Université de Montréal

Réduction du transfert inconscient en d'autres primitives de la théorie de l'information

par
Meriem Debbih

Département d'Informatique et de Recherche Opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en Informatique

Août, 2004

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Réduction du transfert inconscient en d'autres primitives de la théorie de l'information

présenté par:

Meriem Debbih

a été évalué par un jury composé des personnes suivantes:

Alain Tapp
(président-rapporteur)

Stefan Wolf
(directeur de recherche)

Miklós Csűrös
(membre du jury)

Mémoire accepté le:
30 septembre 2004

RÉSUMÉ

Ce mémoire traite de la réduction inconditionnellement sécuritaire du transfert inconscient à des canaux bruyants.
Le transfert inconscient est une version simplifiée de calcul multi-parties sur données privées. Il permet à un acteur cryptographique, *Bob*, de choisir un bit b_c parmi deux bits (b_0, b_1) qui lui sont proposés par un autre acteur cryptographique, *Alice*. Un protocole de transfert inconscient sécuritaire doit (1) empêcher *Bob* d'obtenir de l'information sur le bit qu'il n'a pas choisi, $b_{\bar{c}}$, (2) empêcher *Alice* d'obtenir de l'information sur le choix de *Bob*, c.

Dans [Cré97], Claude Crépeau propose un protocole de réduction du transfert inconscient à un canal binaire symétrique. Cette réduction est inconditionnellement sécuritaire pour des canaux d'erreur $\varphi < 0.1982$. Dans ce mémoire, nous montrerons, en améliorant l'analyse de ce protocole, qu'il est possible de réduire le transfert inconscient à un canal binaire symétrique pour n'importe quel canal d'erreur $\varphi < 1/2$.

Nous présenterons également les derniers résultats concernant la réduction du transfert inconscient présentés par Damgård, Kilian et Salvail dans [DKS99]. Nous décrirons le modèle plus complexe et plus réaliste de réduction du transfert inconscient en un canal bruyant injuste. Un canal bruyant injuste est un canal binaire symétrique d'erreur p évoluant dans un intervalle d'erreurs possibles $[\gamma, \delta]$.

Mots Clés : Cryptologie, transfert inconscient, réduction, canal binaire symétrique, sécurité inconditionnelle, canal injuste.

ABSTRACT

This master's thesis is about reducing oblivious transfer to noisy channels. "One-out-of-Two oblivious transfer" is a cryptographic primitive. It allows *Bob* to choose one bit b_c among two bits (b_0, b_1), proposed by *Alice*. To be unconditionally secure, a One-out-of-Two oblivious transfer protocol must have the following properties (1) when *Bob* learns b_c, he must not be able to get $b_{\bar{c}}$, (2) *Alice* must not be able to learn c, the choice of *Bob* .

In [Cré97], Claude Crépeau proposes a reduction protocol from *binary symmetric channel* to One-out-of-Two oblivious transfer. This reduction is unconditionally secure for any channel with error $\varphi < 0.1982$. In this thesis, we will improve this result and show that a tighter information theoretic analysis allow us to assert that the protocol is unconditionally secure for all $\varphi < 1/2$.

We will also present a more realistic model proposed by Damgård, Kilian and Salvail in [DKS99]. In this article, the authors provide a reduction protocol from unfair noisy channels to One-out-of-Two oblivious transfer. An unfair noisy channel is a binary symmetric channel with error probability p lying in an interval $[\gamma, \delta]$.

Keywords : Cryptology, oblivious transfer, reduction, binary symmetric channel, unconditional security, unfair noisy channel.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES NOTATIONS ET DES SYMBOLES

Il est important de mentionner que les notations suivantes ont été utilisées tout au long de ce mémoire :

- Dans tous les protocoles présentés nous référons à *Alice* par $\mathcal{A}$ et à *Bob* par $\mathcal{B}$.
- X désignera une variable aléatoire discrète, $\mathcal{X}$ son alphabet et P_X sa distribution. $|\mathcal{X}|$ est le nombre d'éléments de l'ensemble $\mathcal{X}$.
- $H(X)$ désignera l'entropie d'une variable aléatoire X.
- $\mathbf{I}(X,Y)$ l'information mutuelle entre les variables aléatoires X et Y.
- $H_2(X)$ désignera l'entropie de Rényi de X.
- $\mathbb{F}_2$ représente le corps à deux éléments muni des opérations $\cdot$ et $\oplus$.
  "$\cdot$" représente la multiplication définie pour a, b, par $a \cdot b = 1$ si et seulement si $a = b = 1$.
  "$\oplus$" représente l'addition définie pour a, b , $a \oplus b = 1$ si et seulement si ($a = 1$ et $b = 0$) ou ($a = 0$ et $b = 1$). $a \oplus b = 0$ si et seulement si $a = b$.
  On pose $\bar{b} = 1 - b$ comme étant l'inverse de b dans $\mathbb{F}_2$.
- "$\odot$" est le produit scalaire sur l'espace vectoriel $\mathbb{F}_2^n$.
- $\mathcal{C}[n, k, d]$ est un code de longueur n, de dimension k et de distance minimale d.
- $\emptyset$ représente l'ensemble vide et ζ représente le mot vide.
- BSC_φ représente un canal binaire symétrique de probabilité d'erreur φ.
- (γ,δ)-UNC représente un canal bruyant injuste d'erreur $p \in [\gamma, \delta]$.
- $\binom{2}{1}$-OT désigne la primitive de transfert inconscient, appelé "One-out-of-Two Oblivious Transfer".
- OT_ε désigne la primitive de transfert équivoque standard.
- (p,q)-WOT et (p,q,ϵ)-WOT désignent un transfert inconscient faible.

- Nous désignerons la réduction d'une primitive en une autre primitive par le symbole $\hookleftarrow$. Par exemple, nous noterons que le transfert inconscient se réduit à un canal binaire symétrique par : $\binom{2}{1}$-OT $\hookleftarrow$ BSC_{φ}.
- La version anglaise des symboles de primitives et de protocoles ont été conservés afin d'assurer la cohérence de ce mémoire avec la littérature étudiée.

"I must study politics and war that my sons may have liberty to study mathematics and philosophy. My sons ought to study mathematics and philosophy,[...] in order to give their children a right to study painting, poetry, music,[...]"

John Adams, 12 Mai 1780.

À Leïla, Nihel, Yamina et Abdelkrim.

REMERCIEMENTS

Je ne peux terminer la rédaction de ce mémoire sans remercier de nombreuses personnes sans qui je ne serais pas là aujourd'hui.

Je commencerais par remercier mon directeur de recherche Stefan Wolf pour son aide et sa patience. Merci pour m'avoir initié à la théorie de l'information. Je tiens à exprimer ma gratitude à Jérome Fournier, Alain Tapp et Khalid BenAbdallah, qui sans le savoir m'ont influencé dans le choix de mon cursus universitaire. Jérome Fournier m'a convaincue de faire mon Baccalauréat en mathématiques fondamentales. Alain Tapp m'a donné envie de découvrir l'informatique théorique lors d'une conférence au département de mathématiques. Khalid BenAbdallah a été le premier à me parler de cryptographie.

Je remercie chaleureusement mes camarades et amis des différents laboratoires du DIRO. Merci à Sébastien Gambs, mon lecteur et correcteur. Merci à Maryam Erfani, mon amie et complice des pauses thé ; à Serge Mani Onana pour les nombreux repas et discussions partagés durant ces dernières années. Enfin merci à Anne Broadbend pour son aide durant la période de rédaction.

Je ne peux oublier de remercier mes amis Nadine Bédard, Étienne Dauphin, Temoojin Chalasani et ma colocataire et amie Rim Boukhssimi, pour leur patience et soutien tout au long de ces longs derniers mois. Je remercie particulièrement Arnaud Lina, Djamel Meddah et Stéphane Baldo pour avoir su me défier. Pari tenu.

Un merci très particulier à mes parents Abdelkrim et Yamina qui m'ont offert le meilleur environnement de travail dont on puisse rêver lors de ces deux derniers

mois. Je n'oublierai pas mes deux adorables soeurs, Nihel et Leïla toujours souriantes.

Il me reste à remercier Jürg Wullschleger sans qui ces dernières semaines n'auraient pas été vivables, pour ses cafés qui donnaient la tremblote, pour son humour et pour son inestimable aide à comprendre la théorie de l'information.

À tous MERCI.

Première partie

Cryptologie et Théorie de l'Information

INTRODUCTION

La cryptologie est : *«La science des écritures secrètes et des messages chiffrés, comprenant la cryptographie et la cryptanalyse»*. La cryptographie, des termes grecs *kruptos* qui signifie *caché* et *graphia* qui signifie *écriture*, est *«l'art de chiffrer, de créer des systèmes de chiffrement»*. La cryptanalyse est *«l'art de déjouer ou de casser ces systèmes»*[1]. En effet, le premier objectif de la cryptologie a été la communication secrète.

La première trace de l'utilisation d'un *système cryptographique*, dans l'histoire de l'homme, remonte à l'Égypte ancienne. À cette époque, l'écriture pouvait jouer le rôle d'un système de chiffrement. Mais c'est chez les Grecs, au VIème siècle avant J.C., qu'on retrouve le véritable premier système de chiffrement, le *scytale*. L'expéditeur enroulait une bandelette autour d'un bâton et écrivait dessus longitudinalement. Puis la bandelette était déroulée et expédiée au destinataire. Ce dernier avait besoin de connaître le diamètre du bâton afin de déchiffrer le message. Le *chiffrement de César* est un autre exemple connu de système cryptographique ancien. Utilisé par l'armée romaine, ce système consistait à décaler l'alphabet de trois lettres.

Aujourd'hui, la définition la plus juste de la cryptologie serait celle que donne Ronald Rivest dans [Riv90] quand il dit que *«la cryptologie concerne la communication en présence d'adversaires»*[2].

---

[1] Ces trois citations sont extraites du *Grand Dictionnaire Terminologique*. Réalisé par l'Office québécois de la langue française, ce dictionnaire peut être consulté à l'adresse http ://www.oqlf.gouv.qc.ca/

[2] *«Cryptography is about communication in the presence of adversaries.» [Riv90].*

En effet, les acteurs cryptographiques, comme *Alice* et *Bob* sont des personnages qui veulent réaliser des *tâches cryptographiques* très variées. En dehors du fait qu'ils n'aimeraient pas être espionnés par *Ève*, qui est malhonnête et trop curieuse, ils aimeraient, avant tout, pouvoir se faire confiance et se protéger l'un de l'autre.

Tâches cryptographiques

Une *tâche cryptographique* est un échange ou un transfert d'information entre deux entités qui offre un service précis et qui possède des propriétés propres de sécurité . Plus exactement, une tâche cryptographique remplit des fonctions spécifiques lors d'un transfert de données.
Afin de mieux comprendre ce concept, présentons quelques exemples simples de tâches cryptographiques.

Communication secrète

Nous avons vu plus haut que le premier objectif de la cryptographie a été la communication secrète. C'est aussi la plus connue des tâches cryptographiques. Si *Alice* veut envoyer un message secret à *Bob*, ils doivent s'assurer que la communication respectera deux propriétés : la *confidentialité* et *l'authenticité*. Plus précisément, *Alice* veut envoyer un message à *Bob* sans utiliser un moyen de communication privé et sans que *Ève* ne puisse en prendre connaissance. C'est ce qu'on appelle la *confidentialité*. D'autre part, *Bob* veut être sûr que le message qu'il reçoit provient effectivement d'*Alice* et non d'*Ève* . C'est ce qu'on appelle *l'authenticité*.

Ces deux propriétés définissent la tâche de communication secrète qui a longtemps été le cheval de bataille des cryptographes. Même si cette tâche cryptographique est aujourd'hui encore très utilisée, il devient de plus en plus nécessaire de pouvoir effectuer d'autres tâches.

Calcul multi-parties sur données privées

Le *calcul multi-parties sur données privées* permet à plusieurs participants d'effectuer un calcul utilisant des informations secrètes provenant de chacun d'eux sans pour autant devoir se les révéler mutuellement.

Le calcul multi-parties est introduit par Yao dans [Yao82] avec le *problème du millionaire.* Ce problème consiste à trouver parmi deux millionnaires lequel est le plus riche sans qu'aucun des deux ne dévoile à l'autre le montant exact de sa fortune. Cette notion est formalisée de la manière suivante : *«N participants avec chacun une entrée secrète x_i désirent calculer $y = F(x_1, x_2, ..., x_N)$ de telle façon que chacun apprend le résultat de la fonction sans rien apprendre au sujet des x_i provenant des autres participants, à l'exception bien entendu, de ce qui peut être déduit par la connaissance de F, y et de son propre x_i.»* [Tap95].

Un autre exemple d'application de cette tâche cryptographique, peut être un peu plus sérieux et vraisemblablement d'actualité, est le *vote en ligne.* Le but ici est de permettre à des citoyens de voter à partir de chez eux tout en s'assurant de l'identité du votant. On souhaite également pouvoir calculer de façon fiable le résultat des élections afin de le communiquer aux électeurs. Chaque électeur veut être sûr que son vote a été comptabilisé et que les élections ne sont pas truquées. Avec un calcul multi-parties il est possible de réaliser cette tâche.

Le Transfert Inconscient

Le *transfert inconscient* est une tâche cryptographique simple et importante. En attendant de donner une définition formelle au chapitre 2, nous pouvons dire qu'il s'agit d'un cas particulier de calcul multi-parties. Sa version la plus simple, le *transfert équivoque*, peut être vu comme un moyen d'introduire une incertitude particulière dans une communication de données[3].

[3] Étant le sujet principal de notre mémoire cette tâche sera définie et étudiée aux chapitres 2,

L'étude du transfert inconscient prend de l'ampleur à partir du moment où Goldreich et Vainish [GV88], et parallèlement Kilian dans [Kil88] démontrent que tout calcul multi-parties peut être réalisé à partir du transfert inconscient.

En résumé, nous pouvons dire qu'une tâche cryptographique est un concept abstrait d'un type de communication idéale qu'*Alice* et *Bob* veulent réaliser entre eux. Afin de concrétiser cette communication, les acteurs cryptographiques doivent construire ce qu'on appelle des *protocoles cryptographiques.*

Procédés cryptographiques ou protocoles

La cryptologie comme toutes les sciences, a un coté abstrait et un coté plus terre à terre et concret. Son coté terre à terre réside sûrement dans les protocoles.

Afin de réaliser une tâche les acteurs cryptographiques doivent poser certaines hypothèses. Cela peut être aussi trivial que d'affirmer qu'ils sont l'un en face de l'autre et qu'ils peuvent se parler. Toutefois, il est souvent plus intéressant et raisonnable de supposer qu'une certaine distance les sépare et qu'ils ont un moyen de communication à leur disposition.A partir de cette hypothèse, *Alice* et *Bob* vont imaginer une succession d'action à effectuer qui leur permettra de réaliser une tâche cryptographique en toute sécurité. C'est ce qu'on a appelé : *protocole* ou *procédé cryptographique.*

En fait, souvent *Alice* et *Bob* ont besoin d'hypothèses plus précises et plus fortes que simplement celle de l'existence d'un moyen de communication. L'existence d'un canal bruyant, l'existence d'un canal quantique, et les hypothèses calculatoires

---

3 et 4.

(comme la difficulté de factoriser les grands entiers, d'extraire le logarithme discret ou les racines carrées modulaires) sont des exemples d'axiomes à partir desquels sont élaborés des protocoles cryptographiques connus. Ils constituent des éléments de bases irréfutables ou évidents qui permettent l'élaboration de modèles mathématiques qui implantent des tâches cryptographiques simples ou complexes.

Un *protocole cryptographique* est donc une recette qui permet d'accomplir concrètement une tâche cryptographique. Il est construit à partir d'axiomes de la cryptologie et de façon à assurer la sécurité de la tâche spécifique qu'il offre.

Primitives cryptographiques et réduction

De façon générale, une *primitive* est un élément de base qui permet d'effectuer une tâche simple. La primitive va généralement servir à construire par *réduction* des éléments plus complexes. Elle peut être vue comme une sorte de brique utilisée pour construire des pans de murs. En cryptographie de nombreux procédés sont conçus à partir d'enchaînements ou d'assemblages de primitives qui effectuent une tâche cryptographique particulière, plus simple que celle offerte par le procédé global. Un générateur pseudo-aléatoire, un canal binaire symétrique, un transfert inconscient, ou encore un canal de communication quelconque sont des exemples de primitives[4].

La construction d'éléments complexes à partir de primitives s'appelle la *réduction*. Conceptuellement, cette notion suppose que des participants ont accès à des *boîtes noires* d'éléments primitifs *idéaux* et qu'à partir de ces éléments, et de préférence sans invoquer aucune autre hypothèse cryptographique, soit construite une tâche cryptographique plus complexe.
La réduction doit être sécuritaire. Nous entendons par là qu'idéalement la sécurité

[4]Nous donnerons au chapitre 1 une définition formelle d'une primitive cryptographique.

du protocole final obtenu doit dépendre uniquement de la sécurité des primitives sur lesquelles il se base.

L'intérêt de l'une réduction réside dans le fait qu'elle permet de diminuer la dépendance de l'implantation au matériel et à la technologie utilisés. En effet, l'objectif ultime d'une réduction serait de parvenir à baser un protocole cryptographique sur une hypothèse très générale et très vague, comme l'existence d'un canal bruyant. Essentiellement, la réduction sert à simplifier les preuves de sécurité des protocoles. Si la réduction est sécuritaire, il suffit de montrer que la primitive sur laquelle elle se base le soit pour prouver que le protocole global l'est aussi. Elle permet parfois, d'améliorer l'efficacité de l'implantation d'une tâche cryptographique.

Sécurité des systèmes cryptographiques

L'élément le plus important d'un système cryptographique est sa sécurité. On dit d'un protocole qu'il est sécuritaire si celui-ci implante une tâche cryptographique empêchant un adversaire d'obtenir plus d'information que celle prévue par cette tâche. Idéalement, l'adversaire ne doit pas pouvoir modifier cette tâche non plus. La sécurité d'un protocole dépend de deux éléments essentiels très liés : le type d'adversaires auxquels il peut être exposé ainsi que son implantation.

Les adversaires

Un adversaire est un personnage cryptographique malhonnête. C'est un être malicieux qui essaiera toujours d'obtenir de l'information à laquelle il n'a pas droit. Il peut également tenter de saboter le fonctionnement du système cryptographique. Un exemple simple serait celui d'un espion en temps de guerre qui essaye de déchiffrer un message secret ou de le modifier pour dérouter ses ennemis. Ou encore, pour reprendre l'exemple du millionaire, un des deux millionaires pourrait vouloir

apprendre le montant de la fortune de l'autre. Un adversaire peut être extérieur aux participants qui tentent d'exécuter un protocole ou il peut faire partie du groupe concerné. Nous nous intéresserons ici à ce dernier type d'adversaire.

Nous serons sans doute d'accord sur le fait que tous les adversaires sont des êtres vils et très curieux. Néanmoins, n'oublions pas que sans eux le cryptographe n'a plus aucune raison d'être. De plus, ils ne sont pas tous de même nature. Lors de l'élaboration d'un protocole, il faut être attentif au genre d'adversaires auxquels on pourrait être exposé. Un adversaire peut être *passif* ou *actif*.

Un adversaire passif[5] : sera celui qui tentera d'obtenir de l'information à partir d'une exécution honnête du protocole. C'est à dire qu'il l'exécutera tout à fait normalement mais pendant l'exécution, ou une fois celle-ci terminée, il tentera d'utiliser les informations obtenues pour en déduire d'autres. Cet adversaire est relativement inoffensif. Nous pouvons le traiter de *curieux*. Cette curiosité peut cependant être dangereuse et non-désirable dans certains cas.

Un adversaire actif : L'adversaire actif, quant à lui, essaiera d'obtenir de l'information par une exécution malhonnête du protocole. Au lieu de se contenter d'exécuter le procédé, il tentera de le modifier sans que les autres participants ne s'en rendent compte. Son but étant d'obtenir plus d'information ou d'induire les autres participants honnêtes en erreur.

Implantation et sécurité

Si l'adversaire est la raison d'être d'un cryptographe, la sécurité d'un protocole est le reflet de son art et de son talent. La sécurité est toujours relative à un adversaire. Certains protocoles peuvent être sécuritaires par rapport à un adversaire

[5] Cette notion apparaît la première fois dans [GMW87]. Il est possible de croiser, dans la littérature, le terme "semi-honest adversary" ou "honest but curious".

passif mais non par rapport à un adversaire actif. Idéalement, un protocole devrait être sécuritaire contre tous types d'adversaires.

