Université de Montréal

Protocoles cryptographiques avec des participants limités en espace mémoire

par

Julien Marcil

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

9 février 2000

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Protocoles cryptographiques avec des participants limités en espace mémoire

présenté par :

Julien Marcil

a été évalué par un jury composé des personnes suivantes :

(président-rapporteur)
Pierre McKenzie

(directeur de recherche)
Claude Crépeau

(membre du jury)
Gilles Brassard

Mémoire accepté le :

Sommaire

CE mémoire explique comment il est possible d'accomplir des tâches cryptographiques, de façon inconditionnellement sécuritaire, sous la seule hypothèse que les participants sont limités en espace mémoire. Des protocoles permettant de réaliser deux primitives très importantes en cryptographie – le partage de clef et le transfert inconscient – sont proposés.

Nous supposons que les participants ont accès à une source qui émet une très grande quantité de bits aléatoires, trop grande pour qu'ils puissent en enregistrer la totalité. La source pourrait être, par exemple, constituée des bits émis par un satellite. Il est également possible d'utiliser une source naturelle de bits aléatoires, publiquement accessible, telle que le bruit de fond de l'univers. Il est par contre nécessaire que les bits ne soient plus accessibles après leur diffusion.

Aucune hypothèse n'est faite sur la puissance de calcul des participants malhonnêtes ; il leur est possible de calculer n'importe quelle fonction, même probabiliste, des bits de la source, en autant que la taille de l'image n'excède pas l'espace mémoire dont ils disposent. La sécurité des protocoles proposés ne dépend que de l'incapacité des participants malhonnêtes à enregistrer tous les bits émis par la source.

L'analyse des protocoles repose grandement sur la théorie des probabilités et sur la théorie de l'information. Bien que les preuves soient très techniques, les méthodes utilisées pour construire les protocoles et prouver leur sécurité sont, quant à elles, assez générales.

Les constructions proposées nécessitent quelques outils ; certains de ceux-ci sont intéressants en eux-mêmes. Entre autres, nous analyserons, à partir de la théorie de l'information, le protocole de hachage interactif de NAOR, OSTROVSKY, VENKATESAN et YUNG [42].

Ce mémoire est largement basé sur le travail de CACHIN et MAURER [17] et sur celui de CACHIN, CRÉPEAU et MARCIL [15].

Table des matières

Liste des protocoles

Table des figures

à Alice et Bob

Remerciements

Ce travail repose sur de nombreuses années d'études. Plusieurs personnes ont été, sur ce parcours, d'une aide et d'un support indispensables. J'aimerais leur offrir mes plus sincères et chaleureux remerciements.

Premièrement, merci à mes parents Réal et Danielle, qui, par leur support et encouragement, m'ont permis de m'investir entièrement dans mes études. Merci également à mon frère Alexandre, qui est pour moi un grand exemple de persévérence et de réussite.

Merci à mes amis : Yannick Delbecque et Jérome Fournier avec qui je partage le grand plaisir des mathématiques ; Alain Tapp pour les nombreuses discussions sur la cryptographie et l'informatique théorique et pour son influence positive sur ma carrière de scientifique ; Jean-François Blanchette pour son gâteau au fromage mais surtout pour sa grande sagesse ; Adam Smith pour des échanges toujours très intéressants sur la théorie de l'information et sur la mécanique quantique ; Adeline Gendron pour m'avoir enduré durant la rédaction de mon mémoire et pour ses commentaires à la lecture de la première version de ce travail ; Sylvie Peltier pour ses commentaires à la lecture de la quinzième version de ce travail.

Un merci spécial à Josianne Lavallée et Sébastien Dubé, qui sont des acteurs un peu plus éloignés de ce travail, mais qui, par leur amitié, ont su me transmettre un sens du dépassement.

J'aimerais remercier les membres du Laboratoire d'informatique théorique et quantique de l'Université de Montréal, tout particulièrement les professeurs Gilles Brassard et Pierre McKenzie qui sont des modèles de leur profession. J'aimerais aussi remercier les membres du jeune *Cryptography and Quantum Information Laboratory* de l'Université McGill.

J'aimerais remercier Christian Cachin, avec qui j'ai travaillé en étroite collaboration sur le protocole de transfert inconcient, pour sa grande patience et son support inestimable.

Finalement, j'aimerais tout particulièrement remercier mon directeur de recherche, mentor et ami, Claude Crépeau, qui, depuis les trois dernières années, m'a donné son soutien et sa confiance. Il m'est très difficile d'imaginer où j'en serais aujourd'hui sans lui. Merci Claude.

Merci à tous.

CHAPITRE 1

Introduction

La *cryptographie* peut se définir comme étant «l'étude des communications dans un environnement hostile»[1]. Elle a pour but d'assurer la sécurité de certains aspects de l'échange d'information[2], telles la confidentialité, l'intégrité et l'authenticité.

Tout au long de cette introduction, plusieurs termes seront régulièrement utilisés. De façon générale, nous parlerons de *tâches cryptographiques*, c'est-à-dire des tâches avec des objectifs de sécurité propres. Les *procédés cryptographiques* constituent, quant à eux, l'ensemble des méthodes concrètes pour accomplir une tâche cryptographique donnée. De façon complémentaire à la cryptographie, la *cryptanalyse* est l'étude des moyens pour faire échouer ou pour briser un procédé. Le terme *cryptologie* est parfois utilisé pour regrouper la cryptographie et la cryptanalyse.

Les procédés cryptographiques existent depuis fort longtemps : l'histoire, particulièrement ses passages militaires, est ponctuée d'exemples d'utilisation de méthodes pour crypter – rendre inintelligible à l'adversaire – un message[3]. Il faut, par contre, attendre le développement de la théorie de l'information par SHANNON [49] pour avoir une véritable *science cryptographique*. Cette théorie donne un cadre d'analyse des procédés cryptographiques et établit quelques premières notions de sécurité. Encore aujourd'hui, une bonne partie de la cryptographie repose sur cette théorie et ce mémoire ne fait pas exception.

C'est l'avènement de l'informatique qui a permis le plus grand développement de la cryptologie. Les raisons sont multiples : premièrement, l'ordinateur facilite grandement la cryptanalyse. Il est maintenant nécessaire de développer des procédés qui viennent avec une certaine *garantie* de sécurité, soit des procédés construits de façon à assurer ses utilisateurs qu'ils sont difficiles à cryptanalyser. Deuxièmement, une foule

1. «*study of communication in an adversarial environment*» RIVEST [46].
2. Le concept d'échange d'information est pris ici dans son sens le plus large.
3. Pour un aperçu détaillé de l'histoire de la cryptologie, l'auteur recommande KAHN [35].

de *technologies* cryptographiques sont apparues pour solutionner les nouveaux problèmes de sécurité tels que l'accès à l'information, les communications entre ordinateurs, le commerce électronique, etc. Les trente dernières années sont donc très riches en nouveaux problèmes, mais aussi en nouveaux résultats [4].

Tâches cryptographiques

Les tâches cryptographiques sont nombreuses et diverses, mais il est toutefois possible de les regrouper dans quelques grandes familles. Mais avant de présenter quelques-unes d'entre elles, il faut tout d'abord introduire nos complices.

La cryptographie décrit des protocoles entre deux ou plusieurs personnes. Afin de rendre les participants un peu plus sympathiques que «Personne A» et «Personne B», nous les appellerons Alice et Bob. Parfois nous aurons besoin d'un troisième participant et, à ce moment-là, nous utiliserons la moins sympathique Ève [5] parce qu'elle est toujours malhonnête.

Parmi toutes les tâches cryptographiques, les plus connues sont celles qui permettent de préserver la *confidentialité*. Par exemple, Alice et Bob veulent discuter ensemble, mais ils n'ont pas un cqanal privé pour transmettre leurs messages, c'est-à-dire qu'Ève peut librement écouter leur conversation. Évidement, ils veulent garder leur discussion secrète. Pour protéger leur confidentialité, Alice et Bob vont devoir utiliser un procédé cryptographique qui empêchera Ève de comprendre, même en partie, leur conversation. Les applications de telles tâches sont évidentes : commerciales, militaires, adultères . . .

De pair avec ces tâches, viennent celles qui assurent l'*authenticité* d'un document. Quand Alice reçoit un message de Bob, elle veut s'assurer qu'il vient effectivement de Bob et non d'Ève essayant de se faire passer pour Bob. Ce type de tâches est très naturel, nous utilisons quotidiennement un procédé fort simple pour authentifier nos documents : notre signature. Les procédés d'authentification ou de signature électronique sont également importants puiqu'ils permettent d'assurer l'*intégrité* d'un document. Alice peut donc vérifier qu'Ève n'a pas essayé de changer le contenu d'un message envoyé et authentifié par Bob.

Les tâches qui préservent la confidentialité et l'authenticité vont de pair puisque bien souvent elles peuvent être réduites à la même primitive : la *distribution de clef*, sujet dont il sera question à la section 1.1.

Bien que les deux scénarios décrits précédemment soient très importants, il ne faut pas pour autant croire que la cryptographie se limite à ces tâches. Il en existe plusieurs

4. Pour une introduction à la cryptographie contemporaine, l'auteur recommande RIVEST [46], BRASSARD [10], SIMMONS [52], STINSON [53] ainsi que MENEZES, VAN OORSCHOT et VANSTONE [41].

5. L'utilisation du nom Ève vient de l'anglais «*eavesdropper*» qui signifie «oreille indiscrète».

autres dont un bon nombre est regroupé dans la famille des *calculs multi-parties sur données privées*.

Par exemple, imaginons que quatre individus veulent jouer au poker ensemble. Ils sont cependant répartis aux quatre coins du globe et il leur est donc difficile de se retrouver tous ensemble pour jouer. Alors, comment leur est-il possible de s'assurer qu'aucun d'eux ne prendra avantage de la situation. Une solution simple à ce problème est d'avoir à leur disposition une cinquième personne en qui les quatre joueurs ont confiance. Ce *maître de jeux* veillera au bon déroulement du jeu. Par contre, si les quatre joueurs ne trouvent personne en qui ils ont tous confiance, leur est-il quand même possible de jouer ? Oui ! ils peuvent, grâce aux calculs multi-parties sur données privées, *simuler* le maître de jeux. Jouer au poker, c'est en fait, faire un gros calcul en prenant en entrée les mains des joueurs et leurs actions.

Ce scénario peut paraître un peu trop ludique, mais il existe beaucoup d'exemples de tâches cryptographiques fort utiles qui peuvent être accomplies grâce aux calculs multi-parties sur données privées, comme des votes secrets, des protocoles de négociation de contrat ou de l'identification mutuelle. Dans notre société où l'information prend une importance grandissante, les calculs multi-parties sont appelés à jouer un rôle capital[6].

Notons que pour ces tâches, l'adversaire n'est plus extérieur au protocole, il fait partie des participants. Ceci change considérablement le cadre d'analyse de la sécurité d'un procédé.

Le *transfert inconscient* est une primitive qui permet de réaliser n'importe quels calculs multi-parties sur données privées. Nous en discuterons plus en détail à la section 1.2.

Sécurité

La sécurité est une des propriétés les plus importantes de tout procédé cryptographique. Mais comment est-il possible de garantir la sécurité d'un procédé ? Avant de répondre à cette question, il est essentiel de comprendre qu'en réalité la sécurité *absolue* n'existe pas. La sécurité d'un procédé cryptographique dépend totalement du modèle – l'ensemble des hypothèses faites sur les participants et leur univers – dans lequel il est proposé.

Dans cette optique, il est donc intéressant d'étudier différents modèles et de voir dans lesquels il est possible, ou non, d'accomplir certaines tâches cryptographiques. Le but est de comparer, non seulement la sécurité, mais également toutes les autres propriétés des procédés cryptographiques jugées importantes : faisabilité, coût en ressource, longévité, etc.

6. Pour une réflexion intéressante sur le sujet, l'auteur recommande de lire le chapitre 5 de TAPP [54].

Selon les hypothèses faites dans leur modèle, il est possible de séparer en deux grandes familles la sécurité des procédés : premièrement la sécurité dite *calculatoire* et deuxièmement la sécurité dite *inconditionnelle*.

La sécurité calculatoire repose toujours sur l'hypothèse que les participants ont une puissance de calcul limitée[7]. Cette limitation empêche les participants de résoudre certains problèmes, c'est-à-dire de réaliser des calculs très complexes dans un temps raisonnable. C'est cette incapacité à solutionner des problèmes qui permet d'accomplir des tâches cryptographiques. La sécurité des procédés de ce type est justifiée en montrant que la capacité d'un adversaire à casser le procédé avec une probabilité significative contredit la difficulté supposée du problème.

Cependant, la complexité des problèmes utilisés – telles la factorisation des grands entiers ou l'extraction des logarithmes discrets – pour construire des procédés cryptographiques n'est pas connue. En fait, très peu de bornes inférieures sont connues en complexité. Il est donc nécessaire de poser l'hypothèse supplémentaire que les problèmes sont en fait *réellement* difficiles.

Bien que la difficulté de ces problèmes ne soit pas mise en doute pour le moment, il n'existe aucune garantie qu'il en sera toujours ainsi. En fait, il existe même des évidences du contraire[8] : SHOR [51] a récemment trouvé un algorithme efficace pour solutionner les problèmes difficiles sur lesquels repose la grande majorité des procédés existants ; il faut cependant un ordinateur quantique[9] qui n'existe pas encore. Mais ceci suggère que, pour ces problèmes, la sécurité calculatoire n'est en fait que *technologique*.

Par opposition à la sécurité calculatoire qui est conditionnelle à la difficulté de résoudre un problème, la sécurité inconditionnelle ne fait, en général, aucune hypothèse sur la puissance de calcul d'un adversaire. En fait, nous donnons souvent à l'adversaire la possibilité de réaliser n'importe quel calcul[10]. Dans cet ordre d'idée, la notion de sécurité inconditionnelle signifie simplement que la sécurité ne dépend d'aucune autre hypothèse que celles spécifiées par le modèle. Évidemment, cette notion de sécurité n'est intéressante que lorsque les hypothèses du modèle sont relativement *faibles*, c'est-à-dire qu'elles sont facilement *réalisables*.

Les preuves de sécurité des procédés inconditionnellement sécuritaires reposent généralement sur la théorie des probabilités et sur la théorie de l'information. L'idée

7. Cette hypothèse, en soi, est assez faible. Nous serions très heureux de vivre dans un monde où il est possible de réaliser des calculs très complexes.

8. Ceci est l'opinion de l'auteur.

9. Pour une introduction à l'informatique quantique, l'auteur recommande BRASSARD [11].

10. Ceci peut paraître déraisonnable, mais imaginez que nous soyions attaqués par des extraterrestres. Quel genre d'hypothèses pourrions-nous faire sur leur puissance de calcul ? (Cette remarque a uniquement pour but d'exciter les amateurs de la série «*The X-Files*».)

est de montrer que les adversaires sont, sauf avec une très faible probabilité, incapables de tirer plus d'information du procédé que celle à laquelle ils ont droit.

Les exemples connus de procédés inconditionnellement sécuritaires tirent avantage d'un phénomène aléatoire. Donc, l'étude de ces procédés est fortement liée aux propriétés aléatoires de la nature.

En pratique, aujourd'hui, la première catégorie est beaucoup plus populaire que la seconde, principalement parce que les procédés cryptographiques que l'on peut construire sous des hypothèses calculatoires sont plus efficaces et plus flexibles. De plus, les hypothèses nécessaires pour utiliser de tels procédés sont automatiquement réalisées : il suffit d'avoir un ordinateur. Il est cependant nécessaire d'être prudent car il est probablement risqué de baser «l'économie mondiale de l'information» sur un nombre restreint de problèmes difficiles à résoudre.

Un point important à souligner en ce qui concerne la sécurité est que généralement un procédé cryptographique est toujours aussi faible que son maillon le plus faible. Heureusement, pour plusieurs tâches cryptographiques, nous savons exactement quel est ce maillon faible. En fait, nous savons qu'en ayant à notre disposition certaines primitives nous pouvons accomplir plusieurs tâches de façon parfaitement sécuritaire ou presque. Alors plusieurs procédés cryptographiques ne font en fait qu'accomplir la primitive et utiliser le système parfaitement sécuritaire. La sécurité du procédé résultant est donc directement liée à la sécurité des sous-procédés qui accomplissent les primitives.

Les protocoles proposés dans cette recherche relèvent de la seconde catégorie. Ce mémoire a pour objectif de montrer comment il est possible d'accomplir deux primitives cryptographiques très importantes – le partage de clef et le transfert inconscient – dans un modèle où les participants possèdent un espace mémoire limité.

Les sections 1.1 et 1.2 introduisent brièvement les primitives en question, leurs applications et les différents procédés qui existent pour les réaliser. Ensuite, la section 1.3 présente une description détaillée du modèle proposé.

1.1 Distribution de clef

La *distribution de clef* est une primitive fondamentale en cryptographie. L'objectif est de permettre à Alice et Bob d'avoir en commun une clef secrète, c'est-à-dire une chaîne binaire au sujet de laquelle Ève n'a aucune information. Nous faisons la distinction entre deux types de distribution de clef : premièrement, l'*échange de clef* [11] où Alice produit une clef et la transmet secrètement à Bob et deuxièmement, le *partage de clef* [12] où Alice et Bob s'entendent sur une clef en utilisant une source d'aléatoires.

11. *key exchange*
12. *key agreement*

Un grand nombre de tâches cryptographiques peuvent être facilement accomplies si Alice et Bob partagent une clef secrète. Reprenons l'exemple d'Alice qui veut envoyer un message secret à Bob. Abstraitement, pour y arriver, Alice et Bob peuvent, à l'aide d'une clef qu'ils sont les seuls à posséder, *verrouiller* et *déverrouiller* le message transmis.

Nous savons, en fait, que si Alice et Bob partagent une clef secrète, ils peuvent l'utiliser pour s'échanger des messages de façon *parfaitement secrète*[13], c'est-à-dire que les messages chiffrés ne révèlent aucune information à Ève sur les messages en clair. Ils peuvent, par exemple, utiliser le *masque jetable*[14] de VERNAM [55].

Par contre, la définition de parfaitement secret est très forte et SHANNON [50] a montré qu'il y a un prix à payer pour cette sécurité : la taille de la clef utilisée doit être au moins aussi grande que tous les messages qu'Alice et Bob veulent s'échanger. Pour cette raison, les procédés inconditionnellement sécuritaires ont longtemps été considérés comme inutilisables en pratique et ils ont été délaissés pour des procédés à sécurité calculatoire.

Les procédés de distribution de clef ne sont pas nécessairement cryptographiques. Voici une façon simple d'en accomplir un : Alice génère une grande quantité de bits uniformément distribués et les enregistre sur un disque dur qu'elle remet à Bob. Si Alice et Bob ne peuvent se rencontrer, il est possible de mettre le disque dans une mallette diplomatique cadenassée à un agent qui se chargera de remettre le disque à Bob. La confiance qu'ils auront dans la sécurité des échanges qu'ils feront par la suite avec cette clef dépend de la confiance qu'ils ont dans le courrier (et dans les autres moyens techniques utilisés pendant le transport du disque). Ce procédé n'est pas très pratique ; les gouvernements ont à leur disposition des agents pour accomplir ce type de tâches, mais ce n'est pas vraiment le cas pour tout le monde.

