Université Paris 13, Institut Galilée
Département de Mathématiques
Licence 2ème année Informatique
2012-2013

Cours de Mathématiques pour l'Informatique
Cryptographie à clés publiques

Sylviane R. Schwer

Leçon du 8 avril 2014

On a vu en TD les fonctions de chiffrements affines, fondées sur les fonctions affines inversibles (ayant une réciproque) et les éléments inversibles de $\mathbb{Z}/n\mathbb{Z}$. Nous travaillons ici sur les fonctions exponentielles de $\mathbb{Z}/n\mathbb{Z}$.

Lorsque l'on (Bob) souhaite transmettre de l'information à un destinataire particulier (Alice) en toute confidentialité, plusieurs questions se posent :

1. Comment s'assurer qu'une tierce personne indélicate (Carl) n'intercepte le message?
2. Comment s'assurer qu'il n'y a pas eu de modification du message au cours de la transmission, soit intentionnellement (par Carl) ou non (problème technique) ?
3. Comment assurer Alice que le message émane bien de Bob ?
4. Comment remédier à des problèmes de transmission ?

L'une des façons de minimiser les problèmes de transmission est de minimiser la taille des messages à transmettre. C'est le même problème que de vouloir transmettre des données trop volumineuses. Il s'agit du problème de la *compression*. Voici quelques formats usuels de compression : pour les images jpeg, pour les sons mp3, pour les vidéos divx ou mp4, pour les données quelconques zip, rar, tar, ... Ce domaine ne sera pas étudié dans ce cours.

Le problème de la reconnaissance de l'émetteur du message se résout par la possibilité qu'à Bob de signer son message. Il s'agit d'ajouter au message une suite de caractères obtenue à l'aide d'algorithmes spécifiques appelés fonction de hachage sécurisée comme MD5 ou SHA-1 dont l'étude ne relève pas non plus de ce cours.

En revanche, les deux premières questions vont être traitées comme des applications de ce cours. La première question relève du *cryptage* ou *chiffrement* des informations, la seconde des codes correcteurs.

On peut sommairement cataloguer les systèmes de chiffrement en deux classes :

- Les codes à clef privée, c'est-à-dire que la fonction de cryptage n'est connu que de Bob et Alice. AES et DES permettent le chiffrement de gros messages rapidement.Mais il faut qu'il y ait alors un échange de ces informations (les clefs).

- Les codes à clef publique comme RSA. Le principe de ces clefs est simple : la fonction de codage est donnée dans une base publique ou envoyée telle qu'elle. C'est une fonction bijective dont la réciproque est plus coûteuse à calculer que l'importance du message (notamment le temps de calcul est plus long que la durée de pertinence du message). Ainsi, si Bob veut envoyer un message chiffré à Alice, il lui demande ou va consulté dans la base publique la clé de chiffrement d'Alice, puis il chiffre avec cette clé. Alice qui possède la fonction inverse de la clé qu'elle a transmise, pourra déchiffrer sans problème.

  Les fonctions utilisées se fonde sur la difficulté de trouver des grands nombres premiers.

1 Le système à clef symétrique El-Gamal (1985)

Ce système convient pour tous les groupes cycliques finis dans lequel le problème du logarithem discret est de grande complexité algorithmique (en temps). Nous nous contentons ici de $\mathbb{F}_p =< (\mathbb{Z}/p\mathbb{Z})^*, \times >$ et des fonctions $f_{p,g} \left| \begin{array}{ccc} \{1, \cdots, p-1\} & \longrightarrow & \mathbf{F}_p \\ x & \longmapsto & g^x \end{array} \right.$ bijectives, soit celles correspondant à g générateurs de $\mathbb{F}_p$, c'est-à-dire tels que $o(g) = p - 1$.

Alice et Bob choisissent un grand nombre premier p (au delà de 1000 bits soit supérieur à 2^{1000} ou 10^{300}) , et un générateur g de $\mathbb{F}_p =< \mathbb{Z}/p\mathbb{Z}, +, x >$, *i.e.* $g \wedge p = 1$ qu'ils publient.

Quand Bob souhaite envoyer un message m à Alice, qui est représenté par un élément $\bar{m} \in \mathbb{F}_p$.

1. Il demande à Alice de choisir secrètement (sans le lui dire) un nombre $\alpha \in [2, p-2]$.

   Elle calcule $\mathbf{a} = [g^\alpha]_p$ et publie son résultat $\mathbf{a}$.

