1) Dossier sur les fonctions booléennes en cryptographie

Les fonctions booléennes de plusieurs variables booléennes, c'est-à-dire les fonctions de la forme f(x1,x2,...,xn) où les variables xi ne prennent que les deux valeurs 0 ou 1, et dont le résultat est 0 ou 1, ont une grande importance en cryptographie. Elles entrent dans la description des systèmes à clé secrète par bloc ou par flot, dans la conception et l'analyse des générateurs pseudo-aléatoires cryptographiques, ainsi que dans d'autres circonstances comme par exemple les systèmes à base de codes correcteurs d'erreurs. On est loin de tout savoir sur ces fonctions, et en particulier, on aimerait savoir mieux construire certains types de fonctions booléennes utiles pour la cryptographie.

Le dossier que nous proposons sur les fonctions booléennes est constitué de liens vers des textes et des mises au point. Tout d'abord au niveau de la recherche actuelle sur ce domaine, nous proposons la page de Claude Carlet , qui est un des meilleurs spécialistes de la question. On y trouvera en particulier son texte "B o olean Functions for Cryptography and Error Correcting Codes " ainsi que son texte "V ectorial (multi-output) Boolean Functions for Cryptography ".

Nous proposons aussi les articles suivants de François Rodier, qui apportent des résultats nouveaux sur le comportement asymptotique de la non-linéarité des fonctions booléennes : " S ur la non-linéaritée des fonctions booléennes, Acta Arithmetica,115, p.1-22, 2004 ", " Asymptotic nonlinearity of Boolean functions, Designs, Codes and Cryptography, 40, p.59-70, 2006 ", " On the nonlinearity of Boolean functions,Proceedings of WCC2003, workshop on coding and cryptography, INRIA, p.397-405, 2003 ".

Pour des questions plus élémentaires, on pourra trouver une présentation de base sur les algèbres de Boole et fonctions booléennes dans le texte de Robert Rolland : " Algèbres de Boole - Fonctions booléennes " et dans les transparents " Algèbres de Boole et Fonctions booléennes ".

D'autres notions interviennent aussi dans le traitement des fonctions booléennes. On trouvera un certain nombre de textes de Robert Rolland sur ces notions :

" Compact de Cantor ", " Transformée de Fourier Discrète ", " signaux finis ", " Fonction de Mobius ", " Outils pour l'étude des propriétés asymptotiques des fonctions booléennes ", " Remarques sur le calcul de la transformation d'Hadamard "

2) Dossier sur l'utilisation des codes correcteurs en cryptographie à clé publique

Nous remercions **Pascal Véron** (Université du Sud Toulon Var) de nous avoir fourni ce dossier sur la théorie du codage et la cryptographie.

La théorie du codage et la cryptographie

La théorie du codage a pour but d'élaborer des mécanismes permettant de transformer avant son envoi un message m de telle sorte que si des perturbations surviennent sur le canal de communication, le destinataire dispose d'une méthode algorithmique lui permettant de reconstituer le message d'origine. Si les perturbations sont provoquées de façon intentionelle par l'expéditeur et si les paramètres de l'algorithme de reconstitution ne sont connus que du destinataire, ce protocole de communication "fiabilisé" se transforme tout naturellement en un système de chiffrement. Evidemment les objets utilisés pour réaliser un tel protocole devront satisfaire un certain nombre de propriétés que nous détaillerons par la suite.

Actuellement, la sécurité de la majorité des protocoles cryptographiques dépend dangereusement de la difficulté de résolution de seulement deux problèmes : la factorisation et le calcul du logarithme discret. Disposer de problèmes difficiles différents permettant d'élaborer

des fonctions à brèche secrète s'avère être une nécessité. Le problème SD (Syndrome Decoding) issu de la théorie du codage est l'un d'entre eux.