Il existe à ce jour deux procédés cryptographique *réalistes* qui permettent à Alice et Bob d'accomplir une distribution de clef de façon inconditionnellement sécuritaire. Premièrement, la distribution de clef quantique développée par BENNETT et BRASSARD [4] où la sécurité est assurée par les lois de physique quantique. Deuxièmement, le partage de clef sur un canal bruyant, aussi appelé canal binaire symétrique, développé par MAURER [40].

Une autre solution ingénieuse au problème de distribution de clef, tel qu'imaginée par DIFFIE et HELLMAN [30], est d'utiliser un système asymétrique, aussi appelé système à clef publique. Dans ce cas, deux clefs différentes sont utilisées : une première pour verrouiller le message et une seconde pour le déverrouiller. La première clef peut être fournie publiquement par Alice qui conservera la seconde secrètement.

---

13. *perfect secrecy*
14. *one-time pad*

Il n'y a donc jamais d'échange privé entre Alice et Bob. La sécurité de ce type de procédé repose nécessairement sur la difficulté à résoudre un problème difficile, puisque l'adversaire n'a qu'à inverser le calcul de Bob avec le message chiffré et la clef publique. Aujourd'hui, la majorité des procédés cryptographiques repose en partie ou totalement sur les systèmes à clef publique. La vedette incontestée des systèmes à clef publique est RSA dû à RIVEST, SHAMIR et ADLEMAN [47].

La distribution de clef est souvent le maillon faible de plusieurs procédés cryptographiques. C'est pour cette raison qu'elle est traité comme une primitive, puisque nous savons que si nous trouvons un procédé pratique et sécuritaire pour distribuer des clefs, alors nous aurons à notre disposition des procédés pour accomplir une foule de tâches cryptographiques.

1.2 Transfert inconscient

Le *transfert inconscient*[15], noté $\binom{2}{1}$-OT, est une primitive très importante en cryptographie moderne. Il permet d'accomplir une foule de tâches cryptographiques tels la *mise en gage de bit*[16], les preuves interactives et les calculs multi-parties sur données privées.

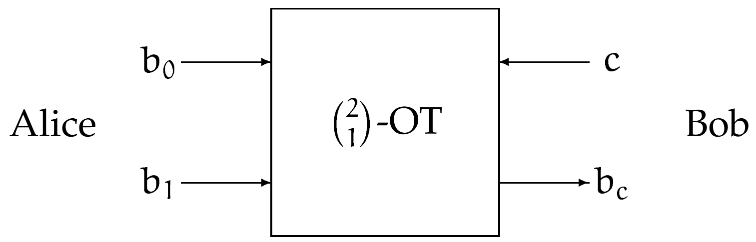

FIG. 1.1 – *Transfert inconscient*

Supposons qu'Alice connaisse (choisisse) deux bits b_0 et b_1 et que Bob connaisse (choisisse) un bit c et veuille apprendre le bit b_c. C'est le scénario décrit à la figure 1.1. Nous ajoutons cependant quelques restrictions : Alice est prête à donner un de ses bits à Bob à la condition que Bob n'apprenne aucune information sur l'autre bit. Plus formellement, nous dirons que Bob n'apprend aucune information sur le bit $b_{\bar{c}}$, même si Alice lui donne, après le transfert, le bit b_c. Bob, lui, est prêt à participer à la condition qu'Alice n'apprenne rien sur son choix, c'est-à-dire son bit c.

15. *one out of two oblivious transfer*
16. *bit commitment*

Cette primitive fut d'abord étudiée par Wiesner [56] sous le nom de «*message multiplexing*» et fut plus tard indépendamment introduite en cryptographie sous différentes variations par Rabin [43] (le transfert équivoque) et par Even, Goldreich et Lempel [31]. Depuis, les relations entre le transfert inconscient et les calculs multi-parties sur données privées ont été beaucoup étudiées : Yao [57], Goldreich, Micali et Wigderson [32], Goldreich, Vainish [33], Kilian [36] et Crépeau, van de Graaf et Tapp [25].

Le transfert inconscient a premièrement été étudié dans des contextes calculatoires tels la factorisation des entiers ou l'existence des permutations à brèche secrète : Even, Goldreich et Lempel [31], Goldreich, Micali et Wigderson [32] et Bellare et Micali [3].

Il est également possible d'accomplir un transfert inconscient de façon inconditionnellement sécuritaire à partir d'un canal bruyant : Crépeau et Kilian [26] et Crépeau [24]. Si Alice et Bob possèdent un canal quantique, c'est-à-dire un canal où ils peuvent s'échanger des bits quantiques, alors il leur est possible de réduire le transfert inconscient à la mise en gage de bit : Bennett, Brassard, Crépeau et Skubiszewska [6] et Crépeau et Kilian [26] Crépeau [22].

Le *transfert équivoque*[17] est une primitive similaire au transfert inconscient.

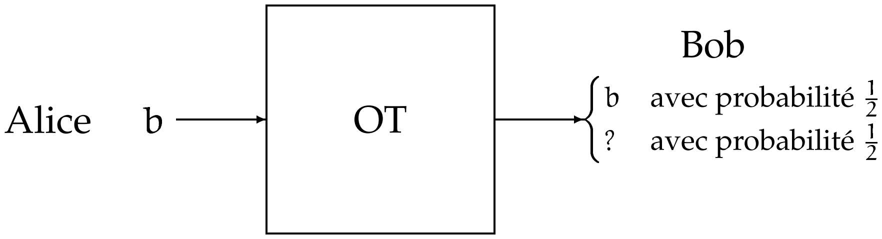

Fig. 1.2 – *Transfert équivoque*

Alice envoie le bit b à Bob. Bob recevra, avec probabilité $\frac{1}{2}$, le bit d'Alice ou, avec probabilité $\frac{1}{2}$, il ne recevra rien (?), mais il sait qu'il n'a rien reçu. Alice, elle, n'apprend pas si Bob a reçu, ou non, son bit. C'est le scénario décrit à la figure 1.2.

Crépeau [23] a montré que le transfert inconscient et le transfert équivoque sont équivalents. L'idée de la réduction du transfert inconscient au transfert équivoque est utilisée dans le protocole proposé pour construire un transfert inconscient dans le modèle des participants à espace mémoire borné. En fait, le protocole de transfert inconscient est beaucoup plus facile à comprendre si l'idée de la réduction est mise en évidence. Vous trouverez donc dans le chapitre 5 la réduction en question.

17. *(Rabin's) oblivious transfer*

Pour une introduction plus complète sur le transfert inconscient et les calculs multi-parties sur données privées, l'auteur recommande de consulter CRAMER [21].

1.3 Modèle des participants à espace mémoire limité

Supposons que les participants ont accès à une source qui émet une très grande quantité de bits aléatoires, trop grande pour qu'ils puissent en enregistrer la totalité. La source pourrait être, par exemple, constituée des bits émis par un satellite. Il est également possible d'utiliser une source naturelle de bits aléatoires, publiquement accessible, telle que le bruit de fond de l'univers. Il est par contre nécessaire que les bits ne soient plus accessibles après leur diffusion. Dans tous les protocoles proposés, il est également possible que ce soit Alice qui génère et envoie des bits uniformément distribués à Bob.

Aucune hypothèse n'est faite sur la puissance de calcul des participants malhonnêtes ; il leur est possible de calculer n'importe quelle fonction, même probabiliste, des bits de la source, en autant que la taille de l'image n'excède pas l'espace mémoire dont ils disposent. Cette hypothèse peut sembler exagérée, car dans ce cas il est possible pour les participants de réaliser des calculs qui requièrent un espace mémoire de taille exponentiellement plus grand que la source, ou encore d'évaluer des fonctions non-calculables. En fait, il est simplement plus facile de prouver la sécurité des protocoles sous cette hypothèse plus forte. La sécurité des protocoles proposés dépend donc uniquement de l'incapacité des participants malhonnêtes à enregistrer tous les bits émis par la source.

Nous supposons que les bits de la source sont transmis aux participants sans erreur. Dans le cas du partage de clef, cette hypothèse n'est présente que pour simplifier les protocoles. Si le canal de transmission n'est pas parfait, il est possible pour Alice et Bob de faire de la réconciliation d'erreurs immédiatement après qu'ils aient sélectionné des bits de la source[18].

Le modèle utilisé fut premièrement motivé par MASSEY et INGEMARSSON [39]. Il fut par la suite utilisé par MAURER [40] pour construire un système de chiffrement qui est sécuritaire s'il existe une grande source qui ne peut être lue entièrement dans un temps raisonnable. Il fut par la suite modifié par CACHIN et MAURER [17] pour accomplir un partage de clef et il a été adapté plus tard pour le transfert inconscient par CACHIN, CRÉPEAU et MARCIL [15].

Les relations entre les limites d'espace mémoire et la cryptographie ont également été étudiées dans d'autres contextes. KILIAN [37], DE SANTIS, PERSIANO et YUNG [29]

18. Cette étape supplémentaire donne un peu d'information à Ève sur la chaîne qu'ils partagent. Des méthodes pour borner l'information ainsi révélée à Ève sont connues CACHIN et MAURER [16].

et AUMANN et FEIGE [2] ont étudié les *preuves interactives à divulgation nulle*[19] dans le cas où le vérificateur est limité en espace mémoire.

Ce modèle semble réaliste en fonction des technologies disponibles en cette fin de siècle. En effet, les réseaux de communication permettent des débits de transmission de l'ordre de plusieurs gigaoctets par seconde. Par ailleurs, le stockage d'information de l'ordre du pétaoctet (un million de gigaoctets) nécessite un assez grand investissement. Mais la sécurité du modèle est principalement du au fait qu'aucun système d'archivage actuelle ne serait capable de soutenir un si grand débit d'information.

Le modèle s'applique également dans le cas où l'espace mémoire des participants serait limité physiquement ; les cartes à puces sont un médium particulièrement bien adapté à ce genre de scénarios[20].

Un grand avantage de ce modèle est que les protocoles sont rétroactivement sécuritaires, c'est-à-dire que si un participant malhonnête n'enregistre pas les bits de la source durant leur transmission, alors ils ne sont plus jamais disponibles et il lui sera impossible de briser le système.

1.4 Organisation des chapitres

Les protocoles présentés dans ce mémoire ne sont pas vraiment compliqués, mais leurs preuves de sécurité sont considérablement techniques. Ces aspects techniques couvrent une très large portion de ce document : premièrement par l'introduction des outils mathématiques nécessaires aux constructions et aux preuves et deuxièmement par la présentation des preuves elles-mêmes. Bien que les preuves soient très techniques, les méthodes utilisées pour construire les protocoles et prouver leur sécurité sont, quant à elles, assez générales.

Le chapitre 2 fait un rappel des notions mathématiques – la théorie des probabilités et la théorie de l'information – nécessaires à l'analyse des protocoles présentés tout au long de ce document. Le chapitre 3 introduit et présente plusieurs résultats – l'amplification de secret et la génération d'ensembles k-indépendants – sur les fonctions de hachage. Il y sera également question d'un protocole de hachage interactif qui sera utilisé plus tard dans le protocole de transfert inconscient.

Les protocoles de partage de clef et de transfert inconscient seront présentés respectivement dans les chapitres 4 et 5. Ils sont accompagnés de leur preuve de sécurité.

Dans l'annexe A, le lecteur trouvera la description d'un encodage simple, efficace et compact des sous-ensembles de k éléments en chaînes binaires.

19. *zero knowledge proofs*

20. Dans ce cas, le partage de clef n'a aucune application, mais le transfert inconscient peut être assez utile. Par exemple, il possible d'utiliser le protocole d'identification mutuelle de CRÉPEAU et SALVAIL [27].

CHAPITRE 2

Préliminaires

DANS ce chapitre, il sera question de la théorie des probabilités et de la théorie de l'information. Une bonne connaissance de ces deux théories est requise pour bien comprendre les preuves de sécurité des protocoles qui sont considérablement techniques. Ce chapitre ne constitue cependant pas une introduction à ces sujets, donc le lecteur devrait être familier avec ces théories. Il existe d'excellents ouvrages d'introduction sur ces sujets[1]. Des notions d'algorithmie seront également nécessaires mais ne sont pas introduites dans ce chapitre[2].

Le but de ce chapitre est double : premièrement, introduire et fixer la notation qui sera utilisée tout au long de ce document et deuxièmement, fournir quelques-uns des théorèmes qui seront utilisés, question de rendre ce document plus complet en soi.

Notation. Tous les logarithmes notés lg sont en base 2. La cardinalité de l'ensemble $\mathcal{S}$ est notée $|\mathcal{S}|$.

Notation. Soit $x \in \{0,1\}^n$ et soit $\mathcal{S} \subseteq \{1,2,\ldots,n\}$. La sous-chaîne de x obtenu en restreignant x aux positions données par l'ensemble $\mathcal{S}$ est notée $x^{\mathcal{S}}$. Plus particulièrement, nous définissons les ensembles suivants : $\langle j] = \{1,2,\ldots,j\}$ et $[j\rangle = \{j,j+1,\ldots,n\}$ (notez que la chaîne originale ne sera pas toujours de longueur n, mais il sera toujours possible à partir du contexte de savoir quelle est sa longueur). Donc la chaîne $x^{\langle j]}$ représente les j premières positions de x et $x^{[j\rangle}$ représente les $n-j+1$ dernières positions de x. Pour éviter toute confusion, x est normalement noté $x^{\langle n]}$.

2.1 Théorie des probabilités

La théorie des probabilités est un outil fondamental à l'analyse des protocoles en cryptographie inconditionnellement sécuritaire. Elle permet principalement de bien modéliser la connaissance des participants, mais elle permet aussi de quantifier les chances d'échec d'un protocole.

1. Pour une introduction à la théorie de l'information, l'auteur recommande ROMAN [48], BLAHUT [9] et COVER et THOMAS [20].

2. Pour une introduction à l'algorithmie, l'auteur recommande BRASSARD et BRATLEY [12].

2.1.1 Variable aléatoire

Définition 2.1. Soit $\mathcal{X}$ un ensemble dénombrable. Soit X une variable aléatoire sur $\mathcal{X}$ et soit $x \in \mathcal{X}$. La distribution de probabilité P_X donne la probabilité $P_X(x) = P[X = x]$ de l'événement $X = x$.

De même avec un vecteur de variables aléatoires $X_1, X_2, \ldots, X_n$, noté $X^{\langle n]}$, la distribution de probabilité $P_{X_1X_2\ldots X_n}$, appelée la distribution conjointe, donne la probabilité $P_{X_1X_2\ldots X_n}(x_1, x_2, \ldots, x_n)$ d'un événement $X_1 = x_1$, $X_2 = x_2$, ..., $X_n = x_n$, noté $X^{\langle n]} = x^{\langle n]}$.

Notation. Une variable aléatoire X est toujours notée par une lettre majuscule. Elle induit une distribution de probabilité notée P_X sur l'ensemble noté par $\mathcal{X}$ la lettre calligraphique correspondante.

Définition 2.2. Soient X et Y des variables aléatoires et soient $x \in \mathcal{X}$ et $y \in \mathcal{Y}$. Alors la distribution de probabilité $P_{X\,|\,Y=y}$ donne la probabilité d'un événement $X = x$ conditionné sur l'événement $Y = y$

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

Définition 2.3. Soient X et Y des variables aléatoires. X et Y sont *indépendantes* si pour tout $x \in \mathcal{X}$ et $y \in \mathcal{Y}$

$$P_{X\,|\,Y=y}(x) \;=\; P_X(x)\ .$$

Les variables aléatoires qui prennent des valeurs réelles ont beaucoup d'importance puisqu'elles ont des propriétés très intéressantes.

Définition 2.4. Soit X une variable aléatoire sur les réels $\mathcal{X} \subset \mathbb{R}$. L'*espérance de* X est définie par

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

Définition 2.5. Soit X une variable aléatoire sur les réels. La *variance de* X est définie par

$$\mathrm{Var}[X] \;=\; \mathrm{E}\left[(X - \mathrm{E}[X])^2\right]\ .$$

Cette définition de la variance n'est pas toujours facile à utiliser. Le lemme suivant (qui se déduit simplement de la définition de la variance et de la linéarité de l'espérance) est souvent plus utile.

Lemme 2.1. $\mathrm{Var}[X] = \mathrm{E}[X^2] - \mathrm{E}[X]^2$.

2.1.2 Quelques inégalités

Les quelques inégalités suivantes sont standard en théorie des probabilités. Nous supposons que $\mathcal{X}$ est un ensemble ordonné.

Théorème 2.2 (Inégalité de Markov). *Soit* X *une variable aléatoire sur les réels positifs et soit* $t \in \mathbb{R}^+$.

$$P[X \geq t] \;\leq\; \frac{E[X]}{t}$$

Théorème 2.3 (Inégalité de Chebychef). *Soit* X *une variable aléatoire sur les réels positifs et soit* $t \in \mathbb{R}^+$.

$$P\left[|X - E[X]| \geq t\right] \;\leq\; \frac{\mathrm{Var}[X]}{t^2}$$

2.2 Théorie de l'information

La théorie de l'information a été développée par Shannon [49]. Cette théorie est un des outils fondamentaux de la théorie des codes et de la cryptographie. La théorie de l'information repose principalement sur la théorie des probabilités.

2.2.1 L'entropie de Shannon

Nous pouvons premièrement définir l'entropie sur une simple variable aléatoire.

Définition 2.6. Soit X une variable aléatoire. L'*entropie (de Shannon) de* X est définie par

$$H(X) \;=\; -\sum_{x \in \mathcal{X}} P_X(x) \lg P_X(x)$$

en posant que $0 \lg 0 = 0$, ce qui se justifie par le fait que $\lim_{p \to 0^+} p \lg p^{-1} = 0$.

Cette définition s'étend aussi aux vecteurs de variables aléatoires $X^{\langle n]}$ en prenant la distribution de probabilité conjointe $P_{X^{\langle n]}}$ du vecteur de variables aléatoires. Nous pouvons aussi définir l'entropie d'une variable aléatoire X conditionnée sur un événement $Y = y$.

Définition 2.7. Soient X et Y des variables aléatoires. L'*entropie conditionnelle de* X *sur l'événement* $Y = y$ est définie par

$$H(X \,|\, Y = y) \;=\; -\sum_{x \in \mathcal{X}} P_{X \,|\, Y=y}(x) \lg P_{X \,|\, Y=y}(x) \;.$$

Notation. Dans le cas particulier où X est une variable aléatoire sur $\{0,1\}$ nous utilisons la fonction d'entropie binaire

$$h(p) \;=\; -p \lg p - (1-p)\lg(1-p)$$

où $p = P_X(1)$ (ou $p = P_X(0)$ par symétrie). Notons que $H(X) = h(p)$.

