Réconciliation et Distillation publiques de secret

Claude Crépeau

École Normale Supérieure *

Résumé

Considérez un scénario où $\mathcal{A}lice$ et $\mathcal{B}ob$ partagent un secret au moyen d'un canal de communication imparfait. On considère, par exemple, la transmission du secret par un canal binaire symétrique [19, 13] ou un canal quantique [1]. Soit X le secret d'$\mathcal{A}lice$, et soit Y la variable aléatoire décrivant la valeur reçu par $\mathcal{B}ob$. Soit Z la variable aléatoire décrivant la valeur reçu par un adversaire $\mathcal{E}ve$ qui a accès à X via un autre canal.

Le but d'un protocole de *réconciliation publique* [4, 6] est de permettre à $\mathcal{A}lice$ et $\mathcal{B}ob$ d'établir une valeur secrète commune à partir de X et Y, sans avoir recourt à un moyen de communication sécurisé. Par simple échange public d'information ils peuvent corriger les différences infligées à Y, mais ils voudront néanmoins minimiser l'information qu'ils donneront ainsi à $\mathcal{E}ve$ sur leur secret commun.

Dans certaines situations il est possible qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ connaisent assez bien la distribution de la variable Z sans pour autant savoir exactement la valeur reçu par $\mathcal{E}ve$. Considérant l'information obtenue par $\mathcal{E}ve$ par le biais de Z et l'information fournie publiquement lors de la réconciliation [4, 7], $\mathcal{A}lice$ et $\mathcal{B}ob$ voudront obtenir par *distillation publique* un secret plus court $f(X)$ sur lequel $\mathcal{E}ve$ n'a pas ou presque pas d'information.

Cette situation, qui est une généralisation du *wire tap channel* de Wyner [19] a d'abord été étudiée de façon spécifique au contexte de la distribution de clef quantique par Bennett, Brassard et Robert [4] et à celui des canaux binaires symétriques par Maurer [13], puis de façon plus générale par Bennett, Brassard, Crépeau et Maurer [2, 14].

Il s'avère qu'en général, il ne suffit pas à $\mathcal{A}lice$ et $\mathcal{B}ob$ de connaître une borne sur la quantité d'information au sens de Shannon donnée par Z sur X pour choisir une fonction $f(X)$ sur laquelle $\mathcal{E}ve$ ne sera que mal informée. L'on montre dans [2] qu'une borne sur l'information au sens de Rényi [16] permettra néanmoins de fournir une telle fonction f. Dans de nombreuses situations une borne sur l'information au sens de Shannon permettra d'obtenir la même chose en gaspillant une moindre fraction du secret.

Ces résultats sont basés sur le concept de fonction de hachage universelle de Wegman et Carter [8] et sur le concept de fonction t-résiliente [4, 9, 18].

* Laboratoire d'Informatique de l'École Normale Supérieure (CNRS URA1327),
45 rue d'Ulm, 75230 Paris CEDEX 05, France. e-mail:crepeau@dmi.ens.fr.

1 Introduction

Considérez un scénario où $\mathcal{A}lice$ et $\mathcal{B}ob$ partagent un secret au moyen d'un canal de communication imparfait. Soit X le secret d'$\mathcal{A}lice$, et soit Y la variable aléatoire décrivant la valeur reçu par $\mathcal{B}ob$. Soit Z la variable aléatoire décrivant la valeur reçu par un adversaire $\mathcal{E}ve$ qui a accès à X via un autre canal.

Le but d'un protocole de *réconciliation publique* [4, 6] est de permettre à $\mathcal{A}lice$ et $\mathcal{B}ob$ d'établir une valeur secrète W commune à partir de X et Y sans avoir recourt à un moyen de communication sécurisé. Par simple échange public d'information ils peuvent corriger les différences infligées à Y, mais ils voudront néanmoins minimiser l'information qu'ils donneront ainsi à $\mathcal{E}ve$ sur leur secret commun.

Dans certaines situations il est possible qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ connaisent assez bien la distribution de la variable Z sans pour autant savoir exactement la valeur reçu par $\mathcal{E}ve$. Considérant l'information obtenue par $\mathcal{E}ve$ par le biais de Z et l'information fournie publiquement lors de la réconciliation [4, 7], $\mathcal{A}lice$ et $\mathcal{B}ob$ voudront obtenir par *distillation publique* un secret plus court $f(X)$ sur lequel $\mathcal{E}ve$ n'a pas ou presque pas d'information.

Il s'avère qu'en général, il ne suffit pas à $\mathcal{A}lice$ et $\mathcal{B}ob$ de connaître une borne sur la quantité d'information au sens de Shannon donnée par Z sur X pour choisir une fonction $f(X)$ sur laquelle $\mathcal{E}ve$ ne sera que mal informée. L'on montre dans [2] qu'une borne sur l'information au sens de Rényi [16] permettra néanmoins de fournir une telle fonction f. Dans de nombreuses situations une borne sur l'information au sens de Shannon permettra d'obtenir la même chose en gaspillant une moindre fraction du secret.

Ces résultats sont basés sur le concept de fonction de hachage universelle de Wegman et Carter [8] et sur le concept de fonction t-résiliente [4, 9, 18].