En sécurité cryptographique, deux écoles de pensées existent : celle des protocoles à *sécurité calculatoire* et celle des protocoles à *sécurité inconditionnelle.*

Protocoles à sécurité calculatoire :

La sécurité calculatoire est basée sur l'hypothèse qu'un adversaire a une capacité de calcul limitée. Cette limitation l'empêche de résoudre en un temps raisonnable certains problèmes mathématiques.

De nos jours, la majorité des systèmes cryptographiques est basée sur des problèmes mathématiques dits *difficiles.* Parmi ces systèmes on peut distinguer deux catégories. Dans la première catégorie, on trouve les systèmes cryptographiques qui ne peuvent être brisés que si et seulement si on peut résoudre le problème mathématique à partir duquel ils sont construits. Nous pouvons donner l'exemple du protocole d'échange de clés Diffie-Hellman [DH76]. Maurer et Wolf ont démontré que sous certaines conditions bien spécifiques la résolution du problème de Diffie-Hellman est équivalente à la résolution du problème du logarithme discret [MW99]. L'autre catégorie est celle des systèmes cryptographiques qui n'ont pas été prouvés équivalents à un problème mathématique difficile. Par exemple le problème de Diffie-Hellman n'est pas encore prouvé équivalent au problème du logarithme discret pour un groupe quelconque.

La factorisation des grands entiers, l'extraction de logarithme discret ou plus généralement les fonctions à sens unique sont des exemples connus de problèmes dits difficiles. Toutefois, la complexité de ces problèmes n'est pas formellement prouvée, ou du moins, elle ne l'est pas encore. Il est donc nécessaire de supposer que ces

problèmes sont *véritablement difficiles* à résoudre.

Ce genre de protocoles a deux inconvénient majeurs. Le premier est que la difficulté des problèmes mathématiques reste étroitement liée à la technologie utilisée. De façon générale, plus notre moyen d'analyse est puissant moins ces problèmes sont difficiles à résoudre. Il a même été démontré il y a quelques années qu'avec un ordinateur quantique, certains de ces problèmes pourraient être résolus efficacement [Sho97].

Le deuxième inconvénient est que la famille des problèmes difficiles est très petite. La majorité des systèmes cryptographiques est basée sur une poignée de problèmes connus. On peut imaginer qu'une catastrophe économique, politique et technologique, s'abattrait sur l'humanité si jamais quelqu'un trouvait une solution intelligente et efficace à l'un de ces problèmes ou s'il construisait un ordinateur quantique. Ce scénario ressemble à un roman de science fiction mais un cryptographe doit imaginer le pire scénario possible afin de produire le meilleur des systèmes cryptographiques.

Protocoles à sécurité inconditionnelle :

Comme son nom l'indique un *protocole à sécurité inconditionnelle* ne se base pas sur l'hypothèse que l'adversaire est limité dans sa capacité de calcul. Il se fonde sur des hypothèses plus générales comme le constat du comportement aléatoire de l'univers.

L'intérêt pour les protocoles à sécurité inconditionnelle naît en même temps que la théorie de l'information de Shannon. La première preuve de sécurité inconditionnelle est celle que donne Shannon concernant les systèmes cryptographiques de chiffrement. Dans sa preuve, il montre qu'il est nécessaire d'avoir une clé de

longueur au moins égale à la longueur du message afin d'obtenir une sécurité parfaite. Plus exactement, Shannon prouve dans [Sha49] que le chiffrement du *masque jetable*[6] est le seul système inconditionnellement sûr, ce qui décourage les cryptographes de l'époque et dirige leurs efforts vers la sécurité calculatoire.

Toutefois, ces dernières années, on remarque un intérêt croissant pour ce type de protocoles. Ce regain d'attention est dû a plusieurs raisons. La technologie avance à très grands pas. Ainsi, beaucoup de systèmes cryptographiques valables il y a vingt ans sont aujourd'hui désuets, car ne résistant pas aux nouvelles ressources matérielles. De plus la possibilité de partager ces ressources et de les cumuler (réseaux, Internet...) facilite le bris de certains de ces systèmes.
En même temps, l'apparition de résultats fondamentaux en cryptographie quantique ainsi que l'incroyable progrès fait dans le domaine de l'informatique quantique motivent les chercheurs à construire des systèmes qui résisteraient à ces nouveaux ordinateurs.

Nous pouvons dire, en citant Ueli Maurer, que ce nouvel intérêt pour la sécurité inconditionnelle est dû, essentiellement, à l'important constat que : *«la cryptographie a lieu dans un monde physique (chaque communication est basée sur un processus physique) dans lequel personne ne peut avoir une information complète sur l'état d'un système, que cela soit dû au bruit ou aux limites théoriques de la physique quantique»*[Mau99][7].

Ce type de sécurité est très alléchant. Il nous promet que pour n'importe quel adversaire et partant de certaines hypothèses il serait possible de construire des systèmes cryptographiques quasiment inviolables.

---

[6] Connu sous le nom de "*One-Time Pad*" et dû à Vernam.
[7] Traduction libre de l'auteur.

Plan du mémoire

Le présent mémoire traitera de la réduction inconditionnellement sécuritaire du transfert inconscient en des primitives plus simples de la théorie de l'information, les canaux bruyants. Notre objectif sera de réaliser cette tâche cryptographique simple, à l'aide de primitives encore plus simples et ce, de façon inconditionnellement sécuritaire.

Dans le premier chapitre, nous présenterons des notions clés des théories des probabilités et de l'information. Ces notions nous seront utiles pour formaliser dans les chapitres suivants la sécurité des protocoles de réduction.
Dans la deuxième partie, qui comprend les chapitres 2, 3 et 4, nous étudierons le transfert inconscient. Au deuxième chapitre, nous commencerons par définir cette primitive et ses différentes versions. Au chapitre suivant, nous montrerons comment il est possible de réaliser un transfert inconscient à partir d'un canal binaire symétrique. Dans ce chapitre nous améliorerons l'analyse du protocole donné par Claude Crépeau [Cré97] et montrerons qu'il est possible de réduire un transfert inconscient à un canal binaire symétrique pour n'importe quel canal d'erreur inférieure à $1/2$. Au dernier chapitre, nous verrons brièvement les derniers travaux concernant la réduction du transfert inconscient à d'autres primitives plus simples. Nous présenterons les résultats de Damgård, Kilian et Salvail qui décrivent une réduction du transfert inconscient à un modèle plus réaliste de canal bruyant, le canal injuste.

CHAPITRE 1

PROBABILITÉS ET THÉORIE DE L'INFORMATION

La notion d'information apparaît au début du siècle dernier dans plusieurs disciplines, qui à l'époque, ne semblaient pas reliées : la physique, la statistique et les télécommunications. Dans le domaine des télécommunications, Claude Elwood Shannon est considéré comme un des fondateurs, si ce n'est le fondateur de ce qu'on appelle aujourd'hui la théorie de l'information. En effet, en 1948, il publie l'article, *"A Mathematical Theory of Communication"*[1] dans lequel, et ce d'après lui, il complète une théorie générale de la communication dont les bases ont été posées par Nyquist [Nyq24] et Hartley [Har28]. En réalité, Shannon pose les fondements d'une nouvelle théorie mathématique qui va largement influencer le domaine d'étude de la communication.

Dans ce chapitre, nous présenterons brièvement des notions essentielles de cette théorie. Dans la première partie, nous donnerons un court rappel de notions de bases de la théorie des probabilités. Dans la seconde partie, nous introduirons la théorie de l'information. Pour une introduction plus approfondie à la théorie de l'information et à la théorie des probabilités il est possible de consulter : [For66, Bla87, CT91].

[1] *«Probably no single work in this century has more profoundly altered man's understanding of communication than C E Shannon's article, "A mathematical theory of communication", first published in 1948[...] The subject thrived and grew to become a well-rounded and exciting chapter in the annals of science.»* D. Slepian (ed.), Key papers in the development of information theory, Institute of Electrical and Electronics Engineers, Inc. (New York, 1974).

1.1 Notions de probabilités

1.1.1 Distribution d'une variable aléatoire

Expérience aléatoire :

Une *expérience aléatoire* est décrite par un ensemble de résultats possibles Ω déterminés au hasard selon une loi de probabilité P.
Chaque *événement élémentaire* a dans *l'ensemble fondamental* Ω est associé à une probabilité $P(a)$. (Ω, P) est appelé un *espace probabilisé.*
$A \subset \Omega$ est un *événement* et $P(A) = \sum_{a \in A} P(a)$ est la probabilité que l'événement A se produise.

Lorsqu'une expérience aléatoire est effectuée, comme le lancer d'un dé plusieurs fois, il peut être intéressant de connaître non pas l'issue de l'experience mais simplement un résultat dépendant de cette issue, comme la somme des résultats de tous les lancers. On parle alors de *variable aléatoire.*

Variable aléatoire :

Définition 1.1 (Variable aléatoire).
Une *variable aléatoire* (v.a.) X est une application de l'ensemble des résultat Ω d'une expérience aléatoire vers un ensemble quelconque $\mathcal{X}$.

$$X : \Omega \to \mathcal{X} \ .$$

Une v.a. X est associée à une loi de probabilités P_X appelée *distribution de la variable aléatoire* telle que :

$$P_X : \mathcal{X} \to [0, 1] \ .$$

$$\forall\, x \in \mathcal{X},\ P_X(x) = \sum_{\{a : X(a) = x\}} P(a) \ .$$

On note $X(\Omega, \mathcal{X}, P_X)$, une v.a. de Ω vers $\mathcal{X}$, de distribution P_X. On dit qu'elle est *discrète* si l'ensemble Ω est fini ou infini dénombrable.

Propriétés de la distribution d'une variable aléatoire discrète :

1. $\forall A \subset \mathcal{X},\ 0 \leq P_X(A) \leq 1$.
2. $P_X(\emptyset) = 0$
3. $P_X(\mathcal{X}) = 1$.
4. Si $A \subset B \ \Rightarrow\ P_X(A) \leq P_X(B)$.
5. $P_X(A \cup B) = P_X(A) + P_X(B) - P_X(A \cap B)$.

Indicateur de distribution d'une variable aléatoire discrète :

Pour une v.a. discrète $X(\Omega, \mathcal{X}, P_X)$, nous introduisons des quantités indicatrices de la nature et des propriétés de sa distribution P_X .

Définitions 1.1.

1. L'*espérance* de X est la moyenne pondérée des valeurs de X. Elle est définie par :

$$E[X] = \sum_{x \in \mathcal{X}} x P_X(x) \ .$$

2. La *variance* V de X donne une indication de la dispersion de la variable aléatoire autour de son espérance. Elle est définie par :

$$V[X] = E[(X - E[X])^2] \ .$$

3. L'*écart type* σ est un autre indicateur de la dispersion de la variable aléatoire autour de son espérance. Il est défini par :

$$\sigma(X) = \sqrt{V[X]} \ .$$

1.1.2 Distributions dans un espace joint

Comme nous verrons plus loin dans l'étude des systèmes de communication, il est important de pouvoir analyser le comportement simultané de plusieurs variables aléatoires. Considérons les v.a. discrètes $X(\Omega, \mathcal{X}, P_X)$ et $Y(\Gamma, \mathcal{Y}, P_Y)$.

Loi de probabilité jointe :

Définition 1.2 (Loi Jointe).
On définit une *loi de probabilité jointe* comme une distribution P_{XY} sur l'espace joint $\mathcal{X} \times \mathcal{Y}$.

$$P_{XY} : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$$

Il n'est pas évident de déterminer directement une loi de probabilité jointe P_{XY} à partir de P_X et P_Y. Par contre, on peut définir de façon immédiate, les *lois marginales* de X et de Y, respectivement P_X et P_Y, à partir de P_{XY} :

$$P_X : \mathcal{X} \rightarrow [0, 1]$$

$$P_X(x) = \sum_{y \in \mathcal{Y}} P_{XY}(x, y) \ .$$

Et

$$P_Y : \mathcal{Y} \rightarrow [0, 1]$$

$$P_Y(y) = \sum_{x \in \mathcal{X}} P_{XY}(x, y) \ .$$

Loi de probabilité conditionnelle :

Nous pouvons, également, nous demander ce que devient la distribution d'une variable aléatoire une fois que le résultat d'une autre est connu. Est ce que la connaissance de l'une influence la distribution de l'autre ? La notion de probabilité conditionnelle répond à cette question.

Définition 1.3 (Loi Conditionnelle).

La probabilité conditionnelle de X sachant Y est définie par :

$$P_{X|Y} : \mathcal{X} \times \mathcal{Y} \to [0,1]$$

Où

$$P_{X|Y}(x,y) = \frac{P_{XY}(x,y)}{P_Y(y)} .$$

Note 1.1. $P_{X|Y}$ n'est pas la distribution d'une variable aléatoire sur $\mathcal{X} \times \mathcal{Y}$, mais pour un y donné, $P_{X|Y=y}$ définit une distribution sur $\mathcal{X}$.

Indépendance de deux variables aléatoires :

Le fait de considérer simultanément le comportement de deux variables aléatoires, introduit la notion d'*indépendance*. On dit que deux événements A et B, sont *indépendants* si aucun n'influence l'autre. En terme de probabilité, on dit que A et B sont indépendants si $P(A \cap B) = P(A)P(B)$. Formellement nous pouvons dire que deux variables aléatoires sont indépendantes si,

Définition 1.4.

Deux v.a. sont *indépendantes* si :

$$\forall\, x \in \mathcal{X}, y \in \mathcal{Y},\ P_{XY}(x,y) = P_X(x)P_Y(y) .$$

1.1.3 Inégalités utiles

Les deux théorèmes suivants seront utilisés dans la preuve de sécurité de certains protocoles. Ils sont cités ici, sans démonstration.

Théorème 1.1 (Loi des Grands Nombres de Bernstein). [2]

[2]Pour plus de détails voir [Rén70].

Soit $X = (X_1, X_2, ..., X_n)$, *tel que* $\forall\, i \in \{1, 2, ..., n\}$ $X_i(\Omega, \{0, 1\}, (P_{X_i}(1) = p, P_{X_i}(0) = 1 - p)$ *est une v.a. discrète alors :*

$$\forall\ \delta \in\ ]0, p(p-1)], \Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} x_i - p\right| \geq \delta\right] \leq 2^{-n\delta^2}\ .$$

Définition 1.5 (Fonction Convexe).

Une fonction f est dite *convexe* sur un intervalle $[a, b]$ si : $\forall\ x_1, x_2 \in [a, b]$ et $0 \leq \lambda \leq 1$:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2)\ .$$

Théorème 1.2 (Inégalité de Jensen). [3]

Pour une fonction convexe f *et une v.a.* X, *on a :*

$$E[f(X)] \geq f(E[X])\ .$$

Note 1.2. $(-\log)$ est une fonction convexe.

[3] Pour la preuve voir [CT91].

1.2 La théorie de l'information

Les systèmes cryptographiques sont un cas particulier de systèmes de communication. Ces derniers traitent et transférent des données, ou plus généralement de l'information.

La formalisation de l'étude des systèmes de communication naît assez tard dans l'histoire de la science. Elle date de quelques décennies seulement. Nous pouvons nous questionner sur la raison de cette naissance tardive alors que la communication et la notion d'information existent depuis que l'homme existe. Alfréd Rényi donne une explication à ce retard dans [Rén84]. Il dit qu'il a d'abord fallu réaliser que *«le flot d'information pouvait être exprimé numériquement de la même façon que la distance, le temps ou la masse...»*[4], avant de pouvoir penser à une théorie mathématique et axiomatique pour étudier les systèmes de communication.

C'est donc en 1948, que C. E. Shannon introduit un formalisme rigoureux d'étude de ces systèmes. Dans son article [Sha48], Shannon pose les fondements de la théorie de l'information. Il commence par définir ce qu'est un système général de communication, puis il introduit de nouveaux outils de mesure de l'information.

[4]Traduction libre de l'auteur.

1.2.1 Système de communication

Un modèle général de système de communication est illustré par le schéma suivant :

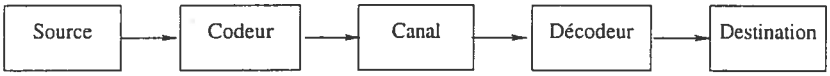

FIG. 1.1 – Modèle général de système de communication.

Chaque système de communication est constitué des éléments suivants :

La source d'information : Elle produit un message qui doit être communiqué à la destination.

Le codeur : Il produit un code du message idéal à être envoyé.

Le canal : Il est le medium qui transmet physiquement le message.

Le décodeur : Il sert à restituer le message envoyé à partir du message reçu.

La destination : Elle est l'entité vers laquelle le message a été envoyé.

Afin d'illustrer ce modèle à l'aide d'un exemple simple, nous pouvons considérer un système cryptographique où *Alice* veut envoyer un message secret à *Bob*. *Alice* détient une *source* lui permettant de générer un message à envoyer. Elle encode ce message à l'aide d'une méthode de chiffrement (le *codeur*), puis le transmet à *Bob* par un moyen de transport quelconque (le téléphone, la poste ou Internet par exemple). *Bob* reçoit le message, transmis à travers le *canal* choisi, et le décode à l'aide de la méthode de déchiffrement correspondante, (le *décodeur*), sur laquelle il s'était entendu avec *Alice* auparavant.

1.2.2 Mesure de l'information

Lorsque nous avons introduit dans le chapitre précédent les tâches cryptographiques, nous avons insisté sur la notion de sécurité des procédés qui les implémentent. Nous avons vu qu'il y avait deux différents types de sécurité, calculatoire et inconditionnelle.
Un protocole inconditionnellement sécuritaire, assure une tâche cryptographique sans faille pour n'importe quelle entrée et pour n'importe quel adversaire (aussi puissant soit-il et participant ou non au procédé). Cela veut dire que le protocole cryptographique permet à *Alice* et à *Bob* de recevoir l'information déterminée par les spécifications de la tâche qu'il implémente et rien d'autre. Quand aux adversaires extérieurs au protocole, ils n'obtiennent aucune information, du moins aucune qui soit significative.

Afin d'évaluer la sécurité des protocoles, il est nécessaire de posséder des outils de mesure de l'information. Lors de la conception d'un protocole cryptographique, nous souhaiterions évaluer la quantité d'information qu'un participant ou un observateur pourrait obtenir à partir d'une exécution, honnête ou non, de ce protocole.

Entropie et Incertitude :

Shannon introduit la notion d'entropie comme mesure et d'information et d'incertitude. Afin de donner une intuition quant à la définition de cette mesure, il faut d'abord comprendre comment est modélisé un système de communication à l'aide d'espaces probabilisés. Prenons l'exemple d'un système {source-canal-destination}.

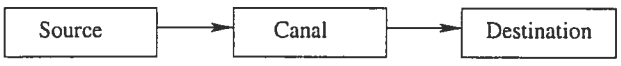

FIG. 1.2 – Modèle simple de système de communication.

Lorsque *Alice* choisit un message à envoyer à *Bob* , on peut voir ce choix comme la réalisation d'un événement x d'une v.a. X d'alphabet $\mathcal{X}$. $\mathcal{X}$ est l'ensemble des messages qu'*Alice* pourrait choisir. Alice choisit un message parmi un nombre fini (ou infini dénombrable) de messages possibles donc $X(\Omega, \mathcal{X}, P_X)$, représentant la source, est une variable aléatoire discrète.
De manière similaire, la destination peut être représentée par une variable aléatoire discrète $Y(\Gamma, \mathcal{Y}, P_Y)$, qui sera en général dépendante de X.
Le canal est représenté par des distributions conditionnelles $\{P_{Y|X=x}\}$, définies sur l'espace $\mathcal{Y}$, telles qu'à chaque $x \in \mathcal{X}$ est associée une distribution de Y, $P_{Y|X=x}$.

Une fois ce système modélisé, plusieurs questions se posent. Par exemple, on peut se demander quelle quantité d'information est contenue dans une source X ou encore, si un événement se produisait quelle quantité d'information apporterait-il ? Si l'expérience n'a pas encore été effectuée, quelle incertitude a-t-on sur son issue ? On peut également se poser la même question sur Y, et sur la connaissance de X si Y est connu. Toutes ces questions vont nous faire aboutir à la notion d'entropie. Il est intéressant de noter que Shannon aboutit à la formulation de cette mesure de façon axiomatique[5].

