Université de Montréal

Évaluation de fonctions sur données privées

par

Alain Tapp

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 1995

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

Évaluation de fonctions sur données privées

présenté par:

Alain Tapp

a été évalué par un jury composé des personnes suivantes:

____

(président-rapporteur)

Geňa Hahn

____

(directeur de recherche)

Claude Crépeau

____

(co-directeur)

Gilles Brassard

____

(membre du jury)

Pierre McKenzie

Mémoire accepté le: 12 décembre 1995

Sommaire

Ce mémoire traite de calculs multi-parties sur données privées (Private multi-party computation, PMPC). Soit N participants désirant évaluer $y = F(x_1, x_2, ..., x_N)$ avec x_i la donnée privée du i ième participant. On aimerait qu'un protocole PMPC réalise cette tâche de façon à ce que: (1) Il soit possible à N participants honnêtes d'apprendre y. (2) Aucune coalition de participants ne pourra apprendre plus au sujet des données des autres participants que ce qui peut être déduit à partir de leurs propres données secrètes et de la connaissance de F et de y. (3) Un participant sait quand le protocole a échoué et qu'il n'apprend pas y. (4) Si un participant apprend y, alors tous l'apprennent.

Dans ce mémoire, un protocole original et efficace permettant d'accomplir cette tâche cryptographique est présenté. Notons qu'aucune supposition ne sera faite quant au nombre de participants honnêtes. Le protocole est réalisable dans une multitude de modèles étant donné qu'il nécessite seulement l'existence d'un transfert inconscient, par extension une mise en gage, et d'un canal de diffusion. Il est le premier protocole de ce genre à être efficace et sûr.

Pour en arriver à ce résultat, une procédure intermédiaire appelée transfert inconscient certifié sera utilisée. Le transfert inconscient certifié est un transfert inconscient où tous les bits en jeu sont des mises en gage. La réalisation efficace de ce sous-protocole est probablement la contribution originale la plus importante de ce mémoire.

Mots clés: Cryptologie, circuit, mise en gage, transfert inconscient, théorie des codes, calcul multi-parties, sécurité.

Table des matières

Table des protocoles

Table des figures

À mon père.

Remerciements

Cette petite section de mon mémoire est le premier endroit où je peux exprimer officiellement mes plus sincères remerciements à tous ceux qui depuis le commencement de mes études ont eu confiance en moi.

Je voudrais remercier Pierre Vaillancourt qui m'a fait aimer les mathématiques, Gaétan Hains qui m'a encouragé lors de mon Bac et m'a ouvert l'esprit à l'informatique théorique, Pierre McKenzie, qui par son enseignement a contribué à faire de moi un étudiant accompli, Geňa Hahn qui m'a encouragé et surtout Claude Crépeau et Gilles Brassard qui m'ont dirigé et ont été des modèles pour moi.

À l'université, il n'y a pas que les professeurs qui ont influencé mon épanouissement intellectuel. Je voudrais remercier Catherine Dufourd qui m'a encouragé et fait profiter de son expérience, Hervé Caussinus qui m'a supporté moralement et intellectuellement de façon constante tout au long de ma maîtrise, Jeroen van de Graaf avec qui j'ai écrit un article pour CRYPTO et qui m'a fait profiter de son expérience, Julie Vachon et Christian Foisy de l'équipe DPML, David Janin pour l'été **COQ** et Alexandre pour ma réputation de savant fou qu'il entretient avec joie.

Je voudrais aussi remercier les gens de l'ENS à Paris qui m'ont si chaleureusement accueilli lors de mon séjour et particulièrement Philippe Béguin.

Je voudrais remercier le CRSNG qui a subventionné mes deux années de recherche et sans lequel beaucoup d'étudiants ne pourraient poursuivre leurs études et ainsi en faire profiter toute la communauté.

Finalement, un merci spécial pour Mélanie qui m'a supporté lors des moments plus difficiles. Je remercie aussi pour leur support mon père, ma mère, mon frère ainsi que tous mes amis.

MERCI!

Chapitre 1

Introduction

Historiquement, la cryptologie s'est surtout intéressée à la protection des communications contre les oreilles indiscrètes. Le scénario classique est le suivant: Alice veut révéler à Bob des informations secrètes dans un contexte où les seuls média de communication à sa disposition sont susceptibles d'être espionnés. À travers les époques, des solutions originales à ce problème ont été formulées. Le 20e siècle est très riche en résultats. Citons par exemple la théorie de l'information de Shannon [50, 51], la cryptologie quantique [10], ainsi que la cryptographie à clef publique [29] qui sont sans aucun doute les réalisations les plus marquantes dans le domaine.

Cependant, le champ d'action de la cryptologie ne se limite pas à cet unique problème. L'informatique, avec l'incroyable quantité d'informations qu'elle nous permet de traiter, menace certaines de nos valeurs démocratiques. En effet, dans notre société moderne, la confidentialité, l'authenticité et l'accès à l'information sont devenues des préoccupations importantes. Paradoxalement, l'ordinateur est à la fois mal et remède, étant devenu l'un des outils indispensables de la cryptologie.

Il est maintenant possible de concilier nos débordants besoins d'information et de confidentialité. C'est d'ailleurs une des préoccupations importantes de la cryptologie contemporaine. Dans cette catégorie de problèmes cryptographiques, on retrouve une multitude de protocoles concrets. *L'identification* permet à deux interlocuteurs de s'assurer de leur identité mutuelle. Elle peut être utilisée entre un client et une banque pour leur permettre à tous deux d'effectuer des opérations bancaires en confiance [27]. La *signature électronique* [30], semblable à son homologue conventionnel, est mieux adaptée aux besoins modernes de communication. L'*argent électronique* [16] pourra remplacer les billets de banque tout en garantissant l'anonymat à leurs utilisateurs. Le *vote* ou l'*élection secrète* permettra, à partir des ressources de télécommunications déjà existantes, de réaliser une élection sans que toute la population ait à se déplacer. Les *statistiques à l'aveugle* permettront aux statisticiens de faire leur travail sans rien apprendre d'autre sur la population étudiée que la statistique qu'il doivent évaluer. Notons encore le tirage au sort [8], le poker à l'aveugle [49, 21] et le problème du millionnaire, problème-jouet introduit par Yao qui consiste pour deux millionnaires à déterminer lequel est le plus riche sans toutefois révéler leur fortune respective.

Beaucoup de ces problèmes sont en fait des cas particuliers de calculs multiparties sur données privées (PMPC). En général, N participants avec chacun une entrée secrète x_i désirent calculer $y = F(x_1, ..., x_N)$ 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.

Voici cinq caractéristiques propres à ce genre de protocole.

- On dit que le protocole est *correct* si N participants qui respectent le protocole arrivent bien à apprendre $F(x_1, ..., x_N)$. Les protocoles cités et décrits dans ce mémoire sont tous *corrects*.

- On dit que le protocole est *privé* si la seule information qu'un groupe de participants G peut apprendre au sujet des x_i des autres participants est ce qui peut en être logiquement déduit à partir de y, F et des x_i appartenant aux membres du groupe. Un protocole est *t-privé* s'il résiste à n'importe quel groupe d'au plus t participants malhonnêtes.

- Le protocole est *honnête* si chaque participant sait quand il n'apprend pas le résultat de la fonction.
- Le protocole est *juste* si le fait qu'un participant apprenne $y = F(x_1, ..., x_N)$ implique que tous l'apprennent.
- Finalement, un protocole est *robuste* (*résiliant*) si les participants malhonnêtes ne peuvent le faire avorter une fois qu'il est bien engagé.

Une solution simple consiste à faire confiance à une tierce partie à qui tous révéleront en privé leur x_i. Il devra calculer la fonction, révéler sa valeur à chaque participant et tout oublier par la ensuite. Malheureusement une telle solution est en général irréaliste ou insatisfaisante, c'est pourquoi beaucoup de recherches ont été réalisées à la fois sur le problème général et pour des cas particuliers comme ceux cités plus haut.

Le reste de ce document traitera du problème général de l'évaluation multi-parties de fonctions sur données privées. On trouve dans [32] un survol détaillé de l'ensemble des résultats dans ce domaine. Seuls les résultats les plus pertinents seront présentés dans cette introduction. Les protocoles se divisent en trois familles: ceux sans hypothèse cryptographique, ceux basés sur des hypothèses calculatoires et les protocoles basés sur le transfert inconscient. Il ne sera question ici que de protocoles où aucune restriction n'est faite quant au comportement que pourraient avoir des participants malhonnêtes.

Ben-Or, Goldwasser et Wigderson (1988) [3] et de façon indépendamment Chaum, Crépeau et Damgård (1988) [18] ont présenté une solution $(N/3)$-privée, juste, honnête et robuste. Leurs protocoles nécessitent qu'au moins 2/3 des participants soient honnêtes et qu'un canal privé existe entre chaque paire de participants. Si, en plus des canaux privés, les participants disposent d'un canal de diffusion, Tal Rabin et Ben-Or (1989) [47] et, de façon indépendante, Beaver (1991) [1] ont montré comment obtenir des résultats similaires en se contentant seulement d'une majorité de participants honnêtes. Ces protocoles ont l'avantage de ne reposer sur aucune hypothèse cryptographique et d'être robustes. Ils sont malheureusement très inefficaces.

En faisant l'hypothèse de l'existence de *fonctions à brèches secrètes* [55], Goldreich, Micali et Wigderson (1987) ont présenté un protocole privé, juste, honnête et robuste. Ce protocole nécessite lui aussi un canal de diffusion et une majorité de participants honnêtes. Chaum, Damgård et van de Graaf (1987) [19] ont présenté indépendamment un protocole privé, juste et honnête basé sur des hypothèses calculatoires. Ce protocole nécessite un canal de diffusion. Il a la caractéristique intéressante de camoufler les données d'un des participants de façon inconditionnelle. Chaum a aussi présenté une solution hybride [17] qui pour certains aspects repose sur des hypothèses calculatoires mais qui pour d'autres est inconditionnellement sûre.

Il est aussi possible de réaliser des protocoles multi-parties en s'appuyant sur l'existence d'un transfert inconscient et par extension d'une mise en gage. Ces procédures cryptographiques bien connues peuvent être implantées dans une multitude de modèles et dans le modèle quantique en particulier. Elles seront décrites en détail dans le prochain chapitre. Notons immédiatement que le protocole original présenté dans ce mémoire appartient à cette catégorie. Le premier résultat de ce genre est celui de Kilian (1988) [39], qui donna une solution pour deux participants. Malheureusement, ce protocole nécessite une transformation complexe du circuit à évaluer et n'a pas été généralisé au cas où le nombre de participants est supérieur à deux. Goldreich et Vainish (1987) [35] ont présenté un protocole privé et honnête fondé sur le transfert inconscient où on suppose que les participants ne trichent pas activement. Le protocole présenté dans ce document lui emprunte plusieurs idées.

Ce mémoire présente un protocole correct, privé, juste et honnête permettant l'évaluation de fonctions multi-parties ne faisant aucune supposition quant à l'honnêteté des participants. Ce protocole est fondé sur l'existence d'un canal de diffusion et d'un canal privé de transfert inconscient entre chaque pair de participants. Pour réaliser cela, il sera utile de construire un protocole intermédiaire de transfert inconscient de mise en gage. Cette tâche cryptographique introduite par Crépeau [24] sera appelé transfert inconscient certifié (COT). Une version *améliorée* de ce protocole sera combinée avec des idées de Goldwasser et Levin [36] pour réaliser le premier protocole multi-parties sûr et efficace de ce genre.

1.1 Conventions et Notations

Dans l'ensemble de ce document, certaines conventions seront utilisées. Les noms de certaines variables et tous les noms des protocoles sont fixés. Notez que des protocoles différents ont parfois des propriétés de même nom. Ces propriétés doivent être interprétées dans le contexte bien précis du protocole auquel elles sont associées.

Notez que:

- P désigne un ensemble de participants.
- $N = |P|$ est le nombre de participants.
- m représente la taille d'un circuit.
- Nous qualifierons d'*actifs* les participants qui ont un rôle direct dans un

protocole par opposition aux participants *passifs* qui ne feront que vérifier que tout se déroule convenablement.

- n est le paramètre de sécurité. Les caractéristiques de tous les protocoles seront vérifiées sauf avec probabilité inférieure à $1/c^n \quad (c > 1)$.
- Nous utiliserons les mots *certitude* et *nécessairement* même dans le cas où il existe une probabilité d'erreur exponentiellement faible.
- Dans tous les protocoles, quand une condition n'est pas satisfaite, le protocole avorte.
- On note une mise en gage (BC) à un bit b par $\widehat{b}$ et un XOR-camouflage (BCX) à un bit b par $\boxed{b}$ (voir chapitre 2 sections 2 et 3). Dans le chapitre 3, on écrit $\boxed{w}$ pour représenter $\{\boxed{w_0}, \boxed{w_2}, ..., \boxed{w_n}\}$. Cet abus de langage ne sera utilisé que lorsque le contexte ne laissera aucun doute quant à la nature de la mise en gage.
- $x \in_R A$ signifie: x est choisi aléatoirement dans A selon la distribution uniforme.
- $A \subset_R B$ signifie: A est un sous-ensemble choisi aléatoirement parmi les éléments de B selon la distribution uniforme.
- On note la négation booléen de A par $\neg A$.
- Pour des raisons de cohérence avec les différentes références, les noms abrégés des protocoles sont inspirés de leurs noms anglais.