2 Scénarios

Ils exsitent de nombreux scénarios où la situation décrite précédemment peut se produire. Historiquement, un premier scénario est celui de Wyner, dit du "Wire-Tap Channel" [19]. Un second scénario est celui de la cryptographie quantique [1] et un troisème est celui des canaux indépendants de Maurer [13, 14].

2.1 Wire-Tap Channel

Dans ce scénario, X est uniformément distribué sur $\{0,1\}^n$, et Y est l'image de X par un canal binaire symétrique avec un taux d'erreur $\varepsilon > 0$, c'est-à-dire que

$$\text{Prob}[Y = y|X = x] = (1-\varepsilon)^{n-d_H(x,y)}\varepsilon^{d_H(x,y)}$$

où $d_H(x, y)$ dénote la distance de Hamming entre x et y. La variable Z est définie de la même façon à partir de Y mais pour un taux d'erreur $\delta > 0$.

2.2 Cryptographie Quantique

Dans ce cas $\mathcal{E}ve$ a le choix de plusieurs stratégies pour obtenir sa variable Z et la relation entre X et Y en dépendra. Afin de simplifier on ne considérera pas toutes les possibilités pour $\mathcal{E}ve$ et nous contenterons de seulement quelques unes particulièrement repésentatives. On permet à $\mathcal{E}ve$ de connaître t bits de X, mais au prix de provoquer en moyenne $t/2$ erreurs entre X et Y. On notera $Z_{i_1,i_2,\ldots,i_t}$ le choix d'$\mathcal{E}ve$ qui satisfera alors la distribution:

$$\text{Prob}[Z_{i_1,i_2,\ldots,i_t} = z|X = x] = \begin{cases} 2^{-n+t} & \text{si } z_{i_1}z_{i_2}\ldots z_{i_t} = x_{i_1}x_{i_2}\ldots x_{i_t} \\ 0 & \text{sinon.} \end{cases}$$

Comme dans le scénario précedent X est uniformément distribué sur $\{0,1\}^n$, et Y est l'image de X par un canal binaire symétrique avec un taux d'erreur $t/2n$, c'est-à-dire que

$$\text{Prob}[Y = y|X = x] = (1 - t/2n)^{n-d_H(x,y)}(t/2n)^{d_H(x,y)}.$$

Les fondements quantiques de ce scénario est un principe d'incertitude à la Heisenberg qui dit que le nombre de bit obtenu par $\mathcal{E}ve$ est proportionel au nombre d'erreurs qu'elle produira par sa mesure sur les bits qu'$\mathcal{A}lice$ transmet à $\mathcal{B}ob$.

2.3 Variables Indépendantes

Ce scénario, similaire à celui de Wyner, est définie à partir d'une source S uniformément distribué sur $\{0,1\}^n$, qui peut être observée indépendamment par $\mathcal{A}lice$, $\mathcal{B}ob$ et $\mathcal{E}ve$ à travers des canaux binaires symétriques ayant des taux d'erreurs α, β et ε positifs, mais quelconques, i.e.

$$\begin{aligned} \text{Prob}[X = x|S = s] &= (1-\alpha)^{n-d_H(x,s)}\alpha^{d_H(x,s)}, \\ \text{Prob}[Y = y|S = s] &= (1-\beta)^{n-d_H(y,s)}\beta^{d_H(y,s)}, \\ \text{Prob}[Z = z|S = s] &= (1-\varepsilon)^{n-d_H(z,s)}\varepsilon^{d_H(z,s)}. \end{aligned}$$

En particulier il se peut que $\alpha, \beta \gg \varepsilon$. On imagine facilement un tel scénario en pratique: $\mathcal{A}lice$, $\mathcal{B}ob$ et $\mathcal{E}ve$ observant un même phénomène astronomique avec des instruments de qualités variées par exemple. On permet à $\mathcal{E}ve$ d'utiliser la meilleur technologie des trois participants.

3 Réconciliation

Le but d'un protocole de *réconciliation publique* [4, 6] est de permettre à $\mathcal{A}lice$ et $\mathcal{B}ob$ d'établir une valeur secrète commune à partir de X et Y sans avoir recourt à un moyen de communication sécurisé. Par simple échange public d'information ils peuvent corriger les différences infligées à Y, mais ils voudront néanmoins minimiser l'information qu'ils donneront ainsi à $\mathcal{E}ve$ sur leur secret commun.

Une première approche non efficace en pratique, utilise les codes correcteurs d'erreurs, et une seconde méthode interactive nous donnera une façon plus efficace de résoudre ce problème.

On supposera qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ on une bonne idée du nombre d'erreur qui séparent X et Y. Ils peuvent obtenir une telle approximation en comparant une fraction des bits de X choisis au hasard avec ceux d'Y, rendant ainsi cette fraction d'entre eux inutilisable dans l'avenir. Ils pourront ensuite faire la correction d'erreur sur la fraction restante des bits de chacun des mots. Soit ε le taux d'erreur tel qu'observé par $\mathcal{A}lice$ et $\mathcal{B}ob$.