Entropie de Shannon :

Définition 1.6 (Entropie de Shannon).
Soit $X(\Omega, \mathcal{X}, P_X)$ une v.a. discrète, l'entropie de X est définie par :

$$H(X) = -K \sum_{x \in \mathcal{X}} P_X(x) \log P_X(x) \ .$$

où K est lune constante positive qui fixe l'unité de mesure. Pour $K = 1$ et le

[5]Pour plus de détails consulter [Sha48] ou [Sal91].

logarithme[6] en base 2 on parle de mesurer l'entropie en bits.

Interprétation :

L'entropie peut être vue comme une mesure de quantité d'information ou une mesure d'incertitude d'une variable aléatoire. En effet, elle est une mesure de la quantité moyenne d'information qu'apporte une variable aléatoire discrète lorsqu'une expérience se réalise. Mais aussi lorsque le résultat de l'experience est inconnu on peut voir l'entropie comme une mesure de l'incertitude que nous avons sur cette v.a.. Donc selon le point de vue où on se situe, l'entropie de Shannon est un indicateur d'une quantité d'information ou d'une quantité d'incertitude.

Propriétés de l'entropie de Shannon :

- Pour $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ on a :

$$0 \leq H(X) \leq \log |\mathcal{X}| = \log n \ .$$

  $H(X)$ atteint son maximum pour une distribution uniforme sur $\mathcal{X}$ c'est à dire si $\forall\ x \in \mathcal{X},\ P_X(x) = \frac{1}{n}$.
  $H(X) = 0$ si et seulement si X a une issue unique. C'est à dire qu'il n'y a pas d'incertitude concernant l'issue de l'expérience.

- La distribution binomiale est importante et particulière. Soit X une variable aléatoire de distribution $P_X = (\varphi, 1-\varphi)$ pour $\varphi \in [0,1]$, $|\mathcal{X}| = 2$. Alors on note $h(\varphi)$ l'entropie de cette v.a. et $h(\varphi) = -\varphi \log \varphi - (1-\varphi) \log (1-\varphi)$.
  La Fig.1.3 illustre la variation de l'entropie d'une binomiale en fonction de φ.

Entropie conjointe et conditionnelle

[6] Tout au long de ce mémoire log représentera le logarithme en base 2.

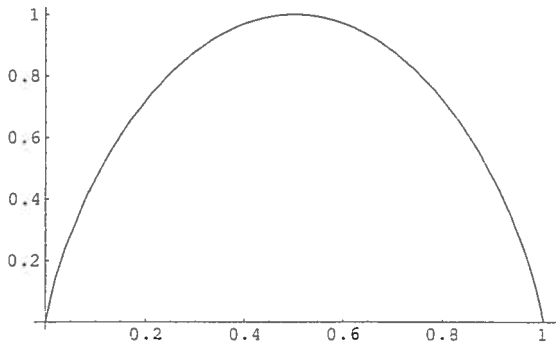

FIG. 1.3 – Entropie de Shannon d'une variable aléatoire de distribution binomiale.

De façon analogue à la définition d'une distribution conjointe, nous introduisons l'entropie conjointe $H(X,Y)$ sur l'espace joint $\mathcal{X} \times \mathcal{Y}$ et l'entropie conditionnelle $H(X|Y)$.

Définition 1.7 (Entropie Conjointe).
L'entropie conjointe est l'incertitude moyenne sur la réalisation d'un événement dans l'espace $\mathcal{X} \times \mathcal{Y}$. Elle est définie par :

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log\left(P_{XY}(x,y)\right) .$$

Définition 1.8 (Entropie Conditionnelle).
L'entropie conditionnelle est définie par :

$$H(X|Y) = \sum_{y\in\mathcal{Y}} P_Y(y)H(X|Y=y) .$$

Où $H(X|Y=y) = -\sum_{x\in\mathcal{X}} P_{X|Y}(x,y)\log\left(P_{X|Y}(x,y)\right)$ est la quantité d'incertitude moyenne que contient X lorsque Y est connue.

Information mutuelle

On appelle information mutuelle, la mesure quantifiant le changement d'incertitude d'une variable aléatoire lorsqu'on en apprend une autre. Cette grandeur nous

renseigne sur la quantité d'information qu'une v.a. Y apporte sur une autre v.a. X.

Définition 1.9 (Information Mutuelle).

Soit X et Y deux v.a. discrètes. On définit l'information mutuelle de X et Y par :

$$\mathbf{I}(X;Y) = H(X) - H(X|Y)$$

Propriétés :

- Règle de la chaîne : $H(X,Y) = H(X|Y) + H(Y)$.
- $\forall$ X et Y v.a. discrètes, $H(X|Y) \leq H(X)$ avec égalité si et seulement si X et Y sont indépendantes.
- $\forall X, Y$ v.a. discrètes $\mathbf{I}(X;Y) \geq 0$ avec égalité si et seulement si X et Y sont indépendantes. Ce qui correspond à l'intuition qu'on a que si deux v.a. sont indépendantes, la connaissance de l'une n'apporte aucune information sur l'autre.

- $\forall$ X, Y v.a. discrètes $\mathbf{I}(X;Y) = \mathbf{I}(Y;X)$.

Entropie de Rényi et Min-entropie

L'entropie de Shannon a été le premier indicateur de quantité d'information utilisé. Elle possède de multiples applications. Entre autres, elle permet d'avoir un indicateur de la longueur minimum à laquelle une source peut être compressée sans erreur de décompression[7]. Elle est également un indicateur du maximum de la quantité d'information qui peut être transmise par un canal avec une erreur négligeable[8]. Toutefois, il y a des situations où l'utilisation de cette mesure n'est pas

[7] Théorème de codage de Shannon.

[8] Deuxième théorème de Shannon. Voir section 1.2.3

très adéquate. C'est la raison pour laquelle d'autres mesures de l'information ont été développées.

Définition 1.10 (Entropie de Rényi).
Soit $X(\Omega, \mathcal{X}, P_X)$ une v.a. discrète et $\alpha \geq 0$, $\alpha \neq 1$. On définit l'entropie de Rényi d'ordre α de X par :

$$H_\alpha(X) := \frac{1}{1-\alpha} \log \sum_{x \in \mathcal{X}} P_X(x)^\alpha \ .$$

Note 1.3. Nous utiliserons essentiellement l'entropie de Rényi d'ordre 2, notée $H_2(X) = -\log P_c(X)$ où $P_c(X) = \sum_{x \in \mathcal{X}} P_X(X)^2$ est appelée *probabilité de collision* de X.

Définition 1.11 (Min-entropie).
Soit $X(\Omega, \mathcal{X}, P_X)$ une v.a. discrète. On définit la min-entropie par :

$$H_\infty(X) := -\log \max_{x \in \mathcal{X}}(P_X(x)) \ .$$

C'est l'information que contient la valeur la plus probable de X. Nous pouvons l'interpréter comme une indication que la distribution de X est au moins autant aléatoire qu'une distribution uniforme sur un alphabet de cardinalité H_∞[9].

Note 1.4. Les différentes mesures d'entropie vérifient la relation suivante :
$\forall$ X v.a. discrète,

$$0 \leq \frac{H_2(X)}{2} \leq H_\infty(X) \leq H_2(X) \leq H(X) \leq \log|\mathcal{X}| \tag{1.1}$$

[9] Voir [HILL91] pour plus de détails.

Les trois égalités de droite sont vérifiées pour une distribution de X uniforme.

1.2.3 Deuxième théorème de Shannon

Afin de pouvoir énoncer le deuxième théorème de Shannon nous devons introduire les notions de primitive, de canal et de codes.

Primitive :

Nous reprenons ici, la définition d'une primitive donnée par Maurer dans [Mau99].

Définition 1.12 (Primitive).

Une *primitive* est un mécanisme abstrait (qui peut être vu comme un service offert par une personne de confiance) auquel chaque joueur $A_1, A_2, ..., A_n$ a accès. A chaque appel de la primitive, chaque joueur peut fournir une entrée (secrète ou pas) X_i d'un certain domaine, et reçoit une sortie (secrète ou pas) Y_i d'un certain rang en fonction d'une certaine distribution de probabilité conditionnelle (généralement publique) $P_{Y_1,...,Y_n|X_1,...,X_n}$ des sorties étant données les entrées.

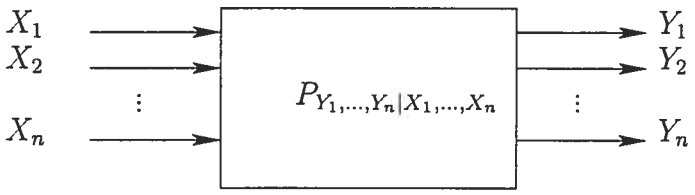

FIG. 1.4 – Primitive en théorie de l'information.

Canaux discrets :

Définition 1.13 (Canal Discret).

Un *canal discret* est une primitive dont l'entrée est une variable aléatoire discrète

X d'alphabet $\mathcal{X}$. La sortie est une variable aléatoire discrète Y d'alphabet $\mathcal{Y}$. Et de probabilité conditionnelle $P_{Y|X}$ qui exprime la probabilité d'observer la sortie y étant donné l'entrée x.

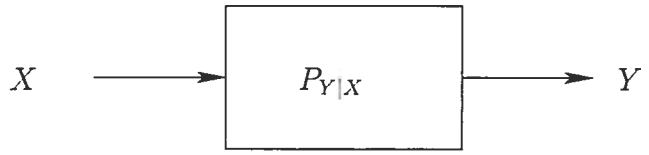

FIG. 1.5 – Un canal discret.

On parle de *canal discret sans mémoire*, si la sortie du canal ne dépend que de la dernière entrée et d'aucune autre entrée ou sortie précédente. Nous utiliserons dans les chapitres suivants uniquement des canaux sans mémoire, plus précisément les canaux bruyants sans mémoire. De façon générale, un *canal bruyant* est un canal qui modifie aléatoirement le signal. Nous verrons à la fin de ce chapitre quelques exemples de canaux discrets sans mémoire.

Définition 1.14 (Capacité d'un Canal).
Étant donné un canal, sa capacité est définie comme étant le maximum de l'information que Y peut produire à propos de X pris sur toutes les distributions possibles de X. En d'autres termes :

$$C = \max_{P_X} \mathbf{I}(X;Y) \ .$$

Les codes correcteurs :

Les codes correcteurs permettent de coder une source X de manière à pouvoir ensuite détecter et corriger les erreurs induites par un canal bruyant. Dans ce qui suit nous considérerons uniquement les codes correcteurs binaires de longueur fixe. On suppose que l'alphabet d'entrée des canaux est toujours $\mathcal{X} = \mathbb{F}_2$ et on pose $\mathcal{X}^n = \mathbb{F}_2^n$.

Définition 1.15 (Distance de Hamming).

On définit la distance de Hamming sur $\mathbb{F}_2^n$ entre deux éléments c_1 et c_2 comme étant le nombre de positions où ils diffèrent. On la note $d_H(c_1, c_2)$.

Définition 1.16 (Code Linéaire).

On définit un *code binaire linéaire* $\mathcal{C}[n, k, d]$ comme un sous-espace linéaire de $\mathbb{F}_2^n$ de dimension k, de cardinalité 2^k et de distance minimale $d = min\{d_H(c_1, c_2)|c_1, c_2 \in \mathcal{C}, c_1 \neq c_2\}$. Nous appellerons un élément c de $\mathcal{C}$ un *mot de code.*

Définition 1.17 (Taux d'un code).

Le taux d'un code binaire de longueur n et de cardinalité M est égal à

$$R = \frac{\log M}{n} .$$

Lorsque $M = 2^k$, $R = \frac{k}{n}$.

Définitions 1.2 (Matrice Génératrice et Matrice de Contrôle).

- G est une *matrice génératrice* d'un code linéaire $\mathcal{C}[n, k, d]$ si ses lignes forment une base de $\mathcal{C}$. Une matrice génératrice est de taille $k \times n$ et de rang k.
- $\mathcal{H}$ est une *matrice de contrôle* (ou de *parité*) de $\mathcal{C}$ si $\forall\ c \in \mathcal{C}\ \mathcal{H}c^t = 0$. i.e. $\mathcal{C} = \ker(\mathcal{H}^t)$.

Un algorithme de décodage de $\mathcal{C}$ à *vraisemblance maximale* est une procédure qui associe à chaque message reçu par un canal bruyant le mot de code le plus proche, en terme de distance de Hamming. Nous utiliserons dans les protocoles du chapitre 3, la méthode de décodage par syndrome.

Définition 1.18 (Syndrome d'un Mot de Code).

Soit $\mathcal{C}[n, k, d]$ un code de matrice de parité $\mathcal{H}$ et $y \in \mathbb{F}_2^n$. Le syndrome Syn(y) de y

est défini par :

$$\begin{aligned} \text{Syn} &: \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^{n-k} \\ \text{Syn}(y) &= \mathcal{H}y^t. \end{aligned}$$

Par définition, le syndrome d'un mot de code est le vecteur nul. La méthode de décodage par syndrome utilise le fait que $\text{Syn}(e+y) = \text{Syn}(e) + \text{Syn}(y)$. Ainsi le syndrome d'un mot de code altéré permet de retrouver l'erreur produite, puis de retrouver le mot de code envoyé. Pour une introduction aux codes nous référons le lecteur à [MS77, Mac03, Sen04].

Le deuxième théorème de Shannon :

En quelques mots, ce théorème surprenant, énonce qu'il est possible d'envoyer de l'information à travers un canal bruyant de manière fiable. C'est à dire que pour un taux inférieur à la capacité du canal en question, il est possible de trouver une séquence de codes avec une probabilité d'erreur de décodage qui tend vers zéro lorsque la longueur du code tend vers l'infini. Nous énonçons ce théorème fondamental de la théorie de l'information sans en donner la preuve. Nous présentons le théorème suivant la formulation donnée dans [Mac03].

Théorème 1.3.

1. *Pour un canal discret de capacité C,*

   Pour $R < C$ et $\delta > 0$, il existe un code $\mathcal{C}(n,k,d)$ de taux $\frac{n}{k} \geq R$ et tel que, pour n suffisamment grand, la probabilité de décoder incorrectement un mot reçu soit inférieure à δ.

2. *Si une probabilité d'erreur par bit p_b est acceptable, des taux jusqu'à $R(p_b)$ sont réalisables, où*

$$R(p_b) = \frac{C}{1 - h(p_b)}$$

3. *Pour une probabilité quelconque p_b les taux supérieurs à $R(p_b)$ ne sont pas réalisables.*

Exemples de canaux discrets :

Nous présentons dans la dernière section de ce chapitre, quelques exemples de canaux binaires qui nous seront utiles dans les chapitres suivants.

Un *canal binaire sans erreur* est un canal qui transmet chaque bit sans aucune modification. La capacité de ce canal est $C = 1$.

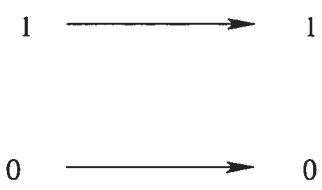

FIG. 1.6 – Canal binaire sans erreur.

Un *canal binaire symétrique* est un canal qui avec une probabilité de φ inverse le bit envoyé, et avec une probabilité de $1 - \varphi$ ne le modifie pas. La capacité d'un tel canal est $C_\varphi = 1 - h(\varphi)$.

Définition 1.19 (Canal Binaire Symétrique).
Un *canal binaire symétrique* ou BSC_φ[10] est un canal bruyant avec entrée $X(\Omega, \mathbb{F}_2, P_X)$, et de sortie $Y(\Gamma, \mathbb{F}_2, P_Y)$, telle que sa matrice de transition $(P_{Y|X})$ est :
$Pr(Y = \bar{x}|X = x) = \varphi$ et $Pr(Y = x|X = x) = 1 - \varphi$.

Lorsqu'un bit est envoyé par un *canal à effacement*, avec une probabilité égale à $1 - \varepsilon$ il est reçu inchangé et avec probabilité ε il est perdu. La capacité d'un tel canal est $C_\varepsilon = 1 - \varepsilon$.

[10]Binary Symmetric Channel.

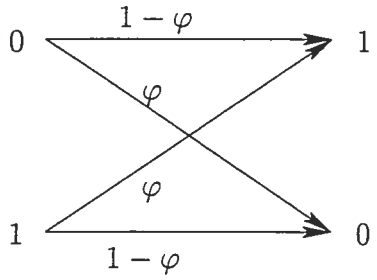

FIG. 1.7 – Canal binaire symétrique.

Définition 1.20 (Canal Binaire à Effacement).
Un *canal binaire à effacement* est un canal bruyant avec entrée $X(\Omega, \mathbb{F}_2, P_X)$ et de sortie $Y(\Gamma, \mathbb{F}_2 \cup \{\zeta\}, P_Y)$, où ζ représente le *mot vide*, avec une matrice de transition $(P_{Y|X})$:
$Pr(Y = x|X = x) = 1 - \varepsilon$ et $Pr(Y = \zeta|X = x) = \varepsilon$.

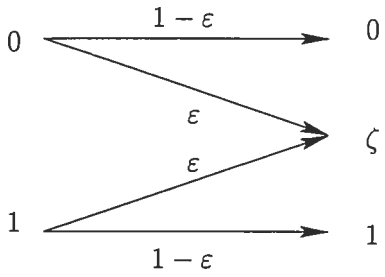

FIG. 1.8 – Canal binaire à effacement.

Un bit qui passe par un *canal binaire à erreur et à effacement* est soit perdu ou inversé. La capacité d'un canal binaire à erreur et à effacement est c.

Définition 1.21 (Canal Binaire à Erreur et à Effacement).
Un *canal binaire à erreur et effacement* est un canal bruyant avec entrée $X(\Omega, \mathbb{F}_2, P_X)$, et sortie $Y(\Gamma, \mathbb{F}_2 \cup \{\zeta\}, P_Y)$. Sa matrice de transition est $(P_{Y|X})$:

$Pr(Y = x|X = x) = 1 - \varphi - \varepsilon,$

$Pr(Y = \bar{x}|X = x) = \varphi,$

Et $Pr(Y = \zeta|X = x) = \varepsilon.$

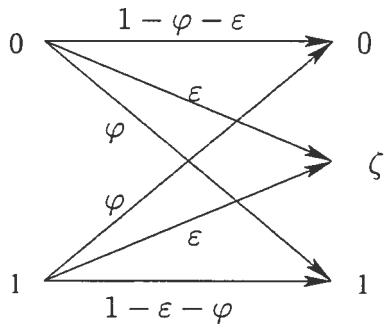

FIG. 1.9 – Canal binaire à erreur et effacement.

Deuxième partie

Transfert Inconscient

CHAPITRE 2

DÉFINITION DU TRANSFERT INCONSCIENT

Le *transfert inconscient* est introduit pour la première fois en 1981 par Rabin dans [Rab81] qui définit une version simplifiée du transfert inconscient : le transfert équivoque. Wiesner généralise cette notion en l'introduisant sous l'appellation *"message multiplexing"* [Wie83]. En réalité, Wiesner est le premier à élaborer une idée de transfert inconscient dès le début des années 70 mais son travail ne sera publié qu'en 1983. En parallèle et de manière indépendante, Even, Goldreich et Lempel définissent une généralisation du transfert inconscient en présentant $\binom{2}{1}$-OT[1] dans [EGL83].

Le transfert inconscient est une primitive simple. Elle est intéressante à utiliser comme brique pour construire d'autres primitives cryptographiques plus complexes. L'étude du transfert inconscient prend réellement de l'ampleur à la fin des années 80 après que Goldreich et Vainish [GV88] et Killian [Kil88] démontrent que toute fonction f de domaine fini et d'image finie peut être *inconsciemment évaluée* si on suppose l'existence d'un protocole de transfert inconscient. Pour être plus précis, cela veut dire que tout calcul multi-parties peut être réduit à un transfert inconscient.

Dans la première partie de ce chapitre, nous définirons les différentes versions du transfert inconscient. Dans la seconde partie nous donnerons une définition de cette même primitive du point de vue de la théorie de l'information. Finalement, dans la troisième et dernière partie, nous verrons quelles ont été les différentes

[1]Pour *One-Out-of-Two Oblivious Transfer*.

pistes suivies pour l'implanter.

2.1 Différentes versions du transfert inconscient

De façon générale, le transfert inconscient est une primitive qui implique deux acteurs cryptographiques *Alice*($\mathcal{A}$) et *Bob* ($\mathcal{B}$).

Définition 2.1 (Transfert Équivoque).
Le transfert équivoque que nous désignerons par OT[2] est une primitive définie pour deux joueurs $\mathcal{A}$ et $\mathcal{B}$ et qui offre le service suivant :

1. $\mathcal{A}$ envoie un bit $b \in \mathbb{F}_2$.
2. $\mathcal{B}$ reçoit b avec probabilité $\frac{1}{2}$ et ζ avec probabilité $\frac{1}{2}$.