1.2 Organisation des chapitres

Le chapitre 2 sera consacré au transfert inconscient ($\binom{1}{2}$-OT) et à la mise en gage (BC). Ces protocoles seront les briques de base à partir desquelles tous les autres protocoles seront construits. Ils y seront définis formellement et les implantations dans les modèles les plus courants y seront brièvement discutées. De plus, nous introduirons dans ce chapitre une forme spéciale de mise en gage appelée XOR-camouflage (BCX). Ses propriétés y seront décrites et une partie importante du chapitre sera consacrée à leur implantation à partir de la mise en gage conventionnelle.

Le chapitre 3, qui est techniquement le plus difficile, est probablement aussi le plus important. Dans ce chapitre, nous décrivons en détail la procédure de transfert inconscient certifié. Elle sera construite à partir des XOR-camouflage (BCX) et du transfert inconscient ($\binom{1}{2}$-OT).

C'est à partir de la procédure de transfert inconscient certifié que nous construirons, dans le chapitre 4, le protocole de calculs multi-parties sur données privées (PMPC). Les détails du protocole, de sa fiabilité et de son efficacité y seront discutés.

Suivra ensuite la conclusion dans laquelle nous discuterons de l'importance qu'occupera ce type de cryptologie dans les années à venir.

La figure 1.1 montre les dépendances logiques entre les différents protocoles présentés dans ce mémoire. Le lecteur pourra y référer pour conserver une vision globale tout au long de la lecture.

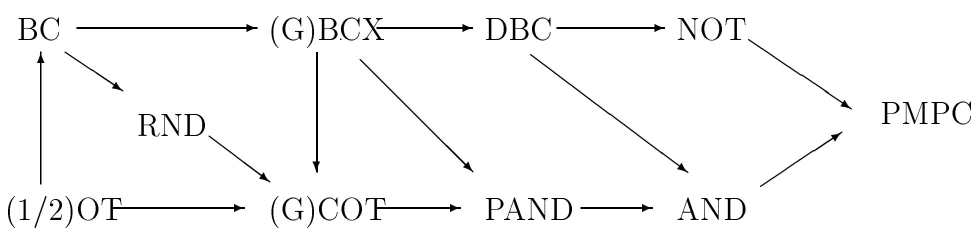

FIG. 1.1 - *Relations entre les protocoles*

Chapitre 2

Fondations

Tous les domaines des mathématiques reposent sur des vérités non démontrées: les axiomes. Ces axiomes ne sont pas choisis au hasard, on exige d'eux plusieurs propriétés. On les souhaite non-contradictoires mais suffisamment puissants pour engendrer toutes les "vérités" que l'on veut mettre en lumière. Cette approche est aussi très utilisée en science et en particulier en physique, comme en témoignent brillamment la théorie de la relativité et la mécanique quantique.

La cryptologie ne fait pas exception. Les cryptologues fondent la sécurité de leurs protocoles sur des hypothèses non-démontrées. Parmi ces hypothèses, on retrouve l'existence d'un canal bruyant, l'existence d'un canal quantique, et de façon plus courante, les hypothèses calculatoires comme la difficulté de factoriser les grands entiers, d'extraire le logarithme discret ou les racines carrées modulaires. Il existe une multitude de suppositions permettant de garantir aux protocoles une sécurité relative. Cette variété est la bienvenue, néanmoins il est intéressant pour des protocoles cryptographiques de haut niveau de résister aux changements de modèle. Il est fastidieux de créer de nouveaux protocoles chaque fois que l'on utilise un modèle différent ou même que l'on adapte un modèle déjà existant.

Dans le but de conserver toute la généralité possible, les protocoles que nous présentons dans ce document sont fondés sur deux procédures de base: le transfert inconscient ($\binom{1}{2}$-OT) et la mise en gage (BC). On connaît des implantations sécuritaires de ces protocoles cryptographiques dans une multitude de modèles. En fait, la mise en gage peut être implantée à partir du transfert inconscient et même à partir d'une version plus faible appelée transfert équivoque. En pratique, comme la mise en gage possède des implantations efficaces dans tous les modèles où l'on sait implanter le transfert inconscient ($\binom{1}{2}$-OT) nous considérerons ces deux procédures de façon indépendante. Ceci facilitera l'analyse des protocoles et donnera une idée plus juste de leurs complexités intrinsèques.

Les deux premières sections du chapitre sont consacrées au transfert inconscient et à la mise en gage. Nous consacrerons la dernière partie de ce chapitre à l'introduction du XOR-camouflage qui est une extension de la mise en gage. Le transfert inconscient et le XOR-camouflage seront les protocoles de base du transfert inconscient certifié présenté dans le chapitre suivant.

2.1 Transfert Inconscient $\binom{1}{2}$-OT

Le transfert inconscient pourrait être considéré comme une panacée étant donné qu'à partir de ce simple outil on peut régler une multitude de problèmes cryptographiques. Ce protocole est en l'occurrence suffisamment puissant pour permettre l'implantation de toutes les procédures qui seront décrites dans les pages qui suivent.

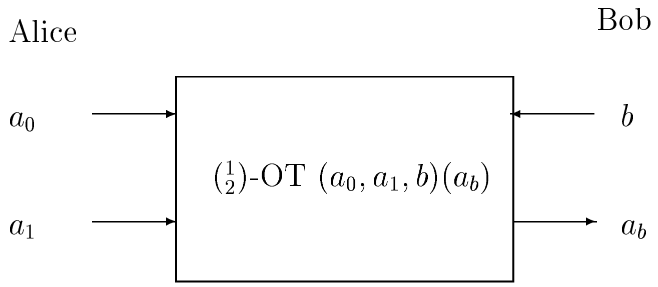

FIG. 2.1 - *Transfert inconscient*

Formellement, supposons qu'Alice connaît (choisit) deux bits a_0 et a_1 et que Bob connaît (choisit) un bit b. Après l'exécution de $\binom{1}{2}$-OT, Bob apprend $c = a_b$. Le protocole doit aussi être *privé*, c'est-à-dire que peu importe leur comportement, Alice n'apprend rien au sujet de b et Bob n'apprend rien au sujet de $a_{\neg b}$ même s'il connaissait déjà a_b. On appelle ce protocole *Transfert inconscient* ("One out of two Oblivious Transfer", $\binom{1}{2}$-OT). Il a été introduit par Wiesner sous le nom "message multiplexing" [5] et redécouvert par Even, Goldreich et Lempel [31]. La multitude des publications qui en traitent témoigne de sa grande utilité [26, 39, 22, 2, 46].

Ce protocole est conceptuellement simple. Supposons qu'Alice et Bob ont un ami commun, Charles, et que tous deux lui font entièrement confiance. L'implantation de $\binom{1}{2}$-OT$(a_0, a_1, b)(c)$ d'Alice vers Bob devient un jeu d'enfant. Le protocole devrait s'exécuter comme suit.

Protocole: $\binom{1}{2}$-OT naïf

Entrées	**Sorties**
Alice : a_0, a_1 : BIT	Bob : c : BIT
Bob : b : BIT	
Relations	**Propriétés**
$c = a_b$	*privé*

1: Alice révèle secrètement a_0 et a_1 à Charles.

2: Bob révèle secrètement b à Charles.

3: Charles révèle secrètement $c = a_b$ à Bob.

4: Charles oublie a_0, a_1 et b.

Il est facile de voir que cette implantation du protocole est correcte si Charles agit convenablement. Comme avec toutes les procédures cryptographiques, le fait d'avoir à sa disposition une tierce partie fiable en laquelle tous les protagonistes ont confiance permet une résolution simple du problème. Heureusement pour les cryptologues, il est parfois impossible (et surtout irréaliste) d'avoir confiance en un tel tiers.

On connaît une multitude de modèles dans lesquels $\binom{1}{2}$-OT peut être implanté, chacun avec ses caractéristiques propres. Voyons rapidement quelques-uns de ces modèles.

2.1.1 $\binom{1}{2}$-OT à partir du transfert équivoque

La première apparition du transfert équivoque dans la littérature nous vient de Rabin [46]. Le transfert équivoque qu'il présentait est une version plus simple du transfert inconscient présenté précédemment. Supposons qu'Alice connaît (choisit) un message w. En exécutant OT(w) avec Alice, Bob a une chance sur deux d'apprendre w sans qu'Alice ait la moindre information sur le fait que Bob l'ait appris ou non. Par contre, Bob, lui, sait s'il a appris w. Ce protocole sera appelé transfert équivoque ("Oblivious Transfer", OT).

Il n'est pas tellement difficile pour Alice et Bob en utilisant $O(n)$ OT d'effectuer un $\binom{1}{2}$-OT d'un bit [23]. La sécurité d'$\binom{1}{2}$-OT implanté de cette façon repose évidemment sur la sécurité de l'implantation de l'OT utilisé.

2.1.2 $\binom{1}{2}$-OT avec hypothèses calculatoires

Plusieurs problèmes cryptographiques calculatoires n'ont aucune solution efficace connue. Malheureusement, la difficulté intrinsèque de ces problèmes n'a pas encore été démontrée. Lorsque la sécurité d'un protocole dépend d'un tel problème, on dit que le protocole repose sur une hypothèse calculatoire. En fait, on limite la capacité de calcul d'un (ou même des deux) participants.

De façon générale, Goldreich, Micali et Wigderson ont montré que l'existence d'une permutation à brèche secrète est suffisante pour implanter $\binom{1}{2}$-OT [34]. Le problème de la factorisation des grands entiers et en particulier du résidu quadratique [37] permet une implantation intéressante de $\binom{1}{2}$-OT [14]. Cette technique peut être généralisée à d'autres hypothèses calculatoires comme RSA [48] et le problème du logarithme discret.

L'avantage de faire reposer un protocole sur une hypothèse calculatoire réside dans la simplicité de son implantation. Il est facile de mettre à la disposition des participants des machines et des canaux de communication classiques. Par contre, comme la sécurité intrinsèque de ces problèmes n'a pas été prouvée, ils sont potentiellement à la merci d'un adversaire ayant une puissance algorithmique supérieure.

2.1.3 $\binom{1}{2}$-OT avec canal bruyant

Les propriétés d'un canal bruyant sont très semblables à celles du transfert équivoque de Rabin [46], il n'en demeure pas moins que l'implantation de $\binom{1}{2}$-OT à partir d'un canal binaire symétrique (canal bruyant) d'erreur ϵ nécessite un travail non négligeable. Le problème a été énoncé pour la première fois dans [26] et une solution plus efficace est à paraître dans [25]. L'algorithme utilise $O(n^3)$ transferts de bits sur le canal pour obtenir un seul $\binom{1}{2}$-OT. L'avantage de cette implantation est de ne pas reposer sur des hypothèses calculatoires.

2.1.4 $\binom{1}{2}$-OT avec canal quantique

L'introduction du canal quantique [54, 5, 4] a été une véritable révolution dans le domaine de la cryptologie. Avec le canal quantique la sécurité de $\binom{1}{2}$-OT repose sur la validité même de la mécanique quantique et non sur des hypothèses calculatoires ou d'autres techniques non démontrées. Sans être absolue, cette notion de sécurité est probablement la plus forte que l'on puisse imaginer dans le contexte d'un $\binom{1}{2}$-OT.

Le protocole est présenté dans [6] et un premier pas vers la démonstration de sa sécurité se trouve dans [56]. Le néophyte trouvera une foule de références utiles dans [10] de même que dans [11] sa version *World Wide Web* (**WWW**) régulièrement mise à jour.

2.2 Mise en Gage

La *mise en gage* ("Bit Commitment", BC) est, tout comme $\binom{1}{2}$-OT, une procédure fondamentale de la cryptologie. Elle est utilisée dans une multitude de protocoles cryptologiques et en particulier dans les protocoles où des preuves à divulgation nulle [38] sont nécessaires.

Par mise en gage (BC), on réfère en réalité à deux procédures distinctes mais complémentaires, soit l'*engagement* et l'*ouverture* (ou *révélation*). Supposons qu'Alice a un bit a en tête. On dit qu'elle s'*engage* à a envers Bob par $\widehat{a}$ si le protocole exécuté est:

correct: il existe une procédure $Ouvre(\widehat{a}) = a$ qui révèle a à Bob.

confidentiel: Bob n'apprend rien au sujet de a avec $\widehat{a}$.

et *liant*: il n'est pas possible pour Alice de convaincre Bob que $Ouvre(\widehat{a}) = \bar{a}$.

Par exemple, pour s'engager au sujet d'un certain bit, Alice pourrait remettre un coffre-fort à Bob dont elle seule connaît la combinaison. À l'intérieur du coffre se trouverait un bout de papier sur lequel serait écrit **0** ou **1**. Pour ouvrir la mise en gage, Alice n'aurait qu'à donner la combinaison du coffre à Bob qui pourrait alors apprendre la valeur du bit qui s'y trouve camouflé. Cet exemple-jouet illustre bien les trois caractéristiques d'une mise en gage.

FIG. 2.2 - *Mise en gage*

Tout comme $\binom{1}{2}$-OT, BC ne peut être implanté sans hypothèses. Dans l'exemple précédent, c'est la confiance envers le coffre-fort qui permet sa réalisation. De multiples modèles existent dans lesquels il est possible d'implanter BC. En fait, on peut utiliser $\binom{1}{2}$-OT et même le *transfert équivoque* pour le réaliser. Comme dans la plupart des modèles où $\binom{1}{2}$-OT existe, on peut implanter BC directement de façon plus efficace et parfois même avec des hypothèses plus faibles, BC sera aussi considéré comme procédure de base. Voici quelques modèles où des implantations intéressantes sont connues.

