Algorithme quantique de transmission de clés

Valentin Feray

Introduction (modélisation de la cryptographie)

Le but de la cryptographie est que **A** puisse envoyer un message à **B** sans qu'une personne quelconque **C** (appelée *espion*) qui intercepte le message ne puisse le comprendre.

Pour cela, **A** et **B** se donnent une fonction $f : \mathcal{M} \to \mathcal{C}$ injective, $\mathcal{M}$ étant l'ensemble des messages en clair m et $\mathcal{C}$ celui des cryptogrammes c.

Problème : si on envoie de nombreux messages, il ne faut pas employer systématiquement la même fonction f. En effet :

- **C** peut avoir besoin de communiquer par ailleurs avec **A** ou **B**.
- **C** peut deviner le sens de cryptogrammes peu importants.

Pour diversifier la manière de coder, on choisit f dans une famille de fonctions $(f_k)_{k\in\mathcal{K}}$ k est appelée la *clé*. $\mathcal{K}$ est l'ensemble des clés.

Apparaît alors une autre façon de voir les choses : on appelle système cryptographique la donnée de $\mathcal{M}$, $\mathcal{C}$, $\mathcal{K}$ et $f : \mathcal{M}\times\mathcal{K}\to\mathcal{C}$ vérifiant : $\forall k\in\mathcal{K}$, $m \to f(m,k)$ est injective.

Par exemple, k peut être une permutation de l'alphabet que l'on applique à chacune des lettres de m pour déterminer c.

Actuellement on a besoin d'envoyer de nombreux messages secrets avec de nombreuses personnes différentes (en particulier dans les échanges commerciaux). On considérera donc que l'espion connaît le système cryptographique utilisé.

1. Clé publique, clé privée

a) 2 principes totalement différents

L'espion connaît le cryptogramme c qu'il a intercepté ainsi que la fonction f qui fait partie du système cryptographique dont l'utilisation est largement répandue. Si le système est efficace, il ne connaît pas m et est incapable de le déterminer. Mais rien ne nous indique s'il connaît k.

A priori, k fixée, $m \to c = f(m,k)$ étant injective, quelqu'un connaissant c et k peut déterminer m. La clé k doit donc rester secrète. On parle de *cryptographie à clé privée*. L'espion ne connaissant pas k ne peut trouver m de manière sure.

Mais l'avènement de l'informatique a fait apparaître l'existence de fonctions qui, bien qu'elles soient inversibles en théorie, sont très difficiles à inverser (il est difficile de définir proprement et de manière efficace cette caractéristique, d'autant plus que certains problèmes d'existence d'algorithme efficace reste ouvert : cas du problème du voyageur de commerce). Utilisant ces fonctions, certains systèmes cryptographiques sont efficaces même lorsque l'espion connaît la clé k. On parle de *cryptographie à clé publique*.

b) exemple de cryptographie à clé publique.

Le système R.S.A. (Rivest, Shamir, Adleman) est actuellement le plus utilisé. Voici comment il fonctionne.

Il repose sur la propriété arithmétique suivante :

Soient p et q deux nombres premiers distincts.

On pose $n = p.q$, $\varphi(n) = (p-1).(q-1)$ (φ est la fonction indicatrice d'Euler)

Alors $\forall k \in \mathbb{N}$, $\forall a \in F_n$, $a^{k.\varphi(n)+1} \equiv a\ [n]$

Système cryptographique :

$\mathcal{K} = (n,e)$ avec n construit comme ci-dessus et e premier avec $\varphi(n)$

$\mathcal{M} \subset F_n$ (on peut par exemple le mettre sous forme de 0 et de 1 et couper le message en blocs de taille inférieur à $\log_2(n)$).

$\mathcal{C} = F_n$

$f(m,(n,e)) = m^e$

B qui a construit n est le seul à connaître p et q ainsi que $\varphi(n)$. Seul **B** est donc capable de déterminer d tel que $e.d = k.\varphi(n)+1$. **B** calcule c^d et retrouve m d'après la proposition ci-dessus.