Lorsqu'il reçoit un bit b, $\mathcal{B}$ est sûr que c'est ce qui a été envoyé par $\mathcal{A}$. Lorsqu'il reçoit ζ, il ne doit obtenir aucune information sur b. D'autre part, $\mathcal{A}$ ne sait jamais si $\mathcal{B}$ reçoit b ou pas.

$$\mathrm{OT}(b) = \begin{cases} b & \text{avec probabilité } \frac{1}{2} \\ \zeta & \text{avec probabilité } \frac{1}{2} \end{cases}$$

Définition 2.2 (Transfert Équivoque Standard).
Le transfert équivoque standard est un transfert équivoque tel que la probabilité que $\mathcal{B}$ reçoive le bit b est $\varepsilon \in]0,1[$ et la probabilité qu'il ne le reçoive pas est $1-\varepsilon$. Nous le désignerons par OT_ε.

$$\mathrm{OT}_\varepsilon(b) = \begin{cases} b & \text{avec probabilité } \varepsilon \\ \zeta & \text{avec probabilité } 1-\varepsilon \end{cases}$$

[2]Pour Oblivious Transfer.

Note 2.1. Nous pouvons voir le transfert équivoque standard comme un canal binaire à effacement où l'erreur est nécessaire et elle est exactement ε.

Définition 2.3 (Transfert Inconscient).
$\mathcal{A}$ possède deux éléments $b_0, b_1 \in \mathbb{F}_2$. Elle veut donner à $\mathcal{B}$ le choix d'obtenir un seul des deux bits. Plus exactement, un transfert inconscient $\binom{2}{1}$-OT est défini comme suit :

1. $\mathcal{A}$ envoie b_0, b_1.
2. $\mathcal{B}$ envoie c.
3. $\mathcal{B}$ reçoit b_c et aucune information sur $b_{\bar{c}}$.
4. $\mathcal{A}$ n'apprend aucune information sur c.

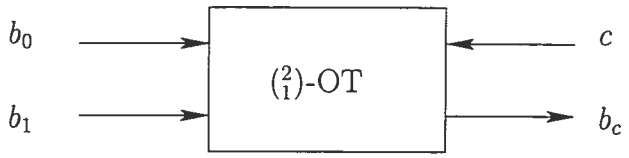

FIG. 2.1 – Transfert inconscient.

Note 2.2. Il est à noter que dans les trois définitions précédentes, il est possible de remplacer $\mathbb{F}_2$ par un alphabet fini ou dénombrable quelconque $\mathcal{X}$.
Brièvement, nous pouvons ajouter qu'il existe plusieurs autres transferts inconscients connus dont le transfert inconscient de chaînes[3] qui sera noté $\binom{2}{1}$-OT^k , pour un choix d'une chaîne parmi deux, de longueur k. Un autre exemple, est le transfert inconscient où le nombre d'entrées est n. Il est noté $\binom{n}{1}$-OT . Dans ce mémoire cependant, nous nous limiterons à l'étude de ceux définis plus haut.

[3] One-out-of-Two String Oblivious Transfer.

2.2 Le transfert inconscient selon la théorie de l'information

Les définitions données ci-haut sont des définitions de tâches cryptographiques abstraites. Nous avons vu dans le premier chapitre qu'une tâche est concrétisée par un protocole cryptographique. Il est donc intéressant et important de définir les particularités d'un protocole implantant un transfert inconscient.
Dans [BCS96], Brassard, Crépeau et Sántha affirment qu'un protocole de transfert inconscient doit être *correct* et *privé*. Ils font appel à la théorie de l'information afin de formaliser ces deux notions. Dans leur article il est question d'un protocole pour $\binom{2}{1}$-OT^k . Nous adapterons ces définitions au $\binom{2}{1}$-OT tout en gardant les notations proposés par les auteurs.

Avant de donner ces définitions, nous devons préciser que le transfert inconscient sous toutes ces formes impliquent deux participants *Alice* et *Bob* . Contrairement à beaucoup d'autres tâches cryptographiques les adversaires d'un protocole de transfert inconscient sont les participants eux-même. Ici nous ne sommes pas intéresser au comportement d'un espion, mais simplement au comportement honnête ou non d'*Alice* et de *Bob* .
D'autre part, le cas où les deux participants trichent en même temps n'est pas intéressant. Le but de la cryptographie est toujours de protéger les personnes honnêtes de celles qui sont malhonnêtes.

2.2.1 Un protocole de $\binom{2}{1}$-OT correct

Notations :

Soit A et B les programmes exécutés par $\mathcal{A}$ et $\mathcal{B}$ respectivement, lors du protocole.
$\widetilde{\mathrm{A}}$ est un programme exécuté par $\mathcal{A}$ malhonnête.
$\left[\mathrm{A}, \mathrm{B}\right]$ représente le protocole réalisé par $\mathcal{A}$ et $\mathcal{B}$. On désigne par $(E, F) = \left[\mathrm{A}, \mathrm{B}\right](b_0, b_1)(c)$

la variable aléatoire qui représente le résultat de l'exécution du protocole $\left[\mathrm{A},\mathrm{B}\right]$ avec les entrées b_0, b_1 et c. Où E représente l'issue du protocole pour $\mathcal{A}$ et F celle pour $\mathcal{B}$.

$\left[\mathrm{A},\mathrm{B}\right]^*(b_0,b_1)(c)$ est la variable aléatoire représentant toute l'information qui peut être générée par l'exécution des programmes A et B mais aussi par tout autre expérience aléatoire additionnelle. Enfin, nous définissons la vue qu'à un participant $\mathcal{P}$ du protocole, par les variables aléatoires marginales : $\left[\mathrm{A},\mathrm{B}\right]_{\mathcal{P}}(b_0,b_1)(c)$ et $\left[\mathrm{A},\mathrm{B}\right]^*_{\mathcal{P}}(b_0,b_1)(c)$.

Il faut mentionner également que *Bob* peut toujours ignorer l'issue d'un protocole, c'est la raison pour laquelle il est parfois utile de mentionner que *Bob* a accepté l'issue du protocole.

Définition 2.4 ($\binom{2}{1}$-OT Correct).

[A,B] est un protocole correct pour $\binom{2}{1}$-OT si :

1. $\forall\, b_0, b_1 \in \mathbb{F}_2$ et $c \in \mathbb{F}_2$,

$$\Pr\Big(\left[\mathrm{A},\mathrm{B}\right](b_0,b_1)(c) \neq (\zeta, b_c)\Big) = 0\ , \tag{2.1}$$

2. $\forall\, \widetilde{\mathrm{A}}, \exists\, \widetilde{\mathrm{S}}$, un programme probabiliste, t.q. $\forall\, b_0, b_1 \in \mathbb{F}_2, c \in \mathbb{F}_2$,

$$\Big(\left[\widetilde{\mathrm{A}},\mathrm{B}\right]_{\mathcal{B}}(b_0,b_1)(c), \mathrm{B\ accepte}\Big) = \Big(\left[\mathrm{A},\mathrm{B}\right]_{\mathcal{B}}(\widetilde{\mathrm{S}}(b_0,b_1))(c), \mathrm{B\ accepte}\Big)\ . \tag{2.2}$$

L'égalité de la condition 2.2 est une égalité de distribution.

Interprétation :

De façon générale, un protocole est correct si le résultat obtenu lors d'une exécution est celui prévu par la tâche cryptographique. Dans le cas d'un transfert inconscient, la condition 2.1, exige que l'issue d'une exécution honnête des deux

participants soit ζ, le mot vide, pour *Alice* et b_c pour *Bob*.

La condition 2.2 signifie que si *Alice* trichait et que *Bob* acceptait le résultat du protocole, elle ne pourrait pas induire une distribution des issues de *Bob* différente que celle qu'elle induirait simplement en changeant ses entrées et en étant honnête. C'est à dire, il existerait une entrée d'*Alice* avec une exécution honnête du protocole qui donnerait le même résultat obtenu en étant malhonnête pendant l'exécution du protocole. Cette deuxième condition est moins intuitive que la première. Il n'est peut être pas évident de comprendre pourquoi il est permis à Alice de tricher. En fait, la condition sous entend qu'il est impossible d'obliger *Alice* à ne pas envoyer ce qu'elle veut à *Bob* . Elle pourra toujours choisir ses entrées.

2.2.2 Un protocole de $\binom{2}{1}$-OT privé :

Définition 2.5 ($\binom{2}{1}$-OT $\mathcal{A}$-**privé**).
Un protocole $\left[\mathrm{A},\mathrm{B}\right]$ est $\mathcal{A}$-*privé* pour $\binom{2}{1}$-OT si $\forall\ (B_0,B_1)$ v.a. de $\mathbb{F}_2^2$ et C v.a. de $\mathbb{F}_2$:
$\forall\ b_0,b_1 \in \mathbb{F}_2$ et $\forall\ \widetilde{\mathrm{A}}$

$$\mathbf{I}\left(C;\left[\widetilde{\mathrm{A}},\mathrm{B}\right]^*_{\mathcal{A}}(B_0,B_1)(C)\middle|(B_0,B_1)=(b_0,b_1)\right)=0\ . \tag{2.3}$$

Définition 2.6 ($\binom{2}{1}$-OT $\mathcal{B}$-**privé**).
Un protocole $\left[\mathrm{A},\mathrm{B}\right]$ est $\mathcal{B}$-*privé* pour $\binom{2}{1}$-OT si $\forall\ (B_0,B_1)$ v.a. de $\mathbb{F}_2^2$ et C v.a. de $\mathbb{F}_2$:
$\forall\ c \in \mathbb{F}_2$ et $\forall\ \widetilde{\mathrm{B}}$, $\exists\ \widetilde{B}_{\widetilde{C}}$ v.a. de $\mathbb{F}_2$ t.q.

$$\mathbf{I}\left(B_{\overline{C}};\left[\mathrm{A},\widetilde{\mathrm{B}}\right]^*_{\mathcal{B}}(B_0,B_1)(C)\middle|C=c,\widetilde{B}_{\widetilde{C}}\right)=0\ . \tag{2.4}$$

Définition 2.7 ($\binom{2}{1}$-OT **privé**).
On dit que le protocole $\left[\mathrm{A}, \mathrm{B}\right]$ est *privé* pour $\binom{2}{1}$-OT s'il est $\mathcal{A}$-privé et $\mathcal{B}$-privé.

Interprétation :

Les conditions 2.3 et 2.4 signifient respectivement que *Alice* et *Bob* n'obtiennent aucune quantité d'information supplémentaire par rapport à ce que le protocole $\left[\mathrm{A}, \mathrm{B}\right]$ leur permet d'obtenir de façon honnête. *Alice* ne doit pas apprendre le choix c de *Bob* même si elle trichait. *Bob* ne doit pas apprendre $b_{\bar{c}}$ même s'il trichait. Les auteurs nous font remarquer dans [BCS96], qu'il n'est pas nécessaire d'exiger que *Bob* connaisse B_C. Il pourrait connaître un $\widetilde{B}_{\widetilde{C}}$, un bit obtenu d'une exécution précédente d'$\binom{2}{1}$-OT. Cependant, dans ce qui suivra on considérera toujours une unique exécution d'$\binom{2}{1}$-OT donc *Bob* obtiendra uniquement B_C.

2.2.3 Protocole statistiquement correct et statistiquement privé

Nous avons défini plus haut les notions de protocole *parfaitement correct* et *parfaitement privé*. Toutefois, ces conditions sont souvent trop restrictives. Dans sa thèse [Cré90] Crépeau définit les notions de protocole *statistiquement correct* et *statistiquement privé*. Ce sont des notions mieux adaptées au monde réel. Nous pouvons dire que des protocoles *statistiquement correct* et *statistiquement privé* sont en quelques sortes *"presque correct"* ou *"presque privé"*.

Définition 2.8 ($\binom{2}{1}$-OT **Statistiquement Correct**).
$\left[\mathrm{A}, \mathrm{B}\right]$ est un protocole *statistiquement correct* pour $\binom{2}{1}$-OT si pour un paramètre de sécurité s,

1. $\forall\, b_0, b_1 \in \mathbb{F}_2$ et $c \in \mathbb{F}_2$,

$$\Pr\Big(\Big[\mathrm{A},\mathrm{B}\Big](b_0,b_1)(c) \neq (\zeta, b_c)\Big) < 2^{-s}\ , \tag{2.5}$$

2. Et $\forall\, \widetilde{\mathrm{A}}, \exists\ \widetilde{\mathrm{S}}$, un programme probabiliste, t.q. $\forall\, b_0, b_1 \in \mathbb{F}_2, c \in \mathbb{F}_2$,

$$\Big(\Big[\widetilde{\mathrm{A}},\mathrm{B}\Big]_{\mathcal{B}}(b_0,b_1)(c), \mathrm{B\ accepte}\Big) = \Big(\Big[\mathrm{A},\mathrm{B}\Big]_{\mathcal{B}}(\widetilde{\mathrm{S}}(b_0,b_1))(c), \mathrm{B\ accepte}\Big)\ . \tag{2.6}$$

Note 2.3. Notons que l'égalité dans la condition 2.6 signifie que les deux distribution sont *statistiquement non-distingable.* Intuitivement, cela veut dire que pour un paramètre s, assez grand, deux familles de distributions $\{X_s\}$ et $\{Y_s\}$ sont statistiquement non-distingable si elles peuvent être remplacées l'une par l'autre sans qu'un juge, limité dans le nombre d'échantillons qu'il peut obtenir, ne puisse s'en rendre compte. Cette notion est définie formellement dans [Cré90].

Définition 2.9 ($\binom{2}{1}$-OT Statistiquement $\mathcal{A}$-privé).
Un protocole $\Big[\mathrm{A},\mathrm{B}\Big]$ est *statistiquement $\mathcal{A}$-privé* pour $\binom{2}{1}$-OT si pour un paramètre de sécurité s, $\forall\ (B_0, B_1)$ v.a. de $\mathbb{F}_2^2$ et C v.a. de $\mathbb{F}_2$:
$\forall\ b_0, b_1 \in \mathbb{F}_2$ et $\forall\, \widetilde{\mathrm{A}}$ t.q. :

$$\mathbf{I}\Big(C; \Big[\widetilde{\mathrm{A}},\mathrm{B}\Big]^*_{\mathcal{A}}(B_0,B_1)(C)\Big|(B_0,B_1) = (b_0,b_1)\Big) < 2^{-s}\ . \tag{2.7}$$

Définition 2.10 ($\binom{2}{1}$-OT Statistiquement $\mathcal{B}$-privé).
Un protocole [A,B] est *statistiquement $\mathcal{B}$-privé* pour $\binom{2}{1}$-OT si pour un paramètre

de sécurité s, $\forall\ (B_0, B_1)$ v.a. de $\mathbb{F}_2^2$ et C v.a. de $\mathbb{F}_2$:
$\forall\, c \in \mathbb{F}_2$ et $\forall\, \widetilde{\mathrm{B}}$, $\exists\, \widetilde{B}_{\widetilde{C}}$ v.a. de $\mathbb{F}_2$,

$$\mathbf{I}\Big(B_{\overline{C}}; \Big[\mathrm{A}, \widetilde{\mathrm{B}}\Big]^*_{\mathcal{B}}(B_0, B_1)(C)\Big| C = c, \widetilde{B}_{\widetilde{C}}\Big) < 2^{-s}\ . \tag{2.8}$$

Définition 2.11 ($\binom{2}{1}$-OT Statistiquement privé).
On dit que le protocole $\Big[\mathrm{A}, \mathrm{B}\Big]$ est *statistiquement privé* pour $\binom{2}{1}$-OT s'il est statistiquement $\mathcal{A}$-privé et statistiquement $\mathcal{B}$-privé.

2.3 Implantation du transfert inconscient

Conceptuellement cette primitive est simple. Le plus difficile reste de l'implanter.

Remarquons d'abord, qu'elle est facile à réaliser en présence d'une tierce personne. Supposons qu'*Alice* et *Bob* connaissent une personne du nom de *Camélia*[4], en qui ils ont tous les deux confiance. Dans ce cas, ils peuvent facilement réaliser un transfert inconscient.

Protocole 2.1. [Protocole Naïf [Tap95]]

1. *Alice* envoie b_1 et b_2 à *Camélia.*
2. *Bob* choisit aléatoirement un bit $c \in \mathbb{F}_2$ et l'envoie à *Camélia.*
3. *Camélia* envoie b_c à Bob.

[4]Traditionnellement la personne de confiance s'appelle *Ted* pour "Trusted Party" ou *Charles* pour confiance. J'ai arbitrairement choisi de la nommer *Camélia* car ce prénom signifie perfection en arabe.

Nous pouvons voir immédiatement que lorsque *Camélia* est une personne de confiance *Bob* reçoit b_c et n'apprend rien sur $b_{\bar{c}}$ et *Alice* n'apprend rien sur c. Cependant, il n'est pas évident de trouver une personne de confiance. Il faut donc trouver une autre solution.
Plusieurs manières d'implanter le transfert inconscient et équivoque ont été développées jusqu'à ce jour. Toutes les implantations connues du transfert inconscient ont finalement le même objectif : remplacer *Camélia* par un protocole interactif entre *Alice* et *Bob*.

2.3.1 Implantation à sécurité calculatoire

Suivant leur tendance à préférer la sécurité calculatoire, les cryptographes ont d'abord élaboré des implantations du transfert inconscient à sécurité calculatoire. Nous citons quelques uns de ces protocoles sans entrer dans les détails.

Rabin, qui a été le premier à introduire le transfert inconscient, proposait un protocole basé sur le problème de factorisation. Ce protocole s'appuie sur l'hypothèse que la factorisation d'un grand nombre entier est un problème difficile à résoudre mais il avait une lacune. Aucune preuve n'a été donnée permettant de montrer qu'il est correct lorsque *Bob* triche. Une quinzaine d'années plus tard, Fisher, Micali et Rackoff [FMR96] proposent un meilleur algorithme toujours basé sur le problème de factorisation, et cette fois, ils donnent la preuve qu'il est correct.

Auparavant Even, Goldreich et Lempel dans [EGL83] avaient proposé une implantation de $\binom{2}{1}$-OT en utilisant un système de chiffrement à clé publique. Une autre implantation de $\binom{2}{1}$-OT est donnée par Goldreich, Micali et Widgerson dans [GMW87]. Ce protocole est basé sur l'hypothèse de l'existence des fonctions

à brèches et est prouvé correct pour des participants passifs seulement.

Nous avons mentionné précédemment que la sécurité calculatoire dépendait généralement de plusieurs hypothèses. Avec le temps et l'évolution de la technologie beaucoup de ces systèmes cryptographiques risquent de devenir obsolètes[5]. C'est pourquoi il est intéressant de se tourner vers des protocoles de transfert inconscient inconditionnellement sécuritaire.

2.3.2 Implantation à sécurité inconditionnelle

Claude Crépeau et Joe Kilian introduisent en 1988 [CK88], la réduction du transfert inconscient en une primitive plus simple, le canal binaire symétrique. Par la suite, le protocole est amélioré dans [Cré97].

Dans le chapitre suivant nous montrerons comment il est possible, à partir de primitives élémentaires, de construire un transfert inconscient. Nous commencerons par donner la preuve de Crépeau sur l'équivalence des transfert équivoque et transfert inconscient. Ensuite, nous présenterons le protocole de réduction du transfert inconscient à un canal binaire symétrique.

[5]Nous concédons que ces systèmes ne seront peut être pas briser dans un futur proche mais ils le seront un jour.

CHAPITRE 3

RÉDUCTIONS DU TRANSFERT INCONSCIENT

Supposons qu'*Alice* et *Bob* possèdent un canal de communication sans erreur à leur disposition. À partir de cette hypothèse, nous aimerions savoir s'il leur est possible de réaliser un protocole de transfert inconscient inconditionnellement sécuritaire.
Énoncé dans la littérature, le fait suivant est connu et admis par tous : *«Il est impossible de construire un transfert inconscient inconditionnellement sécuritaire avec une communication sans bruit. C'est à dire avec un canal sans erreur.»* [DKS99]. Il faut donc, essayer de réduire le transfert inconscient à une autre primitive.

Une des primitives les plus simples et les plus courantes dans les modèles de communication est le canal binaire symétrique. Dans ce chapitre nous commencerons par montrer que $\binom{2}{1}$-OT est équivalent au transfert équivoque. Nous présenterons également un protocole permettant de réduire $\binom{2}{1}$-OT à un canal binaire symétrique.