3.1 Codes Correcteurs

Nous supposons ici que le lecteur est familier avec les codes correcteurs d'erreur. $\mathcal{A}lice$ et $\mathcal{B}ob$ peuvent publiquement se mettre d'accord sur un code linéaire de capacité R et pouvant corriger en moyenne plus de εn erreurs. $\mathcal{A}lice$ peut ensuite communiquer à $\mathcal{B}ob$ le syndrome de X en lui transmettant $(1-R)n$ bits publiquement. $\mathcal{B}ob$ grace à sa connaissance de Y et du syndrome de X peut retrouver X comme le mot le plus près de Y ayant le bon syndrome. Ce mot sera X avec très grande probabilité. À la fin de ce procédé de correction d'erreur, $\mathcal{E}ve$ connait le Z initial ainsi que le syndrome de X.

Depuis Shannon [15], il est connu qu'un code corrigeant εn erreurs peut être obtenu à condition que $R < 1 - h(\varepsilon)$, où $h(x) = -x \log_2 x - (1-x)\log_2(1-x)$ est la fonction d'entropie. Ce résultat s'applique également aux codes linéaires par un théorème d'Elias [15]. Malheureusement, en pratique on ne peut utiliser n'importe quel code linéaire car l'étape de décodage ne peut être réalisée efficacement. Il faudra se restreindre à des classes de codes linéaires pour lesquels on dispose d'algorithme efficace de décodage. On pourra utiliser, par exemple, des codes concaténés [12] ou des codes superconcentrateurs [17] mais que pour des valeurs de n très grandes dans le premier cas et au prix d'une capacité de correction réduite d'un facteur constant $(\varepsilon/8)$ dans le second cas.

Bien que de façon théorique on puisse utiliser les codes concaténés afin d'approcher la borne de Shannon avec des codes efficacement décodables, pour des exemplaires de taille raisonable dès que le taux d'erreur dépassera un certain seuil, aucun code linéaire connu efficacement décodable ne permettra de se réconcilier sans révéler publiquement tous les bits du secret. Ces codes ne peuvent donc être utilisés que dans le cas où le taux d'erreur est faible ou dans le cas où n est vraiment très grand.

3.2 Réconciliation Interactive

Une alternative intéressante aux codes correcteurs d'erreurs consiste à tirer profit de la possibilité d'interaction qu'ont $\mathcal{A}lice$ et $\mathcal{B}ob$. Des recherches menées principalement par Bennett, Brassard & Robert [4] et par Brassard & Salvail [6] ont produit d'excellentes méthodes de réconciliation interactive. Un atoût de ces méthodes interactives est que les codes correcteurs requièrent une construction beaucoup plus elaborée. Nous ne décrirons dans cette section qu'une seule méthode interactive: l'algorithme **CASCADE** de Brassard et Salvail du à son efficacité pratique sur des exemplaires de toutes tailles et pour n'importe quels taux d'erreur.

Dans ce cas, comme dans le cas des codes, à la fin de la réconciliation $\mathcal{E}ve$ aura son Z initial ainsi qu'un certain nombre de bits qui seront le résultat de fonction linéaires appliquées à X.

3.2.1 DICHOT, PARITÉ et DIPAR

Soit X le mot d'$\mathcal{A}lice$ et Y le mot de $\mathcal{B}ob$. L'algorithme **CASCADE** repose d'abord sur l'existence d'un algorithme **DICHOT** qui fait une recherche dichotomique sur deux mots A, B afin d'identifier une position où ils diffèrent et finalement de corriger cette position.

primitive **DICHOT**(A, B):

1. Si $|A| = 1$, alors $\mathcal{B}ob$ pose $B = A$, en demandant publiquement la valeur explicite de A. L'exécution est terminée.
2. Sinon, $\mathcal{A}lice$ calcule secrètement:
$$\widehat{A} \leftarrow A_1, A_2, \ldots, A_{\lfloor \frac{|A|}{2} \rfloor}$$
et
$$\widehat{A'} \leftarrow A_{\lfloor \frac{|A|}{2} \rfloor + 1}, A_{\lfloor \frac{|A|}{2} \rfloor + 2}, \ldots, A_{|A|}.$$
3. $\mathcal{B}ob$ calcule secrètement:
$$\widehat{B} \leftarrow B_1, B_2, \ldots, B_{\lfloor \frac{|B|}{2} \rfloor}$$
et
$$\widehat{B'} \leftarrow B_{\lfloor \frac{|B|}{2} \rfloor + 1}, B_{\lfloor \frac{|B|}{2} \rfloor + 2}, \ldots, B_{|B|}.$$
4. $\mathcal{A}lice$ envoie publiquement à $\mathcal{B}ob$:
$$p_A \leftarrow \bigoplus_{i=1}^{|\widehat{A}|} \widehat{A_i}.$$
5. $\mathcal{B}ob$ calcule secrètement:
$$p_B \leftarrow \bigoplus_{i=1}^{|\widehat{B}|} \widehat{B_i}.$$
6. $\mathcal{B}ob$ compare p_A et p_B et l'annonce à $\mathcal{A}lice$.
7. Si $p_A \neq p_B$, $\mathcal{A}lice$ et $\mathcal{B}ob$ exécutent **DICHOT** $(\widehat{A}, \widehat{B})$.
8. Sinon $\mathcal{A}lice$ et $\mathcal{B}ob$ exécutent **DICHOT** $(\widehat{A'}, \widehat{B'})$.