Il est très difficile pour l'espion d'inverser $m \to f(m,k)$ (décomposer un nombre en facteurs premiers prend un temps exponentiel en la taille du nombre et bien qu'on ait pas démontré qu'il n'en existe pas, on ne connaît pas d'algorithme permettant d'inverser cette fonction sans factoriser n). En revanche il existe des critères de primalité efficaces. On peut donc construire « facilement » des entiers $n = p.q$ que l'espion aura beaucoup de mal à factoriser. C'est donc un système efficace en pratique bien que théoriquement une machine surpuissante le casse sans problème.

Un gros avantage est qu'il n'est pas nécessaire pour **A** et **B** d'avoir auparavant échangé une clé.

c) *clé privée*

Exemple simple : $\mathcal{M} = \{0;1\}^n$

$\mathcal{K} = \{0;1\}^p$

$C = \{0;1\}^n$

$m = m_0 m_1 \ldots m_{n-1}$, $m_i \in \{0;1\}$

$k = k_0 k_1 \ldots k_{p-1}$, $k_i \in \{0;1\}$

$f(m,k) = c = c_0 c_1 \ldots c_{n-1}$, $c_i \in \{0;1\}$, $c_i = m_i + k_r$ [2] où r est le reste de la division de i par p.

Si la clé est trop courte, les caractéristiques de la langue utilisée (fréquence des lettres par exemple) vont permettre de casser facilement le code.

Si $p = n$, le cryptogramme peut représenter n'importe quel message en clair. C'est le *système de Vernam* ou *one-time pad*.

L'énorme avantage est qu'un espion ne possédant pas la clé n'a aucun moyen même en disposant d'ordinateurs surpuissants de trouver le message en clair. Mais il y a un défaut énorme : ce système nécessite de s'échanger au préalable des clés très longues.

Transition :

On a vu qu'il existait deux types de systèmes cryptographiques. Nous allons dorénavant nous intéresser uniquement aux systèmes à clé privée. On ne se pose donc plus de questions de complexité des calculs : on supposera qu'il est toujours possibles de trouver les antécédents de c par f.

La théorie de Shannon va permettre d'étudier ces systèmes.

2. Théorie de Shannon – systèmes parfaits

a) approche probabiliste de la cryptographie

L'idée de Shannon est de considérer $\mathcal{M}$, $\mathcal{K}$ et $\mathcal{C}$ comme des variables aléatoires.

On va ainsi s'intéresser à $\mathbf{P}(\mathcal{M} = x \mid \mathcal{C} = y)$. L'espion ayant intercepté le cryptogramme regarde quel message il peut représenter avec quelle probabilité, la clé ayant été choisie aléatoirement (dans le cas où $\mathcal{K}$ est équirépartie, cela revient à rechercher les antécédents (m,k) de c).

Approche probabiliste appliquée au *one-time pad* en annexe.

On retrouve le fait que la connaissance du cryptogramme n'apporte aucune information sur le message en clair.

b) notions d'information et d'entropie

Pour étudier des systèmes plus complexes (où les valeurs prises par les variables aléatoires ne joue pas forcément des rôles symétriques), Shannon introduit :

- l'information associée à un événement $X = x$ définie par $-\log(\mathbf{P}(X = x))$

Ainsi, l'information est d'autant plus importante que l'événement est peu probable et les informations de deux événements indépendants s'ajoutent.

- l'entropie ou incertitude associée à une variable aléatoire

$H(X) = -\sum_x \mathbf{P}(X = x).\log(\mathbf{P}(X = x))$ (information moyenne apportée par X)

- l'entropie conditionnelle de X sachant Y (intéressante en cryptographie)

$H(X|Y) = -\sum_{x,y} \mathbf{P}(X = x, Y = y).\log(\mathbf{P}(X = x \mid Y = y))$

On a alors les résultats suivants :

$H(X,Y) = H(Y) + H(X|Y)$, à savoir l'information apportée par X et Y vaut celle apportée par Y plus celle apportée par X connaissant Y.