Nous devons mentionner avant de commencer qu'*Alice* et *Bob* ont toujours à leur disposition un canal binaire non-bruyant qui leur permet de communiquer en clair et sans erreur en tout temps.

3.1 Équivalence des transfert inconscient et transfert équivoque standard

Le *transfert équivoque* et le *transfert inconscient* sont deux tâches cryptographiques très similaires. À première vue, nous pouvons avoir l'impression que la

réduction d'un protocole à l'autre doit être une formalité mais ce n'est pas tout à fait le cas.

3.1.1 La réduction OT $\hookleftarrow \binom{2}{1}$-OT

Il est simple de voir que lorsque *Bob* et *Alice* ont à leur disposition une "*boîte noire*" implantant $\binom{2}{1}$-OT, ils peuvent aisément simuler un transfert équivoque. Le protocole suivant permet de réduire OT à $\binom{2}{1}$-OT de façon inconditionnellement sécuritaire.

Protocole 3.1. [OT $\hookleftarrow \binom{2}{1}$-OT [Cra99]]

1. $\mathcal{A}$ choisit aléatoirement b_0, b_1 tel que $b_0 \oplus b_1 = b$.
2. $\mathcal{B}$ choisit aléatoirement le bit c.
3. $\mathcal{A}$ et $\mathcal{B}$ exécute $\binom{2}{1}$-OT$(b_0, b_1)(c)$. $\mathcal{B}$ obtient b_c.
4. $\mathcal{A}$ choisit aléatoirement un bit t et envoie (t, b_t) à $\mathcal{B}$.

$$\begin{cases} t = c & \Rightarrow \mathcal{B} \text{ ne peut pas calculer } b & \text{Avec probabilité } 1/2 \\ t \neq c & \Rightarrow \mathcal{B} \text{ calcule } b = b_c \oplus b_t & \text{Avec probabilité } 1/2 \end{cases}$$

Nous voyons immédiatement que ce protocole est correct et privé. *Alice* n'apprend rien sur c car $\binom{2}{1}$-OT est sécuritaire par hypothèse et donc ne peut savoir si *Bob* reçoit le bit b ou pas. *Bob* , quant à lui, n'apprend rien sur $b_{\bar{c}}$ pour la même raison. Ainsi il ne peut obtenir d'information sur b si $c = t$. Ils réussissent donc à simuler un OT avec une seule utilisation de $\binom{2}{1}$-OT.

3.1.2 La réduction $\mathrm{OT}_\varepsilon \hookleftarrow \mathrm{OT}$

Afin de réduire un transfert équivoque standard OT_ε à un transfert inconscient, il suffit d'effectuer la réduction $\mathrm{OT} \hookleftarrow \binom{2}{1}$-OT et ensuite utiliser la primitive de transfert équivoque obtenue.
L'idée du protocole de la réduction $\mathrm{OT}_\varepsilon \hookleftarrow \mathrm{OT}$ est d'utiliser OT un nombre n de fois afin de diminuer la probabilité qu'à *Bob* d'obtenir le bit b.

Protocole 3.2. [$\mathrm{OT}_\varepsilon \hookleftarrow \mathrm{OT}$]

1. $\mathcal{A}$ choisit aléatoirement $b_1, b_2, \ldots, b_n$ tel que $\oplus_{i=1}^{n} b_i$.
2. Pour $1 \leq i \leq n$ $\mathcal{A}$ envoie b_i à travers OT. $\mathcal{B}$ reçoit $b'_i = \mathrm{OT}(b_i)$.
3. Si pour $1 \leq i \leq n$ $b'_i \neq \zeta$, alors $\mathcal{B}$ calcule $b = \oplus_{i=1}^{n} b'_i$ sinon il recommence le protocole à l'étape 1.

L'analyse du protocole est immédiate. La sécurité de la nouvelle primitive OT_ε dépend de la sécurité de OT. Avec probabilité $\epsilon = \frac{1}{2^n}$ *Bob* reçoit le bit b.

3.1.3 La réduction $\binom{2}{1}$-OT $\hookleftarrow \mathrm{OT}_\varepsilon$

Il est également possible d'effectuer la réduction inverse $\binom{2}{1}$-OT $\hookleftarrow \mathrm{OT}_\varepsilon$ mais celle-ci est moins évidente.
Dans [Cré88], Claude Crépeau est le premier à prouver la sécurité inconditionnelle de la réduction $\binom{2}{1}$-OT $\hookleftarrow \mathrm{OT}_\varepsilon$ et de ce fait, l'équivalence des deux primitives. Nous présentons ici ce protocole et la preuve qu'il est correct et privé.

On suppose qu'*Alice* et *Bob* veulent réaliser un $\binom{2}{1}$-OT et ils ont un OT_ε à leur disposition.

Protocole 3.3. [$\binom{2}{1}$-OT $\hookleftarrow$ OT_ε [Cré88]]

Pour n assez grand,

1. $\mathcal{A}$ choisit n bits $r_1, r_2, ..., r_n$ aléatoirement, et les envoie à $\mathcal{B}$ par OT_ε.
2. $\mathcal{B}$ envoie à $\mathcal{A}$ $I_0 = \{i_1, i_2, ..., i_t\}$, $I_1 = \{i_{t+1}, i_{t+2}, ..., i_{2t}\}$, tels que $I_0 \cap I_1 = \emptyset$, $\forall\, i_j \in I_c$ $\mathcal{B}$ connaît r_{i_j} et $2t < n$
3. $\mathcal{A}$ calcule et envoie $y_0 = b_0 \oplus \left(\bigoplus_{i \in I_0} r_i \right)$ et $y_1 = b_1 \oplus \left(\bigoplus_{i \in I_1} r_i \right)$.
4. $\mathcal{B}$ calcule et obtient $b_c = \left(\bigoplus_{i \in I_c} r_i \right) \oplus y_c$.

Analyse du protocole

Dans l'analyse d'un protocole il y a deux étapes distinctes. La première étape est la vérification que le protocole implante correctement la tâche de transfert inconscient. La seconde étape est de s'assurer qu'il est sécuritaire contre des participants malhonnêtes.

En premier lieu, nous montrerons que le protocole 3.3 est correct, au moins statistiquement puis qu'il est bien statistiquement privé.

Le protocole 3.3 est statistiquement correct s'il satisfait la condition 2.5. C'est à dire que l'issue d'une exécution pour des participants, *Alice* et *Bob*, tous les deux honnêtes doit être (ζ, b_c). Nous affirmons le résultat suivant :

Lemme 3.1.

Si Alice *et* Bob *ne trichent pas. Pour* $0 < \varepsilon < 1$ *et un paramètre de sécurité* $s > 1$,

si $n > \left\lceil \frac{s}{(\varepsilon-\varepsilon^2)^2} \right\rceil$ *et* $\left\lceil n\varepsilon^2 \right\rceil < t < \left\lfloor n\left(\varepsilon - \sqrt{\frac{s}{n}}\right) \right\rfloor$, *le protocole 3.3 est statistiquement correct.*

Preuve :

Il est évident qu'*Alice* ne reçoit rien à la fin d'une exécution de ce protocole. Cependant, il faut montrer que *Bob* reçoit effectivement b_c. C'est à dire qu'à partir des deux bits y_0 et y_1 il puisse calculer b_c.

Afin de calculer b_c à partir de y_c, *Bob* doit recevoir au moins t bits r_i envoyés par *Alice*. Introduisons les deux v.a. suivantes x_i et X_j :

$$x_i = \begin{cases} 0 & \text{Si } Bob \text{ ne reçoit pas } r_i \\ 1 & \text{Si } Bob \text{ reçoit } r_i \end{cases}$$

Et

$$X_j = \sum_{i=1}^{j} x_i$$

Où la v.a. X_j donne le nombre de bits reçus par *Bob* sur j envoies d'*Alice* . Nous voulons évaluer la probabilité que *Bob* reçoive au moins t bits r_i, donc la probabilité que $X_n \geq t$.

$$\begin{aligned} \Pr(X_n \geq t) &= 1 - \Pr(X_n < t) \\ &= 1 - \Pr(\varepsilon - \frac{X_n}{n} > \varepsilon - \frac{t}{n}) \\ &\geq 1 - \Pr\left(\left|\frac{X_n}{n} - \varepsilon\right| > \varepsilon - \frac{t}{n}\right) \\ &\geq 1 - 2^{-n(\varepsilon - \frac{t}{n})^2} \end{aligned}$$

Où la dernière inégalité est obtenue par application du théorème de Bernstein[1]. Pour un paramètre de sécurité $s > 1$, le théorème de Bernstein est satisfait pour

[1] Théorème 1.1.

$n > \left\lceil \frac{s}{(\varepsilon-\varepsilon^2)^2} \right\rceil$ et $\left\lceil n\varepsilon^2 \right\rceil < t < \left\lfloor n\left(\varepsilon - \sqrt{\frac{s}{n}}\right) \right\rfloor$.
Ainsi on a montré que le protocole est statistiquement correct si *Alice* et *Bob* sont honnêtes. □

Le protocole doit satisfaire également la condition 2.6 pour être correct. Si *Alice* est malhonnête lors de l'exécution, *Bob* accepterait l'issue du protocole seulement s'il existe des entrées qui donneraient le même résultat avec une exécution honnête d'*Alice* .

Lemme 3.2.

Si Alice *est malhonnête. Pour* $0 < \varepsilon < 1$ *et un paramètre de sécurité* $s > 1$, *si* $n > \left\lceil \frac{s}{(\varepsilon-\varepsilon^2)^2} \right\rceil$ *et* $\left\lceil n\varepsilon^2 \right\rceil < t < \left\lfloor n\left(\varepsilon - \sqrt{\frac{s}{n}}\right) \right\rfloor$, *le protocole 3.3 est statistiquement correct.*

Preuve :

Alice peut simplement tricher en envoyant y_0 et y_1 ne correspondant pas à $b_0 \oplus \left(\bigoplus_{i\in I_0} r_i\right)$ et à $b_1 \oplus \left(\bigoplus_{i\in I_1} r_i\right)$ respectivement. Elle enverrait n'importe quels bits aléatoires. Ce qui est équivalent à envoyer d'autres b_0 et b_1.
Donc même si elle triche le protocole reste statistiquement correct pour les mêmes conditions du Lemme 3.1. □

Le protocole doit être $\mathcal{A}$-privé, et donc satisfaire la condition 2.7. Pour cela la quantité d'information qu'*Alice* obtient sur c en trichant doit être nulle ou exponentiellement décroissante en fonction d'un paramètre de sécurité s.

Lemme 3.3.

Le protocole 3.3 est $\mathcal{A}$*-privé.*

Preuve :

La seule information sur c qu'*Alice* obtient sont les deux ensembles I_0 et I_1. Or, la

probabilité que $c = 0$ et que $c = 1$ sont égales. Le contenu des deux ensembles I_0 et I_1 ne donne aucune information sur c. Ainsi :

$$\mathrm{I}\left(C; \left[\widetilde{\mathrm{A}}, \mathrm{B}\right]^*_{\mathcal{A}}(B_0, B_1)(c) | (B_0, B_1) = (b_1, b_2)\right) = 0$$

□

Enfin le protocole doit être $\mathcal{B}$-privé. C'est à dire que *Bob* ne doit pas recevoir d'information sur b_c, ou presque pas (Condition 2.8). Nous affirmons qu'il l'est.

Lemme 3.4.

Pour $0 < \varepsilon < 1$ *et un paramètre de sécurité* $s > 1$. *Si* $n > \left\lceil \frac{s}{(\varepsilon - \varepsilon^2)^2} \right\rceil$ *et* $\left\lceil \frac{n\varepsilon^2}{2} \right\rceil < t < \left\lfloor \frac{n}{2}\left(\varepsilon - \sqrt{\frac{s}{n}}\right) \right\rfloor$ *le protocole est statistiquement* $\mathcal{B}$*-privé.*

Preuve :

Le protocole est $\mathcal{B}$-privé si, même en trichant *Bob* n'obtient pas $b_{\bar{c}}$. C'est à dire qu'une fois qu'il aura déterminé l'ensemble I_c, il ne puisse pas à l'aide de l'ensemble des indices $I_{\bar{c}}$ et de $y_{\bar{b}}$ calculer $b_{\bar{c}}$. En d'autres mots, *Bob* ne doit pas recevoir suffisamment de bits, plus que $2t - 1$, pour calculer les deux bits b_0 et b_1. Toujours en utilisant la v.a. discrète X_j définie précédemment, évaluons :

$$\begin{aligned} \Pr(X_n \geq 2t) &= \Pr(\varepsilon - \frac{X_n}{n} \leq \varepsilon - \frac{2t}{n}) \\ &\leq \Pr\left(\left|\frac{X_n}{n} - \varepsilon\right| \geq \varepsilon - \frac{2t}{n}\right) \\ &\geq 2^{-n(\varepsilon - \frac{2t}{n})^2} \end{aligned}$$

Où la dernière inégalité est obtenue par application du théorème de Bernstein[2]. Pour un paramètre $s > 1$, ce théorème est satisfait pour :

[2]Théorème 1.1.

$$n > \left\lceil \frac{1}{(\varepsilon-\varepsilon^2)^2} \right\rceil \text{ Et } \left\lceil \frac{n\varepsilon^2}{2} \right\rceil < t < \left\lfloor \frac{n}{2}(\varepsilon - \sqrt{\frac{1}{n}}) \right\rfloor.$$

□

En utilisant les Lemmes 3.1, 3.2, 3.3 et 3.4, nous pouvons énoncé le théorème suivant :

Théorème 3.1.

Pour un paramètre de sécurité $s > 1$, $n > \left\lceil \frac{s}{(\varepsilon-\varepsilon^2)^2} \right\rceil$ *et* $\left\lceil n\varepsilon^2 \right\rceil < t < \left\lfloor \frac{n}{2}\left(\varepsilon - \sqrt{\frac{s}{n}}\right) \right\rfloor$ *le protocole 3.3 qui implante la réduction* $\binom{2}{1}\text{-OT} \leftarrow \text{OT}_\varepsilon$ *est statistiquement correct et privé.*

3.2 Réduction du transfert inconscient à un canal binaire symétrique

Nous avons décrit plus haut comment il est possible de construire un transfert inconscient à partir d'un transfert équivoque standard. Cependant, même si cette dernière primitive est très simple, il est intéressant de voir s'il est possible de réduire $\binom{2}{1}$-OT en une primitive encore plus simple comme un canal bruyant. Plus exactement, nous présenterons le travail de Claude Crépeau énoncé dans l'article *"Efficient Cryptographic Protocols Based on Noisy Channels"* [Cré97] dans lequel il propose un protocole de réduction de $\binom{2}{1}$-OT en un canal binaire symétrique BSC_φ.

Nous apporterons des précisions quant à l'analyse de ce protocole et nous montrerons essentiellement que le protocole fonctionne pour n'importe quel canal binaire symétrique BSC_φ d'erreur $\varphi < 1/2$.
Nous commencerons par introduire une notion importante qui sera utilisée dans ce protocole, la distillation de secrets ainsi que des théorèmes utiles pour la preuve de la sécurité inconditionnelle du protocole de réduction.

3.2.1 Distillation de secrets

La méthode de distillation de secrets est introduite par Bennett, Brassard et Robert dans [BBR88]. Elle permet, à *Alice* et *Bob*, de construire à partir d'une chaîne partiellement secrète et d'un protocole interactif, une autre chaîne *"plus secrète"*. Cette méthode est très utile dans les protocoles d'échange de clés secrètes. Par la suite, la technique de distillation de secrets est généralisée par Bennett, Brassard, Crépeau et Maurer dans [BBCM94].

La méthode de distillation de secrets est basée sur les fonctions de hachage de classe universelle et le résultat essentiel s'y rattachant est présenté ci-dessous.

Définition 3.1 (Classe de Fonctions de Hachage Universelle [CW79]).
Une classe $\mathcal{G}$ de fonctions g de $\mathcal{X}$ vers $\mathcal{Y}$ est universelle si pour deux éléments distincts x_1 et x_2 dans $\mathcal{X}$, la probabilité que $g(x_1) = g(x_2)$ est au plus $\frac{1}{|\mathcal{Y}|}$ lorsque g est choisi aléatoirement dans $\mathcal{G}$ selon une distribution uniforme.

Théorème 3.2. [Distillation de Secrets [BBCM94]]
Soit les deux variables aléatoires discrètes $X(\mathcal{X})$ et $Y(\mathcal{Y})$ de distribution jointe quelconque P_{XY}, et x une valeur particulière dans $\mathcal{X}$. Et Soit G une variable aléatoire correspondant au choix uniforme d'une fonction de hachage g dans une classe de fonctions de hachage universelle de $\mathcal{Y}$ vers $\mathbb{F}_2^l$. Si $H_2(Y|X = x) \geq t$ alors :

$$H(G(Y)|G, X = x) \geq H_2(G(Y)|G, X = x) \geq l - \frac{2^{l-t}}{\ln(2)}.$$

Nous introduisons également la notion essentielle d'*ensemble typique*. Cette notion est importante dans ce qui suivra, car elle nous donne une idée du comportement *type* d'une séquence $X^n = (X_1, X_2, ..., X_n)$ de variables aléatoires X_i indépendantes et identiquement distribuées selon P_X lorsque n est grand. En quelques mots, le théorème 3.3 nous indique que *presque* toute instance $(x_1, x_2, ..., x_n)$ de X^n appartient à l'ensemble typique. Et que finalement toutes les *séquences typiques* sont équiprobables. Plus formellement [CT91] :

Définition 3.2.
Soit X^n une séquence de v.a. X i.i.d.[3] selon P_X. Un ensemble typique $\mathcal{S}^n(\epsilon)$ est l'ensemble des séquences $(x_1, x_2, ..., x_n)$ qui satisfont :

$$2^{-n(H(X)+\epsilon)} \leq P_{X^n}(x_1, x_2, ..., x_n) \leq 2^{-n(H(X)-\epsilon)}$$

[3] i.i.d. pour indépendantes et identiquement distribuées.

Note 3.1. Dans tout ce qui suivra, on défini $o(n)$ comme une fonction de n, telle que $\lim_{n\to\infty} o(n) = 0$.

Théorème 3.3.

Soit X^n une séquence de v.a. X i.i.d. selon P_X. Alors les propriétés suivantes sont respectées,

1. *Si $(x_1, x_2, ..., x_n) \in \mathcal{S}^n(\varepsilon)$, alors $P_{X^n}(x_1, x_2, ..., x_n) = 2^{-nH(X)+o(n)}$.*
2. *Pour tout $\delta > 0$, $\Pr[X^n \in \mathcal{S}^n(\varepsilon)] > 1 - \delta/n$, pour n suffisamment grand.*
3. *$|\mathcal{S}^n(\varepsilon)| \leq 2^{nH(X)+o(n)}$.*

Nous utiliserons également le résultat suivant extrait de [CM94] qui donne une idée de la quantité espérée de réduction de l'entropie de Rényi d'une v.a. lorsqu'on apprend l'issue d'une v.a. qui lui est corrélée. Ce théorème indique qu'en moyenne cette réduction ne dépasse pas l'entropie de Shannon de la v.a. connue.

Théorème 3.4. [CM94]

Soit X et U deux variables aléatoires d'alphabet $\mathcal{X}$ et $\mathcal{U}$ respectivement. Alors la réduction espérée de l'entropie de Rényi de X, lorsque U est donné, ne dépasse pas l'entropie de Shannon de U.

$$H_2(X) - H_2(X|U) \leq H(U)$$

3.2.2 La réduction $\mathrm{OT}_\varepsilon \hookleftarrow \mathrm{BSC}_\varphi$

Globalement, la réduction $\binom{2}{1}$-OT $\hookleftarrow \mathrm{BSC}_\varphi$ s'effectue en deux étapes. Une première réduction $\mathrm{OT}_\varepsilon \hookleftarrow \mathrm{BSC}_\varphi$ est exécutée puis un protocole similaire au protocole 3.3 est réalisé.

Alice et *Bob* ont à leur disposition un canal binaire symétrique de probabilité

d'erreur $\varphi < 1/2$ qu'on notera BSC_φ et ils veulent réaliser un OT_ε correct et privé.

Protocole 3.4. [$\widehat{\mathrm{OT}}_\varepsilon \hookleftarrow \mathrm{BSC}_\varphi$]

1. $\mathcal{A}$ envoie bb à travers BSC_φ et $\mathcal{B}$ reçoit $b'_0 b'_1$.
2. Si $b'_0 = b'_1$ alors $\mathcal{B}$ reçoit b' sinon il reçoit ζ.