2.2.1 BC à partir de $\binom{1}{2}$-OT ou du transfert équivoque

Comme nous l'avons dit plus tôt, la mise en gage peut être réalisée en utilisant $\binom{1}{2}$-OT ou le transfert équivoque comme sous-protocole. On trouve dans [12] comment le réaliser à partir de $\binom{1}{2}$-OT. Sa réalisation à partir du transfert équivoque est décrite dans [40] et attribuée à Crépeau. Dans les deux cas, le protocole nécessite $O(n)$ transferts pour effectuer une mise en gage et l'ouverture se fera en envoyant un message classique de taille $O(n)$. La sécurité de ces implantations repose sur la sécurité du transfert utilisé.

2.2.2 BC avec hypothèse calculatoire

On peut implanter la mise en gage en faisant des suppositions de nature calculatoire. En fait, il a été démontré par Naor [44] que l'existence d'un générateur pseudo-aléatoire (et donc d'une fonction à sens unique) est suffisante pour réaliser la mise en gage. On trouve aussi dans [28, 45, 12, 19] d'autres implantations intéressantes.

La dernière partie du chapitre est consacrée à des implantations spéciales de mise en gage permettant des opérations supplémentaires. Notons tout de suite que dans le monde calculatoire, plusieurs mises en gage possèdent déjà ces propriétés de façon intrinsèque.

2.2.3 BC avec canal quantique

De la même façon que pour le transfert inconscient, il est possible d'effectuer la mise en gage en utilisant le *canal quantique* et ainsi de faire reposer sa sécurité sur la validité de la mécanique quantique elle-même. Ce résultat de Brassard, Crépeau, Jozsa et Langlois [15] nous donne une implantation de BC utilisant $O(n)$ transferts sur le *canal quantique*. L'ouverture se fait en communiquant un message classique de taille $O(n)$.

2.2.4 BC avec canal bruyant

Il est possible d'implanter la mise en gage si l'on dispose d'un canal bruyant. Une légère modification du protocole basé sur le *canal quantique* [13] ou de celui basé sur le transfert équivoque [40] permet l'implantation de BC. Le protocole se retrouvera aussi dans [25]. L'avantage de l'utilisation du canal bruyant est encore une fois d'éviter les hypothèses calculatoires.

2.2.5 Application de BC au tirage aléatoire

Il est possible, en utilisant la mise en gage, à un groupe de N participants de choisir un bit aléatoirement. Le bit pourra être choisi de façon à ce que même une coalition de $N - 1$ participants ne puisse prédire avec plus d'une chance sur deux le bit qui sera choisi, et ce en autant qu'au moins un participant soit honnête.

Nous présentons ici une solution simple qui sera suffisante pour les besoins des protocoles qui suivent. Pour tirer un bit aléatoirement en groupe, chaque participant choisit lui même un bit aléatoire et s'y engage envers tous les autres participants. Ensuite, chaque participant ouvre toutes ses mises en gage. Le bit aléatoirement tiré par le groupe est le OU-EXCLUSIF de la valeur que chaque participant avait aléatoirement choisie.

Protocole: RND

Entrées	**Sorties**
Nil	x : BIT
Relations	**Propriétés**
Nil	*aléatoire*

1: $\forall i \in P$, i choisit $b_i \in_R \{0,1\}$

2: $\forall i, j \in P$ tel que $i \neq j$, i s'engage à $\widehat{b_i^j} = b_i$ envers j.

3: $\forall i, j \in P$ tel que $i \neq j$, i ouvre $\widehat{b_i^j}$ à j puis j diffuse b_i.

4: $\forall i \in P, i$ vérifie que $\forall j \ b_j^i = b_j$.

5: $x = \bigoplus_{i=1}^{N} b_i$

Il est facile de voir que ce protocole répond bien aux exigences décrites précédemment. Tous les participants s'engagent à leur choix aléatoire pour éviter que les participants malhonnêtes choisissent la valeur de leur bit en fonction des valeurs déjà révélées. Il suffit qu'un bit parmi les N soit aléatoire pour que le résultat du OU-EXCLUSIF le soit aussi.

Ce protocole nécessite un canal de diffusion et $O(N)$ opérations sur des mises en gage par participant.

2.3 XOR-camouflage (BCX)

Dans cette section, nous introduisons un type spécial de mise en gage. En plus des propriétés habituelles des mises en gage, à savoir être confidentielles et liantes, elle seront linéairement corrélables et jumelables.

linéairement corrélable: Si Alice est engagée au sujet de $\{\boxed{b_1}, \boxed{b_2}, ..., \boxed{b_k}\}$ via des BCX, il lui sera possible de prouver à Bob que $\bigoplus_{i=1}^{k} \boxed{b_i} = c$ sans ouvrir aucune des mises en gage ni révéler aucune autre information à leur sujet.

jumelable: Il est possible à Alice de s'engager envers plusieurs participants à une même valeur de telle façon que les participants honnêtes soient convaincus qu'Alice s'est effectivement engagée à la même valeur envers chacun d'eux.

On appellera ces mises en gage des XOR-camouflages (BCX). Les propriétés supplémentaires des BCX en font des outils puissants. Il seront, avec $\binom{1}{2}$-OT, les briques de base du protocole de transfert inconscient certifié (GCOT) présenté dans le prochain chapitre.

2.3.1 BCX calculatoire

Dans le modèle calculatoire, il existe des implantations de la mise en gage ayant d'office les propriétés des BCX. Sans entrer dans les détails, notons que plusieurs mises en gage obtenus à partir d'hypothèses calculatoires sont déjà linéairement corrélables et jumelables [19]. Aucun BC ayant directement ces propriétés n'est encore connu dans les autres modèles. Heureusement on peut réaliser des BCX à partir de BC aux propriétés conventionnelles. Le reste de ce chapitre sera d'ailleurs consacré à cette réalisation.

2.3.2 Obtenir des BCX à partir de BC

Voyons maintenant comment, à partir des BC conventionnels corrects, confidentiels et liants, on peut construire des BCX qui seront en plus linéairement corrélables et jumelables. La technique présentée ici est directement inspirée de travaux de Kilian [41] qui lui même s'est inspiré de Rudich et de Bennett.

Les BCX seront constitués de n paires de BC conventionnels, chaque paire représentant la valeur auquel le participant est engagé. On dit qu'une paire *représente* une certaine valeur si le OU-EXCLUSIF de ses deux composantes égale cette valeur. Formellement un BCX $\boxed{b}$ à une valeur b est un ensemble de n paires de BC

$$\boxed{b} = \{(\widehat{b_{iG}}, \widehat{b_{iD}}) | 1 \leq i \leq n, \forall i \; b_{iG} \oplus b_{iD} = b\}$$

Si toutes les paires n'ont pas la même valeur, on dira que le BCX est inconsistant. La valeur d'un tel BCX sera égale à la valeur de la majorité des paires et on dira que les paires telles que $\widehat{b_{iG}} \oplus \widehat{b_{iD}} \neq b$ sont inconsistantes. Nous verrons plus de détails au sujet de leur création et des inconsistances ultérieurement.

2.3.3 Linéairement corrélable

Nous présentons maintenant un protocole permettant à Alice de convaincre Bob que le OU-EXCLUSIF de k BCX est c. Le protocole sera

- *privé*: aucune autre information au sujet des k BCX ne sera révélée.
- *honnête*: Alice ne pourra convaincre Bob que le OU-EXCLUSIF des k BCX est b si ce n'est pas le cas.

Pour ce faire, Bob choisira aléatoirement une paire de BC pour chaque BCX et demandera à Alice de le convaincre que ces paires ont la bonne parité c. Pour ce faire Alice révélera la parité des BC de gauche à Bob qui pouras alors calculer la parité des BC de droite. Bob lui demandera ensuite selon un choix aléatoire d'ouvrir tous les BC de gauche ou au contraire tous les BC de droite. Il s'assurera finalement que leur valeur est consistante avec ce qu'Alice avait prétendu. Toutes ces opérations devront être répéter n fois de façon à ce que Bob acquière la certitude qu'Alice n'a pu le trompé.

Protocole: BCX-OU-EXCLUSIF

Entrées	**Sorties**
Alice : $\boxed{b^1}, ..., \boxed{b^k}$:BCX	Nil
Alice : b : BIT	
Relations	**Propriétés**
Etablir: $\bigoplus_{i=1}^k \boxed{b^i} = b$	*privé*

1: Alice révèle $b = \bigoplus_{i=1}^k b^i$ à Bob.

2: Bob permute aléatoirement l'ordre des paires dans chaque BCX.

3: Pour j de 1 à n

3.1: Alice révèle $b_{jG} = \bigoplus_{i=1}^k b^i_{jG}$.

3.2: Bob fixe $b_{jD} = b_{jG} \oplus b$

3.3: Bob choisit aléatoirement $X_j \in_R \{G, D\}$ et révèle son choix à Alice.

3.4: Pour i de 1 à k Alice ouvre $\widehat{b^i_{jX_j}}$.

3.5: Bob vérifie que $b_{jX_i} = \bigoplus_{i=1}^k b^i_{jX_j}$

Il n'est pas trop difficile de se convaincre que si Alice et Bob suivent le protocole et que $\bigoplus_{i=1}^k \boxed{b^i} = b$, le protocole s'exécutera sans encombres. Par contre, si les BCX sont construits de façon inconsistante ou si leur parité n'est pas b, Bob aura une probabilité exponentiellement faible de ne pas avorter le protocole. En effet, si $\bigoplus_{i=1}^{k} b^i \neq b$ on aura que $\forall i(\bigoplus_{j=1}^{k} b_{jG}^i \neq \bigoplus_{j=1}^{k} b_{jD}^i \oplus b)$. Si Alice révèle la véritable valeur de $b_{jG} = \bigoplus_{i=1}^{k} b_{jG}^i$, Bob fixera $b_{jD} = b_{jG} \oplus b \neq \bigoplus_{i=1}^{k} b_{jD}^i$ et Alice sera prise en défaut si $X_i = D$. De la même façon, si Alice ment et prétend que $b_{jG} = \bigoplus_{j=1}^{k} b_{jD}^i \oplus b \neq \bigoplus_{j=1}^{k} b_{jG}^i$ elle sera prise en défaut si $X_i = G$. Donc peu importe la stratégie d'Alice, si Bob choisit les X_i de façon aléatoire, il n'aura qu'une probabilité de $1/2^n$ de ne pas se rendre compte qu'Alice tente de le berner.

La même chose se produit si un des BCX est inconsistant, mais cette fois seulement pour certaines lignes (certaines valeurs de j). Si un très petit nombre d'inconsistances existent, il se peut que le protocole réussisse mais la probabilité qu'il avorte s'approche exponentiellement de 1 avec le nombre d'inconsistances. Les BCX très inconsistants seront donc inutilisables. Nous reviendrons sur ce point.

Nous savons maintenant qu'Alice peut convaincre Bob de n'importe quelle relation linéaire entre des BCX. Malheureusement, cette opération consomme toutes les paires de BC qui le constituent. Pour être en mesure d'effectuer plusieurs preuves sur le même BCX, il faudra le copier. Les copies devront se faire elles aussi de façon privée et honnête, c'est-à-dire que le protocole ne devra révéler aucune information sur le BCX qui est copié et garantir que les copies ont toutes la même valeur que le BCX original.

Pour faire k copies d'un BCX $\boxed{b}$, Alice construira $n(k+1)$ paires de BC représentant b et Bob les partitionnera aléatoirement en $k+1$ BCX. Il en choisira ensuite un et demandera à Alice de montrer qu'il est égal à $\boxed{b}$.

Protocole: BCX-COPIE

Entrées	**Sorties**
Alice : $\boxed{b}$:BCX	Bob : $\boxed{b_1}, ..., \boxed{b_k}$: BCX
Relations	**Propriétés**
$\forall i \; b_i = b$	*privé, honnête*

1: Alice construit l'ensemble $E = \{(\widehat{b_{iG}}, \widehat{b_{iD}}) | 1 \leq i \leq n(k+1), b_{iG} \oplus b_{iD} = b\}$.

2: Bob partitionne aléatoirement E en $k+1$ ensembles de n paires qui formeront les BCX $\{\boxed{b_0}, ..., \boxed{b_k}\}$.

3: Alice prouve à Bob que $\boxed{b} \oplus \boxed{b_0} = 0$.

Ce protocole est évidemment correct, si Alice et Bob coopèrent, il y aura bien k copies de $\boxed{b}$ à la fin du protocole. Le plus important est qu'Alice soit incapable de construire un BCX $\boxed{b_i} \neq b$. Comme Bob partitionne les paires de façon aléatoire, à peu près autant de paires inconsistantes se retrouveront dans chaque BCX et donc aussi dans $\boxed{b_0}$. Si Alice construit assez de paires inconsistantes pour avoir une certaine probabilité de construire un BCX consistant ayant comme valeur $\bar{b}$ elle échouera nécessairement le test de l'étape 3. De plus, si $\boxed{b}$ est dès le départ inconsistant, Alice ne réussira tout simplement pas à le copier. Dans le pire cas, Alice réussira à construire des copies contenant de très petits nombres d'inconsistances, ce qui n'est pour elle d'aucun intérêt.

Avant d'effectuer une preuve sur des BCX, Alice en fera une copie, ce qui lui permettra de conserver les BCX pour un usage ultérieur. On obtient, preuve et copie confondues, que k BCX sont linéairement corrélables au prix de l'utilisation de $O(nk)$ opérations élémentaires sur les BC qui les constituent.