Figure 1: **DICHOT**

La procédure **DICHOT** est utilisée dans un contexte où il existe une différence entre A et B. Afin de découvrir l'existence d'une telle différence, $\mathcal{A}lice$ et $\mathcal{B}ob$ utilisent la procédure **PARITÉ**. Cette dernière détermine si le nombre de différence entre A et B est pair (0) ou impair (1). On nomme **DIPAR** la combinaison de ces deux procédures: **DIPAR**(A, B): **PARITÉ**(A, B); Si $p \neq p'$ alors **DICHOT**(A, B).

primitive **PARITÉ**(A, B)

1. $\mathcal{A}lice$ envoie à $\mathcal{B}ob$ publiquement:
$$p \leftarrow \bigoplus_{i=1}^{n} A_i$$
2. $\mathcal{B}ob$ calcule secrètement:
$$p' \leftarrow \bigoplus_{i=1}^{n} B_i$$
3. $\mathcal{B}ob$ apprend que la parité du nombre d'erreurs entre A et B est $p' \oplus p$.

Figure 2: **PARITÉ**

3.2.2 CASCADE

L'algorithme **CASCADE** permute, à chaque itération i, les mots à traiter X, Y pour en faire des A_i, B_i qu'il scinde ensuite en $\lceil n/k_i \rceil$ blocs de taille k_i ($A_i^{k_i}[1], ..., A_i^{k_i}[\lceil n/k_i \rceil]$ et $B_i^{k_i}[1], ..., B_i^{k_i}[\lceil n/k_i \rceil]$), puis applique à chacun des blocs l'algorithme **DIPAR** afin de corriger d'éventuelles erreurs.

Protocole 1 (CASCADE(X, Y)**)**

1. $\mathcal{A}lice$ pose $A_0 = X$.
2. $\mathcal{B}ob$ pose $B_0 = Y$.
3. Pour $i \leftarrow 1$ à ρ faire:
   (a) $\mathcal{A}lice$ et $\mathcal{B}ob$ pigent aléatoirement et publiquement une permutation σ_i de $\{1, 2, \ldots, n\}$.
   (b) $\mathcal{A}lice$ calcule $A_i = \sigma_i(A_{i-1})$.
   (c) $\mathcal{B}ob$ calcule $B_i = \sigma_i(B_{i-1})$.
   (d) Pour $j \leftarrow 1$ à $\lceil n/k_i \rceil$ faire:
      i. $\mathcal{A}lice$ et $\mathcal{B}ob$ exécutent **DIPAR**$(A_i^{k_i}[j], B_i^{k_i}[j])$.
      ii. Si $\mathcal{B}ob$ corrige une erreur e dans B,
      $\mathcal{A}lice$ et $\mathcal{B}ob$ exécutent **CASCOR**$(A, B, i, e, \sigma_1, \sigma_2, \ldots, \sigma_i)$.

Figure 3: **CASCADE**

Seul k_1 est à déterminer en fonction d'ε, car les suivants sont obtenus par $k_{i+1} = 2k_i$. Le nombre global d'itérations ρ est le plus petit nombre tel que $2^\rho k_1 > n$.

Une fois que **DIPAR** a corrigé une erreur e, **CASCADE** utilise la procédure **CASCOR** afin de corriger de nouvelles erreurs dans des itérations précendentes découvertes grace à cette erreur.

L'algorithme **CASCOR** fonctionne en identifiant tous les blocs auquels l'erreur e a appartenu à des itérations précédentes afin d'aller y corriger une nouvelle erreur (après chaque application de **DIPAR** ou **DICHOT** à un bloc, son nombre d'erreurs devient nécessairement pair donc la découverte d'une erreur dans un tel bloc implique qu'il en existe encore une autre!).

Afin de trouver toutes les occurrences de l'erreur e dans les itérations précédentes, **CASCOR** forme une liste $\mathcal{B}^{(i,0)}(e)$ de couples $\langle j', i' \rangle$ signifiants que la position e était dans le bloc j' à l'itération i', pour $0 \leq i' \leq i$. Ensuite **DICHOT** est appliqué successivement à chacun des blocs en question en prenant soin d'ajouter et de retirer de la liste les blocs dont la parité deviendra impaire ou paire suite à chaque nouvelle correction effectuée.

primitive CASCOR$(A, B, i, e, \sigma_1, \ldots, \sigma_i)$

1. *Alice* et *Bob* calculent $\mathcal{C} \leftarrow \mathcal{B}^{(i,0)}(e)$ à l'aide des permutations $\sigma_1, \ldots, \sigma_i$.
2. Tant que $\mathcal{C} \neq \emptyset$ faire:
   (a) *Alice* et *Bob* calculent:
   $$\langle \bar{j}, \bar{i} \rangle = \min_{i'} \{ \langle j', i' \rangle \in \mathcal{C} \}$$
   (b) *Alice* et *Bob* exécutent **DICHOT**$(A_{\bar{i}}^{k_{\bar{i}}}[\bar{j}], B_{\bar{i}}^{k_{\bar{i}}}[\bar{j}])$ pour trouver une erreur en position $\bar{e}$ dans B.
   (c) *Alice* et *Bob* calculent $\mathcal{C} \leftarrow \mathcal{C} \nabla \mathcal{B}^{(i,0)}(\bar{e})$.

