Université de Bordeaux
Mathématiques

Préparation à l'agrégation
2013–2014

Logarithme discret et cryptographie

Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury demande que la discussion soit accompagnée d'exemples traités sur ordinateur. Il est souhaitable que vous organisiez votre présentation comme si le jury n'avait pas connaissance du texte. Le jury aura néanmoins le texte sous les yeux pendant votre exposé.

[*Texte adapté du texte homonyme dans* Modélisation mathématique : un autre regard*, édité par A. Lichnewsky, collection Scopos, vol.* ***16****. La section sur RSA a été supprimée*]

1. Introduction

Les méthodes empiriques traditionnelles de chiffrement étaient fondamentalement symétriques. Elles reposaient par exemple souvent sur le schéma suivant : on choisit une permutation $\sigma \in S_{26}$ sur 26 lettres une fois pour toutes ; le codage d'un texte consiste alors à appliquer cette permutation à chaque groupe de 26 lettres successifs ; le décodage consiste à appliquer la permutation réciproque σ^{-1}. Numériquement, on peut aussi, si le message est codé par exemple sur 64 bits, employer cette technique en choisissant une permutation $\sigma \in S_{64}$. Outre les fragilités exhibées par l'analyse fréquentielle (certains motifs d'un texte écrit en français sont plus fréquents que d'autres), la symétrie du système est un talon d'Achille : si quelqu'un sait coder, il sait du même coup décoder, car il est facile de trouver l'inverse d'une permutation sur 26, 64, ou même 10^6 lettres.

Les cryptosystèmes à clé publique, comme le nom l'indique, sont tel que le *codage* de l'information est public : tout le monde connaît l'algorithme pour coder le message, qui dépend d'une clé publique. Mais on ne peut pas en déduire le décodage, qui dépend d'une clé secrète mathématiquement liée à la clé publique, sans qu'on connaisse d'algorithme praticable permettant d'expliciter ce lien. En fait cela revient à construire une permutation σ d'un ensemble à N élements, mais ici N est gigantesque (de l'ordre de 10^{500}, par exemple). On ne peut même pas écrire la liste des ses éléments, et la méthode habituelle de calcul de la permutation réciproque ne peut plus s'appliquer.

Un problème voisin du précédent consiste à s'assurer de l'identité d'un interlocuteur. Par exemple, si A introduit une carte bancaire dans un lecteur, celui-ci doit s'assurer que A est bien le possesseur de la carte en lui demandant son code secret. La méthode utilisée est la suivante : sur la carte figure le code secret *codé* par un algorithme à clé publique. Lorsque l'on tape le code sur le lecteur, celui-ci code ce nombre, et se contente de vérifier qu'il est identique au mot de passe codé qu'il possède. Une alternative générique est la suivante : supposons que A connaisse la clé publique P d'un algorithme à clé publique, et que B soit le seul à en connaître la clé secrète S. On convient de considérer P et S comme deux permutations réciproques : $P \circ S$ comme $S \circ P$ sont l'application identique.

Par suite A a une méthode simple pour s'assurer de l'identité de B : il transmet un court texte t, reçoit en retour la signature $S(t)$, et il lui suffit de vérifier que $P(S(t)) = t$.

2. Le logarithme discret

Soit G un groupe cyclique d'ordre N, et soit g l'un de ses générateurs. Le *logarithme discret* de G en base g est l'application $\log_g \colon G \to \mathbb{Z}/N\mathbb{Z}$ définie par $\log_g(g^m) = m$. C'est donc un isomorphisme de groupes, d'application réciproque $m \mapsto g^m$.

Pour certains groupes G, le calcul de $\log_g$ est trivial : par exemple pour $G = (\mathbb{Z}/N\mathbb{Z}, +)$. Par contre, si $G = (\mathbb{Z}/p\mathbb{Z})^*$, où p est un nombre premier, trouver un générateur de G n'est déjà pas facile : il faut savoir factoriser $p-1$ pour garantir le résultat. Et le calcul du logarithme discret semble très difficile.