2.3.4 Jumelable

Dans le cas où Alice désire s'engager à la même valeur envers plusieurs participants, il faudra porter une attention particulière lors de la création des BCX pour s'assurer qu'ils soient jumelables. Rappelons qu'un BCX est *jumelable* s'il est possible à Alice de s'engager envers plusieurs participants à une même valeur de telle façon que les participants honnêtes soient convaincus qu'Alice s'est effectivement engagée à la même valeur envers chacun d'entre eux. La consistance n'est pas très importante lors de la création. Nous avons vu que des BCX inconsistants, ne pouvant être ni ouverts, ni utilisés dans une preuve, sont aussi utiles que des BCX qu'Alice refuse d'utiliser.

La création de $N-1$ BCX à la même valeur par Alice, un à chacun des autres participants d'un groupe[1] P, se fera comme suit. Alice s'engage envers des mises en gage conventionnelles à $2n$ paires représentant le même bit b envers chaque participant. Le reste du protocole sera répété n fois. Tous les participants choisissent ensemble et aléatoirement une paire de BC appartenant a chaque BCX. Alice classe ces pairs en deux groupes de paires identiques. Puis, tous les participants choisissent aléatoirement de demander à Alice d'ouvrir les parties gauches ou les parties droites de chaque paire. Enfin les participants vérifient que tous les BC ouvert d'un même groupe sont égaux. Après avoir exécuté cela n fois, les participants honnêtes de P sont convaincus que les paires qui restent constituent des BCX ayant la même valeur pour chacun d'entre eux.

Une fois les BCX construits de façon jumelable, les opérations de copies et de preuves se feront individuellement avec chaque participant.

1. Pour simplifier la notation on suppose momentanément Alice exclu de P

Protocole: CRÉER-BCXS

Entrées

Alice : b : BIT

Sorties

$\forall i \in P$: $\boxed{b^i}$:BCX

Relations

$\forall\ b = b^i$

Propriétés

privé, jumelable

1: $\forall i \in P$, pour j de 1 à n Alice s'engage envers i à $(\widehat{b^i_{jG}}, \widehat{b^i_{jD}})$ tel que $b^i_{jG}, b^i_{jD} \in_R \{0,1\}$ et $b^i_{jG} \oplus b^i_{jD} = b$.

2: $\forall i \in P$, i choisit aléatoirement une permutation de ses $2n$ paires, la rend publique et renomme ses paires suivant la permutation.

3: Pour j de 1 à n

3.1: Alice construit

$$V_0 = \{\ i \mid (\widehat{b^i_{jG}}, \widehat{b^i_{jD}}) = (\widehat{0}, \widehat{b})\ \} \text{ et}$$
$$V_1 = \{\ i \mid (\widehat{b^i_{jG}}, \widehat{b^i_{jD}}) = (\widehat{1}, \widehat{\bar{b}})\ \}.$$

3.2: Alice révèle $E \in_R \{V_0, V_1\}$.

3.3: $x = \text{RND}(\{G, D\})$

3.4: $\forall i \in P$, Alice ouvre $\widehat{b^i_{jX}}$ à i qui diffuse sa valeur.

3.5: $\forall i \in P$, i vérifie que

$$\forall j, k \in E, b^i_{jX} = b^i_{kX} \text{ et}$$
$$\forall j, k \notin E, b^i_{jX} = b^i_{jX}.$$

4: $\forall i \in P$, $\{(\widehat{b^i_{jG}}, \widehat{b^i_{jD}}) | s+1 \leq j \leq 2s\}$ formera le BCX $\boxed{b^i}$ du participant i.

Chapitre 3

Transfert Inconscient Certifié

Ce chapitre est le plus important de ce mémoire. Nous y présentons le protocole de *transfert inconscient certifié* ("Commited Oblivious Transfer", COT). Ce protocole cryptographique est d'une utilité incontestable, comme en témoignera le chapitre suivant où nous verrons comment il est le coeur du protocole d'évaluation multi-partie d'une fonction sur données privées.

Le transfert inconscient certifié a été introduit en 1989 par Claude Crépeau [24] sous l'appellation "Verifiable Oblivious Transfer". Un protocole en apparence plus efficace a été publié en 1989 par Goldwasser et Levin sous le nom de "Preprocess-Oblivious-Transfer" [36]. Malheureusement, une analyse attentive de leur protocole montre qu'il est possible pour les participants de tricher sans être repérés avec probabilité non négligeable. Bien qu'il ne soit pas trop difficile de remédier à ce problème, le protocole qui en résulterait serait encore moins efficace.

Le protocole COT que nous présentons dans ce chapitre est efficace et sûr. Il utilise la mise en gage (via les BCX) et le transfert inconscient ($\binom{1}{2}$-OT) sans toutefois faire de restriction sur leur mode d'implantation. Ceci permet d'adapter les protocoles utilisant COT aux multiples modèles dans lesquels évolue la cryptologie contemporaine. En l'occurrence, avec un transfert inconscient et une mise en gage basés sur l'existence d'un canal quantique, on réalise COT sans hypothèses calculatoires.

L'efficacité du protocole est liée à son utilisation avantageuse de propriétés des codes correcteurs. Ils sont utilisés dans le protocole, non pas pour corriger d'éventuelles erreurs de transmission, mais dans un but de sécurité et d'efficacité.

Dans les prochaines sections, nous allons définir formellement le transfert inconscient certifié (COT) et expliquer en détail son implantation pour deux participants ou à l'intérieur d'un groupe. Nous discuterons aussi de sa validité et de son efficacité.

3.1 Définition

Le transfert inconscient certifié ("Commited Oblivious Transfer", COT) est la réunion logique de la mise en gage (BC) et du transfert inconscient ($\binom{1}{2}$-OT). En effet, un COT n'est rien d'autre qu'un $\binom{1}{2}$-OT où les bits en jeu sont des BCX.

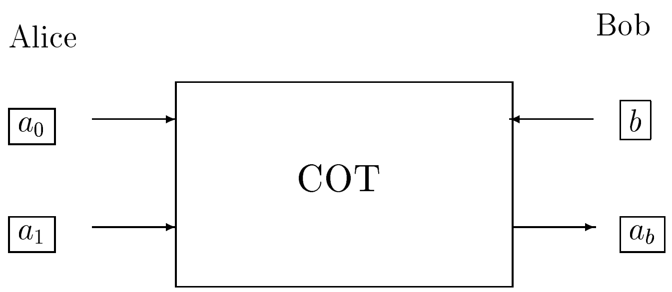

FIG. 3.1 - *Transfert inconscient certifié*

Supposons qu'Alice soit engagée à $\boxed{a_0}$ et $\boxed{a_1}$ envers Bob et que lui soit engagée à $\boxed{b}$ envers Alice. Après COT$(\boxed{a_0}, \boxed{a_1}, \boxed{b})(\boxed{c})$ Bob est engagée à $\boxed{c}$. De plus, le protocole sera:

- *correct*: Si Alice et Bob respectent le protocole, Bob sera engagée envers Alice à $\boxed{c} = a_b$.
- *honnête*: Si Alice respecte le protocole, Bob ne peut être engagée à $\boxed{c} = \bar{a_b}$.
- *privé*: Si Alice respecte le protocole, Bob n'apprendra rien au sujet de $a_{\bar{b}}$, et si Bob respecte le protocole, Alice n'apprendra rien au sujet de b.

Voyons une implantation de COT s'appuyant sur une tierce partie honnête nommée Charles. Cette implantation naïve illustre bien les caractéristiques du protocole.

Protocole: COT naïf

Entrées

Alice : $\boxed{a_0}, \boxed{a_1}$: BCX

Bob : $\boxed{b}$: BCX

Sorties

Bob : $\boxed{c}$: BCX

Relations

$c = a_b$

Propriétés

privé, honnête

1: Alice s'engage à $\boxed{a_0}$ et $\boxed{a_1}$ envers Bob et Charles et
Bob s'engage à $\boxed{b}$ envers Alice et Charles
si ce n'est pas déjà fait.
(n.b. des mises en gage jumelables sont utilisées)

2: Alice ouvre $\boxed{a_0}$ et $\boxed{a_1}$ à Charles en privé.

3: Bob ouvre $\boxed{b}$ à Charles en privé.

4: Charles révèle a_b à Bob en privé.

5: Bob s'engage à $\boxed{c} = a_b$ envers Alice et Charles.

6: Bob ouvre $\boxed{c}$ à Charles en privé qui vérifie que $c = a_b$.

7: Charles oublie a_0, a_1, b et c.

3.2 Description informelle de COT

Intuitivement, pour accomplir COT($\boxed{a_0}$, $\boxed{a_1}$, $\boxed{b}$)($\boxed{c}$) avec Bob, Alice réalise de façon imparfaite un *transfert inconscient de chaîne* sur deux mots de code w_0 et w_1. On dit de ces transferts qu'ils sont imparfaits du fait que même si Bob est honnête, il apprendra de l'information au sujet des deux mots. Bob apprendra tous les bits du mot w_b et quelques bits de $w_{\bar{b}}$. Cela lui permettra de s'assurer que s'il avait effectué le protocole avec b ayant la valeur opposée, tout aurait aussi bien fonctionné. Si Bob n'effectuait pas cette vérification, Alice pourrait se comporter de telle façon que le protocole fonctionne seulement pour une valeur de b, ce qui lui permettrait, par la réussite ou l'échec du protocole, d'apprendre b. À la fin du protocole, Alice montrera une relation entre $\boxed{a_0}$ et $\boxed{w_0}$ ainsi qu'entre $\boxed{a_1}$ et $\boxed{w_1}$, ce qui permettra à Bob d'apprendre a_b et de s'y engager via $\boxed{c}$. Il faut donc aussi s'assurer que Bob ne puisse avoir suffisamment d'information au sujet des deux chaînes pour apprendre les deux bits d'Alice.

Voici de façon plus détaillée les différentes étapes du protocole. Au début, Alice est engagée à $\boxed{a_0}$ et $\boxed{a_1}$, Bob est engagée à $\boxed{b}$. Bob choisit un code avec de bonnes propriétés (voir sec. 3.4). Alice s'engage à $\boxed{w_0}$ et $\boxed{w_1}$, deux mots de code choisis aléatoirement, et prouve qu'ils sont bien des mots de code. Ensuite, Alice transfère w_0 et w_1 de telle façon qu'en chaque position, Bob puisse choisir entre apprendre un bit de w_0 ou de w_1, mais pas les deux. Bob choisira d'apprendre la plupart des bits du côté de w_b mais aussi quelques bits de $w_{\bar{b}}$. Pour se convaincre qu'Alice a bel et bien transmis les bits conformément à $\boxed{w_0}$ et $\boxed{w_1}$, Bob lui demandera, pour un certain nombre de positions, d'ouvrir des BCX de $\boxed{w_0}$ et $\boxed{w_1}$. Si Alice a triché lors de plusieurs transferts, Bob aura une très grande probabilité de s'en rendre compte à ce moment et d'avorter le protocole. Jusqu'à maintenant aucune information concernant a_0 ou a_1 n'a encore été révélée. De plus, si Bob refuse de continuer cela ne donne aucune information à Alice au sujet de b. Par contre, si Bob ne constate aucune inconsistance, il est certain qu'Alice n'a pu tricher sur un grand nombre de positions. Il utilisera le fait que w_0 et w_1 sont des mots de code pour repérer ces positions et les corriger, puis il s'engagera à $\boxed{w} = w_b$.

C'est maintenant à Bob de convaincre Alice que le mot auquel il est engagée est bien w_b. Pour ce faire, Bob prouve à Alice que $\boxed{w}$ est un mot de code. Ensuite, Alice ouvre un certain nombre des positions de w_0 et w_1 et Bob lui prouve que pour ces positions, les bits de $\boxed{w}$ sont égaux à ceux de w_b et ce, sans ouvrir ses BCX ni donner d'informations au sujet de b.

Même si Bob est engagé à $\boxed{w} = w_b$, il possède tout de même de l'information au sujet de $w_{\bar{b}}$, information qu'il obtient par les bits qu'il a appris et du fait qu'il sait que c'est un mot de code. Alice doit maintenant utiliser les deux mots w_0 et w_1 pour révéler à Bob a_b, sans toutefois donner d'information au sujet de $a_{\bar{b}}$. Pour ce faire, elle utilisera une fonction de distillation de secret (*"privacy amplification"* [7]) h tel que $a_0 = h(w_0)$ et $a_1 = h(w_1)$. Cette fonction, en l'occurrence le produit scalaire avec un mot m approprié, sera révélé à Bob. Il pourra alors calculer $c = a_b = m \cdot w$ et s'y engager sans pour autant pouvoir l'utiliser pour apprendre $a_{\bar{b}}$. Finalement, Bob doit prouver à Alice qu'il s'est bien engagé à $\boxed{c} = m \cdot \boxed{w}$ et donc que $\boxed{c} = a_{\boxed{b}}$.

3.3 Description Formelle

Avant le protocole, Alice est engagée à $\boxed{a_0}$ et $\boxed{a_1}$. De son coté, Bob est engagée à $\boxed{b}$ et à la fin du protocole, il sera engagée à $\boxed{c} = a_b$. Alice et Bob s'engagent toujours à un bit en utilisant un BCX et ils s'engagent à une chaîne en s'engageant à chacun de ses bits. Le choix du code qui influence le choix des constantes α et ϵ est discuté dans la prochaine section et les preuves dans la section suivante. À n'importe quel moment du protocole, si une preuve ou une vérification échoue, le protocole est avorté.

Protocole: COT

Entrées

Alice : $\boxed{a_0}, \boxed{a_1}$: BCX

Bob : $\boxed{b}$: BCX

Sorties

Bob : $\boxed{c}$: BCX

Relations

$c = a_b$

Propriétés

privé, honnête, juste

1: Bob choisit et annonce à Alice un code linéaire C $[n, k, d]$ avec $k > (1/2 + 2\sigma)n$ et $d > \epsilon n$, corrigeant efficacement $t \in \Omega(n)$ erreurs.