$H(X|Y) \leq H(X)$, à savoir l'incertitude associée à X diminue quand on connaît Y (information supplémentaire).

c) définition et caractéristique des systèmes parfaits en cryptographie

Shannon qualifie un système cryptographique de parfait si $H(\mathcal{M}|C) = H(\mathcal{M})$

Exemple : le *one-time pad* est parfait au sens de Shannon (cf. annexe)

Proposition : Pour un système parfait, $H(\mathcal{M}) \leq H(\mathcal{K})$.

En effet pour n'importe quel système, on a :

$$H(\mathcal{M}|C) \leq H((\mathcal{M},\mathcal{K})|C) = H(\mathcal{K}|C) + H(\mathcal{M}|(C,\mathcal{K})) = H(\mathcal{K}|C) \leq H(\mathcal{K})$$

Explications :

- La première inégalité est due au fait que $\mathcal{M}$ et $\mathcal{K}$ apportent plus d'informations que $\mathcal{M}$ seul.
- Les étapes 2 et 4 ont été expliquées plus haut ($H(X,Y) = H(Y) + H(X|Y)$ et $H(X|Y) \leq H(X)$)
- La deuxième égalité vient du fait que C et $\mathcal{K}$ déterminent $\mathcal{M}$ entièrement (on voit ici que la théorie ne peut s'appliquer à un système à clé publique).

Ainsi tout système parfait pouvant être utilisé pour tous les types de message ($\mathcal{M}$ équirépartie) vérifie $\#\mathcal{K} \geq \#\mathcal{M}$ (en effet, le nombre des valeurs qu'elle peut prendre étant fixée,

l'entropie d'une variable est maximale lorsqu'elle est équirépartie). Ainsi le *one-time pad* est optimal.

Transition :

Pour mettre au point un système cryptographique sûr qui ne se fonde pas sur la puissance de calcul limitée de l'espion, il faut donc trouver un moyen de se transmettre des clés de la taille des messages (se les transmettre lors d'une rencontre préalable n'est pas très efficace car on ne peut pas toujours prévoir qu'on aura besoin d'envoyer un message crypté).

Néanmoins, la transmission de clé présente deux avantages sur la transmission de messages :

- **A** ne doit pas forcément connaître la clé avant l'échange d'informations avec **B**.
- Si $\mathcal{K}$ est intercepté alors que **A** ou **B** s'en rend compte, on change de clé et le message en clair n'est pas découvert.

3. La cryptographie quantique

a) propriétés intéressantes des photons

La lumière étant une onde transverse, les photons qui la composent ont une infinité de directions de vibration (appelées *polarisations*) possibles. Mais étant donnée une base de directions orthogonales, chaque polarisation peut être vue comme une combinaison linéaire de ces polarisations orthogonales. Ainsi un polarisateur qui laisse passer les photons oscillants verticalement mais pas ceux oscillants horizontalement laissera passer une partie des photons ayant une direction quelconque. Lorsqu'on envoie un seul photon sur un filtre polaroïd de telle sorte que la direction du photon fasse un angle de 45° avec celle du filtre, celui-ci a une chance sur deux de passer et ressort avec la polarisation du filtre (annexe).