Analyse du protocole :

Lemme 3.5.

Le protocole 3.4 est incorrect et il n'est pas $\mathcal{A}$-privé.

Preuve :

Protocole Incorrect :

On note la primitive réalisé par ce protocole de réduction $\widehat{\mathrm{OT}}_\varepsilon$. Soit $\widehat{\mathrm{OT}}_\varepsilon(b)$ la v.a. qui représente l'issue du protocole 3.4 avec l'entrée b.

Lorsque le protocole est exécuté et que les deux participants sont honnêtes, *Alice* ne reçoit rien et *Bob* obtient le résultat suivant :

$$\widehat{\mathrm{OT}}_\varepsilon(b) = \begin{cases} b & \text{avec probabilité } (1-\varphi)^2 \\ \bar{b} & \text{avec probabilité } \varphi^2 \\ \zeta & \text{avec probabilité } 2\varphi(1-\varphi) \end{cases}$$

La probabilité que *Bob* reçoive un bit b' est $\varepsilon = \varphi^2 + (1-\varphi)^2$. Remarquons que pour $0 < \varphi < 1/2$ on a $1/2 < \varepsilon < 1$.

Ce protocole simule, incorrectement, un OT_ε. En effet, lorsque *Bob* reçoit $b' = \widehat{\mathrm{OT}}_\varepsilon(b)$: avec probabilité $\frac{\varphi^2}{\varepsilon}$ ce bit a réellement été envoyé par *Alice* et avec pro-

babilité $1 - \frac{\varphi^2}{\varepsilon}$ ce bit est erroné, *Bob* reçoit en fait $\bar{b}$ mais il ne le sait pas.

$\mathcal{A}$-privé :

D'autre part, *Alice* peut tricher aisément. En envoyant $b\bar{b}$, elle sait qu'avec probabilité $\varepsilon > 1/2$, *Bob* ne reçoit rien. Comme *Alice* obtient plus d'information que prévu, ce protocole n'est donc pas $\mathcal{A}$-privé.

Nous verrons dans le protocole suivant, $\binom{2}{1}\text{-}\widehat{\text{OT}} \hookleftarrow \widehat{\text{OT}}_\varepsilon$, que cela pourrait même lui permettre d'obtenir suffisamment d'information pour savoir avec une bonne probabilité quel bit *Bob* a choisi.

□

Crépeau résout le premier problème de ce protocole par l'utilisation des codes correcteurs. Dans la section suivante, nous présentons le protocole et montrons que contrairement au résultat donné dans [Cré97] ce protocole est correct pour n'importe quel canal binaire symétrique d'erreur $\varphi < 1/2$.

3.2.3 $\binom{2}{1}$-$\widehat{\mathrm{OT}} \hookleftarrow \widehat{\mathrm{OT}}_\varepsilon$: un protocole correct

Protocole 3.5. [$\binom{2}{1}$-$\widehat{\mathrm{OT}} \hookleftarrow \widehat{\mathrm{OT}}_\varepsilon$]

Pour $\gamma > 1$,

1. Pour n assez grand, et pour tout $i \in \mathbb{N}$ et $0 < i \leq 2n$,

   $\mathcal{A}$ choisit aléatoirement un bit r_i et l'envoie à $\mathcal{B}$ par $\widehat{\mathrm{OT}}_\varepsilon$. $\mathcal{B}$ reçoit $r'_i = \widehat{\mathrm{OT}}_\varepsilon(r_i)$.

2. $\mathcal{B}$ choisit deux ensembles disjoints I_0, I_1 t.q. $|I_0| = |I_1| = n$, et $(\forall\, i \in I_c \;\; [r'_i \neq \zeta])$.

3. $\mathcal{A}$ et $\mathcal{B}$ se mettent d'accord sur une matrice de contrôle $\mathcal{H}$ d'un code concaténé $\mathcal{C}$ de paramètres $[n, k, d]$ et qui corrige $\gamma\frac{\varphi^2}{\varepsilon}n$.

4. $\mathcal{A}$
   - Calcule et envoie $s_0 \leftarrow \mathrm{Syn}(r_{I_0})$ et $s_1 \leftarrow \mathrm{Syn}(r_{I_1})$,
   - Choisit une chaîne aléatoire $m \in \mathbb{F}_2^n$ et l'envoie à Bob,
   - Calcule et envoie $\hat{b}_0 \leftarrow b_0 \oplus (m \odot r_{I_0})$ et $\hat{b}_1 \leftarrow b_1 \oplus (m \odot r_{I_1})$.

5. $\mathcal{B}$
   - Corrige les erreurs de r'_{I_c} à l'aide de s_c. Il obtient ainsi r_{I_c},
   - Calcule et obtient $b_c = \hat{b}_c \oplus (m \odot r_{I_c})$.

Analyse du protocole :

Sommairement, l'idée du protocole 3.5 est d'utiliser la distillation de secrets et les codes correcteurs pour s'assurer que lorsque *Bob* reçoit les deux bits envoyés par *Alice*, $\hat{b}_0$ et $\hat{b}_1$, il puisse retrouver uniquement un seul des deux bits b_0 et b_1 avec une probabilité d'erreur négligeable. Nous allons voir en détails ici pourquoi

ce protocole est correct et aussi pourquoi il n'est pas $\mathcal{A}$-privé.

Pour cela, nous aurons besoin de considérer les v.a. discrètes et la remarque suivantes :

- $\mathrm{R}_{\mathrm{I_c}}$ et $\mathrm{R}_{\mathrm{I_{\bar{c}}}}$ correspondent aux choix des chaînes r_{I_c} et $r_{I_{\bar{c}}}$. R_i est la v.a. qui correspond au bit à la position i dans les chaînes R_I.
- Les v.a. $\mathrm{Syn}(\mathrm{R}_{\mathrm{I_c}})$ et $\mathrm{Syn}(\mathrm{R}_{\mathrm{I_{\bar{c}}}})$ correspondent respectivement aux syndromes de R_{I_c} et $R_{I_{\bar{c}}}$ et $\mathfrak{S} = \{s | \mathcal{H} r_I^t = s \ , \ r_I \in \mathcal{C}\}$. $|\mathfrak{S}| = n - k$
- Afin de simplifier les notations dans la preuve, nous noterons $\widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I_c}})$ et $\widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I_{\bar{c}}}})$ les v.a. des chaînes reçues par *Bob* à travers $\widehat{\mathrm{OT}}_\varepsilon$.[4]
- La v.a. discrète M correspond au choix de la chaîne binaire m de longueur n.
- L'hypothèse essentielle de cette réduction est que n est grand. Ainsi nous pouvons supposer, avec grande probabilité, que les séquences de bits considérées sont des séquences typiques.

Le protocole 3.5 doit satisfaire aux deux conditions 2.5 et 2.6 pour être statistiquement correct. Nous montrerons en premier lieu que si *Alice* et *Bob* sont honnêtes le protocole est correct. Puis si *Alice* triche, le protocole l'est aussi.

Lemme 3.6.

Si Alice et Bob ne trichent pas. Pour un paramètre de sécurité $\gamma > 1$ et pour un code concaténé $\mathcal{C}[n, k, d]$ tel que $k < (1 - h(\gamma\frac{\varphi^2}{\varepsilon}))n$ et qui corrige $\gamma\frac{\varphi^2}{\varepsilon}$ erreurs, le protocole est statistiquement correct.

Preuve :

[4] Au lieu de noter $(\widehat{\mathrm{OT}}_\varepsilon(r_{i_1}), \widehat{\mathrm{OT}}_\varepsilon(r_{i_2}), ..., \widehat{\mathrm{OT}}_\varepsilon(r_{i_n}))$ et $(\widehat{\mathrm{OT}}_\varepsilon(r_{i_{n+1}}), \widehat{\mathrm{OT}}_\varepsilon(r_{i_{n+2}}), ..., \widehat{\mathrm{OT}}_\varepsilon(r_{i_{2n}}))$ pour $1 \leq j \leq n$, $i_j \in I_c$ et $n+1 \leq j \leq 2n$, $i_j \in I_{\bar{c}}$.

À la fin d'une exécution honnête du protocole, *Bob* doit pouvoir connaître un des deux bits b_0 et b_1 et *Alice* ne doit rien recevoir, ce qui est le cas pour les deux. Ce qui est évident dans le cas d'*Alice* l'est un peu moins dans celui de *Bob* .

Afin d'obtenir b_c, *Bob* doit recevoir suffisamment de bits pour avoir une chaîne r'_{I_c} de longueur n. Il doit également, pouvoir corriger les erreurs contenus dans la chaîne r'_{I_c} avec une faible probabilité d'erreur.
La première condition est satisfaite pour tout $\varphi < \frac{1}{2}$. On a $\varepsilon > \frac{1}{2}$ et donc *Bob* reçoit en moyenne $2\varepsilon n > n$ bits.
Si *Bob* et *Alice* sont honnêtes le nombre d'erreurs contenus dans r'_{I_c} sera en moyenne $\frac{\varphi^2}{\varepsilon}n$. *Bob* veut pouvoir les corriger à l'aide de s_c. Dans [Cré97], des codes concaténés sont utilisés car ils possèdent la propriété suivante :

Proposition 3.1 (Codes Concaténés).
Pour une assez grande longueur de code n, des codes concaténés[5] *peuvent être construits de telle sorte que pour tout $\epsilon > 0$ il existe une constante $\rho > 1$ telle que pour des valeurs de taux $R < C_\epsilon$, les codes peuvent corriger ϵn erreurs et ont une probabilité d'erreur de correction bornée par $\rho^{(R-C_\epsilon)n}$ où C_ϵ est la capacité d'un canal binaire symétrique d'erreur ϵ.*

Donc la probabilité que le protocole donne le bon résultat (ζ, b_c) est la probabilité que *Bob* n'obtiennent pas d'erreur de correction de r'_{I_c}. Par la proposition 3.1 *Alice* et *Bob* choisissent un code $\mathcal{C}[n, k, d]$, qui corrige plus que $\frac{\varphi^2}{\varepsilon}n$, et tel qu'il

[5] *Les codes concaténés sont utilisés pour leur efficacité de codage et de décodage. Pour de plus amples informations sur les codes concaténés voir [For66]. Cependant, il semble possible de remplacer les codes concaténés par des "Low-Density Parity-Check" codes afin d'obtenir de meilleurs performances. Voir [Mac03]*

existe $\rho > 1$, satisfaisant :

$$\Pr\Big(\Big[\mathrm{A},\mathrm{B}\Big](b_0,b_1)(c) \neq (\zeta, b_c)\Big) \;\leq\; \rho^{\left(R - C_{\frac{\varphi^2}{\varepsilon}}\right)n}$$

Ce qui est satisfait pour

$$R < \; C_{\frac{\varphi^2}{\varepsilon}} = 1 - h\left(\frac{\varphi^2}{\varepsilon}\right) \tag{3.1}$$

$$\Rightarrow \quad k < \; \left(1 - h\left(\frac{\varphi^2}{\varepsilon}\right) + \delta\right) n. \tag{3.2}$$

Ce qui prouve que si les deux participants sont honnêtes, le protocole est statistiquement correct pour les paramètres de code ainsi définis.

□

Lemme 3.7.

Sous les hypothèses du lemme 3.6, si Alice triche le protocole 3.5 est statistiquement correct.

Preuve :

Nous constatons immédiatement que la condition 2.6 est satisfaite. En effet, *Alice* peut tricher de deux manières différentes.

Elle peut envoyer un faux syndrome qui mènera *Bob* à effectuer une erreur de correction mais il acceptera quand même le bit b_c. Par contre, il existe des entrées b'_0, b'_1 qui donneront un même résultat pour *Bob* avec une exécution honnête d'*Alice*.

D'autre part, *Alice* peut envoyer des *"mauvaises"*[6] paires $r_i\bar{r}_i$. À ce moment-là, peut être obtiendra-t-elle suffisamment d'information pour deviner c, mais *Bob* recevra quand même le bon bit b_c.

Ainsi quoi que fasse *Alice* le protocole reste correct sous les conditions du Lemme 3.6.

[6] Cette expression est empruntée à Crépeau et Kilian dans [CK88] qui parlent des bits reçus comme étant les "bons" bits, même s'ils contiennent des erreurs, et les "mauvais" bits font référence aux bits perdus donc issus des "mauvaises" pairs $r_i\bar{r}_i$ envoyées par le canal binaire symétrique.

□

Lemme 3.8.

Pour tout canal binaire symétrique d'erreur $\varphi < 1/2$, et pour $\mu > 1$, un code concaténé $\mathcal{C}[n,k,d]$ avec $k > n\varepsilon(h(\frac{\varphi^2}{\varepsilon}) - 1) + \mu$, le protocole 3.5 est statistiquement $\mathcal{B}$-privé.

Preuve :

Pour être statistiquement $\mathcal{B}$-privé, le protocole doit satisfaire la condition 2.8. C'est à dire que la quantité d'information que *Bob* doit obtenir sur $b_{\bar{c}}$, s'il a appris b_c, doit être exponentiellement décroissante en fonction d'un paramètre de sécurité.

Commençons par une remarque importante. Si *Bob* est malhonnête il essaiera d'obtenir les deux bits b_0 et b_1. Afin d'y arriver il tentera de corriger les erreurs contenus dans les deux chaînes r'_{I_c} et $r'_{I_{\bar{c}}}$. Ainsi il pourra maximiser la quantité d'information qu'il peut obtenir sur les deux bits en même temps. Car s'il réussit à diminuer l'incertitude qu'il a sur $\mathrm{R}_{\mathrm{I}_c}$ et $\mathrm{R}_{\mathrm{I}_{\bar{c}}}$ en même temps, le théorème de distillation de secrets ne sera plus applicable, et il pourra aisément déduire b_c et $b_{\bar{c}}$.

Il est évident que la meilleure stratégie à adopter est de distribuer les bits reçus, et par conséquence les erreurs, équitablement entre les deux chaînes r'_{I_c} et $r'_{I_{\bar{c}}}$. *Bob* sait que sur $2n$ bits envoyés il y aura en moyenne $2\varepsilon n$ reçus, si *Alice* ne triche pas. *Bob* choisira I_c et $I_{\bar{c}}$ de façon à avoir en moyenne εn bits reçus dans chacun. Parmi ces bits reçus ou "bons" bits il y a des erreurs, en moyenne on peut s'attendre à $\varphi^2 n$ erreurs.

Dans la preuve ci-après il sera montré que même en utilisant cette stratégie, le code $\mathcal{C}$ peut être choisi, et doit l'être, de façon à ce que *Bob* ne puisse pas retrouver les deux bits b_0 et b_1.

Évaluons $\forall c \in \mathbb{F}_2$ et $\forall \widetilde{\mathrm{B}}$,

$$\mathbf{I}\left(B_{\overline{C}}; \left[\mathrm{A}, \widetilde{\mathrm{B}}\right]^*_{\mathcal{B}}(B_0, B_1)(C) \middle| C = c, \widetilde{B}_{\widetilde{C}}\right) .$$

Par définition on a :

$$\begin{aligned}
&\mathbf{I}\left(B_{\overline{C}}; \left[\mathrm{A}, \widetilde{\mathrm{B}}\right]^*_{\mathcal{B}}(B_0, B_1)(C) \middle| C = c, \widetilde{B}_{\widetilde{C}}\right) \\
&\qquad = H(B_{\overline{C}}) \\
&\qquad - H\left(B_{\overline{C}} \middle| \left[\mathrm{A}, \widetilde{\mathrm{B}}\right]^*_{\mathcal{B}}(B_0, B_1)(c), \widetilde{B}_{\widetilde{C}}\right) \\
&\qquad = 1 - H\left(B_{\overline{C}} \middle| M, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = s_{\bar{c}}, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}\right)
\end{aligned}$$

La dernière égalité est vérifiée car les bits choisis par *Alice* (b_0, b_1) sont indépendants l'un de l'autre.

Par le théorème 3.2, si $H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = s_{\bar{c}}, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}) \geq r$, on obtient :

$$H\left(B_{\overline{C}} \middle| M, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = s_{\bar{c}}, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}\right) \geq 1 - \frac{2^{(1-r)}}{\ln 2} .$$

Or par le Théorème 3.4,

$$\begin{aligned}
H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}})) &\geq H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}) - H(\mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}})) \\
&\geq H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}) - \log(2^{n-k}) \\
&\geq H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_{\varepsilon}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}) - (n-k)
\end{aligned}$$

De plus,

$$
\begin{aligned}
H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}) &= H_2(R_{I_\zeta} | \, \widehat{\mathrm{OT}}_\varepsilon(r'_{I_\zeta}) = r'_{I_\zeta}) \\
&+ H_2(R_{I_{(0,1)}} | \, \widehat{\mathrm{OT}}_\varepsilon(r'_{I_{(0,1)}}) = r'_{I_{(0,1)}}) \qquad (3.3) \\
&= (1-\varepsilon) n H(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = \zeta) \\
&+ \varepsilon n H(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = 0) + o(n) \qquad (3.4) \\
&= n\Big((1-\varepsilon) + \varepsilon h\Big(\frac{\varphi^2}{\varepsilon}\Big)\Big) + o(n)\ . \qquad (3.5)
\end{aligned}
$$

Où $I_\zeta = \{i \in I_{\bar{c}} | r'_i = \zeta\}$ et $I_{(0,1)} = \{i \in I_{\bar{c}} | r'_i \neq \zeta\}$. L'égalité 3.3 est vérifiée car les v.a. $(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = r'_i)$ sont indépendantes.

Comme on considère toujours les séquences typiques et qu'on s'attend à perdre $(1-\varepsilon)n$ bits et à recevoir en moyenne εn, et les v.a. $(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = 0)$ et $(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = 1)$ sont identiquement distribuées, l'égalité 3.4 est vérifiée.

De plus, $(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = \zeta)$ est une v.a. uniformément distribuée donc $H(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = \zeta) = h(\frac{1}{2}) = 1$.

Enfin, pour $\varphi < 1/2$, $H(R_i | \, \widehat{\mathrm{OT}}_\varepsilon(r'_i) = 0) = h\Big(\frac{\varphi^2}{\varepsilon}\Big)$.

Ainsi,

$$
H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}})) \geq n\Big((1-\varepsilon) + \varepsilon h\Big(\frac{\varphi^2}{\varepsilon}\Big)\Big) - (n-k) + o(n)
$$

Étant donné que Syn est une fonction déterministe de $\mathrm{R}_{\mathrm{I}_{\bar{c}}}$ et on considère les séquences typiques de $\mathrm{R}_{\mathrm{I}_{\bar{c}}}$, de $(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}})$, et donc de $\mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}})$ il est évident que :

$$
\begin{aligned}
&H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}})) \\
&\qquad = \sum_{s_{\bar{c}} \in \mathfrak{S}} P(s_{\bar{c}}) H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = s_{\bar{c}}) \\
&\qquad = H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \,|\, \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = r'_{I_{\bar{c}}}, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = s_{\bar{c}})
\end{aligned}
$$

Donc,

$$H_2(\mathrm{R}_{\mathrm{I}_{\bar{c}}} \mid \widehat{\mathrm{OT}}_\varepsilon(\mathrm{R}_{\mathrm{I}_c}) = r'_{I_{\bar{c}}}, \mathrm{Syn}(\mathrm{R}_{\mathrm{I}_{\bar{c}}}) = s_{\bar{c}}) \geq n\Big((1-\varepsilon) + \varepsilon h\Big(\frac{\varphi^2}{\varepsilon}\Big)\Big) - (n-k) + o(n)$$

On en conclut que pour un code $\mathcal{C}[n,k,d]$ avec $k > n\varepsilon(h(\frac{\varphi^2}{\varepsilon}) - 1) + \mu$ où $\mu > 1$ est un paramètre de sécurité, *Bob* aura une incertitude sur $b_{\bar{c}}$ proche de 1.

□

Jusqu'à présent, nous avons montré que le protocole 3.5, de réduction $\binom{2}{1}$-$\widehat{\mathrm{OT}} \hookleftarrow \widehat{\mathrm{OT}}_\varepsilon$, est statistiquement correct et $\mathcal{B}$-privé. Par contre, il n'est pas $\mathcal{A}$-privé. Nous avons mentionné précédemment qu'*Alice* pouvait envoyer des mauvaises paires $r_i\bar{r}_i$. Dans ce cas, elle sait qu'avec probabilité $\varepsilon > 1/2$, *Bob* reçoit $\widehat{\mathrm{OT}}_\varepsilon(r_i) = \zeta$. Elle peut ainsi avoir une bonne idée de la position des bits perdus et donc de celles des bits reçus. En connaissant le contenus des ensembles I_c et $I_{\bar{c}}$, elle peut savoir lequel des deux bits b_c ou $b_{\bar{c}}$ *Bob* a choisi.