Il est possible de calculer l'espérance de l'entropie conditionnelle de X sur la variable aléatoire Y.

Définition 2.8. Soient X et Y des variables aléatoires. L'*entropie conditionnelle de* X *sur* Y est définie par

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

Théorème 2.4. *Soit* X *une variable aléatoire. L'entropie de Shannon respecte les propriétés suivantes :*

1° $H(X) \geq 0$ *avec égalité si et seulement s'il existe* $x \in \mathcal{X}$ *tel que* $P_X(x) = 1$.

2° $H(X) \leq \lg |\mathcal{X}|$ *avec égalité si et seulement si* X *est uniformément distribué sur* $\mathcal{X}$.

Les théorèmes suivants sont des propriétés importantes de l'entropie de Shannon.

Théorème 2.5 (règle de chaîne).

$$H(XY) \;=\; H(Y) + H(X \mid Y)$$

Théorème 2.6.

$$H(X \mid Y) \;\leq\; H(X)$$

avec égalité si et seulement si X *et* Y *sont indépendantes.*

2.2.2 L'entropie de Rényi

Cette section couvre une généralisation de la théorie de l'information développée par Rényi [44, 45].

Définition 2.9. Soit X une variable aléatoire et soient $\alpha \geq 0$ et $\alpha \neq 1$. L'*entropie de Rényi d'ordre* α *de* X est définie par

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

De même que pour l'entropie de Shannon, il est possible de définir l'entropie de Rényi sur un vecteur de variables aléatoires et sur une variable aléatoire conditionnée sur un événement.

Définition 2.10. Soit X une variable aléatoire et soient $\alpha \geq 0$ et $\alpha \neq 1$. L'*entropie conditionnelle de Rényi d'ordre* α *de* X *sur* Y est définie par

$$H_\alpha(X \mid Y) \;=\; \sum_{y \in \mathcal{Y}} P_Y(y) H_\alpha(X \mid Y = y) \;.$$

Remarque. Il est important de noter que cette dernière définition n'est pas toujours acceptée, principalement parce qu'elle ne préserve pas les propriétés 2.5 et 2.6 de l'entropie de Shannon.

Une autre notion d'entropie fort importante est la min-entropie.

Définition 2.11. Soit X une variable aléatoire. La *min-entropie de* X est définie par

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

Le lemme suivant montre que l'entropie de Rényi n'est en fait qu'une généralisation de l'entropie de Shannon.

Lemme 2.7. *Soit* X *une variable aléatoire*

1. $H_0(X) \;=\; \lg |\mathcal{X}|$
2. $\lim_{\alpha \to 1} H_\alpha(X) \;=\; H(X)$
3. $\lim_{\alpha \to \infty} H_\alpha(X) \;=\; H_\infty(X)$

Théorème 2.8. *Soit* X *une variable aléatoire et soient* α, β *deux réels tels que* $0 \leq \alpha < \beta$ *alors*

$$H_\alpha(X) \;\geq\; H_\beta(X) \;.$$

avec égalité si et seulement si X *est uniformément distribué sur* $\mathcal{X}$ *quand* $\alpha = 0$ *ou* X *est uniformément distribué sur un sous-ensemble de* $\mathcal{X}$ *quand* $\alpha > 0$.

Pour terminer cette section, voici un théorème fort important sur la min-entropie d'une variable aléatoire conditionnée sur un événement. Ce théorème sera utilisé à plusieurs occasions dans les preuves de sécurité.

Théorème 2.9. *Soient* X *et* V *des variables aléatoires et soit* $r > 0$. *Alors, avec une probabilité supérieure à* $1 - 2^{-r}$, V *prend une valeur* v *tel que*

$$H_\infty(X \mid V = v) \;\geq\; H_\infty(X) - \lg |\mathcal{V}| - r \;.$$

Preuve. Posons $p_0 = \frac{2^{-r}}{|\mathcal{V}|}$. Alors

$$\sum_{z : P_V(z) < p_0} P_V(z) \; < \; 2^{-r}.$$

Donc, pour tous les autres $v \in \mathcal{V}$ tel que $P_V(v) \geq p_0$, nous avons que

$$\begin{aligned} H_\infty(X \mid V = v) &= -\lg \max_{x \in \mathcal{X}} P_{X \mid V = v}(x) \\ &= -\lg \max_{x \in \mathcal{X}} \frac{P_X(x) P_{V \mid X = x}(v)}{P_V(v)} \\ &\geq -\lg \max_{x \in \mathcal{X}} \frac{P_X(x)}{p_0} \\ &= H_\infty(X) - \lg |\mathcal{V}| - r \; . \end{aligned}$$

$\square$

Chapitre 3

Fonctions de hachage

Les fonctions de hachage ont plusieurs utilités en informatique : structures de données, applications en informatique théorique[1], etc. La cryptographie ne fait pas exception ; les fonctions de hachage sont régulièrement utilisées dans les procédés d'authentification ou de signature électronique.

Dans ce chapitre, il sera particulièrement question des utilisations des fonctions de hachage en cryptographie inconditionnellement sécuritaire. Les classes de fonctions de hachage k-universelles introduites par Wegman et Carter [18] ont des propriété fort intéressantes et elles seront présentées à la section 3.1.

Dans les sections 3.2 et 3.3, il sera question d'applications particulièrement puissantes de ces classes de fonctions, respectivement la distillation de secret et la génération d'ensemble k-indépendant.

Finalement, à la section 3.4, nous voyons et analysons un premier protocole : le hachage interactif. Ce protocole nous sera nécessaire pour accomplir une étape importante du protocole de transfert inconscient (protocole 5.2).

3.1 Définitions et exemples

Les fonctions de hachage sont simplement des fonctions d'un alphabet $\mathcal{X}$ à un alphabet $\mathcal{Y}$ tel que $|\mathcal{X}| > |\mathcal{Y}|$. L'idée de base est d'utiliser la valeur de hachage $y = h(x)$ comme un court représentant de son entrée x.

Puisque le domaine est plus grand que l'image, alors il se produit inévitablement des *collisions*, c'est-à-dire qu'il existe des valeurs d'entrée x_0, x_1 distinctes qui donnent la même valeur de hachage $h(x_0) = h(x_1)$.

Nous nous interessons à des classes de ces fontions.

Définition 3.1. Une classe $\mathcal{H}$ de fonctions de hachage de $\mathcal{X} \to \mathcal{Y}$ est k-*universelle* si pour chaque ensemble d'éléments distincts $\{x_1, x_2, \ldots, x_k\} \subseteq \mathcal{X}$ il y a au plus $\frac{|\mathcal{H}|}{|\mathcal{Y}|^{k-1}}$ fonctions $h \in \mathcal{H}$ tel que $h(x_1) = h(x_2) = \cdots = h(x_k)$.

1. Pour plus d'information sur le sujet, l'auteur recommande de lire Luby et Wigderson [38].

Définition 3.2. Une classe $\mathcal{H}$ de fonctions de hachage de $\mathcal{X} \to \mathcal{Y}$ est *fortement* k*-universelle* si pour chaque ensemble d'éléments distincts $\{x_1,x_2,\ldots,x_k\} \subseteq \mathcal{X}$ et pour chaque $y_1,y_2,\ldots,y_k \in \mathcal{Y}$ (non nécessairement distinct) il y a exactement $\frac{|\mathcal{H}|}{|\mathcal{Y}|^k}$ fonctions $h \in \mathcal{H}$ tel que $h(x_1) = y_1, h(x_2) = y_2, \ldots, h(x_k) = y_k$.

Exemple 3.1. Soit $\mathbb{F}_q$ un corps fini. L'ensemble des matrices m par n sur $\mathbb{F}_q$ constitue une classe 2-universelle de fonctions de hachage de ${\mathbb{F}_q}^n \to {\mathbb{F}_q}^m$.

Exemple 3.2. Soit $\mathbb{F}_q$ un corps fini. Soit l'ensemble des fonctions

$$\mathcal{H} \;=\; \left\{ h(x) = \mathrm{eps}_m\Big(\sum_{i=0}^{k-1} a_i x^i\Big) \;\middle|\; a_i \in \mathbb{F}_{q^n} \text{ pour } i = 1,2,\ldots,k-1 \text{ et } a_{k-1} \neq 0 \right\}$$

où $\mathrm{eps}_m(x)$ sont les m éléments de $\mathbb{F}_q$ les plus significatifs de x. $\mathcal{H}$ constitue une classe fortement k-universelle de fonctions de hachage de $\mathbb{F}_{q^n} \to \mathbb{F}_{q^m}$.

Ces exemples ne sont pas choisis au hasard, ils nous seront utiles plus tard. Généralement, le corps fini sera $\mathbb{F}_2$, qui peut être vu comme l'ensemble $\{0,1\}$ avec les opérations *et* et *ou-exclusif*.

3.2 Distillation de secret

La *distillation de secret*[2] est un outil fort puissant de la cryptographie basé sur la théorie de l'information ; c'est un élément clef de plusieurs protocoles en cryptographie inconditionnellement sécuritaire. Elle fut premièrement introduite par BENNETT, BRASSARD et ROBERT [7, 8] et par la suite généralisée par, entre autres, BENNETT, BRASSARD, CRÉPEAU et MAURER [5], CACHIN et MAURER[16] et CACHIN [14]. Notons également que HÅSTAD, IMPAGLIAZZO, LEVIN et LUBY [34] sont indépendemment arrivés à des résultats similaires mais dans un contexte assez différent.

Supposons qu'Alice et Bob partagent une certaine chaîne binaire. Plus formellement nous dirons qu'ils partagent une variable aléatoire X. Ève, quant à elle, possède un peu d'information sur X, nous dirons qu'elle a une variable aléatoire V corrélée avec X. La distribution P_{XV} n'est pas connue d'Alice et Bob, c'est-à-dire qu'Alice et Bob ne savent pas quelle information Èveà sur leur chaîne. mais ils savent que l'entropie de Rényi d'ordre 2 d'Ève sur X conditionnée sur l'événement $V = v$ est bornée par une certaine valeur.

En discutant publiquement, Alice et Bob désirent, à partir de X, produire une plus petite chaîne sur laquelle Ève n'a aucune information. Plus précisément, ils veulent s'entendre sur une fonction de compression g tel qu'Ève n'a presque pas d'information sur la chaîne $Y = g(X)$.

2. *privacy amplification*

Le besoin de distiller un secret apparaît habituellement vers la fin d'un protocole, lorsqu'une clef secrète doit être extraite (distillée) d'un ensemble de bits partiellement secrets.

Le théorème suivant, dû à BENNETT, BRASSARD, CRÉPEAU et MAURER [5], montre que si Alice et Bob choisissent une fonction g de façon uniformément aléatoire dans une classe de fonctions de hachage 2-universelle $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$, pour un bon $\mathcal{Y}$ (c'est-à-dire $\lg|\mathcal{Y}| < H_2(X \mid V = v)$), alors, avec une bonne probabilité, l'information d'Ève sur $Y = g(X)$ est négligeable.

Théorème 3.1 (Distillation de secret). *Soit* X *une variable aléatoire, soit* V *une variable aléatoire corrélée avec* X*, soit* G *une variable aléatoire uniformément distribuée sur* $\mathcal{G}$*, une classe 2-universelle de fonctions de hachage de* $\mathcal{X} \to \mathcal{Y}$*, et soit* $Y = G(X)$*. Alors*

$$H(Y \mid G, V = v) \;\geq\; H_2(Y \mid G, V = v) \;\geq\; \lg|\mathcal{Y}| - \frac{2^{\lg|\mathcal{Y}| - H_2(X \mid V = v)}}{\ln 2}\,.$$

Il est possible que $H(Y \mid G = g, V = v) = H(g(X) \mid V = v)$ diffère de façon significative de $\lg|\mathcal{Y}|$ pour certains g, mais la probabilité de choisir un tel g est, quant à elle, négligeable quand $\lg|\mathcal{Y}| \ll H_2(X)$. Donc l'entropie de Y est presque maximale, ce qui veut dire que la distribution de Y est presque uniforme sur $\mathcal{Y}$.

Par contre, le théorème 3.1 ne peut pas être généralisé à l'entropie de Rényi conditionnée sur une variable aléatoire, c'est-à-dire que

$$H(Y \mid G, V) \;\geq\; \lg|\mathcal{Y}| - \frac{2^{\lg|\mathcal{Y}| - H_2(X \mid V)}}{\ln 2}$$

est faux en général.

3.3 k-indépendance

Les classes de fonctions de hachage fortement k-universelles peuvent être utilisées pour générer des ensembles k-indépendants de variables aléatoires uniformément distribuées.

Dans les protocoles de partage de clef, nous aurons besoin de sélectionner un sous-ensemble de bits de façon aléatoire. De plus, il est nécessaire de mémoriser la description de ce sous-ensemble. Évidemment, si nous choisissons le sous-ensemble de façon uniformément aléatoire, la description des positions des bits prendra plus d'espace mémoire que les bits eux-mêmes. Il est possible de réduire l'espace mémoire requis en perdant de l'indépendance entre les bits. Au lieu d'avoir un ensemble de variables aléatoires parfaitement indépendantes, nous aurons, un ensemble k-indépendant.

Définition 3.3. Un ensemble de variables aléatoires est k-indépendant si tous ses sous-ensembles de k éléments ou moins ne contiennent que des variables aléatoires indépendantes mais sans répétition.

Soit $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$ une classe fortement k-universelle de fonctions de hachage (voir l'exemple 3.2). Nous pouvons construire un ensemble k-indépendant de variables aléatoires de la façon suivante. Choisir $g \in \mathcal{G}$ de façon uniformément aléatoire et l'appliquer à un ensemble, préalablement fixé $x_1, x_2, \ldots, x_l$ (normalement $1, 2, \ldots, l$) d'éléments distincts de $\mathcal{X}$. Soit $Y_j = G(x_j)$ pour $j = 1, 2, \ldots, l$. Dans le cas où il y se produirait des collisions, nous pouvons soit éliminer les répétitions à la fin ou encore évaluer g sur la prochaine valeur de la suite $x_1, x_2, \ldots, x_l$. Il est facile de vérifier que $Y_1, Y_2, \ldots, Y_l$ forment un ensemble de variables aléatoires uniformément distribuées et k-indépendant.

L'avantage de cette technique, comparé à celle de choisir l éléments indépendants de $\mathcal{Y}$, est qu'elle ne nécessite que $k \lg |\mathcal{Y}|$ bits (plutôt que $l \lg |\mathcal{Y}|$ bits dans le cas où le sous-ensemble est choisi de façon uniformément aléatoire) pour représenter un sous-ensemble.

Nous voulons maintenant montrer que l'intersection de deux ensembles de variables aléatoires 2-indépendants forme également un ensemble 2-indépendant. Ce théorème nous sera utile pour prouver la sécurité du protocole de partage de clef par discussion publique (protocole 4.2).

Théorème 3.2. *Soient $A_1, A_2, \ldots, A_n$ et $B_1, B_2, \ldots, B_n$ deux ensembles 2-indépendants de variables aléatoires uniformément distribuées sur $\mathcal{Y}$. Posons*

$$I_i = \begin{cases} A_i & \textit{si il existe un } 1 \leq j \leq n \textit{ tel que } I_i = B_j \\ \tau & \textit{sinon} \end{cases}$$

Les variables aléatoires $I_1, I_2, \ldots, I_n$ restreintes aux positions différentes de τ forment un ensemble 2-indépendant de variables aléatoires uniformément distribuées sur $\mathcal{Y}$.

Preuve. Les variables aléatoires $A^{\langle n]}$ et $B^{\langle n]}$ sont uniformément distribuées sur les sous-ensembles de n éléments de $\mathcal{Y}$ et forment des ensembles 2-indépendants, alors pour tout $1 \leq i < j \leq n$ et pour tout $y_1, y_2 \in \mathcal{Y}$ tel que $y_1 \neq y_2$

$$P[A_i = y_1 \text{ et } A_j = y_2] = \frac{1}{|\mathcal{Y}| \left(|\mathcal{Y}| - 1\right)}$$

La séquence $I^{\langle n]}$ satisfait pour tout $1 \leq i < j \leq n$ et pour tout $y_1, y_2 \in \mathcal{Y}$ tel que $y_1 \neq y_2$

$$\begin{aligned}
P[I_i = y_1 \text{ et } I_j = y_2 \,|\, I_i \neq \tau \text{ et } I_j \neq \tau] \\
&= \frac{P[I_i = y_1 \text{ et } I_j = y_2]}{P[I_i \neq \tau \text{ et } I_j \neq \tau]} \\
&= \frac{P[A_i = y_1 \text{ et } \exists_{k_1} B_{k_1} = y_1 \text{ et } A_j = y_2 \text{ et } \exists_{k_2} B_{k_2} = y_2]}{P[\exists_{k_1} \exists_{k_2} B_{k_1} = A_i \text{ et } B_{k_2} = A_j]} \\
&= \frac{P[A_i = y_1 \text{ et } A_j = y_2] \cdot P[\exists_{k_1} \exists_{k_2} B_{k_1} = y_1 \text{ et } B_{k_2} = y_2]}{P[\exists_{k_1} \exists_{k_2} B_{k_1} = A_i \text{ et } B_{k_2} = A_j]} \\
&= P[A_i = y_1 \text{ et } A_j = y_2] \\
&= \frac{1}{|\mathcal{Y}| \left(|\mathcal{Y}| - 1\right)}
\end{aligned}$$

□

3.3.1 Min-entropie d'une sous-chaîne

Une étape très importante de tous les protocoles proposés est la sélection d'un sous-ensemble des bits émis par la source. Un adversaire qui serait incapable d'enregistrer tous les bits de la source aurait une certaine incertitude sur les bits de la source. Le théorème suivant, dû à ZUCKERMAN [58], nous donne une borne sur la min-entropie de la sous-chaîne choisie par les participants.

Théorème 3.3 (Zuckerman). *Soit $X^{\langle n]}$ une variable aléatoire sur l'ensemble $\{0,1\}^n$ tel que $H_\infty(X^{\langle n]}) \geq \delta n$. Soit $\mathcal{S} = \{S_1, S_2, \ldots, S_l\}$ un ensemble de variables aléatoires uniformément distribuées sur $\{1,2,\ldots,n\}$ et 2-indépendant. Soit $\rho = \frac{c\delta}{\lg \delta^{-1}}$ pour une constante c. Soit $\epsilon = \frac{6}{\sqrt{\rho l}}$. Alors avec une probabilité supérieure à $1 - \epsilon$, $\mathcal{S}$ prend une valeur s tel que*

$$H_\infty(X^s) \;\geq\; \rho l.$$

3.4 Hachage interactif

Nous nous intéressons ici à une application toute particulière des fonctions de hachage. Imaginons le scénario suivant : Bob a une chaîne de bits, s, qu'il veut annoncer avec une autre chaîne quelconque à Alice. Alice ne doit pas apprendre laquelle des deux chaînes est s. Jusque-là ce scénario est très simple à réaliser. Maintenant imaginons que Bob a en fait un ensemble préféré de chaînes $\mathcal{P}$ et qu'il voudrait annoncer à Alice deux chaînes qui font partie de $\mathcal{P}$. S'il arrive à faire ceci, alors il arrivera à tricher.