   Ainsi, α est la clé secrète, même pour Bob et $(p, g, \mathbf{a})$ est la clé publique, accessible à Bob.

2. Bob choisit secrètement un entier naturel $\beta \in [2, p-2]$, il calcule $\mathbf{b} = [g^\beta]_p$ et $m' = m\mathbf{a}^\beta$ puis il publie le couple $(\mathbf{b}, m')$

3. Alice peut déchiffrer le message en calculant l'inverse de $\mathbf{a}^\beta$.

   Or $\mathbf{a}^\beta = (g^\alpha)^\beta = g^{\alpha \times \beta} = (g^\beta)^\alpha = \mathbf{b}^\alpha$ avec $\mathbf{b}$ public et α qu'elle a choisi.

   Retrouver $\bar{m}$, c'est trouver dans $\mathbb{Z}/p\mathbb{Z}$ l'inverse de $\mathbf{b}^\alpha$.

   La fonction $f_{p,g^{ab}}$ est donc connue et d'Alice et de Bob. Elle est utilisée à la fois pour coder et décoder, c'est pourquoi on l'appelle *symétrique*.

Le système suivant permet à quiconque veut envoyer un message chiffré à Alice de le faire sans avoir de contact préalable avec elle.

2 Le système à clef disymétrique RSA

La création de la cryptographie à clé publique par Diffie et Hellman en 1976 l'invention du RSA par Rivest, Shamir, et Adleman in 1978 ont constitué un tournant dans la longue histoire des échanges d'informations secrètes.

Bob veut envoyer un message à Alice. Il faut au préalable qu'Alice ait publié sa clef publique RSA. Pour cela, elle procède ainsi :

1. Alice choisit deux grands nombres premiers p et q et calcule $n = pq$.
2. Alice choisit un entier e tel que $e \wedge \varphi(n) = 1$ et calcule son inverse d dans $\mathbb{Z}_n$.
3. Alice publie (n, e) et garde privé d

Ainsi, pour envoyer un message crypté à Alice, il suffit d'utiliser la fonction bijective

$$f_{n,e} \left| \begin{array}{ccc} \mathbb{Z}_n & \longrightarrow & \mathbb{Z}_n \\ x & \longmapsto & x^e \end{array} \right.$$

dont la réciproque est la fonction $f_{n,d}$.

Bob veut transmettre en toute sécurité à Alice le message représenté par un nombre $m \in [0, n-1]$. Il lui envoie $m' \equiv m^e \quad (n)$ et $m' \in [0, n-1]$.

Vérifions qu'Alice calcule $m \equiv [m']^d \quad (n)$.

$[m']^d = m^{ed} \quad (pq)$. Donc $[m']^d = m^{ed} \quad (p)$ et $[m']^d = m^{ed} \quad (q)$ par le théorème des restes chinois.

Soit $m \equiv 0 \quad (p)$ [resp. $m \equiv 0 \quad (q)$] , alors $m^{ed} \equiv 0 \quad (p)$ [resp. $m^{ed} \equiv 0 \quad (q)$].

Sinon, d'après le petit théorème de Fermat, $m^{p-1} \equiv 1 \quad (p)$ [resp. $m^{q-1} \equiv 1 \quad (q)$].

Or $ed \equiv 1 \quad (p-1)(q-1)$, c'est-à-dire qu'il existe un entier $k \in \mathbb{Z}$ tel que $ed = 1 + k(p-1)(q-1)$. Donc $m^{ed} = m^{1+k(p-1)(q-1)} = m.m^{k(p-1)(q-1)} \equiv m \quad (p)$ et de même $m^{ed} \equiv m \quad (q)$. D'après le théorème des restes chinois, $m^{ed} \equiv m \quad (pq)$.

exemple

1. Eve dépose la clé (9797,181). C'est une clé RSA, car de la forme $(p \times q, e)$, avec e un nombre premier avec $\varphi(pq) = (p-1)(q-1)$.
2. Bob veut transmettre une information par la méthode RSA à Eve. Il utilise donc la fonction $[x^{181}]_{9797}$. La fonction utilisée par Eve pour retrouver l'information initiale est la fonction réciproque de $[x]_{9797} \to [x^{181}]_{9797}$, c'est la fonction $[x]_{9797} \to [x^u]_{9797}$, avec u est l'inverse de 181 dans $\mathbb{Z}/\varphi(n)\mathbb{Z}$.