Dans le protocole suivant, nous verrons comment par un test statistique *Bob* peut s'assurer qu'*Alice* ne triche pas.

3.2.4 $\binom{2}{1}$-OT $\hookleftarrow$ $\binom{2}{1}$-$\widehat{\text{OT}}$: un protocole statistiquement $\mathcal{A}$-Privé

Protocole 3.6. [$\binom{2}{1}$-OT $\hookleftarrow$ $\binom{2}{1}$-$\widehat{\text{OT}}$]

1. $\mathcal{A}$ choisit n bits aléatoires $b_{1,0}, b_{2,0}, ..., b_{n,0}$ et calcule pour tout l, $1 \leq l \leq n$ $b_{l,1} = b_0 \oplus b_1 \oplus b_{l,0}$.
2. $\mathcal{B}$ choisit n bits aléatoires $c_1, c_2, ..., c_n$.
3. Pour tout entier l, $0 < l \leq n$ faire ce qui suit :

   $\mathcal{A}$ et $\mathcal{B}$ exécutent le protocole $\binom{2}{1}$-$\widehat{\text{OT}}(b_{l,0}, b_{l,1})(c_l)$ et $\mathcal{B}$ obtient b'_l.
4. Si $\left(\left|\{r'_{l,i_l} \mid 0 < l \leq n \text{ et } 0 < i \leq n\}\right| < 2\varepsilon n^2 - \frac{(2\varepsilon-1)}{2}n\right)$ alors $\mathcal{B}$ stoppe le protocole.

   Sinon $\mathcal{B}$ calcule et envoie $c' \leftarrow c \oplus \left(\bigoplus_{i=1}^{n} c_i\right)$.
5. $\mathcal{A}$ calcule et envoie à $\mathcal{B}$ les deux bits suivants : $\hat{b}_0 \leftarrow b_0 \oplus \left(\bigoplus_{i=1}^{n} b_{i,c'}\right)$ et $\hat{b}_1 \leftarrow b_1 \oplus \left(\bigoplus_{i=1}^{n} b_{i,\bar{c}'}\right)$.
6. $\mathcal{B}$ calcule le résultat final : $\hat{b}_c \leftarrow b_0 \oplus \left(\bigoplus_{i=1}^{n} b'_i\right)$.

Analyse du protocole :

L'objectif de ce protocole est d'empêcher *Alice* de tricher en envoyant de mauvaises paires. L'idée du protocole 3.6 est d'exécuter la primitive $\binom{2}{1}$-$\widehat{\text{OT}}$ un grand nombre de fois afin que *Bob* puisse utiliser le nombre de mots perdus espérés pour tester l'honnêteté d'*Alice* .

Dans [Cré97], la primitive $\binom{2}{1}$-$\widehat{\text{OT}}$ est utilisée n^2 fois mais nous pouvons remarquer qu'il est suffisant de l'utiliser n fois, si n est assez grand.

Lemme 3.9.

Le protocole 3.6 est statistiquement $\mathcal{A}$-privé.

Preuve :

En effet, considérons la v.a. $x_{l,i} = \begin{cases} 1 & \text{Si } r'_{l,i_l} \neq \zeta \\ 0 & \text{Si } r'_{l,i_l} = \zeta \end{cases}$

Si Alice ne triche pas, alors

$$E\left(\sum_{l=1}^{n}\sum_{i=1}^{2n} x_{l,i_l}\right) = 2\varepsilon n^2$$

Si Alice triche,

$$E\left(\sum_{l=1}^{n}\sum_{i=1}^{2n} x_{l,i_l}\right) \leq 2\varepsilon n^2 - (2\varepsilon - 1)n$$

Car elle doit tricher à chaque exécution de $\binom{2}{1}$-$\widehat{\text{OT}}$ pour obtenir tous les c_l et enfin avoir c'.

Ainsi par le théorème de Bernstein, on peut affirmer qu'il existe $\delta < 1$ tel que,

Si *Alice* n'envoie aucune mauvaise paire :

$$\Pr\left[\sum_{l=1}^{n}\sum_{i=1}^{2n} x_{l,i_l} < 2\varepsilon n^2 - \frac{(2\varepsilon - 1)}{2}n\right] < 2^{(-n\delta^2)}$$

Et si *Alice* envoie au moins une mauvaise paire par exécution de $\binom{2}{1}$-$\widehat{\text{OT}}$:

$$\Pr\left[\sum_{l=1}^{n}\sum_{i=1}^{2n} x_{l,i_l} > 2\varepsilon n^2 - \frac{(2\varepsilon - 1)}{2}n\right] < 2^{(-n\delta^2)}$$

Avec cette condition, dès que *Bob* a un doute sur l'honnêteté d'*Alice* il peut arrêter le protocole en cours d'exécution et ainsi *Alice* n'obtiendra pas l'information qu'elle recherche. □

En récapitulant les conditions des lemmes 3.8, 3.6 et 3.9, il existe un code concaténé $\mathcal{C}[n, k, d]$ qui satisfait les conditions des trois lemmes, et donc on peut affirmer le théorème suivant :

Théorème 3.5.

Le protocole 3.6 permet de réduire un transfert inconscient en un canal binaire symétrique BSC_φ *et ce pour toute probabilité d'erreur* $\varphi < 1/2$. *Cette réduction est statistiquement correct et privé.*

Note 3.2. Nous devons mentionner que dans son article qui introduit cette réduction Claude Crépeau ajoute une étape au protocole 3.6 pour vérifier que le syndrome envoyé par *Alice* est le bon. Nous estimons que cette étape n'est pas nécessaire étant donnée la définition d'un protocole correct pour le transfert inconscient. En effet, l'envoie d'un mauvais syndrome par *Alice* serait équivalent à une exécution honnête du protocole avec des entrées différentes. Le fait qu'*Alice* triche à l'envoi du syndrome ne peut lui donner aucun avantage sur *Bob*.

CHAPITRE 4

RÉDUCTION DU TRANSFERT INCONSCIENT EN UN CANAL BRUYANT INJUSTE

Un canal binaire symétrique est un des modèles les plus simples de canaux de communication. Cependant, ce modèle de communication n'est pas très représentatif de la réalité. En effet, dans ce modèle l'erreur de transmission engendrée par ce canal est stable et toujours connue des deux participants. Nous avons vu dans le chapitre précédent qu'*Alice* et *Bob* se servent justement de cette erreur pour réaliser un transfert inconscient inconditionnellement sécuritaire. Le protocole 3.6 de réduction de $\binom{2}{1}$-OT à BSC_φ est inconditionnellement sécuritaire si les deux participants connaissent l'erreur φ du canal et choisissent les bons paramètres de sécurité en fonction de cette erreur.

En fait, souvent *Alice* et *Bob* n'ont pas réellement accès à un tel canal. Le canal qui les relie est souvent moins bruyant pour l'un d'entre eux. Un exemple simple est le cas des fibres optiques. *Alice* et *Bob* communiquent à l'aide de systèmes reliés par une chaîne de fibres optiques, composée de plusieurs fils séparés par des répéteurs. Il est très plausible qu'un des deux puisse avoir accès aux données lorsqu'elles passent par un des répéteurs. Ce participant, malhonnête, pourrait à ce moment là tirer parti de cette information supplémentaire pendant ou après l'exécution du protocole. Les protocoles présentés dans le chapitre précédent ne tiennent pas compte de ce cas de figure.

Jusqu'à présent notre objectif a été de construire un transfert inconscient à partir de primitives plus simples. Pour être plus précis, nous devrions dire que notre objectif est de réduire le transfert inconscient à des hypothèses cryptographiques

élémentaires comme l'existence d'un canal binaire symétrique ou encore à des hypothèses encore plus génériques comme l'existence d'un *canal bruyant injuste*[1].

Dans ce chapitre nous présenterons le travail de Damgård, Kilian et Salvail présenté dans l'article *"On the (Im)possibility of Basing Oblivious Transfer and Bit Commitment on Weakened Security Assumptions"* [DKS99]. Les auteurs présentent une nouvelle primitive le *canal bruyant injuste*, et construisent un protocole de réduction du transfert inconscient à ce nouveau type de canal. Nous présenterons ici les résultats essentiels de cet article ainsi que les différents protocoles et primitives intermédiaires qui permettent d'effectuer cette réduction. Toutefois nous ne présenterons pas les preuves de sécurité de ces protocoles[2].

4.1 Primitives faibles

4.1.1 Canal bruyant injuste

La primitive que présentent les auteurs est un canal avec un taux de bruit (ou d'erreur) p variable. Les deux participants, *Alice* et *Bob*, ont accès à un canal qu'ils ne connaissent pas parfaitement ou au moins un des deux ne connaît pas l'erreur exact de ce canal. Par contre, chacun d'eux connaît l'intervalle $[\gamma, \delta]$ dans lequel se situe cette erreur p. De plus, un participant malhonnête pourrait choisir le taux d'erreur. Il choisira naturellement la plus basse erreur afin de maximiser la quantité d'information qu'il obtient à partir de la communication. De façon un peu plus formelle, un *canal bruyant injuste* est défini comme suit :

Définition 4.1 (Canal Bruyant Injuste).
Soit $\gamma, \delta < 1/2$, un *canal bruyant injuste* (γ,δ)-UNC est un canal binaire symétrique d'erreur $p \in [\gamma, \delta]$ tel que p est inconnu d'un honnête participant et peut

[1] Unfair Noisy Channel.
[2] Pour les preuves de sécurité des protocoles se référer à l'article [DKS99].

être déterminé par un participant malhonnête.

Un autre type de canaux bruyants légèrement différent est présenté dans [DKS99], le *canal injuste passif.*

Définition 4.2 (Canal Injuste Passif).
Soit $\gamma < \delta < 1/2$, un canal injuste passif est un canal binaire symétrique d'erreur δ, et lorsque *Alice* ou *Bob* triche de manière passive alors l'erreur diminue à γ. Nous y référerons par (γ,δ)-PassiveUNC.

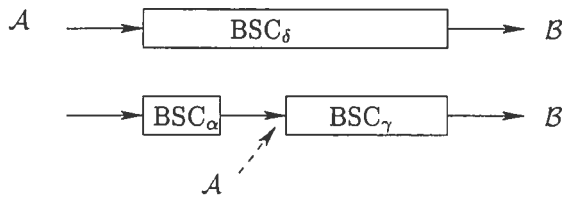

FIG. 4.1 – Modèle de canal injuste passif.

4.1.2 Transfert inconscient faible

Le *transfert inconscient faible*[3] est une version généralisée de transfert inconscient où un des deux participants ou les deux obtiennent plus d'information que prévu. Dans [Cré88], Crépeau pose le problème de la réduction d'un transfert inconscient $\binom{2}{1}$-OT en un transfert inconscient où *Alice* apprendrait avec une certaine probabilité q le choix de *Bob* et où ce dernier obtiendrait de l'information sur le bit $b_{\bar{c}}$ qu'il n'a pas choisi. Damgård, Kilian et Salvail généralisent cette notion en définissant (p,q,ϵ)-WOT .

[3] *Weak Oblivious Transfer (WOT).*

Définition 4.3.

Un *transfert inconscient faible* ou (p,q,ϵ)-WOT est un protocole de transfert inconscient $\binom{2}{1}$-OT tel que :

1. Un expéditeur, *Alice*, malhonnête apprend avec une probabilité de (au plus) q, le choix du receveur *Bob*.
2. Si *Bob* est malhonnête il apprend avec une probabilité de p les deux bits b_0, b_1 envoyés par *Alice*.
3. Si *Bob* est honnête avec probabilité ϵ, il reçoit $\bar{b}_c$ au lieu de recevoir b_c.

Au chapitre précédent, le protocole 3.4 est un exemple de reduction $\mathrm{OT}_\varepsilon \hookleftarrow \mathrm{BSC}_\varphi$ qui implante un transfert inconscient faible $(0,\varepsilon,\varphi^2)$-WOT .

4.2 Des réductions impossibles

Les premiers résultats importants de l'article [DKS99] sont les deux théorèmes énonçant l'impossibilité de réduire le transfert inconscient à un transfert inconscient faible ou à un canal bruyant injuste pour certaines valeurs de (p, q, ϵ) ou de (γ, δ), respectivement.

4.2.1 Impossibilité de la réduction $\binom{2}{1}$-OT $\hookleftarrow$ (p, q, ϵ)-WOT

Théorème 4.1.

Pour $p + q + 2\epsilon \geq 1$, il n'existe pas de réduction de $\binom{2}{1}$-OT à (p,q,ϵ)-WOT inconditionnellement sécuritaire.

La preuve de la véracité de ce théorème est une preuve par contradiction. Les auteurs montrent que pour des paramètres p, q et ϵ ayant la propriété $p+q+2\epsilon \geq 1$, il est possible de simuler un transfert inconscient faible à partir d'un canal non-bruyant.

Donc, si on fait l'hypothèse qu'il existe un protocole de réduction de $\binom{2}{1}$-OT à (p,q,ϵ)-WOT alors on aurait montré que $\binom{2}{1}$-OT se réduit à un canal sans erreur

pour des valeurs de p, q et ϵ satisfaisant $p + q + 2\epsilon \geq 1$. Ce résultat contredit le fait connu qu'il n'existe pas de réduction de transfert inconscient à un canal non-bruyant. Le protocole de simulation de (p,q,ϵ)-WOT est présenté en annexe à la section I.1.

4.2.2 Impossibilité de la réduction $\binom{2}{1}$-OT $\hookleftarrow (\gamma, \delta)$-UNC

Théorème 4.2.

Pour $\delta \geq 2\gamma(1-\gamma)$, il n'existe pas de réduction de $\binom{2}{1}$-OT à (γ,δ)-UNC.

La preuve de ce théorème est semblable à celle du théorème précédent. Pour $\delta \geq 2\gamma(1-\gamma)$, on peut simuler un canal bruyant injuste à partir d'un canal sans erreur. Le protocole de simulation de (γ,δ)-UNC est présenté en annexe à la section I.2. S'il est possible, pour ces valeurs de γ et δ de réduire un transfert inconscient à un canal bruyant injuste alors on aurait réduit $\binom{2}{1}$-OT à un canal sans erreur ce qui est impossible.

4.3 La réduction $\binom{2}{1}$-OT $\hookleftarrow$ (p, q)-WOT

La primitive (p,q)-WOT est un transfert inconscient faible sans erreur. C'est à dire que si *Bob* reçoit un bit à travers cette primitive il sait qu'il est bon. Il est intéressant de voir comment il est possible de réduire $\binom{2}{1}$-OT à cette primitive. Afin d'y parvenir, il suffirait de réaliser un protocole qui réduirait les probabilités qu'ont *Alice* et de *Bob* d'obtenir c et $b_{\bar{c}}$ respectivement. Le lemme suivant affirme qu'il existe un tel protocole pour certaines valeurs de p et q.

Lemme 4.1.

Il existe une réduction de $\binom{2}{1}$-OT à (p,q)-WOT qui est sécuritaire pour tout p et q tels que $p + q < 1$.

Le protocole de réduction de $\binom{2}{1}$-OT à (p,q)-WOT est une composition de deux protocoles simples et connus. Le premier protocole 4.1 réduit la probabilité qu'*Alice* puisse connaître c.

Protocole 4.1. [S-Red(n, *(p,q)-WOT*)]

1. Pour $1 \leq i \leq n$
   - $\mathcal{A}$ choisit aléatoirement et uniformément (b_{0_i}, b_{1_i}) tel que $b_0 = \oplus_{i=1}^{n} b_{0_i}$ et $b_{1_i} = b_{0_i} \oplus b_0 \oplus b_1$.
   - $\mathcal{B}$ choisit aléatoirement et uniformément c_i tel que $c = \oplus_{i=1}^{n} c_i$.
   - $\mathcal{A}$ et $\mathcal{B}$ exécute (p,q)-WOT$(b_{0_i}, b_{1_i})(c_i)$.
2. $\mathcal{B}$ calcule $b_c = \oplus_{i=1}^{n} b_{c_i}$.

Le protocole 4.1 implante un $(p^n, 1-(1-q)^n)$-WOT sécuritaire contre des adversaires actifs si le protocole (p,q)-WOT donné l'est. En premier lieu, remarquons que ce protocole est correct. Si *Alice* et *Bob* sont honnêtes alors ils reçoivent respectivement les bonnes sorties, c'est à dire *Alice* ne reçoit rien et *Bob* reçoit b_c. De plus si *Alice* tente de tricher durant l'exécution du protocole, elle enverra de "mauvais" bits b'_{0_i}, b'_{1_i} et cela est équivalent au choix d'une autre paire b_0, b_1.
D'autre part, afin de connaître c, *Alice* doit apprendre chaque c_i pour $1 \leq i \leq n$. Par la sécurité de (p,q)-WOT, *Alice* a une probabilité p^n d'obtenir c à la fin du protocole 4.1. *Bob* quant à lui peut obtenir les deux bits b_0, b_1 en apprenant seulement une paire b_{0_i}, b_{1_i} ceci se produit avec une probabilité de $(1-(1-q)^n)$.

Le second protocole, 4.2, réduit la probabilité que *Bob* malhonnête obtienne $b_{\bar{c}}$.

Protocole 4.2. [R-Red(n, *(p,q)-WOT*)]

1. Pour $1 \leq i \leq n$, $b_0 = \oplus_{i=1}^{n} b_{0_i}$, $b_1 = \oplus_{i=1}^{n} b_{1_i}$ et $c_i = c$, $\mathcal{A}$ et $\mathcal{B}$ exécutent (p,q)-WOT$(b_{0_i}, b_{1_i})(c_i)$.
2. $\mathcal{B}$ calcule $b_c = \oplus_{i=1}^{n} b_{c_i}$.

Pour n donné et une primitive (p,q)-WOT, le protocole 4.2 implante un transfert inconscient faible $(1-(1-p)^n, q^n)$-WOT sécuritaire contre des adversaires actifs si (p,q)-WOT l'est.

Ce protocole est correct. Si les deux participants sont honnêtes, *Alice* ne reçoit rien et *Bob* reçoit b_c et si *Alice* triche, en exécutant (p,q)-WOT avec de mauvaises paires (b'_{0_i}, b'_{1_i}), différentes de (b_{0_i}, b_{1_i}), l'exécution du protocole est statistiquement équivalente à une exécution honnête avec une autre paire b_0, b_1.

D'autre part, afin d'apprendre c, *Alice* a besoin d'apprendre simplement un des c_i, ce qui est possible avec probabilité $1-(1-p)^n$. *Bob* a besoin, par contre, d'apprendre chaque $b_{\bar{c}_i}$ afin d'obtenir $b_{\bar{c}}$ ce qu'il peut obtenir avec probabilité q^n.

Afin de réaliser la réduction $\binom{2}{1}$-OT à (p,q)-WOT, il suffit d'utiliser les deux protocoles précédents, S-Red(n, *(p,q)-WOT*) et R-Red(n, *(p,q)-WOT*). L'idée de la réduction proposée par les auteurs de [DKS99] est d'utiliser plusieurs fois les protocoles S-Red(n, R-Red(n, *(p,q)-WOT*)) ou R-Red(n, S-Red(n, *(p,q)-WOT*)) jusqu'à ce que les probabilités que *Bob* et *Alice* obtiennent $b_{\bar{c}}$ et c, respectivement soient suffisamment petites. Ensuite, ils démontrent qu'il est possible d'effectuer cette réduction en un nombre d'appels à (p,q)-WOT polynômial en n.

Les protocoles présentés dans cette section et le résultat du théorème 4.1 impliquent directement le théorème suivant.

Théorème 4.3 ($\binom{2}{1}$-OT $\hookleftarrow$ (p, q)-WOT).
$\binom{2}{1}$-OT *peut se réduire à (p,q)-WOT si et seulement si* $p + q < 1$.

4.4 La réduction $\binom{2}{1}$-OT $\hookleftarrow$ (p, q, ϵ)-WOT

La réduction du transfert inconscient $\binom{2}{1}$-OT à la primitive de transfert inconscient faible (p,q,ϵ)-WOT s'effectue à l'aide de trois protocoles. L'idée de combiner les deux protocoles R-Red et S-Red est réutilisée pour cette nouvelle réduction et un dernier protocole la complète qui sert à diminuer la probabilité d'erreur ϵ.