Donc Alice ne peut tout simplement pas accepter deux chaînes de bits choisies par Bob.

Puisqu'Alice ne doit pas apprendre laquelle des deux chaînes est s, une solution simple pour que la seconde chaîne ne fasse pas partie de l'ensemble $\mathcal{S}$ avec bonne probabilité est de la choisir de façon aléatoire parmi toutes les chaînes possibles.

3.4.1 Une première solution

Une première solution à ce problème est d'utiliser les fonctions de hachage. Alice prend une fonction de hachage de m bits dans $m-1$ bits. Cette fonction a la propriété que pour chaque image il existe deux pré-images possibles. Donc Alice choisit une fonction g dans une classe 2-universelle et annonce cette fonction à Bob.

Paramètres

Alice et Bob se mettent d'accord sur les paramètres suivants du protocole :

1° $\mathcal{G}$: une classe 2-universelle de fonctions de hachage de $\{0,1\}^m \to \{0,1\}^{m-1}$.

Protocole 3.1 Hachage simple

1° Alice choisit une fonction $g \in \mathcal{G}$ de façon uniformément aléatoire sur l'ensemble $\mathcal{G}$ et annonce g à Bob.
2° Bob calcule $y = g(s)$ et annonce y à Alice.
3° Alice calcule s_0, s_1 tel que $s_0 \neq s_1$ et $g(s_0) = g(s_1) = y$. Bob calcule $\bar{s}$ tel que $\bar{s} \neq s$ et $g(\bar{s}) = y$.

Vous trouverez à la fin de ce chapitre une version plus schématique de ce protocole.

Le problème est que dans ce cas Bob n'a qu'à chercher une collision dans son ensemble préféré de chaînes $\mathcal{P}$. Le théorème suivant permet d'établir quelle taille $\mathcal{P}$ peut avoir pour qu'il n'y ait pas de collision.

Théorème 3.4. *Soit $\mathcal{G}$ une classe de fonctions de hachage 2-universelle de $\{0,1\}^m$ dans $\{0,1\}^d$. Soit $\mathcal{P} \subseteq \{0,1\}^m$ tel que $|\mathcal{P}| = 2^{\nu m}$ pour $0 < \nu < 1$. Soit $a : \mathcal{G} \to \mathbb{N}$ qui donne le nombre de collisions dans $\mathcal{P}$ pour un $g \in \mathcal{G}$ particulier :*

$$a(g) \ = \ \left|\left\{(x_1,x_2) \in \mathcal{P}^2 \,\middle|\, g(x_1) = g(x_2),\, x_1 < x_2\right\}\right|$$

Soit G une variable aléatoire uniformément distribuée sur $\mathcal{G}$ et soit $A = a(G)$. Alors

$$E[A] \ \leq \ 2^{2\nu m - d - 1}$$

Preuve. Posons

$$c(x_1,x_2) = \begin{cases} |\{g \in \mathcal{G} \mid g(x_1) = g(x_2)\}| & \text{si } x_1 < x_2 \\ 0 & \text{sinon} \end{cases}$$

Puisque que $\mathcal{G}$ est 2-universelle, nous avons pour tout $x_1,x_2 \in \{0,1\}^m$ que $c(x_1,x_2) \leq \frac{|\mathcal{G}|}{2^d}$. Nous avons donc que

$$\begin{aligned} \sum_{g\in\mathcal{G}} a(g) &= \sum_{(x_1,x_2)\in\mathcal{P}^2} c(x_1,x_2) \\ &\leq \frac{1}{2}|\mathcal{P}|^2 \frac{|\mathcal{G}|}{2^d} \end{aligned}$$

et alors

$$\begin{aligned} E[A] &\leq \frac{|\mathcal{P}|^2}{2^{d+1}} \\ &= 2^{2\nu m-d-1}. \end{aligned}$$

□

Corollaire 3.5. *Supposons qu'Alice et Bob font le hachage simple d'une chaîne de* m *bits à une chaîne de* $m-1$ *bits comme décrit dans le protocole 3.1. Soit* $\mathcal{P} \subset \{0,1\}^m$ *tel que* $|\mathcal{P}| = 2^{\nu m}$ *pour* $0 < \nu < 1$*. Si* $\nu < \frac{1}{2} - \frac{r}{2m}$ *alors la probabilité qu'il existe deux éléments distincts* $s_1,s_2 \in \mathcal{P}$ *tel que* $g(x_1) = g(x_2)$ *est inférieure à* 2^{-r}

Preuve. Par le lemme 3.4 nous savons que

$$E[A] \leq 2^{2\nu m-d-1}.$$

Dans le protocole de hachage simple $d = m-1$, donc

$$E[A] \leq 2^{(2\nu-1)m}.$$

Par l'inégalité de Markov (théorème 2.2) et puisque $2\nu < 1 - \frac{r}{m}$ nous avons que

$$\begin{aligned} P[A \geq 1] &\leq P[A \geq 2^{r+(2\nu-1)m}] \\ &\leq 2^{-r}. \end{aligned}$$

□

Le protocole de hachage simple ne fonctionne plus à partir du moment où $\nu \geq \frac{1}{2}$. En modifiant quelque peu le protocole pour le rendre interactif, il y a moyen de faire beaucoup mieux (voir le théorème 3.8).

3.4.2 Protocole de hachage interactif

Le hachage interactif est un protocole entre un challenger Alice et un répondeur Bob. Alice n'a aucune entrée et Bob a une chaîne $s \in \{0,1\}^m$. Le but du protocole est d'isoler deux chaînes: une est la chaîne s entrée par Bob et l'autre est choisie aléatoirement. Alice ne doit pas apprendre quelle chaîne est s.

L'idée originale de ce protocole est dû à NAOR, OSTROVSKY, VENKATESAN et YUNG [42].

Paramètres

Alice et Bob se mettent d'accord sur les paramètres suivants du protocole :

1° $\mathcal{G} = \left\{ g(x) = a \odot x \,\middle|\, a \in \{0,1\}^m \right\}$: une classe 2-universelle de fonctions de hachage de $\{0,1\}^m \to \{0,1\}$.

Protocole 3.2 Hachage interactif

1° Pour $i = 1$ jusqu'à $m - 1$ faire
   (a) Alice choisit une fonction $g_i \in \mathcal{G}$. Soit a_i associé à g_i. Si a_i est linéairement dépendant de $a_1, a_2, \ldots, a_{i-1}$ alors Alice recommence cette étape, sinon elle annonce g_i à Bob.
   (b) Bob calcule $b_i = g_i(s)$ et annonce b_i à Alice.

2° Alice calcule s_0 et s_1 tel que $s_0 \neq s_1$ et $\forall_{i=1,2,\ldots,m-1}\ g_i(s_0) = g_i(s_1) = b_i$.
Bob calcule $\bar{s}$ tel que $\bar{s} \neq s$ et $\forall_{i=1,2,\ldots,m-1}\ g_i(\bar{s}) = b_i$

Vous trouverez à la fin de ce chapitre une version plus schématique de ce protocole.

3.4.3 Preuve de sécurité

Lemme 3.6. *Supposons que nous soyons à la* i^e *étape du protocole de hachage interactif. Soit* $\mathcal{P} \subseteq \{0,1\}^m$ *tel que* $|\mathcal{P}| = 2^{\nu m}$ *pour* $0 < \nu < 1$. *Soit* c *un entier positif tel que* $c \leq \nu m/3$. *Soit* G *une variable aléatoire avec distribution uniforme sur* $\mathcal{G}$. *Alors pour tout* $b \in \{0,1\}$, *avec une probabilité supérieure à* $1 - 2^{-c}$, G *prend un valeur* g *tel que*

$$\frac{\left|\left\{ s \in \mathcal{P} \,\middle|\, g(s) = b \right\}\right|}{|\mathcal{P}|} < \frac{1}{2} + 2^{-c}.$$

Preuve. Posons

$$Z = \left|\left\{ s \in \mathcal{P} \,\middle|\, G(s) = 0 \right\}\right| \qquad \overline{Z} = \left|\left\{ s \in \mathcal{P} \,\middle|\, G(s) = 1 \right\}\right| \tag{3.1}$$

Maintenant posons

$$X = \max\{Z, \overline{Z}\} \qquad\qquad Y = \begin{cases} Z & \text{avec probabilité } \frac{1}{2} \\ \overline{Z} & \text{avec probabilité } \frac{1}{2} \end{cases} \tag{3.2}$$

Nous voulons donc, pour prouver le lemme, montrer que

$$P\left[\frac{X}{|\mathcal{P}|} < \frac{1}{2} + 2^{-c}\right] > 1 - 2^{-c}. \tag{3.3}$$

Maintenant, remarquons que par (3.1)

$$\left|Z - \frac{|\mathcal{P}|}{2}\right| = \left|\overline{Z} - \frac{|\mathcal{P}|}{2}\right|$$

et donc par (3.2) nous avons que

$$\left|X - \frac{|\mathcal{P}|}{2}\right| = \left|Y - \frac{|\mathcal{P}|}{2}\right|.$$

Alors pour tous $\sigma > 0$ nous savons que

$$P\left[\left|X - \frac{|\mathcal{P}|}{2}\right| \geq \sigma\right] = P\left[\left|Y - \frac{|\mathcal{P}|}{2}\right| \geq \sigma\right].$$

Il est donc suffisant pour montrer (3.3) de prouver qu'avec une probabilité inférieure ou égale à 2^{-c}

$$\left|Y - \frac{|\mathcal{P}|}{2}\right| \geq 2^{-c}. \tag{3.4}$$

De la définition de Y, il est clair que $E[Y] = \frac{|\mathcal{P}|}{2}$. Nous voulons montrer que $\mathrm{Var}[Y] \leq \frac{|\mathcal{P}|}{4}$.

$$\begin{aligned}
\mathrm{Var}[Y] &= E[Y^2] - E[Y]^2 \\
&= \frac{E[Z^2]}{2} + \frac{E[\overline{Z}^2]}{2} - \left(\frac{|\mathcal{P}|}{2}\right)^2 \\
&= \frac{E[(Z + \overline{Z})^2 - 2Z\overline{Z}]}{2} - \frac{|\mathcal{P}|^2}{4} \\
&= \frac{|\mathcal{P}|^2}{2} - \frac{|\mathcal{P}|^2}{4} - E[Z\overline{Z}] \\
&= \frac{|\mathcal{P}|^2}{4} - E[Z\overline{Z}].
\end{aligned}$$

Il ne reste plus qu'à trouver la valeur de $E[Z\overline{Z}]$. Posons

$$K_{s_1,s_2} = \begin{cases} 1 & \text{si } G(s_1) \neq G(s_2) \\ 0 & \text{sinon} \end{cases}$$

et la somme

$$K = \sum_{(s_1,s_2)\in\mathcal{P}^2} K_{s_1,s_2} = \left|\left\{(s_1,s_2) \in \mathcal{P}^2 \,\middle|\, G(s_1) \neq G(s_2)\right\}\right|.$$

Remarquons que $K = Z\overline{Z} + \overline{Z}Z$, puisque pour une paire $(s_1,s_2) \in \mathcal{P}^2$ qui contribue à la somme de K, il y a Z choix de s_1 et $\overline{Z}$ choix de s_2 si $G(s_1) = 0$ et $G(s_2) = 1$ et inversement si $G(s_1) = 1$ et $G(s_2) = 0$. Nous avons donc

$$E[Z\overline{Z}] = \frac{E[K]}{2}. \tag{3.5}$$

De plus

$$E[K] = \sum_{(s_1,s_2)\in\mathcal{P}^2} E[K_{s_1,s_2}]. \tag{3.6}$$

Si $s_1 = s_2$ alors $E[K_{s_1,s_2}] = 0$, sinon $E[K_{s_1,s_2}] > \frac{1}{2}$ car G est choisie à chaque étape du protocole parmi une classe de fonctions de hachage 2-universelle.

Nous supposons que nous sommes à la i^e étape du protocole. Alice a déjà envoyé les fonctions $g_1, g_2, \ldots, g_{i-1}$ représentées par les chaînes $a_1, a_2, \ldots, a_{i-1} \in \{0,1\}^m$. La chaîne $a_i \in \{0,1\}^m$ doit être linéairement indépendante des chaînes précédentes $a_1, a_2, \ldots, a_{i-1}$. Il ne reste donc que $2^m - 2^{i-1}$ chaînes valides parmi lesquelles a_i doit être choisie. De plus, les chaînes dans $\mathcal{P}$ ont la propriété que

$$\forall_{j=1,2,\ldots,i-1} \forall_{(s_1,s_2)\in\mathcal{P}^2} \quad a_j s_1 = a_j s_2 \tag{3.7}$$

c'est-à-dire qu'elles sont consistantes avec les réponses de Bob aux questions d'Alice pour les étapes 1 à $i-1$.

Soit $(s_1,s_2) \in \mathcal{P}^2$ une paire de chaînes dans l'ensemble préféré de Bob. Nous voulons maintenant compter le nombre de chaînes a_i qui séparent la paire (s_1,s_2), c'est-à-dire qui font qu'après la i^e étape du protocole elles ne seront plus consistantes. Par la propriété (3.7)

$$\left\{a_i \in \{0,1\}^m \,\middle|\, (s_1,s_2) \in \mathcal{P}^2, a_i s_1 = a_i s_2\right\} = 2^{m-1} - 2^{i-1}$$

comparativement à

$$\left\{a_i \in \{0,1\}^m \,\middle|\, (s_1,s_2) \in \mathcal{P}^2,\, a_i s_1 \neq a_i s_2\right\} \;=\; 2^{m-1}$$

Donc

$$\begin{aligned} E[K_{s_1,s_2}] &= P[a_i s_1 \neq a_i s_2] \\ &= \frac{2^m}{2^{m-1} - 2^i} \\ &> \frac{1}{2} \end{aligned}$$

Nous avons alors que

$$E[K] \;\geq\; \frac{1}{2}\left|\left\{(s_1,s_2) \in \mathcal{P}^2 \,\middle|\, s_1 \neq s_2\right\}\right|$$

ce qui implique que

$$E[Z\overline{Z}] \;\geq\; \frac{|\mathcal{P}|^2 - |\mathcal{P}|}{4}$$

et donc

$$\begin{aligned} \mathrm{Var}[Y] &= \frac{|\mathcal{P}|^2}{4} - E[Z\overline{Z}] \\ &\leq \frac{|\mathcal{P}|^2}{4} - \frac{|\mathcal{P}|^2 - |\mathcal{P}|}{4} \\ &= \frac{|\mathcal{P}|}{4}. \end{aligned}$$

Par le théorème de Chebychef (théorème 2.3) nous avons que

$$P\left[\left|Y - \frac{|\mathcal{P}|}{2}\right| \geq \sigma\right] \;\leq\; \frac{|\mathcal{P}|}{4\sigma^2}.$$

En substituant σ par $\sqrt{2^c\,|\mathcal{P}|/4}$ nous avons que

$$P\left[\left|Y - \frac{|\mathcal{P}|}{2}\right| \geq 2^{\frac{c+\nu m-2}{2}}\right] \;\leq\; 2^{-c}$$

et nous avons qu'avec une probabilité supérieure à $1 - 2^{-c}$

$$\begin{aligned} \frac{Y}{|\mathcal{P}|} &\leq \frac{1}{2} + 2^{\frac{c+\nu m-2}{2} - \nu m} \\ &< \frac{1}{2} + 2^{-c}. \end{aligned}$$

□

Lemme 3.7. *Soit $\mathcal{P} \subseteq \{0,1\}^m$ tel que $|\mathcal{P}| = 2^{\nu m}$ pour $0 < \nu < 1$. Soit $c,d \leq m$ deux entiers positifs tel que $2\nu m < d - c$. Soit $\mathcal{G} : \{0,1\}^m \to \{0,1\}^d$ une classe de fonctions de hachage 2-universelles. Soit G une variable aléatoire avec distribution uniforme sur $\mathcal{G}$. Alors la probabilité que G prenne une valeur g tel qu'il existe $s_1,s_2 \in \mathcal{P}$ distincts tel que $g(s_1) = g(s_2)$ est au plus 2^{-c}.*

Preuve. Soit la fonction $a : \mathcal{G} \to \mathbb{N}$ qui donne le nombre de collisions dans $\mathcal{P}$ pour un g particulier :

$$a(g) \;=\; \left|\left\{(s_1,s_2) \in \mathcal{P}^2 \;\middle|\; g(s_1) = g(s_2),\, s_1 < s_2\right\}\right|$$

et soit la variable aléatoire $A = a(G)$. Par le lemme 3.4 nous savons que

$$E[A] \;\leq\; 2^{2\nu m - d - 1}.$$

Par l'inégalité de Markov (théorème 2.2), nous avons

$$\begin{aligned} P[A \geq 1] \;&\leq\; P[A \geq 2^{c+2\nu m-d-1}] \\ &\leq\; 2^{-c} \end{aligned}$$

puisque $2\nu m < d - c$. □

Théorème 3.8. *Supposons qu'Alice et Bob font le hachage interactif d'une chaîne de m bits à une chaîne de $m - 1$ bits comme décrit dans le protocole 3.2. Soit $\mathcal{P} \subset \{0,1\}^m$ tel que $|\mathcal{P}| = 2^{\nu m}$ pour $0 < \nu < 1$. Soit $r \geq \lg m$. Si $\nu < 1 - \frac{8r+4}{m}$ alors la probabilité que Bob puisse répondre aux questions d'Alice tel qu'il existe deux éléments distincts $s_1,s_2 \in \mathcal{P}$ consistants avec les réponses de Bob est inférieure à 2^{-r}.*

Preuve. Posons $\mathcal{P}_0 = \mathcal{P}$ et pour $j = 1,\ldots,m-1$ posons

$$\mathcal{P}_j \;=\; \left\{s \in \mathcal{P}_{j-1} \;\middle|\; g_j(s) = b_j\right\}$$

Les $\mathcal{P}_j$ représentent l'évolution de l'ensemble préféré de Bob.

Tant que $\mathcal{P}_j$ est assez grand, nous pouvons borner la taille de $\mathcal{P}_{j+1}$ par le lemme 3.6. Ensuite, nous appliquons le lemme 3.7 pour les étapes restantes. Posons $c = 2r$ et soit j_t un entier tel que

$$\nu m - 3c + 1 \;\geq\; j_t \;>\; \nu m - 3c \tag{3.8}$$

qui marquera la transition.