Figure 4: CASCOR

3.2.3 Choix de k_1

Un choix de k_1 approprié nous donnera une probabilité d'erreur résiduelle à la fin du protocole exponentiellement faible en n. Une analyse détaillée du choix de k_1 se trouve dans [6]. Afin de ne pas alourdir induement ce texte, nous ne fournirons ici le résultat empirique du choix de k_1:

$$k_1 \approx \frac{0.73}{\varepsilon}.$$

3.2.4 Performances de CASCADE

En pratique **CASCADE** a des performances proches de la borne de Shannon. La table suivante nous donne les résultats de l'application de **CASCADE** sur des échantillons de $n = 10000$ bits, pour quatre taux d'erreurs (0.01, 0.05, 0.10, 0.15). La table nous indique le nombre de bits échangés publiquement lors du protocole afin de réconcilier les mots d'*Alice* et de *Bob*. La valeur $nh(\varepsilon)$ est celle prévue par la borne de Shannon, la valur $I(4)$ est le nombre estimé de bits pour 4 itérations de **CASCADE**, et $\hat{I}(4)$ est le nombre obtenu en pratique sur 10 tests empiriques.

ε	k_1	$\hat{I}(4)$	$nh(\varepsilon)$	$I(4)$
0.01	73	886.3	806.8	932.9
0.05	14	3285.7	2857.1	3314.3
0.10	7	5442.9	4685.7	5700
0.15	5	7600	6010	8240

4 Distillation de Secret

Une fois qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ sont arrivés à une chaîne commune, disons W, s'ils connaissent assez bien la ditribution de la variable Z connue d'$\mathcal{E}ve$, ils pourront *distiller* à partir de W une nouvelle chaîne beaucoup plus secrète que l'originale. Notons qu'à partir de maintenant nous considérons que Z ne décrit plus seulement la variable aléatoire connue d'$\mathcal{E}ve$ avant réconciliation, mais l'information globale qu'elle connait sur W **après** réconciliation. Pour fin de distillation, $\mathcal{A}lice$ et $\mathcal{B}ob$ choisiront une fonction $f : \{0,1\}^n \to \{0,1\}^r$ et l'appliqueront à W pour obtenir $U \leftarrow f(W)$. Si r est suffisamment petit par rapport à n, en fonction de la distribution de Z, U sera à toute fin pratique inconnue d'$\mathcal{E}ve$.

4.1 fonctions t-résilientes

Considérons d'abord un premier cas où $\mathcal{A}lice$ et $\mathcal{B}ob$ possèdent une mot commun W uniformément distribué sur $\{0,1\}^n$ et où Z est identique à W dans t positions choisies par $\mathcal{E}ve$ et uniformément distribué sur les autres positions (dans ce cas on suppose que les observations d'$\mathcal{E}ve$ n'ont pas produit d'erreurs). Nous supposerons aussi qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ connaissent seulement t, mais pas les positions où $\mathcal{E}ve$ a obtenu de bons bits. Dans une telle situation, il existe une solution absolue: les fonctions t-résilientes.

Définition 1 *Une fonction f est t-résiliente si la connaissance de n'importe quel t bits de son entrée x, ne donne aucune information sur sa valeur de sortie $f(x)$.*

Ces fonctions peuvent être facilement obtenues, en particulier si l'on se contente de fonctions *linéaires*. Un théorème de Bennett, Brassard et Robert [4] ainsi que de Chor, Goldreich, Håstad, Friedman, Rudich et Smolensky [9] nous dit que

Théorème 1 *Une matrice binaire M de dimensions $n \times k$ définie une fonction t-résiliente $f : \{0,1\}^n \to \{0,1\}^k$ par $f(x) \leftarrow Mx$ exactement si M est la matrice génératrice d'un code linéaire de paramètres $[n,k,t+1]$.*

Considérant qu'il est possible grace au résultat de Warshamow-Gilbert[15] d'obtenir des codes linéaires de paramètres asymptotiques $[n,(1-h(\varepsilon))n,\varepsilon n]$, $\mathcal{A}lice$ et $\mathcal{B}ob$ pourront à partir de W extraire $f(W)$, soit $n - h(\frac{t+1}{n})n$ bits complètement secret d'$\mathcal{E}ve$ qui connaissait t bits de W au départ. Une matrice aléatoire de dimensions $n \times cn$, pour $c < 1 - h(\frac{t+1}{n})$, définira une fonction t-résiliente sauf avec une probabilité qui décroit exponentiellement avec n.

