. Cerf. Unconditional optimality of Gaussian attacks against continuous-variable QKD. *E-print* `quant-ph/0608032`, 2006.

[43] Michael M. Wolf, Geza Giedke, and J. Ignacio Cirac. Extremality of Gaussian Quantum States. *Physical Review Letters*, 96 :080502, 2006.

[44] Renato Renner, Nicolas Gisin, and Barbara Kraus. Information-theoretic security proof for quantum-key-distribution protocols. *Physical Review A*, 72 :012332, 2005.

[45] Matthias Christandl, Renato Renner, and Artur Ekert. A Generic Security Proof for Quantum Key Distribution. *E-print* `quant-ph/0402131`, 2004.

[46] N. Gisin, S. Fasel, B. Kraus, H. Zbinden, and G. Ribordy. Trojan-horse attacks on quantum-key-distribution systems. *Physical Review A*, 73 :022320, 2006.

[47] Ch. Silberhorn, T. C. Ralph, N. Lütkenhaus, and G. Leuchs. Continuous Variable Quantum Cryptography : Beating the 3 dB Loss Limit. *Physical Review Letters*, 89 :167901, 2002.

[48] Ryo Namiki and Takuya Hirano. Security of continuous-variable quantum cryptography using coherent states : Decline of postselection advantage. *Physical Review A*, 72 :024301, 2005.

[49] Matthias Heid and Norbert Lütkenhaus. Security of coherent state quantum cryptography in the presence of excess noise. *E-print* `quant-ph/0608015`, 2006.

[50] Jérôme Lodewyck, Thierry Debuisschert, Rosa Tualle-Brouri, and Philippe Grangier. Controlling excess noise in fiber-optics continuous-variable quantum key distribution. *Physical Review A*, 72 :050303(R), 2005.

[51] H. Hansen, T. Aichele, C. Hettich, P. Lodahl, A. I. Lvovsky, J. Mlynek, and S. Schiller. Ultrasensitive pulsed, balanced homodyne detector : application to time-domain quantum measurements. *Optics Letters*, 26 :1714–1716, 2001.

[52] Jérôme Lodewyck, Raúl García-Patrón, Thierry Debuisschert, Rosa Tualle-Brouri, Nicolas J. Cerf, and Philippe Grangier. Experimental implementation of non-Gaussian attacks on a continuous-variable quantum key distribution system. *Submitted*, 2006.

[53] Jérôme Lodewyck, Thierry Debuisschert, Rosa Tualle-Bouri, and Philippe Grangier. Système de distribution quantique de clé de cryptage à variables continues. *Numéro FR 04 13337*, 2004.

[54] `http://subversion.tigris.org`.

[55] `http://www.stack.nl/~dimitri/doxygen`.

[56] `http://gcc.gnu.org/`.

[57] `http://www.cmake.org`.

[58] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1992.

[59] Douglas R. Stinson. Universal Hashing and Authentication Codes. *Lecture Notes in Computer Science*, 576 :74–85, 1991.

[60] C.H. Bennett, G. Brassard, C. Crepeau, and U.M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6) :1915–1923, 1995.

[61] Thomas J. Richardson et Rüdiger L. Urbanke. Efficient Encoding of Low-Density Parity-Check Codes. *IEEE Transactions on Information Theory*, 47(2).

[62] K.C. Nguyen, G. Van Assche, and N.J. Cerf. Side-Information Coding with Turbo Codes and its Application to Quantum Key Distribution. In *International Symposium on Information Theory and its Applications, ISITA 2004, Parma, Italy*, 2004.

[63] Gilles Brassard and Louis Salvail. Secret-key reconciliation by public discussion. *Lecture Notes in Computer Science*, 765 :410–423, 1994.

[64] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, , and C. G. Peterson. Fast, efficient error reconciliation for quantum cryptography. *Physical Review A*, 67 :052303, 2003.

[65] M. Tüchler and J. Hagenauer. EXIT charts of irregular codes. In *Conference on Information Sciences and Systems, Princeton University*, 2002.

[66] R.G. Gallager. Low density parity check codes. *IRE Trans. Inform. Theory*, 8(21) :21–28, 1962.

[67] D. MacKay and R. Neal. Near Shannon Limit Performance of Low Density Parity Check Codes. *Electronics Letters*, 33(6) :457–458, 1997.

[68] U. Wachsmann and J.B. Huber R.F.H. Fischer. Multilevel codes : theoretical concepts and practical design rules. *IEEE Transactions on Information Theory*, 45(5) :1361–1391, 1999.

[69] T. J. Richardson, M. A. Shokrollahi, , and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2) :619–637, 2001.

[70] `http://lthcwww.epfl.ch/research/ldpcopt`.

[71] Amin Shokrollahi. An efficient message-passing schedule for LDPC decoding. *Proceedings of the 23rd IEEE Convention of Electrical and Electronics Engineers in Israel, 2004*, pages 223–226, 2004.