Le système de chiffrement à clé publique de Mc Eliece (développé en 1978) est le premier protocole cryptographique dont la sécurité dépend en partie de la difficulté de résolution du problème SD. Depuis l'utilisation des codes s'est généralisée à d'autres domaines de la cryptographie : identification, générateurs aléatoires, signature, fonctions de hachage. Tous les protocoles construits autour du problème SD ont l'avantage de n'utiliser que des opérations extrêmement simples (arithmétique du corps à 2 éléments) comparativement à leur homologues basés sur la factorisation ou le logarithme discret. Malgré cela, la communauté cryptographique reste encore assez réticente en ce qui conerne leur utilisation du fait de la taille des clés manipulées dans ces protocoles. Cependant après 30 ans d'existence, force est de constater que la cryptanalyse du système Mc Eliece est toujours exponentielle, ce qui n'est pas le cas du RSA ...

Pour la suite, nous considèrerons que le lecteur est familier avec la théorie du codage. Dans le cas contraire, il pourra consulter [PW]. Seule une connaissance minimale sur les codes linéaires est nécessaire pour aborder sans difficulté la suite de ce dossier.

Le problème SD

Ce problème de décision s'énonce de la façon suivante :

Entrée : H une matrice binaire (r,n) , p un entier , s un vecteur binaire de taille r.
Question : Existe-t-il un vecteur binaire x de longueur n et de poids au plus p tel que Hx = s ?

Ce problème qui peut se réduire en temps polynomial au problème des 3-mariages fait partie de la classe des problèmes NP-complets (cf. [BMT]), même si on suppose que la matrice H est de rang maximum. Reformuler dans le langage de la théorie du codage, ce problème correspond à rechercher un mot de syndrome donné
s
et de poids borné par
p
. Ceci est en lien direct avec la problématique générale du décodage d'un code linéaire binaire. En effet pour décoder un mot de la forme
c+e
où
c
est un mot de code et

e
une erreur de poids inférieur à
t
(la capacité de correction du code), on calcule
s=H(c+e)=He
où
H
est la matrice de contrôle du code. Parmi l'ensemble des solutions
y
vérifiant
Hy=s
,
e
est celle de plus petit poids, toutes les autres sont de poids strictement supérieur à
t
. Décoder
c
+
e
revient donc à rechercher un mot de poids
t
et de syndrome
s
.

Bien que le problème SD soit NP-complet, encore faut-il identifier les instances pour lesquelles en pratique la recherche d'une solution s'avère être impossible en temps raisonnable. Il existe essentiellement six algorithmes (probabilistes) permettant de résoudre dans certains cas le problème SD :
algorithme de Mc Eliece, algorithme de Lee et Brickell, algorithme de Leon, algorithme de Stern, algorithme de Canteaut-Chabaud et algorithme de Johansson et Jönsson.
Tous ces algorithmes s'intéressent à la recherche d'un mot de petit poids dans un code binaire (problème auquel SD peut être réduit en temps polynomial). L'algorithme de Johansson et Jönsson est légèrement différent des autres, son objectif est de décoder un mot à partir d'une liste de mots reçus. Une analyse détaillée des cinq premiers algorithmes est disponible dans [Can]. Il en ressort que les instances difficiles du problème SD sont obtenues lorsque le poids recherché est proche de la distance minimale théorique d du code donnée par la borne de Gilbert-Varshamov : $H_2(d/n) = 1-k/n$.
A titre d'exemple, la recherche d'un mot de poids
33
dans un code de longueur
2048
et de dimension
1685

nécessite de l'ordre de
2^{80}
opérations.

Le cryptosystème à clé publique de Mc Eliece

Le principe de ce système consiste à utiliser un code C (polynomialement décodable) pour fonction de chiffrement. On associe au texte clair, un mot de code (via la matrice génératrice) puis ce mot est volontairement perturbé par des erreurs avant d'être expédié. Le destinataire utilise alors l'algorithme de décodage pour reconstituer le message initial. Cependant, dans le contexte de la cryptographie à clé publique, afin d'éviter qu'un intrus puisse retrouver l'algorithme de décodage à partir de la structure du code utilisé, il faut masquer ce dernier. En l'absence d'informations supplémentaires, le code masqué semblera aléatoire et l'attaquant se retrouvera face au problème général du décodage d'un code binaire quelconque, en d'autres mots face au problème SD. Le système fonctionne de la façon suivante :

Clé secrète:

1) Un code linéaire binaire C(n,k,d) pour lequel il existe un algorithme polynomial A de décodage corrigeant
t
erreurs,
2) S(k,k) une matrice inversible,
3) P(n,n) une matrice de permutation.