2: Alice choisit $w_0, w_1 \in_R C$, s'y engage via $\boxed{w_0}$ et $\boxed{w_1}$ et prouve que $\boxed{w_0} \in C$ et $\boxed{w_1} \in C$.

3: Bob choisit $I_0, I_1 \subset_R \{1...n\}$ avec $|I_0| = |I_1| = \sigma n,\ I_1 \cap I_0 = \emptyset$.

4: Pour i de 1 à n, si $i \in I_0$ alors

Alice exécute $\binom{1}{2}$-OT$(w_0^i, w_1^i, \bar{b})$ avec Bob qui reçoit w^i

sinon Alice exécute $\binom{1}{2}$-OT(w_0^i, w_1^i, b) avec Bob qui reçoit w^i.

5: Bob annonce $I = I_0 \cup I_1$ à Alice puis $\forall i \in I$, Alice ouvre $\boxed{w_0^i}$ et $\boxed{w_1^i}$.

6: $\forall i \in I_0$, Bob vérifie que $w^i = w_{\bar{b}}^i$ et affecte $w^i \leftarrow w_b^i$, $\forall i \in I_1$, Bob vérifie que $w^i = w_b^i$.

7: Bob corrige w en utilisant l'algorithme de décodage de C, s'engage à $\boxed{w}$ et prouve que $\boxed{w} \in C$.

8: Alice choisit $I_2 \subset_R \{1...n\}$ avec $|I_2| = \sigma n$ et $I_2 \cap I = \emptyset$, annonce I_2 et ouvre $\boxed{w_0^i}$ et $\boxed{w_1^i}$, pour $i \in I_2$.

9: $\forall i \in I_2$, Bob prouve que $\boxed{w^i} = w_{\boxed{b}}^i$.

10: Alice choisit $m \in_R \{0,1\}^n$ tel que $\boxed{a_0} = m \cdot \boxed{w_0}$ et $\boxed{a_1} = m \cdot \boxed{w_1}$ et le prouve.

11: Bob s'engage à $\boxed{c} \leftarrow m \cdot \boxed{w}$ et le prouve.

3.4 Choix du code

Pour réaliser COT, certaines propriétés des codes seront mises à profit. Nous présentons ici les concepts essentiels pour la réalisation de COT mais le lecteur désirant avoir plus d'information pourra se référer à [43].

Un code linéaire[1] C $[n, k, d]$ est un sous-ensemble de $\{0, 1\}^n$ tel que

1. $|C| = 2^k$
2. $\forall\ x, y \in C, x \neq y \Rightarrow D(x, y) \geq d$. où $D(x, y)$, la distance de Hamming, représente le nombre de positions différentes entre x et y.
3. $\forall x, y \in C, x \oplus y \in C$.

Pour qu'une famille de codes soit intéressante, il faut qu'un algorithme efficace permettant de corriger αn erreurs lui soit associé, c'est-à-dire un algorithme permettant, à partir d'un mot de code x et d'un mot x' tels que $D(x, x') \leq \alpha n$, de retrouver x.

Dans le protocole COT, Bob devra choisir un code C $[n, k, d]$ tel que

$$k > (1/2 + 2\sigma)n \quad et \quad d > \epsilon n.$$

De plus, il devra exister un algorithme efficace permettant de corriger une quantité d'erreurs proportionnelle à d. De tels codes existent; les codes concaténés [33] sont les codes les plus connus pouvant satisfaire ces exigences. Ils ne sont malheureusement pas suffisamment efficaces pour nos besoins.

Une publication récente de Spielman [53] introduit les "Superconcentrator Codes", une nouvelle famille de codes ayant les propriétés qui nous intéressent et

1. Nous nous intéressons uniquement aux codes linéaires binaires

qui de plus sont très efficaces. Dans cet article, on trouve un algorithme de correction nécessitant uniquement l'évaluation de $O(n)$ OU-EXCLUSIFs d'un nombre constant de bits. On peut aussi vérifier qu'un mot appartient au code avec le même nombre d'opérations.

Comme, à la connaissance de l'auteur, aucune autre famille de codes plus efficace ayant les propriétés désirées n'existe, nous considérerons dans l'analyse de complexité et lors des preuves que les "Superconcentrator Codes" sont utilisés.

3.5 Preuves à divulgation nulle de COT

Dans le protocole, Alice et Bob effectuent un certain nombre de preuves à divulgation nulle. Rappelons qu'une *preuve à divulgation nulle* est un ensemble d'opérations qui permettent à un prouveur de convaincre un vérificateur que des bits sont reliés par une certaine relation sans révéler aucune autre information au sujet de ces bits. Toutes les preuves du protocole sont faciles à réaliser si les mises en gage utilisées sont des BCX. C'est d'ailleurs pourquoi dans le protocole on utilise ce type de mise en gage.

Aux l'étapes 2 et 7 un des participants doit prouver qu'un ensemble de BCX forme un mot de code. Comme nous l'avons dit précédemment, si le bon code est utilisé, cette opération peut être accomplie en montrant $O(n)$ relations linéaires (OU-EXCLUSIF) d'un nombre constant de bits du mot de code. Ceci nécessite au total $O(n)$ opérations élémentaires sur les BCX.

À l'étape 9, pour $i \in I_2$, Bob doit prouver à Alice que $\boxed{w^i} = w^i_{\boxed{b}}$. Pour chaque position où $w^i_0 = w^i_1$, Bob n'a qu'à ouvrir $\boxed{w^i}$ car dans ce cas w^i ne contient aucune information sur b. Si $w^i_0 = 0$ et $w^i_1 = 1$, on a que $w^i_b = b$ et donc Bob prouve que $\boxed{w^i} \oplus \boxed{b} = 0$. De la même façon, si $w^i_0 = 1$ et $w^i_1 = 0$ Bob prouve que $\boxed{w^i} \oplus \boxed{b} = 1$. Aucune de ces preuves ne révèle d'information au sujet de b. Aux étapes 10 et 11, les preuves se font directement en utilisant les propriétés des BCX.

3.6 Validité de COT

Il existe une foule de façons pour Alice ou Bob d'avoir un comportement déviant lors de l'exécution du protocole. À n'importe quel moment, l'un ou l'autre peut décider de cesser la coopération ou tout simplement utiliser des valeurs erronées lors des transferts inconscients.

Dans cette section, nous allons montrer que le protocole est *correct, honnête* et *privé*.

3.6.1 COT est correct

Si Alice et Bob respectent le protocole à la lettre, il est assez facile de se convaincre que le protocole ne va pas avorter et qu'à la fin Bob sera engagée à $\boxed{c} = a_b$. À travers le protocole, Bob apprendra w_b puis a_b, ce qui lui permettra de s'engager à $\boxed{w} = w_b$ et finalement à $\boxed{c} = a_b$.

Nous avons discuté dans la section 3.4 du fait qu'Bob puisse choisir à l'étape 1 un code avec les bonnes propriétés. Aux étapes 3, 4, 6, 7, 8, 10 et 11, Alice et Bob effectuent des calculs ou communiquent de l'information, ce qu'il réussiront sans peine. Les vérifications des étapes 2, 5, 7, 9, 10 et 11 pourront être réalisées correctement si Alice et Bob ont bien transféré et se sont engagés à ce qu'ils doivent.

3.6.2 COT est honnête

Si Alice respecte le protocole, Bob ne pourra s'engager à $\boxed{c} \neq a_b$. Il est important de noter que nous ne pouvons pas forcer Bob à s'engager à $\boxed{c} = a_b$. Ce n'est d'ailleurs pas un problème car même si on le pouvait, en refusant de l'utiliser par la suite, Bob rendrait cette mise en gage bien inutile. C'est pourquoi l'important est que si Bob s'engage, il le fasse à la bonne valeur.

Voyons pourquoi cela est vrai. Supposons au contraire que Bob ait pu s'engager à $\boxed{c} \neq a_b$. Cela signifie qu'il a prouvé à l'étape 11 que $\boxed{c} = m \cdot \boxed{w}$, mais $\boxed{c} \neq a_b$ et $a_b = m \cdot w_b$ alors $\boxed{w} \neq w_b$. Comme Bob a prouvé que $\boxed{w}$ est un mot de code, le nombre de positions différentes entre $\boxed{w}$ et w_b doit être supérieur à $d > \epsilon n$. Mais à l'étape 9, Alice lui a demandé de prouver que $\boxed{w}$ et w_b sont égaux sur σn positions aléatoirement choisies, ce que Bob n'aurait pu faire, excepté avec probabilité exponentiellement faible en n. D'où la contradiction.

Il est aussi impossible à Alice d'utiliser le protocole de façon à ce que Bob s'engage à $\boxed{c} \neq a_b$ s'il est honnête. Du fait qu'Alice ait prouvé que $\boxed{a_b} = m \cdot \boxed{w_b}$, il est évident que si $\boxed{w} = \boxed{w_b}$, Bob pourra s'engager à $\boxed{c} = a_{\boxed{b}}$. Supposons au contraire que $\boxed{w} \neq \boxed{w_b}$. Comme ce sont deux mots de code, ils doivent avoir au moins $d > \epsilon n$ position différentes. Mais si c'est le cas, le test qui s'effectue aux étapes 5 et 6 ne pourra avoir été réussi. Donc si Alice ou Bob est honnête, à la fin du protocole Bob ne pourra être engagée à $\boxed{c} \neq a_{\boxed{b}}$, excepté avec probabilité exponentiellement faible en n.

3.6.3 COT est privé pour Bob

Montrons que, peu importe le comportement d'Alice, si Bob respecte le protocole, elle n'apprendra rien au sujet de b. Bob transmet de l'information à Alice seulement aux étapes 1, 5, 7, 9 et 11. Aux étapes 1 et 5, l'information transmise est indépendante de b. Aux étapes 7, 9 et 11, par définition de BCX et du fait que les preuves sont à divulgation nulle, Alice n'apprend rien au sujet de b. Le seul moyen qui reste à Alice pour obtenir de l'information consiste à faire en sorte que Bob agisse différemment dépendant de la valeur de b. Elle devra le forcer à refuser si $b = 0$ ($b = 1$) aux étapes 7 ou 9.

Heureusement, cela est impossible. Si Bob ne réussit pas les preuves de l'étape 7 ou 9, c'est que $\boxed{w} \neq \boxed{w_b}$. Comme il a été prouvé à l'étape 2 que $\boxed{w_b}$ est un mot de code, $\boxed{w}$ et $\boxed{w_b}$ doivent avoir au moins t positions différentes, sans quoi l'algorithme de décodage aurait trouvé w_b. Comme $t \in \Omega(n)$, la vérification de l'étape 6 ne pourra avoir été réussie qu'avec probabilité exponentiellement faible. Donc, si la vérification de l'étape 6 réussit, les preuves de l'étape 7 et 8 vont aussi réussir. Notons que la vérification de l'étape 6 a la même probabilité de succès peu importe b, et ce dû au fait que $|I_0| = |I_1|$.

3.6.4 COT est privé pour Alice

Si Alice respecte le protocole, Bob ne pourra rien apprendre au sujet $a_{\bar{b}}$, peu importe sa stratégie et ce, même s'il connaît déjà a_b.

Comme Alice choisit w_0 et w_1 aléatoirement, aucune information sur $a_{\bar{b}}$ ne parvient à Bob avant l'étape 10 où Alice révèle m et prouve que $\boxed{a_{\bar{b}}} = m \cdot \boxed{w_{\bar{b}}}$. Pour en arriver là, Bob doit préalablement passer le test de l'étape 9 et convaincre Alice qu'il est engagé à $\boxed{w} = w_b$. Pour que Bob ait moins que $2^{\sigma n/2}$ candidats pour w_b au moment de son engagement de l'étape 6, il doit avoir obtenu au moins $n/2+3\sigma n/2$ des $n+2\sigma n$ bits de w_b durant les étapes 4 et 5. Ceci est dû au fait qu'il y a $2^{n/2+2\sigma n}$ mots de code possibles, et donc après avoir appris $n/2 + 3\sigma n/2$ bits supplémentaires, $2^{\sigma n/2}$ d'entre eux sont encore possibles. Si c'est le cas, il y aura aussi $2^{\sigma n/2}$ candidats pour $w_{\bar{b}}$ même après avoir appris σn bits supplémentaires à l'étape 8.

Donc, si Bob apprend seulement $n/2 + 3\sigma n/2$ bits de w_b, il aura une probabilité d'au plus $2^{-\sigma n/2}$ de s'engager au bon mot de code à l'étape 7. Comme tout mot de code erroné aura au moins ϵn positions différentes du bon mot, l'étape 9 aurait une chance exponentiellement faible d'être réussie dans ce cas. D'un autre côté, si Bob apprend aussi peu que $n/2 + \sigma n/2$ bits de $w_{\bar{b}}$ il aura une probabilité d'au plus $2^{-\sigma n/2}$ d'avoir une information, même minime, au sujet de $a_{\bar{b}} = m \cdot w_{\bar{b}}$ étant donné que m est aléatoire et ce, par un théorème de Bennett, Brassard et Robert [7].