[72] Amin Shokrollahi. LDPC Codes : An Introduction. *http://www.ipm.ac.ir/IPM/homepage/Amin2.pdf*.

[73] A. de Baynast, P. Radosavljevic, A. Sabharwal, and J. Cavallaro. On Turbo-Schedules for LDPC Decoding. *IEEE Communications Letters*, Submitted, 2006.

[74] C. Berrou. The ten-year-old turbo codes are entering into service. *IEEE Communication magazine*, 41(8) :110–116, 2003.

[75] F. Verdier and D. Declercq. A Low Cost Parallel Scalable FPGA Architecture for Regular and Irregular LDPC Decoding. *IEEE Transations on Communications*, 54(7) :1215–1223, 2006.

[76] R. Brent, S. Larvala, and P. Zimmermann. A Fast Algorithm for Testing Irreducibility of Trinomials mod. Technical Report PRG-TR-13-00, Oxford University Computing Laboratory, 2000.

[77] Ueli Maurer and Stefan Wolf. Information-Theoretic Key Agreement : From Weak to Strong Secrecy for Free. *Lecture Notes in Computer Science*, 1807 :351+, 2000.

[78] Rosa Tualle-Brouri. *Mémoire d'habilitation à diriger des recherches. Dispositifs pour la cryptographie quantique.* 2002.

[79] Douglas R. Stinson. Universal Hashing and Authentication Codes. *Lecture Notes in Computer Science*, 576 :74–85, 1991.

[80] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Transactions on Information Theory*, 28 :55–67, 1982.

[81] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20 :284–287, 1974.

[82] Jr G. David Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3) :268, 1973.

[83] P Robertson, P Hoeher, and E Villebrun. A Comparison of Optimal and Sub-optimal MAP Decoding Algorithms Operating in the Log Domain. *IEEE International Conference on Communications, ICC'95 Seatle*, 2 :1009–1013, 1995.

[84] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding : Turbo-codes. *IEEE International Conference on Communications, 1993. ICC 93. Geneva.*, 2 :1064–1070, 1993.

[85] W. Ryan. A turbo code tutorial, 1997. Submitted to Globecom 1997.

Résumé La distribution quantique de clé permet la transmission d'une clé de cryptage secrète entre deux interlocuteurs distants. Les lois de la physique quantique garantissent la sécurité inconditionnelle du transfert.

L'utilisation de variables continues dans le domaine de l'information quantique, récemment apparue, permet de concevoir des systèmes de distribution quantique de clé qui ne nécessitent que des composants standards de l'industrie des télécommunications. Ces composants ouvrent la voie vers les hauts débits caractéristiques des liaisons en fibres optiques.

Nous avons réalisé un système complet de distribution quantique de clé qui utilise l'amplitude et la phase d'états cohérents pulsés de la lumière modulées selon une distribution gaussienne. Notre système est exclusivement réalisé avec des fibres optiques, et atteint un taux de répétition de 1 MHz. Nous avons caractérisé l'information secrète transmise par ce dispositif. Nous avons validé cette caractérisation en réalisant des attaques quantiques originales qui couvrent l'ensemble des perturbations qui peuvent être envisagées sur la transmission.

Nous avons ensuite adapté des algorithmes de correction d'erreur et d'amplification de secret qui produisent une clé secrète à partir des données expérimentales. Enfin, nous avons conçu un ensemble logiciel autonome qui intègre la gestion de l'expérience aux algorithmes de correction d'erreur.

Ces travaux nous ont permis de distribuer une clé secrète sur une fibre de 25 km avec un taux final de 1 kb/s. Le système que nous avons réalisé sera intégré dans un réseau de distribution quantique de clé faisant intervenir plusieurs collaborateurs européens.

Mots clés Cryptographie quantique, variables continues, états cohérents, fibres optiques, bruit de photon, réconciliation, codes LDPC.

Abstract Quantum key distribution enables two distant interlocutors to share a secret key. Laws of quantum physics warrant the unconditonal security of the transmitted message. Continuous variables recently applied to the field of quantum information, enable new key distribution protocols that only require standard, off-the-shelves telecom components. These components yield to the high transmission rates of fiber optics communications.

We build a complete quantum key distribution system that encodes the key in gaussian modulated amplitude and phase of coherent states of light. Our system is entirely made of fiber optics components, and works at a repetition rate of 1 MHz. We measured the secret information transmitted by this device and validated this rate by implementing quantum attacks that cover all the possible channel perturbations.

We adapted error correction and privacy amplification algorithms that produce a secret key from experimental data. Then, we desgined a set of automated software that integrate both experiment control and error correction

This work enabled us to transmit a secret key over a 25 km long fiber with a final key rate of 1 kb/s. Our device will be integrated to a quantum key distribution network that involves several european systems.

Keywords Quantum cyrptography, continuous variables, coherent states, fiber-optics, shot-noise, reconciliation, LDPC codes