Ceci a donné l'idée d'algorithmes d'authentification symétriques : supposons que A et B veuillent s'assurer de leur identité respectives. Ils se donnent un groupe G cyclique de générateur g ; A (resp. B) choisit au hasard un $a \in \mathbb{Z}/N\mathbb{Z}$ (resp. b) et publie g^a (resp g^b) ; A peut s'assurer de l'identité de B en vérifiant que celui-ci sait bien calculer $g^{ab} = (g^a)^b$ (dont A connaît la valeur $= (g^b)^a$). Et vice-versa.

3. Preuves sans apport d'information

Le talon d'Achille de la méthode de signature sécurisée décrite en introduction réside dans le fait qu'à un moment donné, la personne qui doit prouver son identité transmet son code en clair. Il est donc possible à une personne mal intentionnée d'intercepter ce code (par exemple, dans le cas d'un lecteur de cartes, en regardant les chiffres que vous tapez). Plus généralement, supposons que A possède un secret (ici un code de carte bancaire), et que B veuille s'assurer que A possède ce secret. Est-ce possible sans que A ne donne à B une quelconque information sur le secret lui-même ? La réponse est oui, en utilisant un protocole probabiliste, proche d'un QCM particulièrement strict : à la moindre erreur on est rejeté. Nous en donnons deux ci-dessous, assez voisins :

3.1. **Racine carrée.** Soit $(G, \times)$ un groupe dans lequel on sait calculer rapidement, mais dans lequel le calcul d'une racine carrée est difficile (cette fois, $G = \mathbb{F}_p^*$ ne convient pas ; mais $G = (\mathbb{Z}/pq\mathbb{Z})^*$ si, où p, q sont de grands nombres premiers distincts).

Le secret de A est une racine carrée x d'un élément $a \in G$; B connaît a et veut s'assurer que A connaît x. Il demande à A de choisir $y \in G$ à sa guise et de lui envoyer $b = y^2$; il aura alors le choix entre deux questions qu'il pourra poser à A :

- lui demander $c = xy$: il peut maintenant vérifier que $ab = c^2$;
- lui demander y : il peut maintenant vérifier que $b = y^2$.

Supposons que A ne connaisse pas x ; il peut parier que B va lui poser

- la première question : il choisit $z \in G$ et envoie $b = a/z^2$ à B ; si B pose bien la question attendue, le coup de bluff de A aura marché, car il pourra envoyer $c = a/z$; mais sinon, A est piégé car il ne connaît pas la racine carrée de b.
- la deuxième question : il joue le jeu en choisissant y et en envoyant y^2 à B. Mais il est piégé si B choisit la première question.

Après n questions, la probabilité de succès que A ne se soit jamais trompé est 2^{-n}. En tout état de cause, les informations recueillies par B pendant son questionnaire ne lui permettent pas de deviner le secret x.

3.2. **Logarithme discret.** Soit G un groupe cyclique de générateur g dans lequel le problème du logarithme discret est difficile. Ici le secret de A est un entier x, et B connaît g^x. Il doit s'assurer que A connaît bien x. Il demande à A de choisir un entier y et de lui envoyer $b = g^y$. Il a maintenant le choix entre deux questions

- demander y à A : il vérifie que $b = g^y$;
- demander $z = x + y$ à A : il vérifie que $ab = g^z$.

4. Suggestions

Le candidat pourra choisir de développer les points qu'il désire sur le texte.

- Plusieurs points sont affirmés dans démonstration, le candidat est invité à les expliciter.
- Le candidat pourra programmer le calcul du logarithme discret dans le groupe $(\mathbb{Z}/N\mathbb{Z}, +)$, puis dans le cas du groupe multiplicatif $\mathbb{F}_p^*$, avec l'algorithme de son choix. Ou encore programmer une des méthodes de preuve sans apport d'information proposées dans le texte.