Par induction sur $j = 1,2,\ldots,j_t$ en utilisant le lemme 3.6 nous avons que

$$|\mathcal{P}_j| \leq \left(\frac{1}{2} + 2^{-c}\right)^j |\mathcal{P}|$$

sauf avec une probabilité d'au plus $j2^{-j}$. Nous avons que

$$|\mathcal{P}_{j_t}| \leq 2^{\nu m - j_t}\left(1 + 2^{-c+1}\right)^{j_t}$$

et donc

$$\lg|\mathcal{P}_{j_t}| \leq (\nu m - j_t) + j_t \lg\left(1 + 2^{-c+1}\right) < 3c + 1 \tag{3.9}$$

par (3.8) et par le fait que

$$j_t \lg\left(1 + 2^{-c+1}\right) < m2^{-c} < 1.$$

Pour pouvoir appliquer le lemme 3.7 il faut montrer que

$$2\lg|\mathcal{P}_{j_t}| \leq (m - 1 - j_t) - c \tag{3.10}$$

Puisque $\nu < 1 - \frac{4c+4}{m}$ implique que $4c < m - \nu m - 4$ nous avons que

$$\begin{aligned}
2\lg|\mathcal{P}_{j_t}| &< 6c + 2 && \text{par (3.9)}\\
&< 2c + m - \nu m - 2 &&\\
&= m - (\nu m - 3c) - c - 2 &&\\
&< (m - 1 - j_t) - c && \text{par (3.8)}
\end{aligned}$$

et donc l'inégalité (3.10) tient.

La probabilité que le protocole ne fonctionne pas est au plus

$$\begin{aligned}
(j_t + 1)2^{-c} &< m2^{-c}\\
&< 2^{-r}.
\end{aligned}$$

□

Il est donc possible avec le hachage interactif de dépasser la borne $\nu \geq \frac{1}{2}$ (voir le corollaire 3.5).

Alice		Bob(s)
$g \in_{\mathcal{R}} \mathcal{G}$		
	$\xrightarrow{g}$	
		$y = g(s)$
	$\xleftarrow{y}$	
s_0, s_1 tel que $s_0 \neq s_1$ et $g(s_0) = g(s_1) = y$		$\bar{s}$ tel que $\bar{s} \neq s$ et $g(\bar{s}) = y$
s_0, s_1		$s, \bar{s}$

FIG. 3.1 – *Protocole de hachage simple*

Alice		Bob(s)
$g_1 \in_{\mathcal{R}} \mathcal{G}$		
	$\xrightarrow{g_1}$	
		$b_1 = g_1(s)$
	$\xleftarrow{b_1}$	
$\vdots$	$\vdots$	$\vdots$
$g_{m-1} \in_{\mathcal{R}} \mathcal{G} \backslash [a_1, \ldots, a_{m-2}]$		
	$\xrightarrow{g_{m-1}}$	
		$b_{m-1} = g_{m-1}(s)$
	$\xleftarrow{b_{m-1}}$	
s_0, s_1 tel que $s_0 \neq s_1$ et $\forall_i\ g_i(s_0) = g_i(s_1) = b_i$		$\bar{s}$ tel que $\bar{s} \neq s$ et $\forall_i\ g_i(\bar{s}) = b_i$
s_0, s_1		$s, \bar{s}$

FIG. 3.2 – *Protocole de hachage interactif*

CHAPITRE 4

Partage de clef

DANS ce chapitre, nous montrons comment Alice et Bob peuvent accomplir un partage de clef de façon inconditionnellement sécuritaire, sous la seule hypothèse qu'Ève a un espace mémoire borné. Deux protocoles – variations sur le même thème – sont proposés pour des scénarios légèrement différents.

Nous supposons qu'une source émet N bits uniformément distribués dont Alice et Bob sélectionnent et enregistrent n bits. De façon équivalente, Alice peut générer et envoyer les N bits à Bob. Pour tout $\gamma < 1$, si Ève ne peut enregistrer plus que γN bits, alors il lui est impossible de tricher sauf avec une faible probabilité. Alice et Bob ont tous deux besoin d'un espace mémoire de taille $\theta(n)$ pour accomplir les tâches du protocole.

Le premier protocole suppose qu'Alice et Bob partagent déjà une clef initiale. Cette hypothèse est relativement faible, puisqu'ils ont, de toute façon, besoin d'authentifier les messages qu'ils s'échangeront publiquement[1]. Le protocole a donc pour but de générer une clef plus grande que celle qu'ils ont déjà.

Le deuxième protocole ne suppose pas qu'Alice et Bob partagent déjà une petite clef. Ils veulent donc essayer de s'entendre, en discutant publiquement, sur une chaîne au sujet de laquelle Ève n'a aucune information. Par contre, il est nécessaire de supposer qu'Alice et Bob ont à leur disposition un canal authentifié pour discuter publiquement.

Les protocoles présentés et leurs analyses sont ceux de CACHIN et MAURER [17].

4.1 Protocoles

L'idée de base des deux protocoles est fort simple. Lorsque la source émet les bits uniformément distribués, Alice et Bob sélectionnent un sous-ensemble de ces bits et enregistrent leur valeur. Comment ils accomplissent ceci est expliqué en détail dans les protocoles. Puisque les bits sont sélectionnés au hasard, si Ève ne peut enregistrer qu'une fraction des bits de la source, elle ne connaîtra alors qu'environ la même fraction des bits des sous-ensembles d'Alice et de Bob. Par la distillation de secret (voir la

1. Voir la section 1.1 pour plus de détails.

section 3.2) Alice et Bob vont pouvoir éliminer toute l'information partielle qu'Ève a au sujet des bits de leur sous-ensemble.

4.1.1 Recyclage de clef

Dans le scénario suivant Alice et Bob partagent déjà une petite clef secrète k_0. Ils pourront utiliser cette clef initiale pour, entre autres, authentifier les messages qu'ils s'échangeront sur le canal public.

De la clef secrète initiale k_0 qu'ils partagent, Alice et Bob vont donner la description du sous-ensemble de bits qu'ils vont choisir de la source. La description du sous-ensemble doit être compacte, car la description d'un sous-ensemble de n éléments choisi uniformément aléatoirement d'un ensemble de N bits requiert $n \lg N$ bits ce qui est plus de bits que ce qu'Alice et Bob tireront de la sélection. Au prix de l'indépendance des bits choisis, Alice et Bob décriront leur ensemble grâce à la technique décrite à la section 3.3.

Paramètres

Alice et Bob se mettent d'accord sur les paramètres suivants du protocole :

1° n : le nombre de bits choisis et mémorisés par Alice et Bob.

2° N : le nombre de bit émis par la source.

3° $\mathcal{G}$: une classe fortement 2-universelle (voir l'exemple 3.2) de fonctions de hachage de $\{0,1\}^{\lg N} \to \{0,1\}^{\lg N}$.

4° ℓ : la longueur de la clef résultante.

5° $\mathcal{F}$: une classe 2-universelle de fonctions de hachage de $\{0,1\}^n \to \{0,1\}^\ell$.

Protocole 4.1 Recyclage de clef

1° Alice et Bob s'entendent sur $g \in \mathcal{G}$ une fonction de hachage fortement 2-universelle tirée des bits de leur clef secrète initiale k_0.

2° Alice (ou une source indépendante) émet N bits uniformément distribués $r_1, r_2, \ldots, r_N$ (noté $r^{\langle N]}$). Alice et Bob sélectionnent n bits aux positions $\mathcal{I} = \{g(1), g(2), \ldots, g(n)\}$. La sous-chaîne de $r^{\langle N]}$ ainsi formée par l'ensemble $\mathcal{I}$ est notée $r^{\mathcal{I}}$.

3° Alice choisit une fonction de hachage $f \in \mathcal{F}$ de façon uniformément aléatoire et l'annonce à Bob.

4° Alice et Bob calculent $k = f(r^{\mathcal{I}})$.

Vous trouverez à la fin de ce chapitre une version plus schématique de ce protocole.

4.1.2 Partage de clef par discussion publique

Dans ce second scénario, Alice et Bob n'ont pas d'information commune de laquelle ils peuvent tirer la description d'un ensemble de bits à sélectionner. Ils sélectionneront donc des bits chacun de leur coté et garderont seulement ceux qu'ils auront en commun. Pour ce faire, ils devront s'annoncer les ensembles de bits qu'ils ont choisis et, encore une fois, ils devront utiliser une courte description de leur ensemble.

La taille de l'intersection des ensembles d'Alice et de Bob risque d'être quelque peu plus petite. Mais ceci ne pose aucun problème puisque qu'après avoir réalisé ce protocole une fois, Alice et Bob peuvent exécuter le protocole 4.1 à loisir pour générer une clef aussi grande qu'ils le veulent.

Paramètres

Alice et Bob se mettent d'accord sur les paramètres suivants du protocole :

1º n : le nombre de bits choisis et mémorisés par Alice et Bob.

2º N : le nombre de bits émis par la source.

3º $\mathcal{G}$: une classe fortement 2-universelle (voir l'exemple 3.2) de fonctions de hachage de $\{0,1\}^{\lg N} \to \{0,1\}^{\lg N}$.

4º ℓ : la longueur de la clef résultante.

5º $\mathcal{F}$: une classe 2-universelle de fonctions de hachage de $\{0,1\}^n \to \{0,1\}^\ell$.

Protocole 4.2 Partage de clef par discussion publique

1º Alice choisit de façon uniformément aléatoire une fonction de hachage g_a dans l'ensemble $\mathcal{G}$. Bob fait de même et choisit de façon uniformément aléatoire une fonction de hachage g_b dans l'ensemble $\mathcal{G}$.

2º Alice (ou une source indépendante) émet N bits uniformément distribués $r_1, r_2, \ldots, r_N$ (noté $r^{\langle N]}$). Alice choisit n bits aux positions $\mathcal{A} = \{g_a(1), g_a(2), \ldots, g_a(n)\}$, Bob choisit n bits aux positions $\mathcal{B} = \{g_b(1), g_b(2), \ldots, g_b(n)\}$.

3º Alice annonce g_a à Bob et Bob annonce g_b à Alice. Alice et Bob calculent $\mathcal{I} = \mathcal{A} \cap \mathcal{B}$.

4º Alice choisit une fonction de hachage f de façon uniformément aléatoire de l'ensemble $\mathcal{F}$ et l'annonce à Bob.

5º Alice et Bob calculent $K = f(r^{\mathcal{I}})$.

Vous trouverez à la fin de ce chapitre une version plus schématique de ce protocole.

4.2 Preuve de sécurité

La preuve de sécurité consiste à montrer que si Ève ne peut pas enregistrer tous les bits de la chaîne $r^{\langle N]}$ alors elle n'a aucune information sur la clef k que partagent Alice et Bob à la fin du protocole.

Nous représentons la chaîne $r^{\langle N]}$ diffusée à Alice et Bob par la variable aléatoire $R^{\langle N]}$ uniformément distribuée sur $\{0,1\}^N$. Pendant la diffusion de $R^{\langle N]}$, Ève peut calculer n'importe quelle fonction (même probabiliste) de $\{0,1\}^N$ dans $\mathcal{V}$ tel que $\lg|\mathcal{V}| \leq \gamma N$. Nous décrivons l'information qu'Ève a sur la chaîne $R^{\langle N]}$ par la variable aléatoire V.

La preuve de sécurité est divisée en trois parties. Premièrement, nous donnons une borne inférieure sur la min-entropie d'Ève sur la chaîne $R^{\langle N]}$, étant donné qu'elle apprend $V = v$. Deuxièmement, nous montrons qu'avec une bonne probabilité, la min-entropie d'Ève sur la sous-chaîne $R^{\mathcal{I}}$ qu'Alice et Bob partagent est grande. Finalement, nous appliquons le théorème de distillation de secret (théorème 3.1) sur la chaîne $R^{\mathcal{I}}$ pour montrer qu'Ève n'a presque qu'aucune information sur la clef résultante $K = F(R^{\mathcal{I}})$ où F est une variable aléatoire uniformément distribuée sur $\mathcal{F}$.

Dans le premier protocole, Ève n'apprend jamais quel est l'ensemble $\mathcal{I}$. Pour les fins de la preuve nous supposons qu'Ève apprend d'un oracle l'ensemble $\mathcal{I}$ *après* la diffusion des bits par la source. Dans le premier protocole, $|\mathcal{I}| = n$ et dans le deuxième protocole, $|\mathcal{I}| \approx \frac{n}{N}n$.

Théorème 4.1. *Soient ϵ_1,ϵ_2 des paramètres de sécurité. Si l'espace mémoire d'Ève est inférieur à γN pour un $\gamma < 1$, alors, pour un n suffisamment grand, l'information qu'Ève apprend sur la clef $K = F(R^{\mathcal{I}})$ est négligeable avec une probabilité supérieure à $1 - \epsilon_1 - \epsilon_2 - \frac{6}{\sqrt{\rho|\mathcal{I}|}}$. Formellement*

$$P\left[H(K) - H(K \mid F, V = v) \leq \frac{\epsilon_2}{\ln 2}\right] \;>\; 1 - \epsilon_1 - \epsilon_2 - \frac{6}{\sqrt{\rho\,|\mathcal{I}|}}$$

Preuve. Puisque $R^{\langle N]}$ est uniformément distribué sur $\{0,1\}^N$ alors

$$H_\infty(R^{\langle N]}) \;=\; N$$

L'information qu'Ève apprend sur $R^{\langle N]}$ est représentée par la variable aléatoire V tel que $\lg|\mathcal{V}| \leq \gamma N$. Par le lemme 2.9 nous avons que

$$H_\infty(R^{\langle N]} \mid V = v) \;\geq\; N - \gamma N - \lg\frac{1}{\epsilon_1} \tag{4.1}$$

avec une probabilité supérieure à $1 - \epsilon_1$.

Posons $\delta = (1 - \gamma - \frac{1}{N} \lg \frac{1}{\epsilon_1})$ et $\rho = c\delta \lg \delta^{-1}$. Par le théorème de Zuckerman (théorème 3.3) nous avons que

$$\mathsf{H}_\infty(\mathsf{R}^{\mathcal{I}} \mid \mathsf{V} = \nu) \;\geq\; \rho\,|\mathcal{I}| \tag{4.2}$$

avec une probabilité supérieure à $1 - \frac{6}{\sqrt{\rho|\mathcal{I}|}}$.

Posons $\ell = \rho\,|\mathcal{I}| - \lg \epsilon_2$. Par le théorème de distillation de secret (théorème 3.1) nous avons que

$$\mathsf{H}(\mathsf{K} \mid \mathsf{G},\mathsf{V} = \nu) \;\geq\; \ell - \frac{\epsilon_2}{\ln 2}$$

Donc avec une probabilité supérieure à $1 - \epsilon_1 - \epsilon_2 - \frac{6}{\sqrt{\rho|\mathcal{I}|}}$ nous avons que

$$\mathsf{H}(\mathsf{K}) - \mathsf{H}(\mathsf{K} \mid \mathsf{G},\mathsf{V} = \nu) \;\leq\; \frac{\epsilon_2}{\ln 2}$$

Puisqu'Ève n'apprend que ν et g alors son entropie sur la chaîne K est presque maximale. $\square$

$\text{Alice}(k_0)$		$\text{Bob}(k_0)$
$g \leftarrow k_0$		$g \leftarrow k_0$
$\mathcal{A} = \{g(1), \ldots, g(n)\}$		$\mathcal{B} = \{g(1), \ldots, g(n)\}$
$r^{\langle N]} \in_{\mathcal{R}} \{0,1\}^N$		
	$\xrightarrow{r^{\langle N]}}$	
enregistre $r^{\mathcal{A}}$		enregistre $r^{\mathcal{B}}$
$f \in_{\mathcal{R}} \mathcal{F}$		
	$\xrightarrow{f}$	
$k = f(r^{\mathcal{A}})$		$k = f(r^{\mathcal{B}})$
k		k

FIG. 4.1 – *Protocole de recyclage de clef*

Alice		Bob
$g_a \in_{\mathcal{R}} \mathcal{G}$		$g_b \in_{\mathcal{R}} \mathcal{G}$
$\mathcal{A} = \{g_a(1), \ldots, g_a(N)\}$		$\mathcal{B} = \{g_b(1), \ldots, g_b(N)\}$
$r^{\langle N]} \in_{\mathcal{R}} \{0,1\}^N$		
	$\xrightarrow{r^{\langle N]}}$	
enregistre $r^{\mathcal{A}}$		enregistre $r^{\mathcal{B}}$
	$\xrightarrow{g_a}$	
	$\xleftarrow{g_b}$	
$f \in_{\mathcal{R}} \mathcal{F}$		
	$\xrightarrow{f}$	
$k = f\left(r^{\mathcal{A} \cap \{g_b(1), \ldots, g_b(N)\}}\right)$		$k = f\left(r^{\mathcal{B} \cap \{g_a(1), \ldots, g_a(N)\}}\right)$
k		k

FIG. 4.2 – *Protocole de partage de clef par discussion publique*

CHAPITRE 5

Transfert inconscient

DANS ce chapitre, nous montrons comment Alice et Bob peuvent accomplir un transfert inconscient de façon inconditionnellement sécuritaire, sous la seule hypothèse que le receveur (Bob) a un espace mémoire borné.

Nous supposons qu'une source émet $N = n^{2-\alpha-\beta}$ bits uniformément distribués, où α et β sont des paramètres tel que $0 < \alpha < \beta < \frac{1}{2}$ et n est le nombre de bits qu'Alice et Bob sélectionnent de la source. De façon équivalente, Alice peut générer et envoyer les N bits à Bob. Pour tout $\gamma < 1$, si Bob ne peut enregistrer plus que γN bits, alors pour un n assez grand, il lui est impossible de tricher sauf avec une probabilité bornée par l'inverse d'un polynôme en n[1]. Dans tous les cas, il est impossible pour Alice de tricher. Alice et Bob ont tous deux besoin d'un espace mémoire de taille $\theta(n^{2-2\alpha})$ pour accomplir les tâches du protocole.

Le protocole présenté est celui de CACHIN, CRÉPEAU et MARCIL [15]. La preuve de sécurité comporte cependant quelques modifications : le lemme 5.3 tel qu'énoncé dans [15, lemme 9] n'est pas vrai, le lemme et sa preuve ont donc été corrigés.

5.1 Protocoles

Avant de présenter le protocole pour accomplir un transfert inconscient dans le modèle des participants à mémoire bornée, nous étudierons un autre protocole : la réduction de transfert inconscient au transfert équivoque. La raison est que l'idée de base de cette réduction est utilisée dans notre protocole[2].

5.1.1 Réduction de transfert inconscient au transfert équivoque

En utilisant n transferts équivoques, Alice et Bob peuvent accomplir un transfert inconscient. Le protocole présenté est celui de CRÉPEAU [23]. Voici l'idée de base de

1. La probabilité que Bob triche n'est pas exponentiellement faible en n, mais il est possible de construire un transfert inconscient exponentiellement sécuritaire à partir de plusieurs transferts inconscients qui ne le sont pas. Consultez CRÉPEAU et KILIAN [26] et DAMGÅRD, KILIAN et SALVAIL [28] pour plus de détails.

2. Notons que cette même idée est également utilisée par CRÉPEAU [24] pour construire un transfert inconscient à partir d'un canal bruyant.

cette réduction : Alice envoie à Bob n bits par le transfert équivoque. Bob reçoit environ la moitié des bits transmis par Alice, sans qu'Alice ne sache lesquels Bob a reçus.