3.7 COT à l'intérieur d'un groupe (GCOT)

Si Alice et Bob sont engagés à leurs entrées envers tous les participants d'un groupe, il leur est possible d'effectuer GCOT($\boxed{a_0}, \boxed{a_1}, \boxed{b}$)($\boxed{c}$) de façon à ce qu'à la fin du protocole, Bob soit engagé à $\boxed{c}$ envers tous les membres du groupe. Le protocole aura aussi les caractéristiques suivantes:

- *correct*: Si tous les participants respectent le protocole, Bob sera engagée envers tous à $\boxed{c} = a_b$ (via BCX).
- *honnête*: Si au moins un participant respecte le protocole, Bob ne peut être engagée à $\boxed{c} = \bar{a}_b$.
- *privé*: Si Alice respecte le protocole, Bob et les autres participants n'apprendront rien au sujet de $a_{\bar{b}}$, et si Bob respecte le protocole, Alice et les autres participants n'apprendront rien au sujet de b.

Trois modifications au protocole COT sont nécessaires pour le transformer dans un protocole GCOT :

- Le code correcteur C de l'étape 1 doit être accepté par l'ensemble des participants.
- Quand un participant s'engage ou prouve une relation, il le fait envers tous les participants.
- Le sous-ensemble I_2 de l'étape 8 doit être choisi par tous les participants.

Pour choisir I_2, seulement $O(n \log(n))$ bits aléatoires devront être choisis, au coût de $O(Nn \log(n))$ BCs pour chaque participant. Quand plusieurs GCOT sont effectués en parallèle, le même ensemble I_2 peut être utilisé à chaque fois sans perte de sécurité. Ceci permettra une économie de ressources appréciable si les participants préparent I_2 avant d'effectuer les GCOT.

Le protocole de GCOT est très semblable au COT, l'ensemble des participants jouant alternativement le rôle d'Alice et de Bob. Le protocole est *correct, privé* et *honnête* pour les mêmes raisons.

3.8 Complexité de GCOT

La complexité de GCOT dépend du type de BCX et de $\binom{1}{2}$-OT utilisés par les participants. C'est pourquoi nous discuterons de sa complexité en terme des protocoles déjà définis.

GCOT utilise $O(n)$ $\binom{1}{2}$-OT qui sont effectués quand w_0 et w_1 sont transférés. Le protocole nécessite $O(n)$ opérations sur les BCX de la part d'Alice et Bob avec chaque participant pour s'engager respectivement au sujet de w_0, w_1 et w et pour effectuer les différentes preuves. Alice et Bob effectuent donc $O(Nn)$ opérations sur les BCX et les autres participants $O(n)$. Nous n'avons pas tenu compte ici des tirages de $O(n \log n)$ bits aléatoires qui doivent être faits par les participants. Comme nous l'avons dit précédemment, ce tirage peut être effectué une seule fois pour plusieurs GCOT. Nous en tiendrons compte plus tard.

Si les BC disponibles ne sont pas des BCX, alors la technique décrite dans le chapitre 2 permettant leur construction devra être utilisée. Dans ce cas, pour réaliser un GCOT, $O(Nn^2)$ opérations sur les BC seront effectuées par Alice et Bob et $O(n^2)$ par les autres participants. Enfin, si seulement $\binom{1}{2}$-OT est disponible les BC devront être implantés à partir de cette procédure au coût de $O(n)$ $\binom{1}{2}$-OT par BC et pour une complexité totale de $O(Nn^3)$ $\binom{1}{2}$-OT pour Alice et Bob, et de $O(n^3)$ $\binom{1}{2}$-OT pour les autres.

Chapitre 4

Calculs Multi-Parties sur des données Privées

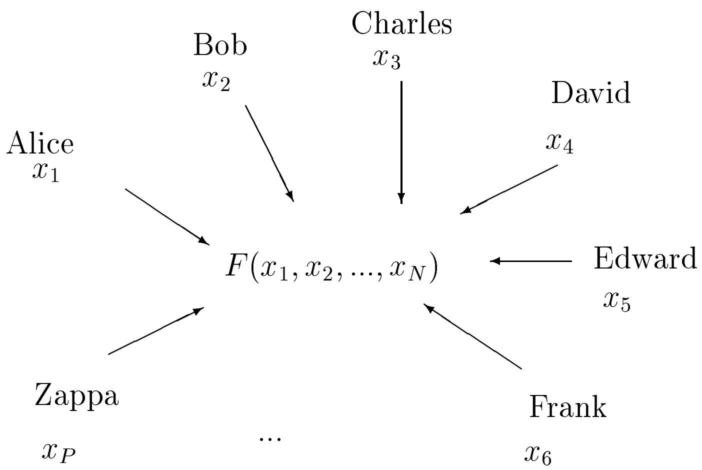

FIG. 4.1 - *Calculs multi-parties*

Dans ce chapitre, nous présentons un protocole de *calculs multi-parties sur données privées* conceptuellement simple et efficace. L'introduction de ce mémoire présente un survol des travaux déjà réalisés dans ce domaine. Rappelons que ce genre de protocole permet l'implantation du vote secret, du poker à l'aveugle ("Mental Poker") [49, 21], des statistiques confidentielles et de tout autre protocole nécessitant des opérations sur des données secrètes réparties entre deux ou plusieurs individus [32].

Dans un protocole multi-parties, N participants ayant chacun une entrée secrète x_i désirent apprendre $y = F(x_1, ..., x_N)$. Rappelons quatre caractéristiques propres à ce genre de protocole:

- On dit que le protocole est *correct* si N participants qui respectent le protocole réussissent à apprendre y.
- On dit que le protocole est *privé* si la seule information qu'un groupe de participants G peut apprendre au sujet des x_i des autres participants est y et ce qui peut en être logiquement déduit sachant F et les x_i tel que $i \in G$.
- Le protocole est *honnête* si un participant sait quand il n'apprend pas un résultat y valable. On dit que y est un résultat valable s'il existe $\{x'_1, ..., x'_N\}$ tel que $y = F(x'_1, ..., x'_N)$ et $\forall i,\ x'_i = x_i$ si le participant i est honnête.
- Le protocole est *juste* si le fait qu'un participant apprenne de l'information au sujet de y à travers le protocole implique que tous les participants apprennent y.

Le reste du chapitre sera consacré à la présentation d'un protocole d'évaluation multi-parties de fonctions sur données privées (PMPC) correct, privé, juste et honnête basé sur l'existence d'un transfert inconscient ($\binom{1}{2}$-OT) entre chaque paire de participants et l'existence d'un canal de diffusion fiable les reliant tous. Aucune supposition ne sera faite quant à l'honnêteté des participants. Le protocole reposera aussi conceptuellement sur les BCX qui peuvent, comme nous l'avons vu dans le chapitre 2, être réalisés à l'aide du $\binom{1}{2}$-OT.

La procédure de transfert inconscient certifié (GCOT) décrite au chapitre précédent et réalisée à partir d'$\binom{1}{2}$-OT et des BCX sera la pierre angulaire du protocole

PMPC. Rappelons que les BCX peuvent (entre autres) être implantés à l'aide des BC qui eux-mêmes peuvent l'être à partir d'$\binom{1}{2}$-OT. De plus, notons que certaines implantations de $\binom{1}{2}$-OT permettent qu'il soit utilisé comme canal privé. Si par contre le $\binom{1}{2}$-OT utilisé n'a pas cette propriété, chaque paire de participants devra en plus disposer d'un canal privé ou d'une chaîne aléatoire secrète commune.

Le protocole PMPC n'est pas robuste, c'est-à-dire qu'un participant peut faire avorter le protocole même si celui-ci est déjà bien entamé. Ceci est inévitable étant donné qu'aucune hypothèse n'est faite quant à l'honnêteté des participants. Pour garantir que le protocole soit juste, chaque participant devra coopérer à chacune des étapes du protocole. Si un groupe de participants pouvait continuer l'évaluation du circuit sans la coopération de certains autres participants (même un seul), ceci leur permettrait de tricher le protocole en évaluant la fonction avec les données de tous sans partager le résultat, ce qui n'est pas permis dans un protocole juste. Ce n'est donc pas un inconvénient propre au protocole présenté ici mais bien une caractéristique intrinsèque de tous les protocoles de ce type.

Le protocole PMPC est constitué de trois phases. Premièrement, l'initialisation où les participants s'entendent sur le circuit et les paramètres du protocole. C'est aussi lors de l'initialisation que les participants vont préparer leurs bits privés de façon à pouvoir effectuer le calcul. Pendant la deuxième phase du protocole, les participants évaluent la fonction proprement dite. Le circuit calculant la fonction sera évalué une porte à la fois par un travail conjoint de tous les participants. La dernière étape est celle de la révélation graduelle du résultat. Cette étape a pour but d'éviter que certains participants n'apprennent le résultat pendant que d'autres restent dans l'ignorance. C'est surtout à ce moment que les participants se soucient du fait que le protocole soit juste.

4.1 Initialisation

L'initialisation est divisée en deux parties. Premièrement, les participants vont s'entendre sur un circuit calculant la fonction F ainsi que sur tous les paramètres des sous-protocoles. Les paramètres sont liés au type de BCX et de $\binom{1}{2}$-OT utilisés ainsi qu'au protocole COT où un code et plusieurs constantes doivent être choisis de façon consistante.

Dans un deuxième temps, les participants vont distribuer leurs bits d'entrée en utilisant des BCX. Ces bits distribués seront appelés DBC ("Distributed Bits Commitment" ou *mise en gage distribuée*) et c'est sur eux que seront effectuées les opérations permettant l'évaluation du circuit.

4.1.1 Mise en gage distribuée (DBC)

Pour chacun de ses bits d'entrée, chaque participant devra, avec la coopération des autres, construire une mise en gage distribuée (DBC). Comme leurs noms l'indiquent, les DBC sont des mises en gage dont la valeur est distribuée entre tous les participants. Un DBC est le OU-EXCLUSIF de N valeurs connues chacune par un participant et sur laquelle il est engagé envers tous par des BCX. Rappelons que les BCX sont jumelables. A partir de ce moment, à moins d'avis contraire, lorsqu'un participant s'engage à une valeur, cela signifie qu'il s'engage envers tous les participants en utilisant des BCX égaux.

Pour camoufler b avec un DBC B, un participant p demande à chaque participant i de s'engager à un bit aléatoire b_i et de l'ouvrir à p et uniquement à lui. Finalement p s'engagera à $\boxed{b_p} = b \oplus (\bigoplus_{i \neq p} b_i)$.

Protocole: CRÉER-DBC

Entrées	**Sorties**
$p : b$ BIT	B : DBC
Relations	**Propriétés**
$B = b$	*privé*

1: $\forall i \in P$ tel que $i \neq p$, i s'engage à $\boxed{b_i} \in_R \{0, 1\}$.

2: $\forall i \in P$ tel que $i \neq p$, i ouvre $\boxed{b_i}$ à p et uniquement à lui.

3: p s'engage à $\boxed{b_p} = b \oplus \bigoplus_{i \neq p} b_i$

4: $B = \{\boxed{b_1}, \boxed{b_2}, ..., \boxed{b_N}\}$

Toutes les valeurs intermédiaires lors de l'évaluation du circuit seront des DBC. Pour avoir de l'information au sujet de la valeur d'un DBC, il faut connaître la valeur de tous les bits qui le constituent. Les DBC restent donc privés même contre une coalition de $N - 1$ participants.

On peut considérer les DBC comme des exemples de "secret sharing" parfait où tous les participants doivent coopérer pour récupérer la valeur du bit distribué. C'est d'ailleurs ce qui sera fait lors de la dernière étape du protocole pour permettre à tous d'apprendre le résultat de la fonction.

4.2 Évaluation

La deuxième étape du protocole PMPC est l'évaluation du circuit. Le circuit composé de NOT et AND sera évalué une porte à la fois. Chaque porte aura comme entrée et comme sortie des DBC. L'évaluation d'une porte sera réalisée de façon à ne pas accroître l'information d'une éventuelle coalition malhonnête.

4.2.1 Évaluation du NOT

L'évaluation du NOT ne pose pas de problème particulier. On désire à partir d'un DBC A camouflant a construire un DBC B camouflant b tel que $b = \bar{a}$. Le protocole devra être:

- *honnête:* $B \neq A$ et tous les participants en seront convaincus.
- *privé:* Aucune coalition de moins de N participants ne peut obtenir d'information supplémentaire au sujet de A (ainsi que B) par le protocole.

Pour construire le DBC B camouflant $b = \bar{a}$, chaque participant s'engagera à une valeur égale à sa part de A, sauf un participant appelé ξ qui, lui, s'engagera à la valeur opposée. Chacun doit évidemment prouver qu'il s'est engagé à la bonne valeur. Comme la valeur d'un DBC est le OU-EXCLUSIF de toutes les parts le constituant, le DBC ainsi construit aura la valeur opposée au DBC original.

Protocole: NOT

Entrées	**Sorties**
$A = \{\boxed{a_1}, ..., \boxed{a_N}\}$:DBC	$B = \{\boxed{b_1}, ..., \boxed{b_N}\}$: DBC
Relations	**Propriétés**
$A \neq B$	*privé, honnête*

1: $\forall i \in P$ tel que $i \neq \xi$, i s'engage à $\boxed{b_i} = \boxed{a_i}$ et le prouve.

2: Le participant ξ s'engage à $\boxed{b_\xi} \neq \boxed{a_\xi}$ et le prouve.

3: $B = \{\boxed{b_1}, \boxed{b_2}, ..., \boxed{b_N}\}$

Il est facile ici de voir que le protocole est privé et honnête.

4.2.2 Évaluation du AND

L'évaluation du AND est la partie la plus difficile d'un protocole PMPC. C'est pour évaluer cette porte asymétrique que le GCOT et les DBC sont nécessaires.