4.2 hachage universel

Malheureusement dans tous les scénarios que nous avons suggérés précédemment, l'information connue d'$\mathcal{E}ve$ ne contient pas que des bits à certaines positions précises de W: en général, $\mathcal{E}ve$ connais également des bits qui sont des fonctions des bits de W. Dans les cas où $\mathcal{A}lice$ et $\mathcal{B}ob$ ont d'abord procédé à une réconciliation publique de leur information commune $\mathcal{E}ve$ connaitra des relations linéaires sur les bits de W, que ce soit sous forme d'un syndrome ou de bits XOR révélés pendant les méthodes interactives. Dans de tels cas, les fonctions t-résilientes ne font plus l'affaire car si $\mathcal{E}ve$ connaît t bits qui sont n'importe quelle fonction de W (contrairement à n'importe quels t bits de W), il n'existe aucune fonction permettant à coup sûr à $\mathcal{A}lice$ et $\mathcal{B}ob$ d'obtenir même un simple bit d'information secrète à partir de W! Il faut alors choisir une fonction qui a une bonne probabilité d'éliminer l'information connue d'$\mathcal{E}ve$.

C'est exactement l'idée qu'ont exploitée Bennett, Brassard et Robert [4]. Au lieu de chercher des fonctions précises permettant d'éliminer complètement l'information connue d'$\mathcal{E}ve$, ils proposent de choisir une fonction f qui a un comportement essentiellement aléatoire. Pour se faire ils ont utilisé la notion de fonctions *universelles* de Wegman et Carter:

Définition 2 ([8]) *Une classe $\mathcal{F}$ de fonctions $\mathcal{A} \to \mathcal{B}$ est* universel$_2$ *(ou simplement "universel") si, pour toute paire x_1, x_2 d'éléments dinctincts de $\mathcal{A}$, la probabilité que $f(x_1) = f(x_2)$ est d'au plus $1/|\mathcal{B}|$ quand f est choisit aléatoirement dans $\mathcal{F}$ uniformément.*

Exemple 4.1 *La classe $\mathcal{F}$ de toute les fonctions $\{0,1\}^n \to \{0,1\}^r$ est universelle. Notez par contre que cette classe est inutile en pratique car pour décrire une telle fonction il faut $r2^n$ bits.*

Exemple 4.2 *La classe $\mathcal{F}$ de toute les fonctions linéaires $\{0,1\}^n \to \{0,1\}^r$ est universelle. Notez que pour décrire une telle fonction il suffit d'une matrice binaire $r \times n$.*

Exemple 4.3 *La classe $\mathcal{F}$ de toute les fonctions $f(x) \leftarrow (a \times x) \bmod 2^r$ où a est une mot de n bits et le produit $\times$ est celui du corps fini à 2^n éléments, est universelle. Notez que pour décrire une telle fonction il suffit d'un mot binaire de taille n.*

4.2.1 Mesures d'Incertitude

Afin de caractériser le nombre de bits qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ peuvent distiller à partir de W il nous faut utiliser une mesure de l'information connue par $\mathcal{E}ve$. Soit W la variable aléatoire précisant le mot commun d'$\mathcal{A}lice$ et $\mathcal{B}ob$, et soit Z la variable aléatoire d'$\mathcal{E}ve$ distribuée selon P_Z. La mesure standard à utiliser pour une telle chose est l'entropie de Shannon.

Nous supposons que le lecteur est familier avec l'entropie de Shannon. Il est clair que l'on voudrait obtenir un critère nous permettant de borner de façon très serrée la valeur $H(F(W)|Z,F)$ qui est l'incertitude sur le nouveau secret $F(W)$ connaissant Z et la fonction de hachage F. Néanmoins, il nous faut utiliser la notion moins bien connue d'entropie de Renyi comme critère d'utilisation.

Définition 3 *L'*entropie de Shannon *d'une variable aléatoire X distribuée selon P_X est*

$$H(X) = -\sum_x P_X(x) \log_2 P_X(x)$$

et conditionnée par une autre variable Y est

$$H(X|Y) = \sum_y P_Y(y) H(X|Y = y).$$

Définition 4 *La* probabilité de collision *d'une variable aléatoire X distribuée selon P_X est donnée par*

$$P_c(X) = \sum_x P_X(x)^2.$$

Définition 5 *L'*entropie de Renyi *d'une variable aléatoire X distribuée selon P_X est*

$$R(X) = -\log_2 P_c(X)$$

et conditionnée par une autre variable Y est

$$R(X|Y) = \sum_y P_Y(y) R(X|Y = y).$$

Afin de situer notre intuition au sujet de ces notions, il est utile de savoir les faits suivants:

Fait 1 *Pour tout variables aléatoires discrètes X, Y*

$$R(X) \leq H(X) \text{ et } R(X|Y) \leq H(X|Y)$$

avec égalité quand X est uniformément distribué sur un certain sous-ensemble.