À partir de ces bits, Bob forme deux sous-ensembles disjoints – un *bon* et un *mauvais* – de taille $\frac{n}{3}$. Le bon ensemble ne contient que des bits dont Bob connaît la valeur. Le mauvais ensemble contient au moins un bit qu'il ne connaît pas. Puisqu'il n'apprend qu'environ la moitié des bits transmis par Alice, Bob ne peut former deux bons ensembles, sauf avec probabilité exponentiellement faible par la loi des grands nombres. Nous tirons avantage du fait que Bob ne connaît que la parité du bon ensemble.

Bob annonce à Alice les ensembles qu'il a formés en lui annonçant les positions des bits des deux ensembles, sans lui dire lequel est le bon et lequel est le mauvais. Ensuite, Alice annonce ses deux bits b_0 et b_1 en les dissimulant avec les bits de parité des sous-ensembles formés par Bob. Puisque Bob ne connaît que le bit de parité du bon ensemble, il n'apprend qu'un seul des bits d'Alice. Puisqu'Alice ne sait pas quel est le bon ensemble, elle n'apprend pas quel bit Bob apprend.

Notons que la réduction fonctionne également dans le cas où la probabilité p que Bob reçoive le bit transmis par Alice est $0 < p < 1$. Dans ce cas, il faut que Bob forme des sous-ensembles de taille $n \cdot \min\left\{\frac{2}{3}p, \frac{1}{2}\right\}$. Il est par contre essentiel que la probabilité soit fixe et connue de Bob.

Le protocole est présenté sans preuve formelle de sécurité. Il est surtout important de comprendre l'idée du *bon* et du *mauvais* sous-ensemble, puisque c'est cet artifice qui sera utilisé plus tard dans le protocole de transfert inconscient avec des participants limités en espace mémoire.

Protocole 5.1 Réduction de transfert inconscient au transfert équivoque

1° Alice envoie à Bob n bits uniformément distribués $t_1, t_2, \ldots, t_n$.

2° Bob forme deux ensembles disjoints $\mathcal{U}_0$ et $\mathcal{U}_1$ tous deux de taille $\frac{n}{3}$ tel qu'il connaît tous les bits de t_i tel que $i \in \mathcal{U}_c$. Bob annonce $\mathcal{U}_0$ et $\mathcal{U}_1$ à Alice.

3° Alice calcule les bits $z_0 = b_0 \oplus \left(\bigoplus_{j \in \mathcal{U}_0} t_j\right)$ et $z_1 = b_1 \oplus \left(\bigoplus_{j \in \mathcal{U}_1} t_j\right)$. Alice annonce z_0, z_1 à Bob.

4° Bob calcule $b_c = z_c \oplus \left(\bigoplus_{j \in \mathcal{U}_c} t_j\right)$

Vous trouverez à la fin de ce chapitre un version plus schématique de ce protocole.

5.1.2 Transfert inconscient

Nous essayons de reproduire l'idée du *bon* et du *mauvais* sous-ensemble. Lorsque la source émet les bits uniformément aléatoires, Alice et Bob sélectionnent n bits de la source et enregistrent leur valeur. Ensuite, Alice annonce à Bob l'ensemble des bits qu'elle a choisi (seulement les positions, pas les bits eux-mêmes). Dans le cas où Bob est honnête et enregistre seulement n bits de la source, nous nous retrouvons dans un scénario fort similaire à la réduction précédente : Bob connaît une certaine proportion $n\frac{n}{N} = n^{\alpha+\beta}$ des bits de l'ensemble qu'Alice a choisis. De façon équivalente, Alice aurait pu envoyer ses bits à Bob par un transfert équivoque où la probabilité que Bob reçoive chacun des bits d'Alice est $\frac{n}{N}$.

Mais le scénario n'est pas équivalent, car la proportion des bits que Bob connaît sera différente dans le cas où il est honnête par rapport au cas où il est malhonnête. Donc la réduction ne fonctionne pas directement, mais il est possible, en travaillant un peu, d'arriver à avoir créer un bon ensemble et un mauvais ensemble.

Supposons que Bob malhonnête enregistre γN des bits de la source. Alors, pour s'assurer qu'il n'annonce pas à Alice deux bons ensembles, c'est-à-dire deux ensembles dont il connaît la parité, nous le forçons à annoncer un des deux ensembles de façon aléatoire parmi tous les ensembles possibles. Avec une très bonne probabilité, cet ensemble sera mauvais. Nous accomplissons ceci à l'aide du hachage interactif (section 3.4) et de l'encodage des sous-ensembles de k éléments à des chaînes binaires (annexe A).

Puisque nous permettons à Bob de calculer n'importe quelle fonction des bits de la source, il faut effectuer, avant le hachage interactif, une distillation de secret sur des blocs de bits. Nous nous retrouvons ensuite dans un scénario où Bob ne peut avoir que de l'information sur certains bits.

Vous trouverez à la fin de ce chapitre un version plus schématique de ce protocole.

Paramètres

Alice et Bob se mettent d'accord sur les paramètres suivants du protocole :

1º α,β tel que $0 < \beta < \alpha < \frac{1}{2}$: les paramètres de mémoire.

2º n : le nombre de bits choisis et mémorisés par Alice et Bob.

3º $N = n^{2-\alpha-\beta}$: le nombre de bit émis par la source.

4º $m = n^{1-\alpha}$: le nombre de blocs (et de bits $t_1, t_2, \ldots, t_m$).

5º $\ell = n^{\alpha}$: la longueur d'un bloc.

6º $k = n^{\beta}$: le nombre de blocs dont Bob connaît tous les bits.

7º $M = \lceil \lg \binom{m}{k} \rceil = \lceil \lg \binom{n^{1-\alpha}}{n^{\beta}} \rceil \leq n^{1-\alpha} h(n^{\alpha+\beta-1})$: la longueur de l'encodage d'un sous-ensemble de k éléments $\{1,2,\ldots,m\}$.

8º $\mathcal{F}$: une classe de fonctions de hachage 2-universelle de $\{0,1\}^{\ell} \to \{0,1\}$.

Protocole 5.2 Transfert inconscient

1º Alice (ou une source indépendante) émet N bits uniformément distribués $r_1,r_2,\dots,r_N$ (notés $r^{\langle N]}$). Alice choisit n bits aux positions $\mathcal{A} = \{a_1,a_2,\dots,a_n\}$ et Bob choisit n bits aux positions $\mathcal{B} = \{b_1,b_2,\dots,b_n\}$. $\mathcal{A}$ et $\mathcal{B}$ sont des ensembles de n entiers distincts choisis de façon uniformément aléatoire parmi $\{1,2,\dots,N\}$. Les deux sous-chaînes de $r^{\langle N]}$ ainsi formées par les ensembles $\mathcal{A}$ et $\mathcal{B}$ sont notées $r^{\mathcal{A}}$ et $r^{\mathcal{B}}$.

2º Alice annonce $\mathcal{A}$ à Bob. Avec les bits de $r^{\mathcal{A}}$, Bob forme m blocs $x_1,x_2,\dots,x_m$ de ℓ bits chacun tel qu'il y est au moins k blocs dont tous les bits se trouvent dans $\mathcal{A} \cap \mathcal{B}$ (donc dont Bob connaît les bits). Si ce n'est pas possible (car l'intersection contient moins que $k\ell$ bits) alors il «*abort*». Soit $\mathcal{S}$ l'ensemble des blocs dont tous les bits font partie de l'intersection. Formellement, Bob construit une permutation de l'ensemble $\mathcal{A}$, soit $\pi : \{1,2,\dots,n\} \to \{1,2,\dots,n\}$ et soit

$$\mathcal{C}_i = \{a_{\pi(\ell(i-1)+1)}, a_{\pi(\ell(i-1)+2)}, \dots, a_{\pi(\ell i)}\}$$

pour $i = 1,2,\dots,m$ tel que $\forall_{i\in\mathcal{S}}\ C_i \subset \mathcal{A} \cap \mathcal{B}$. Bob annonce π à Alice.

3º Alice groupe ses n bits $r^{\langle\mathcal{A}]}$ dans les blocs $r^{\mathcal{C}_1}, r^{\mathcal{C}_2}, \dots, r^{\mathcal{C}_m}$. Alice choisit m fonctions de hachage $f_1,f_2,\dots,f_m$ de façon indépendante et uniformément aléatoire de l'ensemble $\mathcal{F}$ et les annonce à Bob. Alice calcule les bits $t_1,t_2,\dots,t_m$ où $t_i = f_i(r^{\mathcal{C}_i})$.

4º Bob calcule les bits $y_i = f_i(r^{\mathcal{C}_i})$ pour $i \in \mathcal{S}$. Bob calcule aussi la chaîne de bits $s = \sigma(\mathcal{S})$ de longueur M (voir annexe A).

5º Alice et Bob font un hachage interactif de la chaîne s à la chaîne w de longueur $M-1$ (comme décrit à la section 3.4). Alice calcule deux ensembles $\mathcal{U}_0,\mathcal{U}_1 \subset \{1,2,\dots,m\}$ tel que les chaînes $\sigma(\mathcal{U}_0)$ et $\sigma(\mathcal{U}_1)$ ont la même valeur de hachage et $\mathcal{U}_0 < \mathcal{U}_1$ selon un ordre préétabli. Si ce n'est pas possible car une des chaînes qui résulte du hachage interactif n'encode pas un sous-ensemble valide, alors Bob «*abort*».

6º Bob connaît aussi les ensembles $\mathcal{U}_0$ et $\mathcal{U}_1$. Il choisit un bit c' tel que $\mathcal{U}_{c\oplus c'} = \mathcal{S}$ et envoie c' à Alice. Alice calcule les bits $z_0 = b_0 \oplus \left(\bigoplus_{j\in\mathcal{U}_{c'}} t_j\right)$ et $z_1 = b_1 \oplus \left(\bigoplus_{j\in\mathcal{U}_{c'\oplus 1}} t_j\right)$. Alice annonce z_0,z_1 à Bob.

7º Bob calcule $b_c = z_c \oplus \left(\bigoplus_{j\in\mathcal{S}} y_j\right)$

La description des ensembles $\mathcal{A}$ et $\mathcal{B}$ est de taille $O(n \lg n)$. Le nombre d'indices qu'Alice et Bob devraient avoir en commun est $k\ell = \frac{n^2}{N} = n^{\alpha+\beta}$. Ils peuvent donc s'assurer que la taille de leur intersection est au moins $k\ell$ sauf une probabilité très faible en enregistrant quelques bits en extra.

Pour qu'il puissent compléter toutes étapes du protocole Alice et Bob doivent avoir un espace mémoire de taille $\theta(n^{2-2\alpha})$. Ils ont besoin d'enregistrer la valeur de hachage des chaînes $\sigma(\mathcal{U}_0)$ et $\sigma(\mathcal{U}_1)$ et également la fonction de hachages qui elle est de taille quadratique en fonction de la taille de la valeur de hachage donc $M^2 \leq n^{2-2\alpha}$.

Il est facile de voir que le protocole réussi avec avec une probabilité au moins $\frac{1}{2}$ si Alice et Bob sont honnêtes. Les «*aborts*» se produisent aux étapes 2 et 5.

5.2 Preuve de sécurité

Cher lecteur, bienvenue dans une des parties les plus *coriaces* de ce mémoire. Je vous félicite d'ailleurs pour votre persévérance dans la lecture, probablement très ardue, de ce chapitre. Je vous recommande fortement, avant d'entreprendre la lecture de cette section, d'aller vous chercher un café !

La preuve de sécurité est divisée en deux parties bien distinctes. Premièrement, il faut montrer qu'Alice n'apprend aucune information sur le bit c de Bob et deuxièmement, il faut montrer que Bob n'apprend aucune information sur le bit $b_{\bar{c}}$. En fait, il faut prouver un énoncé un peu plus fort : Bob n'apprend aucune information sur le bit $b_{\bar{c}}$, même s'il apprend le bit b_c après l'exécution du protocole[3].

5.2.1 Sécurité pour Bob

Théorème 5.1. *Dans le protocole 5.2, Alice n'apprend aucune information sur le bit* c *de Bob.*

Preuve. La preuve de ce théorème est similaire à la justification que la réduction du transfert équivoque au transfert inconscient fonctionne.

Bob n'annonce qu'à la fin (à l'étape 6 du protocole) quel bit il veut, en annonçant le bit c'. Puisqu'Alice ne sait pas quel est le bon et mauvais ensemble de Bob, elle n'apprend rien sur c en apprenant c'. □

5.2.2 Sécurité pour Alice

Premièrement, $R^{\langle N]}$ correspond à $r^{\langle N]}$. Pendant la diffusion de $R^{\langle N]}$, Bob malhonnête peut calculer n'importe quelle fonction (même probabiliste) de $\{0,1\}^N$ dans $\mathcal{V}$ tel que $\lg |\mathcal{V}| \leq \gamma N$. Nous décrivons l'information que Bob a sur la chaîne $R^{\langle N]}$ par la variable aléatoire V.

3. Voir la section 1.2 pour plus de détails.

Après qu'Alice ait annoncé à Bob l'ensemble $\mathcal{A}$, il doit former m blocs $x_1, x_2, \ldots, x_m$ représentés par $X_1, X_2, \ldots, X_m$. Il est important de voir que la distribution de $X^{\langle m]}$ est donnée par la distribution de $R^{\mathcal{A}}$ conditionnée sur l'événement $V = v$. L'équivalence est donnée par l'équation suivante

$$P_{X^{\langle m]}}(x_1, \ldots, x_m) \; = \; P_{R^{\mathcal{A}} \mid V=v}(r^{C_1}, \ldots, r^{C_m}). \tag{5.1}$$

La preuve de sécurité pour Alice est divisée en trois parties. Premièrement, nous donnons une borne inférieure sur la min-entropie de Bob de la chaîne $X^{\langle m]}$ étant donné qu'il apprend $V = v$ sur la chaîne $R^{\langle N]}$. Deuxièmement, nous montrons que la première distillation génère quelques bits, en fait μm bits, dont Bob n'a aucune information. Troisièmement, nous montrons qu'à partir des bits obtenus de la distillation, il est possible d'effectuer un hachage interactif entre Alice et Bob pour que Bob annonce à Alice deux sous-ensembles : un bon et un mauvais. Finalement, les trois parties sont regroupées dans le théorème 5.4.

Lemme 5.2. *Soit ϵ_1 un paramètre de sécurité. Soit $\delta = 1 - \gamma - \frac{1}{N} \lg \frac{1}{\epsilon_1}$, soit $\rho = c\delta / \lg \delta^{-1}$ pour une constante $c > 0$. Alors avec une probabilité supérieure à $1 - \epsilon_1 - 6/\sqrt{\rho n}$ nous avons que*

$$H_\infty(X^{\langle m]}) \; \geq \; \rho n.$$

Preuve. Puisque $R^{\langle N]}$ est uniformément distribué sur $\{0,1\}^N$ alors

$$H_\infty(R^{\langle N]}) \; = \; N.$$

L'information que Bob apprend sur $R^{\langle N]}$ est représentée par la variable aléatoire V tel que $\lg |\mathcal{V}| \leq \gamma N$. Par le lemme 2.9 nous avons que

$$H_\infty(R^{\langle N]} \mid V = v) \; \geq \; N - \gamma N - \lg \frac{1}{\epsilon_1}$$

avec une probabilité supérieure à $1 - \epsilon_1$.

Considérons maintenant la sous-chaîne $R^{\langle N]}$ donnée par l'ensemble $\mathcal{A}$. Par le théorème de Zuckerman (théorème 3.3) nous avons que

$$H_\infty(R^{\mathcal{A}} \mid V = v) \; \geq \; \rho n$$

avec une probabilité supérieure à $1 - \frac{6}{\sqrt{\rho n}}$.

Pour finir, nous voulons obtenir un énoncé sur la min-entropie de la chaîne $X^{\langle m]}$. Par (5.1) nous avons que

$$H_\infty(X^{\langle m]}) \; = \; H_\infty(R^{\mathcal{A}} \mid V = v).$$

□

Lemme 5.3. *Supposons que* $H_\infty(X^{\langle m]}) \geq \rho n$*. Soient* $\epsilon_2,\epsilon_3,\epsilon_4 > 0$ *des paramètres de sécurité. Il existe un ensemble* $\Omega \subseteq \{1,2,\ldots,m\}$ *tel que* $|\Omega| \in \theta(m)$ *et tel qu'avec probabilité supérieure à* $1-(\epsilon_2 + m2^{-2\ell} + \epsilon_3 + \epsilon_4)$ *pour tout* $j \in \Omega$

$$H\big(T_j \,|\, F_j = f_j, T^{\langle j-1][j+1\rangle} = t^{\langle j-1][j+1\rangle}\big) \;=\; 1 - \frac{2\epsilon_4}{\ln 2}.$$

Nous partons du fait que $H_\infty(X^{\langle m]}) \geq \rho n$ et nous voulons obtenir un énoncé sur la min-entropie de chacun des blocs de $X^{\langle m]}$. Puisque la règle de chaîne (lemme 2.5) ne fonctionne pas avec la min-entropie, il faut travailler un peu. Nous découpons la chaîne $X^{\langle m]}$ en blocs par induction ; pour arriver à ceci nous devons utiliser un argument de *connaissance de gâchis*[4]. Un oracle fournit à Bob une *information subsidiaire*[5] à laquelle il n'a en réalité pas accès. Cet artifice de preuve est nécessaire pour lisser sa distribution de probabilité sur la chaîne.

Après, il est facile par le théorème de distillation de secret (théorème 3.1) de montrer que Bob n'a presque pas d'information sur un ensemble Ω de bits.

Preuve. Premièrement, nous construisons l'argument de l'information subsidiaire. Cet argument n'est utilisé que pour les fins de la preuve. Bob n'apprend pas en réalité l'information subsidiaire.

Supposons que l'information subsidiaire $u_1,u_2,\ldots,u_m$ tel que $u_j \in \{1,2,\ldots,2j\ell\}$ est rendue disponible à Bob par un oracle après que les fonctions de hachage soient annoncées. Soit $U^{\langle m]}$ une variable aléatoire correspondant à la distribution de $u^{\langle m]}$. L'information subsidiaire est définie pour $j = 1,2,\ldots,m$ par $U_j = \lambda_j(X^{\langle j]})$ où

$$\lambda_j(x^{\langle j]}) \;=\; \begin{cases} 2j\ell & \text{si } P_{X^{\langle j]}\,|\,T^{[j+2\rangle}=t^{[j+2\rangle}}\big(x^{\langle j]}\big) \leq 2^{-2j\ell} \\ \Big\lfloor -\lg P_{X^{\langle j]}\,|\,T^{[j+2\rangle}=t^{[j+2\rangle}}\big(x^{\langle j]}\big)\Big\rfloor & \text{sinon} \end{cases}$$

Remarque. Lorsqu'une distribution est conditionnée sur $T_i = t_i$ pour un certain i (de même pour $T^{[i\rangle} = t^{[i\rangle}$ et $T^{\langle i]} = t^{\langle i]}$) elle est en fait conditionnée sur $T_i = t_i, F_i = f_i$ puisque $T_i = F_i(X_i)$.