Pour simplifier la présentation de l'évaluation du AND, nous introduirons un protocole de AND partiel (PAND) réalisé entre deux participants sous la supervision du groupe. Au début du protocole PAND, Alice est engagée à $\boxed{a}$ et Bob à $\boxed{b}$. Après PAND($\boxed{a}$, $\boxed{b}$) Alice sera engagée à $\boxed{c_A}$ et Bob à $\boxed{c_B}$ tels que $\boxed{c_A} \oplus \boxed{c_B} = \boxed{a} \wedge \boxed{b}$. De plus le PAND sera:

- *honnête:* $\boxed{a} \wedge \boxed{b} = \boxed{c_A} \oplus \boxed{c_B}$ et tous les participants en seront convaincus.
- *privé:* peu importe son comportement Alice n'apprend rien sur la valeur de b et de façon similaire Bob n'apprend rien sur la valeur de a.

L'implantation du PAND est assez simple.

Protocole: PAND

Entrées	**Sorties**
Alice : $\boxed{a}$:BCX	Alice : $\boxed{c_A}$:BCX
Bob : $\boxed{b}$:BCX	Bob : $\boxed{c_B}$:BCX
Relations	**Propriétés**
$a \wedge b = c_a \oplus c_b$	*privé, honnête*

1: Alice s'engage à $\boxed{c_A} \in_R \{0, 1\}$.

2: Alice s'engage à $\boxed{t_0} = \boxed{c_A}$ et à $\boxed{t_1} = \boxed{c_A} \oplus \boxed{a}$ et le prouve.

3: Alice et Bob effectuent $GCOT(\boxed{t_0}, \boxed{t_1}, \boxed{b})(\boxed{c_B})$.

Voyons pourquoi le protocole est honnête. Comme GCOT est correct, on a que si $b = 0$ alors $c_B = t_0 = c_A \Rightarrow c_A \oplus c_B = 0 = a \wedge b$ et si $b = 1$ alors $c_B = t_1 = c_A \oplus a \Rightarrow c_A \oplus c_B = a = a \wedge b$. Le protocole est privé du fait que GCOT et BCX le sont.

Nous sommes maintenant en mesure de réaliser le protocole AND qui à partir de deux DBC A et B construira le DBC C tel que $A \wedge B = C$. Le AND sera privé et honnête de la même façon que le NOT.

Pour construire $C =$AND(A, B), toutes les paires ordonnées de participants exécuteront un PAND de leur part de A et B, puis chacun évaluera le OU-EXCLUSIF de toutes les valeurs auxquelles il s'est engagé en sortie des PAND et s'y engagera. Toute ces BCX formeront C.

Protocole: AND

Entrées	**Sorties**
$A = \{\boxed{a_1}, ..., \boxed{a_N}\}$: DBC	: C :DBC
$B = \{\boxed{b_1}, ..., \boxed{b_N}\}$: DBC	
Relations	**Propriétés**
$A \wedge B = C$	*privé, honnête*

1: $\forall i, j \in P$ avec $i \neq j$, i et j effectuent PAND$(\boxed{a_i}, \boxed{b_j}) = (\boxed{g_{ij}}, \boxed{d_{ji}})$

2: $\forall i \in P$, i s'engage à $\boxed{c_i} = \bigoplus_{j=1}^{N}(\boxed{g_{ij}} \oplus \boxed{d_{ji}})$

3: $C = \{\boxed{c_1}, \boxed{c_2}, ..., \boxed{c_N}\}$

Du fait que le PAND est honnête et privé ainsi qu'en regard de l'équation

$$\{\bigoplus_{i=1}^{N} a_i\} \wedge \{\bigoplus_{i=1}^{N} b_i\} = \bigoplus_{i=1,j=1}^{N} (a_i \wedge b_j),$$

il est facile de se convaincre que le AND est lui aussi honnête et privé.

L'évaluation d'un AND nécessitera que chaque participant s'implique de façon active dans $O(N)$ PAND et de façon passive dans les $O(N^2)$ autres. La complexité du PAND étant essentiellement égale à celle du COT, on obtient que les participants seront activement impliqués dans $O(N)$ COT et passivement dans $O(N^2)$. De plus, nous devons tenir compte ici des $O(n \log n)$ BC nécessaires pour le choix du sous-ensemble aléatoire qui sera utilisé dans les $O(N^2)$ COT. En combinant ces trois mesures, on obtient que chaque participant devra effectuer $O(Nn)$ $\binom{1}{2}$-OT , $O(N^2 n)$ opérations sur des BCX et $O(n \log n)$ BC.

4.3 Révélation graduelle du résultat

Nous sommes maintenant au point où l'évaluation du circuit est terminée. Mais comme les bits résultants sont camouflés dans des DBC, personne n'a encore appris quoi que ce soit au sujet de y. Nous désirons que les participants apprennent cette valeur de façon juste, c'est-à-dire que si un participant apprend le résultat, tous l'apprennent. Cette étape est plus difficile qu'elle n'y paraît à première vue.

Si tous les participants ouvrent leur part du DBC simultanément, le protocole sera juste. En pratique, il est difficile d'accomplir une telle tâche, une coalition de participants malhonnêtes pouvant toujours se retirer au moment où ils apprennent les parts des autres participants et ainsi être seuls à apprendre le résultat de la fonction. Pour éviter qu'une telle chose ne se produise, le résultat sera réparti en un nombre polynomial de bits qui seront ouverts les uns après les autres alternativement par chaque participant. Chaque bit donnera peu d'information au sujet du résultat mais peu à peu l'incertitude se dissipera et à la fin tous les participants auront appris la valeur de la fonction avec certitude. À aucune étape, un groupe de participants n'aura d'avantage important sur les autres. Un groupe pourra seulement avoir un avantage de $1/Poly$. Cet avantage peut être considéré par certains comme non négligeable. Cela est malheureusement inévitable, car si à chaque étape un groupe de participants ne peut avoir qu'un avantage exponentiellement faible sur les autres, il faudra une nombre exponentiel d'étapes pour que tous les participants apprennent le résultat de la fonction.

Il existe différentes techniques permettant de rendre un protocole comme PMPC juste. On trouve dans [42] et [20] des solutions intéressantes à ce problème. Ces solutions sont évidemment privées, honnêtes et justes selon des définitions consistantes avec celles du protocole PMPC. Le lecteur est invité à y référer pour plus d'information. La solution naturelle consiste à transformer le circuit S qui calcule F en un circuit S' qui aura pour chaque bit de sortie de S, $O(Poly(n))$ bits de sortie qui ensemble lui seront équivalents. Ces bits de sortie devront être tels que leur révélation séquentielle divulgue lentement le résultat de F à tous.

4.4 Validité

Dans cette section, les propriétés du protocole d'évaluation de circuit multi-parties sur des données secrètes seront étudiées. Nous montrerons que le protocole est correct, privé, honnête et juste tel que défini au début du chapitre. Les arguments ne feront pas appel à des simulateurs comme c'est souvent le cas pour ce genre de protocole. Une approche plus directe sera utilisée. Les propriétés du PMPC seront déduites des propriétés des sous-protocoles qui sont nécessaires à sa réalisation. C'est cette approche qui a aussi été utilisée avec les sous-protocoles eux-mêmes. Ce processus récursif s'arrête avec les procédures de base, soit la mise en gage et le transfert inconscient, décrits dans le chapitre 2. Leur existence et leurs propriétés sont supposées et c'est sur eux que s'appuient tous les protocoles de ce mémoire. Rappelons que l'existence d'un canal de diffusion fiable est aussi supposée.

4.4.1 Le protocole est *correct*

Rappelons que le protocole est correct si en le respectant les P participants arrivent bien à apprendre $y = F(x_1, ..., x_N)$. Les opérations NOT et AND sont correctes et donc, si les participants suivent le protocole et évaluent le circuit calculant la fonction, il leur est garanti que ce qu'ils calculent est bien $F(x_1, ..., x_N)$. Il suffit donc que le protocole de révélation utilisé soit correct pour que le protocole PMPC le soit aussi.

4.4.2 Le protocole est *privé*

Rappelons que le protocole est privé si la seule information qu'un groupe de participants G peut obtenir au sujet des x_i des autres participants est y et ce qui peut en être logiquement déduit connaissant F et les x_i tels que $i \in G$.

Le protocole est privé car avant la révélation, aucune information ne va transpirer. En effet, aucune information ne peut être connue au sujet des bits d'entrée par une coalition de moins de N participants du fait qu'ils sont camouflés dans des DBC. De plus, ni le NOT ni le AND ne pourront être utilisés pour obtenir de l'information du fait que ces protocoles sont eux aussi privés. Il suffit donc que la phase de révélation ne dévoile que le résultat pour que le protocole soit privé.

4.4.3 Le protocole est *honnête*

Rappelons que le protocole est honnête si un participant sait quand il n'apprend pas le résultat de la fonction.

Ce qui est important ici est qu'il soit impossible à un groupe de participants de faire croire à un ou plusieurs participants que la fonction a été évaluée de façon correcte si ce n'est pas le cas. Il ne faut pas oublier que les participants peuvent s'engager à une valeur de leur choix au début du protocole. Ce sera donc la seule façon à leur disposition pour influencer le résultat de l'évaluation.

Une fois un DBC construit, il est impossible d'en changer la valeur. De plus, le AND et le NOT sont honnêtes. Il suffit donc encore une fois que la phase de révélation soit elle-même honnête pour que PMPC le soit.

4.4.4 Le protocole est *juste*

Rappelons que le protocole est juste si le fait qu'un participant apprenne $F(x_1, ..., x_N)$ implique que tous l'apprennent.

Cette caractéristique est surtout liée à la révélation du résultat. En fait, comme nous l'avons vu plus tôt, aucune information quelle qu'elle soit n'est révélée au sujet du résultat avant l'étape de révélation. Si le protocole s'arrête avant cette étape, aucun groupe de moins de N participants ne pourra apprendre le résultat de la fonction, sauf bien entendu si le résultat est parfaitement défini par leurs entrées. Donc le protocole sera juste si l'étape de révélation est faite de façon correcte.

4.5 Complexité

Le transfert inconscient ($\binom{1}{2}$-OT) est la procédure de base sur laquelle repose PMPC. En pratique, il est parfois plus facile d'implanter BC et même BCX que $\binom{1}{2}$-OT. Nous donnerons donc une analyse de complexité pour le cas où seulement $\binom{1}{2}$-OT est directement disponible, le cas où $\binom{1}{2}$-OT et BC sont disponibles et le cas où $\binom{1}{2}$-OT et BCX sont disponibles.

Il est facile de voir que l'évaluation du circuit F composé de m portes nécessitera: CREATION, NOT et AND confondus, $O(m)$ opération sur des DBC. L'opération la plus coûteuse sur les DBC est le AND et donc la complexité du protocole est égale à l'évaluation de $O(m)$ AND.

Voici un tableau récapitulatif des coûts de chacun des sous-protocoles:

Procédure	$\binom{1}{2}$-OT	$\binom{1}{2}$-OT	BC	$\binom{1}{2}$-OT	BCX
BC	$O(n)$		1		1
BCX	$O(n^2)$		$O(n)$		1
GCOT actif	$O(Nn^3)$	$O(n)$	$O(Nn^2)$	$O(n)$	$O(Nn)$
GCOT passif	$O(n^3)$		$O(n^2)$		$O(n)$
PAND passif	$O(Nn^3)$	$O(n)$	$O(Nn^2)$	$O(n)$	$O(Nn)$
PAND actif	$O(n^3)$		$O(n^2)$		$O(n)$
AND	$O(N^2n^3)$	$O(Nn)$	$O(N^2n^2)$	$O(Nn)$	$O(N^2n\log n)$
PMPC	$O(mN^2n^3)$	$O(mNn)$	$O(mN^2n^2)$	$O(mNn)$	$O(m(N^2n + n\log n))$

FIG. 4.2 - *Complexité*

Chapitre 5

Conclusion

Durant le siècle dernier, l'homme a pris pleinement conscience de son pouvoir sur la matière. La vaste étendue de ses constructions en témoigne. Nous modifions notre environnement d'une façon de plus en plus profonde et il ne semble pas y avoir de limite à notre capacité créatrice. Ce n'est que plus récemment, particulièrement avec l'avènement de l'informatique, que nous avons pris conscience de l'importance d'une entité plus subtile mais tout aussi réelle: l'information.

L'information occupe de plus en plus de place dans nos vies. La fulgurante augmentation de notre capacité de communication associée à notre capacité de stockage, de gestion et de manipulation de cette matière malléable nous ouvre de nouveaux horizons. Nous vivons la révolution informatique.

La révolution industrielle a modifié nos vies d'une façon si dramatique qu'il nous est maintenant difficile d'en faire la pleine mesure. La révolution informatique pourrait s'avérer tout aussi marquante. Il est de plus en plus évident que l'information mène le monde moderne. Il faut prendre des décisions rapides et ceux qui contrôlent l'information contrôlent par extension les décisions que nous prenons.

Une savante manipulation de l'information peut changer le cours d'une guerre. Elle peut permettre à des empires de s'écrouler et à d'autres de naître. Sa judicieuse utilisation nous permettra de modeler notre avenir et ainsi, souhaitons-le, de mieux vivre.

Il faudra que la société s'adapte. La quantité de données concernant les individus qui se trouve stockée électroniquement est impressionnante. Si un quelconque être malicieux rassemblait toute l'information qui nous est associée, il connaîtrait sur notre vie des détails que même nos proches ignorent. Il aurait probablement accès à notre adresse, notre revenu, nos dépenses, nos voyages, les restaurants que nous fréquentons, à qui nous téléphonons, à quelle heure et combien de temps, quels sont les films que nous avons emprunter à notre club vidéo, notre dossier fiscal, judiciaire, médical, scolaire ou bancaire, si nous respectons les règles de stationnement, si nous avons eu des contraventions, quels sont les journaux, revues et périodiques auxquels nous sommes abonnés, de quel parti politique nous sommes membre, etc. Si de plus nous sommes connectés à un réseau informatique à notre travail, tout notre courrier, nos fichiers et notre horaire pourraient en être déduits.