4.2.2 Critère d'Utilisation

Revenons à la situation d'$\mathcal{A}lice$, $\mathcal{B}ob$ et $\mathcal{E}ve$. Supposons qu'$\mathcal{A}lice$ et $\mathcal{B}ob$ savent que $H(W|Z) \geq n - t$ pour une certaine constante t. C'est le cas par exemple de la cryptographie quantique où $\mathcal{E}ve$ a obtenu m bits de W directement et l bits de W durant la réconciliation. On pose alors $t = m + l$. Il est montré dans [2] que certaines distributions sur Z telles que $H(W|Z) = n - t$ sont telles que l'incertitude après application d'une fonction quelconque $f : \{0,1\}^n \to \{0,1\}^r$ sera propostionelle à l'incertitude a priori:

$$H(f(W)|Z, f) \approx (1 - t/n)r.$$

C'est à dire que si la seule indication qu'ont $\mathcal{A}lice$ et $\mathcal{B}ob$ sur Z est $H(W|Z)$ ils ne pourront distiller de bits à partir de W. Par contre, si l'on suppose qu'ils savent que $R(W|Z) \geq n - t$ pour une certaine constante t (comme dans l'exemple ci-haut), alors le théorème suivant de Bennett, Brassard, Crépeau et Maurer[2] nous dit combien de bits $\mathcal{A}lice$ et $\mathcal{B}ob$ peuvent distiller à partir de W sur lesquels $\mathcal{E}ve$ sera prèsque totalement ignorante:

Théorème 4.1 *Soit W une variable aléatoire uniformément distribuée. Soit Z une variable aléatoire dépendante de W telle que $R(W|Z) \geq n-t$. Soit F une variable aléatoire uniformément distribuée sur une classe de fonction universelle sur $\{0,1\}^n \to \{0,1\}^r$. On aura alors*

$$R(F(W)|Z,F) \geq r - \log_2(1 + 2^{r+t-n})$$

et donc puisque $H(F(W)|Z,F) \geq R(F(W)|Z,F)$ et $r - \log_2(1+2^{r+t-n}) \geq r - \frac{2^{r+t-n}}{\ln 2}$ on aura

$$H(F(W)|Z,F) \geq r - \frac{2^{r+t-n}}{\ln 2}.$$

Exemple 4.4 *Revenons au scénario 2.2 de la cryptographie quantique. $\mathcal{A}$lice et $\mathcal{B}$ob possèdent un mot secret W de $n = 10000$ bits. De ces 10 000 bits, $\mathcal{E}$ve en a obtenu $m = 1000$ directement et a ainsi provoqué environ 500 erreurs. $\mathcal{A}$lice et $\mathcal{B}$ob qui ont utilisé* **CASCADE** *pour réconcilier leurs mots ont révélé $l = 3286$ bits de plus pendant la phase de réconciliation. Dans ce cas $R(W|Z) = n - t = n - m - l = 5714$ bits. Supposons maintenant qu'ils choisissent une fonction F au hasard dans une classe universelle de fonctions sur $\{0,1\}^{10000} \to \{0,1\}^{5000}$. Si $\mathcal{E}$ve n'avait aucune information sur W on aurait que $H(F(W)|F) = 5000$ bits. Dans notre contexte ce sera prèsqu'aussi bon avec*

$$H(F(W)|Z,F) \geq 5000 - \frac{2^{-714}}{\ln 2} \text{ bits.}$$

Par analogie à $h(x)$, on définie $r(x) = -\log_2(x^2 + (1-x)^2)$ pour ce qui suit.

Exemple 4.5 *$\mathcal{A}$lice et $\mathcal{B}$ob possèdent un mot secret W de $n = 10000$ bits. $\mathcal{E}$ve apprend de l'information sur W au moyen d'un canal binaire symétrique avec probabilité d'erreur 25%. Dans ce cas, $R(W|Z) = n \log_2 r(0.25) \approx 0.678072n \approx 6781$ bits (dans ce cas $H(W|Z) \approx 8113$ bits). Supposons maintenant qu'ils choisissent une fonction F au hasard dans une classe universelle de fonctions sur $\{0,1\}^{10000} \to \{0,1\}^{6000}$. Si $\mathcal{E}$ve n'avait aucune information sur W on aurait que $H(F(W)|F) = 6000$ bits. Dans notre contexte ce sera prèsqu'aussi bon avec*

$$H(F(W)|Z,F) \geq 6000 - \frac{2^{-781}}{\ln 2} \text{ bits.}$$

Dans le cas du canal binaire symétrique de taux d'erreur ε, tel que l'exemple 4.5, il est en fait possible de distiller plus de $R(W|Z) = nr(\varepsilon)$ bits et d'atteindre $H(W|Z) = nh(\varepsilon)$. Consultez [2] pour les détails.

5 Autres Applications

Jusqu'ici toute notre attention s'est portée sur une situation bien précise d'échange d'une clef secrète, potentiellement à des fins cryptographiques. Les résultats précédent de réconciliation et distillation trouve également des applications dans d'autres situations où de l'information secrète est mise en cause. Deux autres exemple d'utilisation sont dans les protocoles cryptographiques de *mise-en-gage* et de *transfert inconscient*. Dans ces protocoles, il n'y a que deux participants $\mathcal{A}$lice et $\mathcal{B}$ob qui ne se font pas mutuellement confiance. Ces outils très puissants peuvent être utilisés pour la construction de nombreux autres protocoles cryptographiques [10].