Cela peut sembler surprenant que l'on donne de l'information en plus à Bob, mais en fait il apprend relativement peu d'information sur $X^{\langle m]}$ simplement parce que l'alphabet de $U^{\langle m]}$ est assez petit. La taille de $U^{\langle m]}$ (en bit) est bornée par $m\lg(2m\ell) = m(\lg n + 1)$ et donc par le lemme 2.9

$$H_\infty(X^{\langle m]} \,|\, U^{\langle m]} = u^{\langle m]}) \;\geq\; H_\infty(X^{\langle m]}) - m(\lg n + 1) - \lg\frac{1}{\epsilon_2} \tag{5.2}$$

4. *spoiling knowledge*
5. *side information*

avec une probabilité supérieure à $1 - \epsilon_2$.

U_j a la propriété de partionner les valeurs de $X^{\langle j]}$ en des ensembles qui ont approximativement la même probabilité. Pour tous les u_j excepté $u_j = 2j\ell$, la valeur de la distribution de probabilité de $P_{X^{\langle j]} \mid U_j = u_j}$ diffère moins qu'un facteur de 2. Donc nous avons que

$$\min_{x^{\langle j]}} P_{X^{\langle j]} \mid T^{[j+2\rangle} = t^{[j+2\rangle}, U_j = u_j}(x^{\langle j]}) \;\geq\; \frac{1}{2} \max_{x^{\langle j]}} P_{X^{\langle j]} \mid T^{[j+2\rangle} = t^{[j+2\rangle}, U_j = u_j}(x^{\langle j]}) \tag{5.3}$$

Cette propriété nous sera utile dans le pas d'induction.

Pour un j particulier, la probabilité que $U_j = 2j\ell$ est inférieure ou égale à $2^{-2j\ell} \leq 2^{-2\ell}$. Donc, la probabilité qu'il existe un j tel que $U_j = 2j\ell$ est inférieure à $m2^{-2\ell}$. La propriété (5.3) est vraie avec une probabilité supérieure à $1 - m2^{-2\ell}$.

Nous voulons obtenir une borne inférieure sur la somme des min-entropies des blocs X_j à partir de la min-entropie de la chaîne formée par tous les blocs $X^{\langle m]}$. Nous aurons besoin de la propriété (5.3) dans le pas d'induction, donc toutes les distributions sont conditionnées sur l'événement $U^{\langle m]} = u^{\langle m]}$.

Nous allons prouver que l'énoncé suivant est vrai avec une probabilité supérieure à $1 - \epsilon_3$.

$$\begin{aligned}\sum_{j=1}^{m} H_\infty(X_j \mid X^{\langle j-1]} = x^{\langle j-1]}, T^{[j+1\rangle} = t^{[j+1\rangle}, U^{\langle m]} = u^{\langle m]}) \\ \geq\; H_\infty(X^{\langle m]} \mid U^{\langle m]} = u^{\langle m]}) - m\left(2 + \lg \frac{m}{\epsilon_3}\right)\end{aligned} \tag{5.4}$$

Ce que l'on fait par induction sur $j = 0, 1, \ldots, m-1$ en utilisant l'hypothèse d'induction suivante

$$\begin{aligned}H_\infty(X^{\langle m-j]} \mid T^{[m-j+1\rangle} = t^{[m-j+1\rangle}, U^{\langle m]} = u^{\langle m]}) \\ + \sum_{i=m-j+1}^{m} H_\infty(X_i \mid X^{\langle i-1]} = x^{\langle i-1]}, T^{[i+1\rangle} = t^{[i+1\rangle}, U^{\langle m]} = u^{\langle m]}) \\ \geq\; H_\infty(X^{\langle m]} \mid U^{\langle m]} = u^{\langle m]}) - j\left(2 + \lg \frac{m}{\epsilon_3}\right)\end{aligned} \tag{5.5}$$

est vrai avec une probabilité supérieure à $1 - j\frac{\epsilon_3}{m}$.

Évident pour $j = 0$. Supposons que (5.5) est vrai pour un $j \geq 0$. Alors pour tout $x^{\langle m-j-1]}$, pour tout x^{m-j} et pour tout $t^{[m-j+1\rangle}$ nous avons que

$$\begin{aligned}
&\max_{\hat{x}^{\langle m-j]}} P_{X^{\langle m-j]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(\hat{x}^{\langle m-j]}) \\
&\qquad \geq P_{X^{\langle m-j-1]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(x^{\langle m-j-1]}) \\
&\qquad\quad \cdot P_{X_{m-j}\,|\,X^{\langle m-j-1]}=x^{\langle m-j-1]},\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(x_{m-j}) \qquad (5.6) \\
&\qquad \geq \min_{\hat{x}^{\langle m-j-1]}} P_{X^{\langle m-j-1]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(\hat{x}^{\langle m-j-1]}) \\
&\qquad\quad \cdot P_{X_{m-j}\,|\,X^{\langle m-j-1]}=x^{\langle m-j-1]},\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(x_{m-j}) \qquad (5.7)
\end{aligned}$$

et par (5.3) nous avons

$$\begin{aligned}
&\geq \frac{1}{2}\max_{\hat{x}^{\langle m-j-1]}} P_{X^{\langle m-j-1]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(\hat{x}^{\langle m-j-1]}) \\
&\quad \cdot P_{X_{m-j}\,|\,X^{\langle m-j-1]}=x^{\langle m-j-1]},\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},\,U^{\langle m]}=u^{\langle m]}}(x_{m-j}) \qquad (5.8)
\end{aligned}$$

Ce qui est équivalent à

$$\begin{aligned}
&H_\infty(X^{\langle m-j-1]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},U^{\langle m]}=u^{\langle m]}) \\
&\quad + H_\infty(X_{m-j}\,|\,X^{\langle m-j-1]}=x^{\langle m-j-1]},T^{[m-j+1\rangle}=t^{[m-j+1\rangle},U^{\langle m]}=u^{\langle m]}) \\
&\qquad \geq H_\infty(X^{\langle m-j]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},U^{\langle m]}=u^{\langle m]}) - 1 \qquad (5.9)
\end{aligned}$$

Par le lemme 2.9

$$\begin{aligned}
&H_\infty(X^{\langle m-j-1]}\,|\,T^{[m-j\rangle}=t^{[m-j\rangle},U^{\langle m]}=u^{\langle m]}) \\
&\qquad \geq H_\infty(X^{\langle m-j-1]}\,|\,T^{[m-j+1\rangle}=t^{[m-j+1\rangle},U^{\langle m]}=u^{\langle m]}) - 1 - \lg\frac{m}{\epsilon_3} \qquad (5.10)
\end{aligned}$$

avec une probabilité supérieure à $1 - \frac{\epsilon_3}{m}$.

Par notre hypothèse d'induction (5.5) et par (5.9) et (5.10) nous avons que

$$\begin{aligned}
&H_\infty(X^{\langle m-j-1]}\,|\,T^{[m-j\rangle}=t^{[m-j\rangle},U^{\langle m]}=u^{\langle m]}) \\
&\quad + \sum_{i=m-j}^{m} H_\infty(X_i\,|\,X^{\langle i-1]}=x^{\langle i-1]},T^{[i+1\rangle}=t^{[i+1\rangle},U^{\langle m]}=u^{\langle m]}) \\
&\qquad \geq H_\infty(X^{\langle m]}\,|\,U^{\langle m]}=u^{\langle m]}) - (j+1)\left(2+\lg\frac{m}{\epsilon_3}\right) \qquad (5.11)
\end{aligned}$$

avec une probabilité supérieure à $1-(j+1)\frac{\epsilon_3}{m}$. Ce qui conclut l'induction.

Nous avons par (5.2) et (5.4)

$$\sum_{j=1}^{m} H_\infty(X_j \,|\, X^{\langle j-1]} = x^{\langle j-1]}, T^{[j+1\rangle} = t^{[j+1\rangle}, U^{\langle m]} = u^{\langle m]})$$
$$\geq \rho n - m\Big(\lg n + 3 + \lg \frac{m}{\epsilon_3}\Big) - \lg \frac{1}{\epsilon_2} \quad (5.12)$$

Ceci implique que la min-entropie d'au moins q blocs $X_1, X_2, \ldots, X_m$ conditionnés sur $X^{\langle j-1]} = x^{\langle j-1]}$, $T^{[j+1\rangle} = t^{[j+1\rangle}$ pour le bloc j et sur $U^{\langle m]} = u^{\langle m]}$ est plus grande que $2\lg \frac{1}{\epsilon_4}$ où

$$q = \rho m - \frac{m(\lg n + 3 + \lg \frac{m}{\epsilon_3}) + \lg \frac{1}{\epsilon_2} + 2m\lg \frac{1}{\epsilon_4}}{\ell}$$

Soit $\mathcal{Q} \subseteq \{1,2,\ldots,m\}$ l'ensemble d'indices j tel que pour tout $j \in \mathcal{Q}$

$$H_\infty(X_j \,|\, X^{\langle j-1]} = x^{\langle j-1]}, T^{[j+1\rangle} = t^{[j+1\rangle}, U^{\langle m]} = u^{\langle m]}) \geq 2\lg \frac{1}{\epsilon_4} \quad (5.13)$$

Par le théorème 3.1, nous avons donc pour tout $j \in \mathcal{Q}$

$$H(T_j \,|\, F_j, X^{\langle j-1]} = x^{\langle j-1]}, T^{[j+1\rangle} = t^{[j+1\rangle}, U^{\langle m]} = u^{\langle m]}) \geq 1 - \frac{2{\epsilon_4}^2}{\ln 2} \quad (5.14)$$

Ce qui implique que

$$H(T_j \,|\, F_j = f_j, X^{\langle j-1]} = x^{\langle j-1]}, T^{[j+1\rangle} = t^{[j+1\rangle}, U^{\langle m]} = u^{\langle m]}) \geq 1 - \frac{2\epsilon_4}{\ln 2} \quad (5.15)$$

avec une probabilité supérieure à $1 - \epsilon_4$. Ce qui implique que

$$H(T_j \,|\, F_j = f_j, T^{\langle j-1][j+1\rangle} = t^{\langle j-1][j+1\rangle}) \geq 1 - \frac{2\epsilon_4}{\ln 2} \quad (5.16)$$

avec une probabilité supérieure à $1 - (\epsilon_2 + m2^{-2\ell} + \epsilon_3 + \epsilon_4)$. □

Théorème 5.4. *Supposons que l'espace mémoire d'un Bob malhonnête est inférieur à γN pour un $\gamma < 1$. Alors pour un n suffisamment grand, l'information que Bob apprenne sur le bit $b_{\bar{c}}$, même si Alice lui donne, après le transfert, le bit b_c est négligeable avec une probabilité supérieure à $1 - (\epsilon_1 + \frac{6}{\sqrt{\rho n}} + \epsilon_2 + m2^{-2\ell} + \epsilon_3 + \epsilon_4) - \frac{1}{M}$.*

Nous voulons maintenant montrer que par le hachage interactif (section 3.4), il est impossible pour Bob d'annoncer deux bons ensembles, c'est-à-dire deux ensembles dont il connaît tous les bits. Il suffit de montrer que les conditions du lemme 3.8 sont remplies.

Preuve. À partir des lemmes 5.2 et 5.3, nous avons montré qu'avec une probabilité supérieure à $1 - (\epsilon_1 + \frac{6}{\sqrt{\rho n}} + \epsilon_2 + m2^{-2\ell} + \epsilon_3 + \epsilon_4)$ où ϵ_1, ϵ_2, ϵ_3 et ϵ_4 sont des paramètres de sécurité, nous pouvons extraire au moins $q \geq (\rho - \mu)m$ bits au sujet desquels Bob n'a aucune information.

Il existe $\binom{m}{k}$ sous-ensembles de k éléments des bits $T_1, T_2, \ldots, T_m$. Les chaînes qui représentent les sous-ensembles de k éléments sont de longueur $M = \lceil \lg \binom{m}{k} \rceil$.

Si Bob ignore au moins q des m bits $T_1, T_2, \ldots, T_m$ alors il connaît tous les bits d'au plus $\binom{m-q}{k}$ des sous-ensembles k éléments.

Posons $r = \lg M$ et $\nu = \frac{1}{M} \lg \binom{m-q}{k}$. Pour appliquer le lemme 3.8, il faut montrer que

$$\nu < 1 - \frac{8r+4}{M} \tag{5.17}$$

ce qui est équivalent à

$$M - \lg \binom{m-q}{k} - 8r - 4 > 0 \tag{5.18}$$

Or $M \geq \lg \binom{m}{k}$ et $r < \lg m$ donc

$$\begin{aligned}
M - \lg \binom{m-q}{k} - 8r - 4 &\geq \lg \binom{m}{k} - \lg \binom{m-q}{k} - 8r - 4 \\
&= \lg \left(\frac{\binom{m}{k}}{\binom{m-q}{k}} \right) - 8r - 4 \\
&= \lg \left(\prod_{i=1}^{q} \frac{m-q-i}{m-q-k-i} \right) - 8r - 4 \\
&\geq q \left(\lg(m-q) - \lg(m-k-q) \right) - 8r - 4 \\
&> (1-\mu)m \left(\lg(m-q) - \lg(m-k-q) \right) - 8 \lg m - 4
\end{aligned}$$

et donc pour un m suffisamment grand

$$> 0$$

Donc l'équation (5.17) est vraie pour un n suffisamment large. Alors avec une probabilité supérieure à $1 - \frac{1}{M}$ Bob ne peut annoncer deux chaînes dont il connaît tous les bits et donc il n'apprend rien sur le bit $b_{\bar{c}}$. □

Alice(b_0, b_1)		Bob(c)
$t_1, t_2, \ldots, t_n \in \{0,1\}$		
	$\xrightarrow{\mathrm{TE}(t_1),\ \mathrm{TE}(t_2),\ \vdots,\ \mathrm{TE}(t_n)}$	
		enregistre $t_{s_1}, t_{s_2}, \ldots, t_{s_i}$
		$\mathcal{U}_c \subseteq \{s_1, s_2, \ldots, s_i\}$ et $\mathcal{U}_{\bar{c}} \subseteq \{1, 2, \ldots, n\}$ tel que $\mathcal{U}_c \cap \mathcal{U}_{\bar{c}} = \emptyset$
	$\xleftarrow{\mathcal{U}_0, \mathcal{U}_1}$	
$z_0 = b_0 \oplus \left(\bigoplus_{j \in \mathcal{U}_0} t_j\right)$ $z_1 = b_1 \oplus \left(\bigoplus_{j \in \mathcal{U}_1} t_j\right)$		
	$\xrightarrow{z_0, z_1}$	
		$b_c = z_c \oplus \left(\bigoplus_{j \in \mathcal{U}_c} t_j\right)$
		b_c

FIG. 5.1 – *Protocole de réduction de transfert inconscient au transfert équivoque*

Alice(b_0,b_1)		Bob(c)
$r^{\langle N]} \in_{\mathcal{R}} \{0,1\}^N$		
$\mathcal{A} \subset^n_{\mathcal{R}} \{1,\dots,N\}$		$\mathcal{B} \subset^n_{\mathcal{R}} \{1,\dots,N\}$
	$\xrightarrow{r^{\langle N]}}$	
enregistre $r^{\mathcal{A}}$		enregistre $r^{\mathcal{B}}$
	$\xrightarrow{\mathcal{A}}$	
		calcule π tel que $\mathcal{C}_i = \{a_{\pi(\ell(i-1)+1)},\dots,a_{\pi(\ell i)}\}$ tel que $\forall_{i\in\mathcal{S}}\ C_i \subset \mathcal{A}\cap\mathcal{B}$
	$\xleftarrow{\pi}$	
$f_1,f_2,\dots,f_m \in_{\mathcal{R}} \mathcal{F}$		
	$\xrightarrow{f_1,f_2,\dots,f_m}$	
$t_i = f_i(r^{C_i})$ pour $i = 1,2,\dots,n$		$y_i = f_i(r^{C_i})$ pour $i \in \mathcal{S}$
		$s = \sigma(S)$
	hachage interactif	
$\mathcal{U}_0,\mathcal{U}_1$ tel que $\mathcal{U}_0 \neq \mathcal{U}_1$ et $h(\sigma(\mathcal{U}_0)) = h(\sigma(\mathcal{U}_1))$		$\mathcal{U}_0,\mathcal{U}_1$ tel que $\mathcal{U}_0 \neq \mathcal{U}_1$ et $h(\sigma(\mathcal{U}_0)) = h(\sigma(\mathcal{U}_1))$
		c' tel que $\mathcal{U}_{c\oplus c'} = \mathcal{S}$
	$\xleftarrow{c'}$	
$z_0 = b_0 \oplus \left(\bigoplus_{j\in\mathcal{U}_{c'}} t_j\right)$ $z_1 = b_1 \oplus \left(\bigoplus_{j\in\mathcal{U}_{c'\oplus 1}} t_j\right)$		
	$\xrightarrow{z_0,z_1}$	
		$b_c = z_c \oplus \left(\bigoplus_{j\in\mathcal{S}} y_j\right)$
		b_c

FIG. 5.2 – *Protocole de transfert inconscient*

CHAPITRE 6

Conclusion

LES chapitres précédents ont expliqué comment il est possible d'accomplir, de façon inconditionnellement sécuritaire, des tâches cryptographiques sous l'hypothèse que les participants sont limités en espace mémoire. Plus précisément, ils ont présenté des protocoles pour réaliser deux primitives cryptographiques : le partage de clef et le transfert inconscient.

Ces résultats sont intéressants puisqu'ils proposent un nouveau modèle *réaliste* où il est possible d'accomplir des tâches cryptographiques qui requièrent très peu de compromis sur la sécurité. Les protocoles présentés ont cependant quelques inconvénients qu'il est important de souligner.

Premièrement, nous supposons l'existence d'une grande source de bits uniformément distribués. Une telle source n'est certainement pas facile à construire. Il est même possible que générer les bits aléatoires soit plus coûteux que les transmettre. Les coûts de construction d'une infrastructure pour réaliser de tels protocoles peuvent être, par contre, amortis par le fait que plusieurs usagers peuvent utiliser la source en même temps. De plus, une source de bits uniformément distribués aurait plusieurs autres applications en cryptographie.

Deuxièmement, la sécurité des protocoles proposés n'est qu'au mieux linéaire. C'est-à-dire que la séparation entre l'espace mémoire requis par les participants honnêtes pour accomplir les protocoles et l'espace mémoire nécessaire aux participants malhonnêtes pour tricher, n'est qu'au mieux linéaire dans le nombre de bits choisis par Alice et Bob.