Bref, aujourd'hui, pratiquement tout ce qui nous concerne est stocké quelque part et tout porte a croire que cela va s'accentuer. L'énumération qui précède est bien courte en comparaison avec ce que pourrait être la réalité. Et ce, sans compter les recoupements qui pourraient aussi être faits entre l'information concernant plusieurs individus et des techniques d'analyse statistique complexes qui pourraient permettre de percer encore plus profondément notre intimité.

Heureusement, il ne semble pas possible à un individu d'accéder à toute cette information. Il n'en demeure pas moins qu'il est assez facile d'en obtenir une bonne partie. Cela a d'ailleurs parfois des conséquences anodines comme la réception de publicité personnalisée ou d'autre forme de sollicitation. Mais jusqu'où cela peut-il aller? Les gens s'interrogent déjà sur ces problèmes. Le cinéma est en général un bon indicateur social et les films comme: “The Conversation”, “Brazil”, “1984”, “Snickers”, “Johnny Mnemonique”, “The Net” témoignent de l’attention portée par la population à cette réalité.

D’un autre côté, il faut aussi réaliser que notre sécurité est parfois liée à la capacité de la société d’accéder à cette information. Nous désirons tous que les criminels dangereux puissent être repérés et mis hors d’état de nuire. Nous voudrions en fait pouvoir les repérer, prévoir et prévenir leurs actions. Mais si cela était possible, il serait insupportable qu’à chaque fois que nous dépassions la limite de vitesse légale en voiture, une contravention nous soit automatiquement facturée. Les lois et la capacité d’intervention des instances policières devront s’harmoniser avec notre capacité grandissante à repérer les contrevenants. Beaucoup d’interdictions dans nos sociétés ont pour but d’éviter les abus. Le contrôle total des individus n’est pas l’objectif à atteindre.

Comment la société peut-elle à la fois prévenir les actions néfastes de certains individus et se garder d’intervenir dans notre vie privée? Comme nous l’avons vu dans les chapitres précédents, la cryptologie apporte des solutions qui vont bien plus loin que la communication secrète. Les preuves à divulgation nulle et le calcul multi-parties pourront permettre à des organismes un traitement de l’information sans y avoir directement accès. Un individu accusé de meurtre pourrait, par exemple, prouver qu’il n’était pas sur les lieux du crime, sans révéler l’endroit où il se trouvait. De façon plus réaliste, on peut imaginer un dossier médical qui répond uniquement aux questions posées par un médecin en service et dûment autorisé. Il sera aussi possible d’effectuer des paiements dans l’anonymat et quand même recevoir la marchandise. Aussi, on pourra faire des statistiques sans poser de questions indiscrètes. En fait, une foule d’opérations qui nécessitent aujourd’hui l’accumulation d’informations pourront être réalisées de façon anonyme et fiable.

Dans un monde idéal, un certain nombre d’organismes, suffisamment large pour satisfaire toute la population et suffisamment restreint pour être pratique, pourrait gérer toute l'information qui nous concerne. Chez nous, ces organismes pourraient par exemple être les différents partis politiques, le gouvernement, les banques, les juges, les médecins, une association de protection des consommateurs, l'église et quelques autres. À eux tous, ils pourraient répondre aux requêtes nous concernant, mais isolés, ils ne pourraient rien manigancer. Ces institutions ne pourraient servir leurs intérêts aux dépends des autres mais surtout, il faudrait infiltrer simultanément tous ces organismes pour accéder à l'information qu'ils possèdent ensemble, ce qui est plus difficile que d'en infiltrer une seule, surtout pour leurs employés.

Il suffira d'avoir confiance en une seule de ces institutions pour dormir tranquille. L'important, c'est que l'information existe et puisse éventuellement être utilisée si cela est nécessaire, justifié et autorisé.

L'implantation de la cryptologie à grande échelle se heurte à des barrières techniques. Heureusement, ces difficultés peuvent être surmontées assez facilement. Le protocole PMPC présenté dans ce mémoire est d'ailleurs assez efficace et général pour répondre à des besoins réels. Il serait concevable pour un groupe d'institutions comme celui cité plus haut d'accumuler dans des DBC de l'information au sujet de chaque individu. Ils pourraient alors d'un commun accord effectuer les opérations nécessaires au bon fonctionnement de la société.

L'explosion démographique à l'intérieur du réseau **Internet** et l'accès plus facile à des ressources cryptographiques permettent d'espérer des changements majeurs dans le traitement de l'information. Nos notions de justice, de tolérance et de propriété devront s'ajuster à tous ces changements, mais tout cela se fera je l'espère dans le respect et le renforcement de la démocratie et de notre intimité.

Bibliographie

[1] D. Beaver. Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2), 1991.

[2] M. Bellare et S. Micali. Non-interactive oblivious transfer and applications. Dans G. Brassard, éditeur, *Proc. CRYPTO 89*, pages 547–559. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[3] M. Ben-Or, S. Goldwasser, et A. Wigderson. Completeness theorems for fault-tolerant distributed computing. Dans *Proc. 20th ACM Symp. on Theory of Computing*, pages 1–10, Chicago, 1988. ACM.

[4] C. H. Bennett et G. Brassard. An update on quantum cryptography. Dans G. R. Blakley et D. C. Chaum, éditeurs, *Proc. CRYPTO 84*, pages 475–480. Springer-Verlag, 1985. Lecture Notes in Computer Science No. 196.

[5] C. H. Bennett, G. Brassard, S. Breidbard, et S. Wiesner. Quantum cryptography, or unforgeable subway tokens. Dans R. L. Rivest, A. Sherman, et D. Chaum, éditeurs, *Proc. CRYPTO 82*, pages 267–275, New York, 1983. Plenum Press.

[6] C. H. Bennett, G. Brassard, C. Crépeau, et M.-H. Skubiszewska. Practical quantum oblivious transfer. Dans J. Feigenbaum, éditeur, *Proc. CRYPTO 91*, pages 351–366. Springer-Verlag, 1992. Lecture Notes in Computer Science No. 576.

[7] C. H. Bennett, G. Brassard, et J.-M. Robert. Privacy amplification by public discussion. *SIAM J. Computing*, 17(2):210–229, Avr. 1988.

[8] M. Blum. Coin flipping by telephone. Dans *Proc. IEEE Spring COMPCOM*, pages 133–137. IEEE, 1982.

[9] G. Brassard. *Cryptologie Contemporaine*. Masson, 1993.

[10] G. Brassard. Cryptology column — quantum cryptography: A bibliography. *Sigact News*, 24(3), 1993.

[11] G. Brassard. A bibliography of quantum cryptography. **http://www.iro.umontreal.ca/people/crepeau/Biblio-QC.html**, Laboratoire d'Informatique Théorique et Quantique, 1995.

[12] G. Brassard, D. Chaum, et C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[13] G. Brassard et C. Crépeau. Quantum bit commitment and coin tossing protocols. Dans A. Menezes et S. A. Vanstone, éditeurs, *Proc. CRYPTO 90*, pages 49–61. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 537.

[14] G. Brassard, C. Crépeau, et J.-M. Robert. All-or-nothing disclosure of secrets. Dans A. Odlyzko, éditeur, *Proc. CRYPTO 86*, pages 234–238. Springer-Verlag, 1987. Lecture Notes in Computer Science No. 263.

[15] G. Brassard, C. Crépeau, R. Jozsa, et D. Langlois. A quantum bit commitment scheme provably unbreakable by both parties. Dans *Proc. 34th IEEE Symp. on Foundations of Comp. Science*, pages 362–371. IEEE, 1992.

[16] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–88, 1981.

[17] D. Chaum. The spymasters double-agent problem: Multiparty computations secure unconditionally from minorities and cryptographically from majorities. Dans G. Brassard, éditeur, *Proc. CRYPTO 89*, pages 591–603. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[18] D. Chaum, C. Crépeau, et I. Damgård. Multi-party unconditionally secure protocols. Dans *Proc.* 20*th ACM Symp. on Theory of Computing*, Chicago, 1988. ACM.

[19] D. Chaum, I. Damgård, et J. van de Graaf. Multiparty computations ensuring privacy of each party's input and correctness of the result. Dans *Proc. CRYPTO 87*. Springer-Verlag, 1988.

[20] R. Cleve. Controlled gradual disclosure schemes for random bits and their applications. Dans G. Brassard, éditeur, *Proc. CRYPTO 89*, pages 573–588. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[21] C. Crépeau. A secure poker protocol that minimizes the effect of player coalitions. Dans H. C. Williams, éditeur, *Proc. CRYPTO 85*, pages 73–86. Springer-Verlag, 1986. Lecture Notes in Computer Science No. 218.

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

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

[24] C. Crépeau. Verifiable disclosure of secrets and applications. Dans C. Pomerance, éditeur, *Proc. EUROCRYPT 89*, pages 181–191. Springer-Verlag, 1990.

[25] C. Crépeau. Cryptographic protocols based on noisy channel. En préparation, 1995.

[26] C. Crépeau et J. Kilian. Weakening security assumptions and oblivious transfer. Dans S. Goldwasser, éditeur, *Proc. CRYPTO 88*, pages 2–7. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 403.

[27] C. Crépeau et L. Salvail. Quantum oblivious mutual identification. Dans *Proc. EUROCRYPT 95*, pages 133–147. Springer-Verlag, 1995.

[28] I. Damgård, T. P. Pedersen, et B. Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. Dans D. R. Stinson, éditeur, *Proc. CRYPTO 93*, pages 250–265. Springer, 1994. Lecture Notes in Computer Science No. 773.

[29] W. Diffie. *The first ten years of public-key cryptography.* 1992. dans [52].

[30] W. Diffie et M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, Nov. 1976.

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

[32] M. Franklin. *Complexity and Security of Distributed Protocols.* PhD thesis, Computer Science Department, Columbia University, New York, 1993.

[33] G.D.Forney. *Concatenated Codes.* M.I.T. Press, 1966.

[34] O. Goldreich, S. Micali, et A. Wigderson. How to play any mental game, or: A completeness theorem for protocols with honest majority. Dans *Proc. 19th ACM Symp. on Theory of Computing*, pages 218–229. ACM, 1987.

[35] O. Goldreich et R. Vainish. How to solve any protocol problem - an efficiency improvement. Dans C. Pomerance, éditeur, *Proc. CRYPTO 87*, pages 73–86. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 293.

[36] S. Goldwasser et L. Levin. Fair computation of general functions in presence of immoral majority. Dans A. Menezes et S. A. Vanstone, éditeurs,

Proc. CRYPTO 90, pages 77–93. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 537.

[37] S. Goldwasser et S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, Avr. 1984.

[38] S. Goldwasser, S. Micali, et C. Rackoff. The knowledge complexity of interactive proof-systems. Dans *Proc. 17th ACM Symp. on Theory of Computing*, pages 291–304, Providence, 1985. ACM.

[39] J. Kilian. Founding cryptography on oblivious transfer. Dans *Proc. 20th ACM Symp. on Theory of Computing*, pages 20–31, Chicago, 1988. ACM.

[40] J. Kilian. *Uses of Randomness in Algorithms and Protocols*, chapitre 3. MIT Press, 1990.

[41] J. Kilian. On the complexity of bounded-interaction and noninteractive zero-knowledge proofs. Dans *Proc. 26th ACM Symp. on Theory of Computing*, pages 466–477, Santa Fe, 1994. IEEE.

[42] M. Luby, S. Micali, et C. Rackoff. How to simultaneously exchange a secret bit by flipping a symmetrically biased coin. Dans *Proc. 24th IEEE Symp. on Foundations of Comp. Science*, pages 11–22, Tucson, 1983. IEEE.

[43] F. J. MacWilliams et N. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1977.

[44] M. Naor. Bit commitment using pseudo-randomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[45] M. Naor, R. Ostrovsky, R. Venkatesan, et M. Yung. Perfect zero-knowledge arguments for np can be based on general complexity assumptions. Dans E. F. Brickell, éditeur, *Proc. CRYPTO 92*, pages 196–214. Springer-Verlag, 1992. Lecture Notes in Computer Science No. 740.

[46] M. Rabin. How to exchange secrets by oblivious transfer. Rapport Technique TR-81, Harvard Aiken Computation Laboratory, 1981.

[47] T. Rabin et M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. Dans *STOC89*, pages 73–85, 1989.

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

[49] A. Shamir, R. L. Rivest, et L. M. Adleman. Mental poker. Dans D. Klarner, éditeur, *The Mathematical Gardner*, pages 37–43. Wadsworth, Belmont, California, 1981.

[50] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:623–656, 1948.

[51] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:657–715, 1949.

[52] G. Simmons, éditeur. *Contemporary Cryptology: The science of Information Integrity.* IEEE Press, 1992.

[53] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. Dans *Proc. 27th ACM Symp. on Theory of Computing*, pages 388–498. ACM, 1995.

[54] S. Wiesner. Conjugate coding. *Sigact News*, 15(1):78–88, 1983.

[55] A. C.-C. Yao. Theory and application of trapdoor functions. Dans *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 80–91, Chicago, 1982. IEEE.

[56] A. C.-C. Yao. Security of quantum protocols against coherent measurements. Dans *Proc. 27th ACM Symp. on Theory of Computing*, pages 67–82. ACM, 1995.