Mise-en-gage $\mathcal{A}lice$ possède un bit b qu'elle veut éventuellement révéler à $\mathcal{B}ob$. Le but de la mise-en-gage est de fixer une fois pour toute la valeur de b mais d'une façon qui ne le donne pas à $\mathcal{B}ob$. La mise-en-gage consiste d'abord à la transmission d'une information à $\mathcal{B}ob$ qui doit être suffisante pour fixer b mais insuffisante pour que $\mathcal{B}ob$ découvre la valeur de b. Ces deux conditions, apparemment paradoxales, peuvent être satisfaites dans des scénarios quantique [5] ou avec canal bruité [11] grace aux techniques présentées dans ce document.

Transfert Inconsicent $\mathcal{A}lice$ possède deux bits b_0, b_1 qu'elle veut éventuellement révéler à $\mathcal{B}ob$. Le but du transfert inconscient est de permettre à $\mathcal{B}ob$ d'obtenir exactement l'un de b_0, b_1, selon son choix $c \in \{0, 1\}$ sans qu'$\mathcal{A}lice$ ne sache lequel il a obtenu. C'est à dire que l'échange d'information entre $\mathcal{A}lice$ et $\mathcal{B}ob$ ne doit contenir que l'information sur b_c, mais pas celle de c, ni celle de $b_{\bar{c}}$. Ces deux conditions, apparemment paradoxales, peuvent être satisfaites dans des scénarios quantique [3] ou avec canal bruité [11] grace aux techniques présentées dans ce document.

6 Conclusion

Nous avons présenté des méthodes de réconciliation publique permettant à deux participants de se mettre d'accord efficacement sur un secret commun à partir d'un secret partiellement erronné.

Nous avons également démontré comment deux participants possédant un secret commun partiellement secret peuvent en extraire un secret plus petit qui est très secret.

On a vu sur certains exemples comment appliquer ces techniques et comment les combiner. Un traîtement général sur la façon de les combiner se trouve dans [7].

References

[1] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, “Experimental quantum cryptography”, *Journal of Cryptology*, Vol. 5, no. 1, 1992, pp. 3–28.

[2] C.H. Bennett, G. Brassard, C. Crépeau, and U. Maurer. Generalized Privacy Amplification, *IEEE Transaction on Information Theory*, 1995. à paraître.

[3] C.H. Bennett, G. Brassard, C. Crépeau, M.-H. Skubiszewska, *Practical Quantum Oblivious Transfer*, Proceedings of CRYPTO'91, Springer Verlag, Berlin, 1992, pp 351–366.

[4] C. H. Bennett, G. Brassard and J.–M. Robert, “Privacy amplification by public discussion”, *SIAM Journal on Computing*, Vol. 17, no. 2, April 1988, pp. 210–229.

[5] G. Brassard, C. Crépeau, R. Jozsa, D. Langlois, *A Quantum Bit Commitment Scheme Provably Unbreakable by both Parties*, 29^{th} Symp. on Found. of Computer Sci., pages 362–371, IEEE,1993.

[6] G. Brassard and L. Salvail, “Reconciliation of a secret key through public discussion”, *Advances in Cryptology – Eurocrypt '93*, pages 410–423, Springer Verlag.

[7] C. Cachin and U. M. Maurer, “Linking information reconciliation and privacy amplification”, Proceedings of Eurocrypt '94, à paraître, 1994.

[8] J. L. Carter and M. N. Wegman, “Universal classes of hash functions”, *Journal of Computer and System Sciences*, Vol. 18, 1979, pp. 143–154.

[9] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich and R. Smolensky, “The bit extraction problem or t–resilient functions”, *Proceedings of 26th IEEE Symposium on Foundations of Computer Science*, October 1985, pp. 396–407.

[10] C. Crépeau. Cryptographic primitives and quantum theory. *Proceedings of Workshop on Physics and Computation, PhysComp 92*, pages 200–204, 1993.

[11] Crépeau, C., *Cryptographic Protocols based on Noisy Channel*, Manuscript, 1995.

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

[13] U. M. Maurer, “Secret key agreement by public discussion from common information”, *IEEE Transactions on Information Theory*, Vol. IT–39, 1993, pp. 733-742.

[14] U. M. Maurer “The strong secret key rate of discrete random triples”, à paraître dans *Proceedings of Symposium on Communications, Coding and Cryptography*, R. Blahut et al. (editors), Ascona, Switzerland, February 10–13, 1994, Kluwer.

[15] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North–Holland, New York, 1977.

[16] A. Rényi, “On measures of entropy and information”, *Proceedings of 4th Berkeley Symposium on Mathematics, Statistics and Probability*, Vol. 1, 1961, pp. 547–561.

[17] D. Spielman, *Linear-Time Encodable and Decodable Error-Correcting Codes*, 27th ANNUAL ACM STOC,1995.

[18] D. R. Stinson and J. L. Massey, “An infinite class of counterexamples to a conjecture concerning non-linear resilient functions” *Journal of Cryptology*, à paraître.

[19] A. D. Wyner, “The wire-tap channel”, *Bell System Technical Journal*, Vol. 54, no. 8, 1975, pp. 1355–1387.