Habituellement, nous voulons une séparation exponentielle (ou au moins superpolynomiale). Par exemple, pour la sécurité calculatoire, l'adversaire doit, pour briser un procédé cryptographique, réaliser un calcul qui prend un temps exponentiel comparé à un temps polynômial pour les calculs des participants honnêtes. Il faut cependant réaliser que la mémoire est présentement une ressource plus coûteuse que le temps, donc cette séparation est tout de même acceptable. Nous proposons, comme suite de ces recherches, de développer des protocoles qui donnent une séparation plus grande.

Une conséquence de ceci est que la sécurité des protocoles décrits repose beaucoup sur l'écart entre deux technologies : celle des communications à haut débit et celle du stockage d'information à large échelle. La technologie actuelle semble suffisante pour supporter les protocoles, mais des développements majeurs sont à prévoir dans ces deux domaines qui rendront soit plus intéressants ou complètement obsolètes les protocoles proposés.

ANNEXE A

Encodage des sous-ensembles de k éléments

DANS cet annexe, nous présentons un encodage simple, efficace et compact des sous-ensembles de k éléments en chaînes binaires. Un tel encodage est requis dans le protocole 5.2 (voir la section 5.1.2 pour plus de détails). Ce type d'encodage est connu depuis quelques temps [1].

Soit l'ensemble $\mathcal{S} = \{1,2,\ldots,n\}$. Sans perte de généralité, $\mathcal{S}$ représente n'importe quel ensemble ordonné. Nous voulons encoder efficacement les sous-ensembles de k éléments de $\mathcal{S}$ en chaînes binaires. De plus, cet encodage doit être dense, plus précisément, nous voulons un encodage qui envoie les sous-ensembles de k éléments sur les chaînes binaires de longueur $\left\lceil \lg \binom{n}{k} \right\rceil \leq nh(k/n)$.

Nous allons décrire une fonction $\sigma_{n,k}$ qui associe à chaque sous-ensemble de k éléments de $\mathcal{S}$ un entier compris entre 0 et $\binom{n}{k} - 1$.

$$\sigma_{n,k} : \left\{ \mathcal{Q} \mid \mathcal{Q} \subseteq \mathcal{S} \text{ et } |\mathcal{Q}| = k \right\} \to \left\{ 0,1,\ldots,\binom{n}{k} - 1 \right\}$$

Pour obtenir l'encodage final d'un sous-ensemble de k éléments, il suffit de prendre la représentation en binaire de l'entier qui lui est associé.

Sans perte de généralité, nous pouvons représenter les sous-ensembles de k éléments de $\mathcal{S}$ par $\mathcal{Q} = \{e_1,e_2,\ldots,e_k\}$ tel que $e_{i-1} < e_i$ pour $i = 1,2,\ldots,k$. À partir de cette représentation, il est très facile de définir l'encodage.

Nous allons donc travailler avec une représentation un peu différente pour montrer comment fonctionne l'encodage. Les sous-ensembles de k éléments de $\mathcal{S}$ correspondent naturellement aux chaînes de bits de longueur n et de poids k. Les valeurs $e_1,e_2,\ldots,e_k$ donnent les positions où l'on trouve des «1» dans ces chaînes.

Exemple A.1. Soient $n = 6$ et $k = 3$. $\mathcal{Q} = \{2,3,6\}$ est un sous-ensemble de 3 éléments de $\mathcal{S} = \{1,2,\ldots,6\}$. $\mathcal{Q}$ est représenté par la chaîne de longueur n et de poids k : 011001.

1. Pour des résultats plus généraux sur ce type d'encodage, l'auteur recommande de consulter COVER [19].

Nous allons encoder les chaînes de longueur n et de poids k dans l'ordre lexicographique inverse (c'est-à-dire, par exemple, 11100, 11010, 11001, 10110, ..., 00111). Soit w_Q la chaîne de longueur n et de poids k qui représente le sous-ensemble de k éléments $Q = \{e_1, e_2, \ldots, e_k\}$. Prenons le premier «1» le plus à gauche de w_Q, il est précédé de $e_1 - 1$ «0». Toutes les chaînes qui ont leur premier «1» plus à gauche que w_Q précèdent donc w_Q dans l'ordre lexicographique inverse. Il est facile de dénombrer ces chaînes.

Pour toutes les positions j tel que $1 \leq j \leq e_1 - 1$, il y a $\binom{n-j}{k-1}$ (combien de façons existe-t-il de placer les $k-1$ «1» restants dans les $n-j$ positions restantes) chaînes de longueur n et de poids k tel que leur «1» le plus à gauche est en position j. En sommant sur $j = 1, \ldots, e_1 - 1$, nous trouvons le nombre de chaînes qui précèdent w_Q dans l'ordre lexicographique inverse pour le premier «1» :

$$\sum_{j=1}^{e_1-1} \binom{n-j}{k-1} .$$

Exemple A.2. Soient $n = 6$ et $k = 3$. Soit w_Q la chaîne 011001. Le premier «1» de w_Q est en position 2. Il y a $\sum_{j=1}^{1} \binom{n-1}{k-1} = \binom{5}{2} = 10$ chaînes de longueur n et de poids k dont le premier «1» précède celui de w_Q :

111000, 110100, 110010, 110001, 101100, 101010, 101001, 100110, 100101, 100011

De même façon, prenons le i^e «1» de la chaîne w_Q et comptons le nombre de chaînes dont le i^e «1» est plus à gauche que celui de w_Q. Pour toutes les positions j comprises entre $e_{i-1} + 1$ et $e_i - 1$, il y a $\binom{n-j}{k-i}$ chaînes de poids k tel que leur i^e «1» est en position j. En sommant sur $j = e_{i-1} + 1, \ldots, e_i - 1$, nous trouvons le nombre de chaînes qui précèdent w_Q dans l'ordre lexicographique inverse pour le i^e «1» :

$$\sum_{j=e_{i-1}+1}^{e_i-1} \binom{n-j}{k-i} .$$

En sommant toutes les chaînes qui précèdent w_Q pour les positions $i = 1, 2, \ldots, k$ nous obtenons l'index $\sigma_{n,k}(Q)$ de w_Q dans l'ordre lexicographique inverse. Nous posons $e_0 = 0$. L'encodage est donc décrit par la fonction suivante :

$$\sigma_{n,k}(Q) = \sum_{i=1}^{k} \sum_{j=e_{i-1}+1}^{e_i-1} \binom{n-j}{k-i} .$$

Exemple A.3. Soient $n = 6$ et $k = 3$. Calculons $\sigma_{6,3}\big(\{2,3,6\}\big)$

$$
\begin{aligned}
\sigma_{6,3}\big(\{2,3,6\}\big) &= \sum_{i=1}^{k} \sum_{j=e_{i-1}+1}^{e_i-1} \binom{6-j}{3-i} \\
&= \sum_{j=1}^{1} \binom{6-j}{3-1} + \sum_{j=3}^{2} \binom{6-j}{3-2} + \sum_{j=4}^{5} \binom{6-j}{3-3} \\
&= \binom{5}{2} + 0 + \binom{2}{0} + \binom{1}{0} \\
&= 12
\end{aligned}
$$

Pour décoder, il suffit d'utiliser l'algorithme vorace suivant :

Algorithme A.1 $\sigma_{n,k}^{-1}(m)$

pour $i = 1$ **jusqu'à** k

$e_i \leftarrow$ le plus grand ℓ tel que $\displaystyle\sum_{j=e_{i-1}+1}^{\ell-1} \binom{n-j}{k-i} \leq m$

$m \leftarrow m - \displaystyle\sum_{j=e_{i-1}+1}^{e_i-1} \binom{n-j}{k-i}$

fin du pour

Bibliographie

[1] ALON, Noga, Oded GOLDREICH, Johan HÅSTAD et René PERALTA. «Simple Construction of Almost k-wise Independent Random Variables», *Random Structures and Algorithms*, vol. 3, nº 3, 1992, p. 289–304.

[2] AUMANN, Yonatan et Uriel FEIGE. «One Message Proof Systems with Known Space Verifiers», *Advances in Cryptology — CRYPTO '93 Proceedings* (1994), D. R. Stinson, éditeur, 773e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 85–99.

[3] BELLARE, Mihir et Silvio MICALI. «Non-interactive Oblivious Transfer and Applications», *Advances in Cryptology — CRYPTO '89 Proceedings* (1990), G. Brassard, éditeur, 435e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 547–557.

[4] BENNETT, Charles H., François BESSETTE, Gilles BRASSARD, Louis SALVAIL et John SMOLIN. «Experimental Quantum Cryptography», *Journal of Cryptology*, vol. 5, nº 1, 1992, p. 3–28.

[5] BENNETT, Charles H., Gilles BRASSARD, Claude CRÉPEAU et Ueli M. MAURER. «Generalized Privacy Amplification», *IEEE Transactions on Information Theory*, vol. 41, nº 6, novembre 1995, p. 1915–1923.

[6] BENNETT, Charles H., Gilles BRASSARD, Claude CRÉPEAU et Marie-Hélène SKUBISZEWSKA. «Practical Quantum Oblivious Transfer Protocols», *Advances in Cryptology — CRYPTO '91 Proceedings* (1992), 576e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 351–366.

[7] BENNETT, Charles H., Gilles BRASSARD et Jean-Marc ROBERT. «How to Reduce your Enemy's Information», *Advances in Cryptology — CRYPTO '85 Proceedings* (1986), H. C. Williams, éditeur, 218e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 468–476.

[8] BENNETT, Charles H., Gilles BRASSARD et Jean-Marc ROBERT. «Privacy Amplification by Public Discussion», *SIAM Journal of Computing*, vol. 17, nº 2, avril 1988, p. 210–229.

[9] BLAHUT, R. E. *Principles and Practice of Information Theory*, Addison-Wesley Publishing Company, 1988.

[10] BRASSARD, Gilles. *Cryptologie contemporaine*, Paris, Masson, 1992.

[11] BRASSARD, Gilles. *Quantum Computing for Computer Scientist*, MIT Press, 1999. Bientôt disponible dans une librairie près de chez vous.

[12] BRASSARD, Gilles et Paul BRATLEY. *Fundamentals of Algorithmics*, Englewoods Cliffs, NJ, Prentice-Hall, 1996.

[13] CACHIN, Christian. *Entropy Measures and Unconditional Security in Cryptography*, 1er volume de *ETH Series in Information Security and Cryptography*, Konstanz, Allemagne, Hartung-Gorre Verlag, 1997. (Réimpression de la thèse de doctorat no 12187, ETH Zürich).

[14] CACHIN, Christian. «Smooth Entropy and Rényi Entropy», *Advances in Cryptology — EUROCRYPT '97 Proceedings* (1997), W. Fumy, éditeur, 1233e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 193–203.

[15] CACHIN, Christian, Claude CRÉPEAU et Julien MARCIL. «Oblivious Transfer with a Memory-Bounded Receiver», *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science (FOCS)* (1998), p. 493–502.

[16] CACHIN, Christian et Ueli M. MAURER. «Linking Information Reconciliation and Privacy Amplification», *Journal of Cryptology*, vol. 10, no 2, 1997, p. 97–110.

[17] CACHIN, Christian et Ueli M. MAURER. «Unconditional Security Against Memory-Bounded Adversaries», *Advances in Cryptology — CRYPTO '97 Proceedings* (1997), B. Kaliski, éditeur, 1294e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 292–306.

[18] CARTER, J. Lawrence et Mark N. WEGMAN. «Universal Classes of Hash Functions», *Journal of Computer and System Sciences*, vol. 18, 1979, p. 143–154.

[19] COVER, Thomas M. «Enumerative Source Encoding», *IEEE Transactions on Information Theory*, vol. 19, no 1, janvier 1973, p. 73–77.

[20] COVER, Thomas M. et Joy A. THOMAS. *Elements of Information Theory*, John Wiley, 1991.

[21] CRAMER, Ronald. «Introduction to Secure Computation», *Lecture on Data Security: Modern Cryptology in Theory and Practice*, I. Damgård, éditeur, 1561e volume de *Lecture Notes in Computer Science*. Springer, 1999, p. 63–86.

[22] CRÉPEAU, Claude. «Quantum Oblivious Transfer», *Journal of Modern Optics*, vol. 41, no 12, décembre 1984, p. 2445–2454.

[23] CRÉPEAU, Claude. «Equivalence Between Two Flavours of Oblivious Transfer», *Advances in Cryptology — CRYPTO '87 Proceedings* (1988), C. Pomerance, éditeur, 273e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 350–354.

[24] CRÉPEAU, Claude. «Efficient Cryptographic Protocols Based on Noisy Channels», *Advances in Cryptology — CRYPTO '97 Proceedings* (1997), W. Fumy, éditeur, 1233e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 306–317.

[25] CRÉPEAU, Claude, Jeroen VAN DE GRAAF et Alain TAPP. «Committed Oblivious Transfer and Private Multi-Party Computations», *Advances in Cryptology — CRYPTO '95 Proceedings* (1995), D. Coppersmith, éditeur, 963e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 110–123.

[26] CRÉPEAU, Claude et Joe KILIAN. «Achieving Oblivious Transfer Using Weakened Security Assumptions», *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science (FOCS)* (1988), p. 42–52.

[27] CRÉPEAU, Claude et Louis SALVAIL. «Quantum Oblivious Mutual Identification», *Advances in Cryptology — EUROCRYPT '95 Proceedings* (1995), 921e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 133–146.

[28] DAMGÅRD, Ivan, Joe KILIAN et Louis SALVAIL. «On the (Im)possibility of Basing Oblivious Tranfer and Bit Commitment on Weakened Security Assuptions», *Advances in Cryptology — EUROCRYPT '99 Proceedings* (1999), J. Stern, éditeur, 1592e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 56–73.

[29] DE SANTIS, Alfredo, Giuseppe PERSIANO et Moti YUNG. «One-message Statistical Zero-Knowledge Proofs with Space-bounded Verifier», *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP)* (1992), 623e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 28–40.

[30] DIFFIE, Whitfield et Martin E. HELLMAN. «New Directions in Cryptography», *IEEE Transactions on Information Theory*, vol. 22, nº 6, novembre 1976, p. 644–654.

[31] EVEN, Shimon, Oded GOLDREICH et Abraham LEMPEL. «A Randomized Protocol for Signing Contracts», *Advances in Cryptology: Proceedings of CRYPTO 82* (1983), R. L. Rivest, A. Sherman, et D. Chaum, éditeurs, Plenum Press, p. 205–210.

[32] GOLDREICH, Oded, Silvio MICALI et Avi WIGDERSON. «How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority», *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)* (1987), p. 218–229.

[33] GOLDREICH, Oded et Ronen VAINISH. «How to Solve any Protocol Problem – An Efficiency Improvement», *Advances in Cryptology — CRYPTO '87 Proceedings* (1988), C. Pomerance, éditeur, 293e volume de *Lecture Notes in Computer Science*, Springer-Verlag, p. 73–86.

[34] HÅSTAD, Johan, Russell IMPAGLIAZZO, Leonid A. LEVIN et Michael LUBY. «A Pseudorandom Generator from any One-way Function», *SIAM Journal on Computing*, vol. 28, nº 4, 1999, p. 1364–1396.

[35] KAHN, David. *The Codebreakers: The Story of Secret Writing*, New York, NY, Macmillan Publishing Company, 1967.

[36] KILIAN, Joe. «Founding Cryptography on Oblivious Transfer», *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)* (1988), p. 20–31.

[37] KILIAN, Joe. «Zero-knowledge with Log-Space Verifiers», *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science (FOCS)* (1988), p. 25–35.

[38] LUBY, Michael et Avi WIGDERSON. «Pairwise Independence and Derandomization». Rapport Technique 95-035, International Computer Science Institute (ICSI), Berkeley, 1995.

[39] MASSEY, James L. et Ingemar INGEMARSSON. «The Rip van Winkle Cipher — a Simple and Provably Computationally Secure Cipher with Finite Key (Abstract)», *IEEE International Symposium on Information Theory* (1985), p. 146.

[40] MAURER, Ueli M. «Secret Key Agreement by Public Discussion from Common Information», *IEEE Transactions on Information Theory*, vol. 39, nº 3, mai 1993, p. 733–742.

[41] MENEZES, Alfred J., Paul C. VAN OORSCHOT et Scott A. VANSTONE. *Handbook of Applied Cryptography*, Discrete mathematics and its application, Boca Raton, FL, CRC Press, 1997.

[42] NAOR, Moni, Rafail OSTROVSKY, Ramarathnam VENKATESAN et Moti YUNG. «Perfect Zero-Knowledge Arguments for *NP* Using Any One-Way Function», *Journal of Cryptology*, vol. 11, nº 2, 1998, p. 87–108.

[43] RABIN, Michael O. «How to Exchange Secrets by Oblivious Transfer». Rapport Technique TR-81, Harvard, 1981.

[44] RÉNYI, Alfréd. «On Measures of Entropy and Information», *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability* (1961), Berkeley, University of California Press, p. 547–561.

[45] RÉNYI, Alfréd. *Probability Theory*, Amsterdam, North-Holland, 1970.

[46] RIVEST, Ronald L. «Cryptography», *Handbook of Theoretical Computer Science (Volume A: Algorithms and Complexity)*, J. van Leeuwen, éditeur. Elsevier and MIT Press, 1990, chapitre 13, p. 717–755.

[47] RIVEST, Ronald L., Adi SHAMIR et Leonard M. ADLEMAN. «A Method for Obtaining Digital Signatures and Public-Key Cryptosystems», *Communications of the ACM*, vol. 21, nº 2, février 1978, p. 120–126.

[48] ROMAN, Steven. *Coding and Information Theory*, 134e volume de *Graduate Texts in Mathematics*, Springer-Verlag, 1992.

[49] SHANNON, Claude E. «A Mathematical Theory of Communication», *Bell System Technical Journal*, vol. 27, 1948, p. 379–423, 623–656.

[50] SHANNON, Claude E. «Communication Theory of Secrecy Systems», *Bell System Technical Journal*, vol. 28, 1949, p. 656–715.

[51] SHOR, Peter W. «Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer», *SIAM Journal of Computing*, vol. 26, nº 5, octobre 1997, p. 1484–1509.

[52] SIMMONS, Gustavus J., éditeur. *Contemporary Cryptology: The Science of Information Integrity*, Piscataway, NJ, IEEE Press, 1992.

[53] STINSON, Douglas R. *Cryptography: Theory and Practice*, Discrete mathematics and its application, CRC Press, 1995.

[54] TAPP, Alain. «Évaluation de fonctions sur données privées». Mémoire de Maîtrise, Université de Montréal, décembre 1995.

[55] VERNAM, Gilbert S. «Cypher Printing Telegraph System for Secret Wire and Radio Telegraphic Communications», *Journal of American Institute of Electrical Engineers*, vol. 55, 1926, p. 109–115.

[56] WIESNER, Stephen. «Conjugate Coding», *SIGACT News*, vol. 15, nº 1, 1983. (Manuscrit original écrit en 1970).

[57] YAO, Andrew C.-C. «How to Generate and Exchange Secrets», *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science (FOCS)* (1986), p. 162–167.

[58] ZUCKERMAN, David. «Simulating BPP Using a General Weak Random Source», *Algorithmica*, vol. 16, 1996, p. 367–391.