On ne peut donc pas connaître la direction de polarisation d'un photon en utilisant un polarisateur. C'est une conséquence du principe d'incertitude d'Heisenberg. De plus, il risque de modifier cette direction de polarisation. On peut donc peut-être utiliser cette propriété pour créer un système cryptographique où il est possible de repérer toute tentative d'espionnage. Le principe d'Heisenberg n'étant pas dû à une limite technologique, la cryptographie quantique ne repose pas sur la faiblesse de l'espion.

b) algorithme

En cryptographie quantique, on ne s'intéressera qu'à quatre polarisations (0°,45°,90°,135°). D'après les observations ci-dessus, on peut imaginer que l'on possède 2 types de détecteurs pour mesurer l'orientation d'un photon : détecteurs + (polarisateur vertical qui mesure 0° si le photon sort et 90° s'il est bloqué) ou x (polarisateur oblique) effectuant les mesures suivantes :

Détecteur	Photon à 0°	Photon à 45°	Photon à 90°	Photon à 135°
+	Mesure la bonne polarisation	Mesure soit 0° soit 90° (au hasard)	Mesure la bonne polarisation	Mesure soit 0° soit 90° (au hasard)
x	Mesure soit 45° soit 135° (au hasard)	Mesure la bonne polarisation	Mesure soit 45° soit 135° (au hasard)	Mesure la bonne polarisation

Quelqu'un qui veut connaître la direction d'un photon doit choisir un des deux détecteur. S'il n'a aucune information sur la polarisation du photon au départ, il ne peut la déterminer avec certitude.

L'algorithme suivant permet de se mettre d'accord sur une suite de 0 et de 1 qui pourra ensuite être utilisée comme clé pour un *one-time pad* classique. On associe aux polarisations 0° et 45° la valeur 0, 1 à 90° et 135°.

Etape 1 : **A** envoie une série de photons ayant des polarisations aléatoires parmi les 4 citées ci-dessus.

Etape 2 : **B** choisit pour chaque photon un détecteur au hasard. Il note les détecteurs utilisés et les polarisations obtenues.

Etape 3 : **B** publie la liste des détecteurs qu'il a utilisé et **A** lui indique pour quel photon il a mesuré la bonne polarisation. **A** et **B** ont maintenant en commun la liste des valeurs correspondants à ces photons. Les photons pour lesquels **B** n'a pas utilisé le bon détecteur ne sont bien sûr pas pris en compte (la liste obtenue n'est pas celle que **B** avait envoyée, on ne peut donc envoyer directement le message mais on peut s'en servir de clé).

Cet algorithme illustré en annexe nécessite que **A** et **B** puissent communiquer sur le canal publique et permet d'échanger des clés de manière sûre comme nous allons le voir dans le paragraphe suivant.

c) tentative d'espionnage.

Imaginons maintenant qu'une tierce personne **E** veuille intercepter la clé. Il intercepte les photons mais est obligé de choisir un détecteur au hasard pour mesurer leur orientation avant de les renvoyer à **B** (il ne peut les copier s'il ne les connaît pas leur orientation). Pour qu'il connaisse la clé avec certitude, **E** doit avoir choisi le bon détecteur à chaque fois que **B** l'a fait ce qui a 1 chance sur 2^n de se produire (n étant la taille de la clé). **E** ne peut donc pas connaître la clé.

Mais la cryptographie quantique possède un autre avantage : si quelqu'un a tenté d'intercepter la clé, **A** et **B** peuvent le savoir. En effet quand **E** se trompe de détecteur, il modifie l'orientation du photon. **B**, en utilisant le bon détecteur a alors une chance sur 2 de trouver la mauvaise valeur. En comparant une partie de leur clé (environ 100 chiffres) qu'ils n'utiliseront pas, ils seront donc sûr, s'il y a des erreurs, que **E** a tenté d'intercepter la clé. Ils doivent alors recommencer.

L'algorithme de transmission de clé à l'aide de photons permet donc d'être sûr que personne ne connaît la clé. En l'utilisant après avec le *one-time pad*, on a mis au point un système de cryptographie parfait protégé contre toute forme de cryptanalyse.

Conclusion :

La cryptographie quantique est un code théoriquement incassable. Mais, en pratique le transport de photons n'est réalisable que sur de courtes distance et l'efficacité du système R.S.A. plus simple à mettre en œuvre ne pousse pas à améliorer la distance maximale de transmission.