Le protocole 4.3 réalise une réduction de (p',q',ϵ')-WOT à (p,q,ϵ)-WOT pour $\epsilon' < \epsilon$. Cependant, cette réduction ne tient pas compte des attaques d'adversaires actifs. Elle est sécuritaire pour des adversaires passifs seulement.

Protocole 4.3. [ErRed(l,(p,q,ϵ)-WOT)]

1. $\mathcal{A}$ et $\mathcal{B}$ choisissent respectivement q_0, q_1 et s aléatoirement dans $\mathbb{F}_2$,
2. $\mathcal{A}$ envoie l fois les bits (q_0, q_1) à travers (p,q,ϵ)-WOT et $\mathcal{B}$ choisit le bit q_s l fois.
3. Si $\mathcal{B}$ ne reçoit pas l fois le même bit $\hat{q}_s$ alors ils répètent le protocole à partir de l'étape 1.
4. $\mathcal{B}$ annonce $y = 0$ si $s = c$ et $y = 1$ sinon.
5. $\mathcal{A}$ annonce r_0 et r_1 tel que $b_y = r_0 \oplus q_0$ et $b_{\bar{y}} = r_1 \oplus q_1$ permettant ainsi à $\mathcal{B}$ de calculer $b_c = \hat{q}_s \oplus r_s$.

La composition des trois protocoles S-Red, R-Red et ErRed permet de réduire $\binom{2}{1}$-OT à (p,q,ϵ)-WOT . Le protocole finale proposé par Damgård, Kilian et Salvail

est le suivant et son analyse implique le théorème 4.4.

Protocole 4.4. [SR_ϵ-Red]

$\mathrm{ErRed}(l_2, \mathrm{R\text{-}Red}(k', \mathrm{ErRed}(l_1, \mathrm{S\text{-}Red}(k, \mathrm{ErRed}(l_0, (p,q,\epsilon)\text{-WOT }))))).$

Théorème 4.4.

Pour $p + q + 2\epsilon < 0.45$, *le protocole 4.5,* SR_ϵ-Red, *réduit* $\binom{2}{1}$-OT *à* (p,q,ϵ)-*WOT* .

4.5 La réduction $\binom{2}{1}$-OT $\hookleftarrow$ (γ, δ)-UNC

Dans cette dernière partie, nous présentons le protocole qui permet de réduire un transfert inconscient faible en un canal bruyant injuste. Afin de réduire un transfert inconscient en un canal injuste, il suffirait donc d'appliquer cette nouvelle réduction pour obtenir un (p,q,ϵ)-WOT et ensuite de se servir du protocole SR_ϵ-Red pour réaliser la réduction $\binom{2}{1}$-OT $\hookleftarrow$ (γ, δ)-UNC . En réalité, la réduction est moins immédiate que cela. Au lieu d'être réduit à (γ,δ)-UNC, (p,q,ϵ)-WOT est d'abord réduit à un canal injuste passif (γ,δ)-PassiveUNC.

Protocole 4.5. [(p, q, ϵ)-WOT $\hookleftarrow$ (γ, δ)-PassiveUNC]

1. $\mathcal{A}$ choisit aléatoirement $x, y \in \mathbb{F}_2$,
2. $\mathcal{A}$ envoie (xx, yy) par (γ,δ)-PassiveUNC et $\mathcal{B}$ reçoit $(\tilde{x}\tilde{x}', \tilde{y}\tilde{y}')$,
3. Si $\mathcal{B}$ reçoit $(\tilde{x} \oplus \tilde{x}', \tilde{y} \oplus \tilde{y}') \notin \{(0,1),(1,0)\}$ alors ils reprennent le protocole à l'étape 1.
4. $\mathcal{B}$ annonce w tel que
   - $w = 0$ si $((\tilde{x} \oplus \tilde{x}' = 0) \wedge (c = 0)) \vee ((\tilde{y} \oplus \tilde{y}' = 0) \wedge (c = 1))$,
   - $w = 1$ si $((\tilde{x} \oplus \tilde{x}' = 0) \wedge (c = 1)) \vee ((\tilde{y} \oplus \tilde{y}' = 0) \wedge (c = 0))$.
5. $\mathcal{A}$ annonce
   - $(a, b) = (x \oplus b_0, y \oplus b_1)$ si $w = 0$,
   - $(a, b) = (y \oplus b_0, x \oplus b_1)$ si $w = 1$.
6. $\mathcal{B}$ calcule
   - $b_0 = (\tilde{x} \oplus a)$ si $c = 0$ et $w = 0$,
   - $b_0 = (\tilde{y} \oplus a)$ si $c = 0$ et $w = 1$,
   - $b_1 = (\tilde{y} \oplus b)$ si $c = 1$ et $w = 0$,
   - $b_1 = (\tilde{y} \oplus b)$ si $c = 1$ et $w = 1$.

Après l'analyse de ce protocole, les auteurs aboutissent au théorème suivant qui délimite une région des valeurs de γ, δ qui permette la réduction du transfert inconscient en un canal injuste passif.

Théorème 4.5.

$\binom{2}{1}$-OT *peut être réduit à* (γ,δ)-PassiveUNC *si* $\alpha^3\beta^3(1 - \mu(1-\alpha)) > \frac{0.775+\epsilon(\delta)}{1-\epsilon(\delta)}$, *où* $\epsilon(\delta) = \frac{\delta^2}{\delta^2+(1-\delta)^2}$, $\alpha = \frac{1-\delta-\gamma}{1-2\gamma}$, $\beta = \frac{1-\gamma}{1-\delta}$, *et* $\mu = \frac{1-\gamma}{\delta}$.

D'après le théorème 4.5, pour un intervalle de valeurs de γ, δ, le protocole 4.5 de réduction du transfert inconscient faible à un canal injuste passif est inconditionnellement sécuritaire contre des adversaires passifs. Par la suite, le protocole , SR_ϵ-Red, est utilisé afin de réaliser la réduction du transfert inconscient au canal injuste passif. À priori cette réduction est sécuritaire uniquement contre des adversaires passifs.

Damgård, Kilian et Salvail, démontrent que le protocole peut être utilisé pour réduire un transfert inconscient faible à un canal bruyant injuste (γ,δ)-UNC et ce pour les mêmes valeurs de γ, δ.[4]

La dernière étape de la réduction du transfert inconscient en un canal bruyant injuste consiste à rendre le protocole sécuritaire contre tout adversaire actif. Afin de réaliser cet objectif les auteurs se servent d'un autre résultat important de leur article. En effet, ils démontrent qu'à partir d'un canal bruyant injuste non-trivial,[5] il est possible de construire une mise en gage inconditionnellement sécuritaire.
La mise en gage est une primitive cryptographique qui permet à *Alice* de mettre en gage un bit b en envoyant, Commit(b) à *Bob*. *Bob* n'apprend rien à propos de b en voyant Commit(b) et si *Alice* découvre le bit en appliquant une fonction d'ouverture Open(Commit(b)), *Bob* est sur que b = Open(Commit(b)).
À l'aide de ce résultat, *Alice* et *Bob* peuvent se prouver mutuellement à partir d'une preuve à divulgation nulle qu'ils ont effectivement exécuter le protocole honnêtement.

[4] Voir Lemme 12 et Lemme 13 de [DKS99]

[5] Un canal bruyant injuste non-trivial est un canal qui ne peut pas être simuler à partir d'un canal sans erreur.

En conclusion, rappelons que deux théorème ont été énoncé dans ce chapitre. Le premier est celui qui délimite une région de l'espace $[0,1] \times [0,1]$ telle que pour (γ, δ) dans cette région, le transfert inconscient ne se réduit pas à un canal bruyant injuste. Le second théorème montre que pour une autre région distincte de la première il existe un protocole de réduction. Cependant, ces deux régions ne couvrent pas tout l'espace des valeurs possibles de γ, δ. Il reste une zone *"grise"* dans laquelle il n'a pas encore été démontré s'il existait ou non de protocole de réduction du transfert inconscient à un canal bruyant injuste. Dans cette dernière partie nous avons simplement voulu donner une idée des preuves présentées par les auteurs de [DKS99] sans vouloir aller dans les détails.

Nous devons néanmoins mentionner que des travaux ont été effectués par Damgård, Fehr, Morozov et Salvail [DFMS04] afin d'améliorer la borne du théorème 4.5, mais que leur résultat n'atteint toujours pas la borne d'impossibilité. Il reste toujours une zone grise, plus petite, dans laquelle la possibilité de réduction de $\binom{2}{1}$-OT à (γ,δ)-UNC est inconnue.[6]

[6]Pour plus de détails sur cette borne voir [DFMS04].

CONCLUSION

Pendant plus de 2000 ans la cryptologie a été réservée au secteur militaire et aux agences gouvernementales secrètes. Aujourd'hui encore lorsqu'on mentionne ce domaine d'étude nos interlocuteurs pensent immédiatement aux agents secrets, aux écoutes téléphoniques et aux films d'espionnage.
Pourtant, la cryptologie ou plus exactement la cryptographie est de plus en plus utilisée au quotidien. En effet, sans s'en rendre compte tout citoyen utilise des systèmes cryptographiques. Les transactions banquaires par guichets automatiques, le commerce électronique ou encore le vote en ligne sont des applications typiques de la cryptographie. La banalisation d'Internet et l'accès aux nouvelles technologies sont en grande partie responsables de la création de nouveaux besoins, civiles cette fois, en cryptographie.

Toutefois, l'expansion de l'utilisation des systèmes cryptographiques n'est pas dûe uniquement aux nouveaux besoins créés, elle est aussi le résultat de plusieurs années de recherche et de travail des cryptographes.

Dans l'histoire de la cryptologie, deux événements majeurs ont changé notre façon de la percevoir. L'introduction de la théorie de l'information par Shannon a été la première révolution. C'est à ce moment précis que la cryptologie est passée de l'état d'art à celui de science. Depuis 1948, elle possède un formalisme et un langage propre qui permettent enfin de parler d'une science avec essentiellement de nouveaux outils mathématiques permettant d'étudier la sécurité des protocoles cryptographiques.

La deuxième révolution a eu lieu dans les années 70 avec l'émergence des systèmes à clés publiques. Nous pensons bien évidemment à l'article de Diffie et Hell-

mann[DH76] et celui de Rivest, Shamir et Adleman [RSA78] qui révolutionnèrent cette science. Ces techniques permettent à tout individu de pouvoir recevoir de l'information chiffrée ou d'échanger une clé secrète au vu et au su de tout le monde et ce de façon sécuritaire. Elle a littéralement fait basculer la cryptologie du domaine militaire au domaine civile.

Aujourd'hui, depuis l'avènement de la cryptographie quantique, nous assistons aux débuts d'une nouvelle ère. Ce nouveau genre de cryptographie introduit plusieurs résultats fondamentaux qui sont optimistes et pessimistes à la fois. En effet, nous avons déjà mentionné le résultat de Shor [Sho97] qui a montré qu'avec un ordinateur quantique il serait possible de résoudre le problème de factorisation efficacement. Heureusement ce n'est pas le seul résultat de la cryptographie quantique. La réalisation de systèmes inconditionnellement sécuritaire, notamment celui proposé par Bennett et Brassard dans [BB84], est un résultat plutôt encourageant de la cryptographie quantique. Dans leur article, Bennett et Brassard proposent un système d'échange de clé secrète qui a été démontré plus tard comme étant inconditionnellement sécuritaire.

Toutefois, malgré les immenses progrès théoriques de l'informatique quantique, l'ordinateur quantique n'est pas encore très performant et reste encore à l'état expérimental. Dans le même temps, les techniques de cryptanalyse évoluent rapidement. Internet et la vulgarisation de l'information permettent aux personnes malhonnêtes d'être mieux armées contre les systèmes cryptographiques à sécurité calculatoire. Il est donc intéressant de penser à des systèmes inconditionnellement sécuritaire qui ne dépendraient pas de la technologie utilisée et qui ne poseraient aucune hypothèse sur la puissance de calcul d'un adversaire. La réduction inconditionnellement sécuritaire est une méthode qui permettrait d'atteindre cet objectif. Elle permet de construire des systèmes à partir de primitives simples ou d'hypo-

thèses pas trop restrictives. Les hypothèses peuvent être aussi simples que supposer l'existence d'une source aléatoire ou l'existence d'un canal de communication bruyant. Dans ce mémoire, un exemple de réduction inconditionnellement sécuritaire a été donné. En effet, nous avons vu qu'il est possible de réaliser un transfert inconscient inconditionnellement sécuritaire à l'aide de canaux binaires symétriques. Le résultat principal de ce mémoire est l'amélioration de l'analyse de la sécurité du protocole présenté par Crépeau dans [Cré97]. Nous avons montré qu'un transfert inconscient peut se réduire à un canal binaire symétrique quelconque. Par ailleurs, nous avons présenté les travaux de Dåmgard, Kilian et Salvail dans [DKS99] qui démontrent qu'il est possible de réduire le transfert inconscient à certains canaux bruyants injustes.

Cependant, ces travaux restent à compléter. Dans un article plus récent, Dåmgard, Fehr, Morozov, et Salvail [DFMS04] améliore la borne présentée dans [DKS99] sur la possibilité de réduire le transfert inconscient à un canal injuste, mais il reste une région floue dans laquelle il n'est pas encore connu s'il est possible ou non d'effectuer cette réduction.

En conclusion nous pouvons dire que le travail d'un cryptographe est passionnant. Il doit penser les nouveaux systèmes cryptographiques et prévenir tous les moyens de les briser. C'est un mathématicien, avant-gardiste qui doit être très rattaché à la réalité qui l'entoure. Il doit être à la fois cryptographe et cryptanalyste. Nous pouvons dire qu'il est ce mathématicien-tailleur fou décrit par Stanislaw Lem dans son roman ***Summa Technologiae*** qui tente de coudre tous les vêtements possibles en espérant réaliser le vêtement parfait. *"Mathematicians are mad tailors : they are making "all the possible clothes" hoping to make also something suitable for dressing..."*

BIBLIOGRAPHIE

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography : Public-key distribution and coin tossing. In *IEEE International Conference on Computers, Systems and Signal Processing*, pages 175 – 179, December 1984. Bangalore, India.

[BBCM94] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer. Generalized privacy amplification. In *Proc. IEEE International Symposium of Information Theory*, pages 350–350, 1994. Trondheim, Norway.

[BBR88] C. H. Bennett, G. Brassard, and J. Robert. Privacy amplification by public discussion. *SIAM J. Computing*, 17(2) :210–229, April 1988.

[BCS96] G. Brassard, C. Crépeau, and M. Sántha. Oblivious transfers and intersecting codes. *IEEE Transactions on Information Theory*, 42(6) :1769–1780, 1996.

[Bla87] R. E. Blahut. *Principles and practice of information theory.* Addison-Wesley Longman Publishing, 1987.

[CK88] C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *Annual IEEE Symposium on Foundations of Computer Science (FOCS '88)*, pages 42–52, 1988.

[CM94] C. Cachin and U. Maurer. Linking information reconciliation and privacy amplification. In *Advances in Cryptology — EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 266–274, May 1994.

[Cré88] C. Crépeau. Equivalence between two flavours of oblivious transfers. In Carl Pomerance, editor, *Proc. CRYPTO 87*, pages 350–354. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 293.

[Cré90] C. Crépeau. *Correct and Private Reductions among Oblivious Transfers*. PhD thesis, Massachusetts Institute of Technology, 1990.

[Cra99] R. Cramer. Introduction to secure computation. In *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, pages 16–62. Springer-Verlag, 1999.

[Cré97] C. Crépeau. Efficient cryptographic protocols based on noisy channels. In *Advances in Cryptology : CRYPTO '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 306–317. Springer-Verlag, 1997.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.

[CW79] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *JCSS*, 18 :143–154, 1979.

[DFMS04] I. Damgård, S. Fehr, K. Morozov, and L. Salvail. Unfair noisy channels and oblivious transfer. In *Lecture Notes in Computer Science*, page To appear. Theory of Cryptography Conference, Springer-Verlag, 2004. TCC '04.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6) :644–654, 1976.

[DKS99] I. Damgård, J. Kilian, and L. Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *Advances in Cryptology : CRYPTO '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 56–73. Springer-Verlag, 1999.

[EGL83] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Proc. CRYPTO 82*, pages 205–210, New York, 1983. Plenum Press.

[FMR96] M. J. Fischer, S. Micali, and C. Rackoff. A secure protocol for the oblivious transfer. *Journal of Cryptology*, 9(3) :191–195, 1996. Présenté à Eurocrypt '84, mais non publié dans les "proceedings".

[For66] D. G. Forney. *Concatenated Codes.* MIT Press, 1966.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Conference on Theory of Computing*, pages 218–229. ACM Press, 1987.

[GV88] O. Goldreich and R. Vainish. How to solve any protocol problem - an efficiency improvement. In Carl Pomerance, editor, *Proc. CRYPTO 87*, pages 73–86. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 293.

[Har28] R.V.L. Hartley. Transmission of information. *Bell System Technical Journal*, 7(4) :535–563, 1928.

[HILL91] J. Hastad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. Technical Report 9-068, ICSI, 1991.

[Kil88] J. Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 20–31, 1988.

[Mac03] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, October 2003.

[Mau99] U. Maurer. Information-theoretic cryptography. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 47–64. Springer-Verlag, August 1999.

[MS77] F. J. MacWilliams and N. J. A. Sloane, editors. *The Theory of Error-Correcting Codes.* North-Holland, 1977. CISM Courses and Lectures No. 279.

[MW99] U. Maurer and S. Wolf. The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. *SIAM Journal on Computing*, 28(5) :1689–1721, 1999.

[Nyq24] H. Nyquist. Certain factors affecting telegraph speed. *Bell System Technical Journal*, 3 :324–346, 1924.

[Rab81] M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[Riv90] R. L. Rivest. Cryptography. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science (Volume A : Algorithms and Complexity)*, chapter 13, pages 717–755. Elsevier and MIT Press, 1990.

[Rén70] A. Rényi. *Probability Theory.* North- Holland, Amsterdam, 1970.

[Rén84] A. Rényi. *A Diary On Information Theory.* 1984.

[RSA78] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) :120–126, 1978.

[Sal91] L. Salvail. Le problème de réconciliation en cryptographie. Master's thesis, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 1991.

[Sen04] N. Sendrier. Introduction à la théorie de l'information. Notes de cours, 2004.

[Sha48] C. E. Shannon. A mathematical theory of communication. 27(3) :379–423, July 1948. Continued 27(4) :623-656, October 1948.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4), 1949.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5) :1484–1509, 1997.

[Tap95] A. Tapp. Évaluations de fonctions sur données privées. Master's thesis, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 1995.

[Wie83] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1) :78–88, 1983. Manuscrit datant des années 70, non-publié avant 1983.

[Yao82] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '82)*, pages 160–164, 1982.

Annexe I

Protocoles de simulation

I.1 Protocole de simulation d'un transfert inconscient faible

Le protocole présenté dans cette annexe permet à *Alice* et *Bob* de simuler une primitive de transfert inconscient faible. Pour des valeurs de p, q et ϵ vérifiant $p + q + 2\epsilon = 1$, le protocole simule (p,q,ϵ)-WOT à partir d'un canal sans erreur.

Protocole I.1. [SimNoisyWOT$[p, q](b_0, b_1)(c)$]

1. Avec probabilité q $\mathcal{A}$ annonce (b_0, b_1), $\mathcal{B}$ calcule b_c et le protocole se termine. Sinon $\mathcal{A}$ annonce "passe".
2. Avec probabilité $\frac{p}{1-q}$, $\mathcal{B}$ envoie c à $\mathcal{A}$ qui lui renvoie b_c. Sinon $\mathcal{B}$ choisit aléatoirement b_c.

I.2 Protocole de simulation d'un canal bruyant injuste

De manière similaire, le protocole I.2 simule un canal bruyant injuste à partir d'un canal sans erreur et pour $\delta = 2\gamma(1 - \gamma)$.

Protocole I.2. [SimUNC[γ](b)]

1. $\mathcal{A}$ et $\mathcal{B}$ choisissent $b_{\mathcal{A}}$ et $b_{\mathcal{B}}$ tels que $\Pr(b_{\mathcal{A}} = 1) = \Pr(b_{\mathcal{B}} = 1) = \gamma$.
2. $\mathcal{A}$ envoie $b' = b \oplus b_{\mathcal{A}}$ à $\mathcal{B}$. $\mathcal{B}$ calcule une sortie $b^* = b' \oplus b_{\mathcal{B}}$, et $\mathcal{A}$ n'a aucune sortie.