Clé publique :

(G'=SGP, t) où G est la matrice génératrice du code C.

Chiffrement:

1) Message : m un élément de $\{0,1\}^k$,
2) Cryptogramme : c=mG' + e où e est un vecteur binaire de poids t.

Déhiffrement:

Étant donné que eP^{-1} et e ont le même poids, calculer :

1) $mS=A(cP^{-1})=A((mS)G + eP^{-1})$,
2) $m = (mS)S^{-1}$.

Le code C doit satisfaire un certain nombre de contraintes :
1) à n, k, et d fixés , il doit appartenir à une classe suffisamment importante afin qu'il ne soit pas possible de le construire via une simple énumération exhaustive (remarquons qu'il suffit en fait de trouver un code équivalent au code C).
2) le code doit être décodable en temps polynomial.3) aucune information ne doit pouvoir être obtenu sur le code secret C à partir de la matrice publique G'.

Il semble qu'actuellement la classe des codes de Goppa classiques (suggérée par Mc Eliece) satisfasse ces 3 conditions. En ce qui concerne la sécurité du système, si l'on admet que le problème de l'indistinguabilité des codes de Goppa (pouvoir affirmer qu'une matrice génératrice G est la matrice génératrice d'un certain code de Goppa) est difficile, un attaquant se retrouve donc face à la résolution d'une instance du problème SD. En réalité le poids de l'erreur n'étant pas réellement aléatoire (du fait des propriétés des codes de Goppa), le problème sous-jacent est dérivé du problème SD et se nomme GPBD (Goppa Parameterized Bounded Deconding) qui est lui aussi NP-complet (cf. [Fin]).

Le tableau ci-dessous établit une comparaison entre le système de Mc Eliece et le système RSA. Les paramètres utilisés sont ceux qui sont recommandés actuellement. Le taux de transmission correspond au rapport (nombre de bits du text clair)/(nombre de bits du cryptogramme) .

	Mac Eliece (2048,1718,30)	RSA-2048, e=2^16+1
Taille clé Publique	179.5 Ko	512 Octets
Taux de transmission	83.9%	100%
Nombre d'op binaires/		

bit d'info (chiffement)

1025 40555

Nombre d'op binaires/

bit d'info (déchiffement)

2311 6557176

Il existe une version duale de l'algorithme de Mc Eliece (dûe à Niederreiter) dans laquelle la matrice publique utilisée est une matrice de contrôle. Pour les mêmes paramètres, la taille de la clé publique est alors de 70867 octets. Cependant l'encodage initial des textes clairs n'y est pas aussi simple que dans le système Mc Eliece.

Le lecteur intéressé trouvera de plus amples informations sur l'interaction entre les codes correcteurs d'erreurs et la cryptographie dans [BRV] au sein duquel sont décrits notamment d'autres protocoles issus du problème SD (schéma d'identification, générateur aléatoire, schéma de signature et fonction de hachage). Finalement, nous terminerons ce dossier en mentionnant les travaux récents de Pierre-Louis Cayrel, Philippe Gaborit et Marc Girault sur l'utilisation des codes pour les schémas d'authentification basés sur l'identité [CGG].

Bibliographie:

[BMT] Berlekamp E.R., McEliece R.J., Van Tilborg H.C.A., "On the intractability of certain coding problems", IEEE Trans. on info. theory, vol. 24, n. 3, p. 384-386, 1978.
[BRV] Barthélemy P., Rolland R., Véron P., Cryptographie, principes et mises en oeuvre, Hermès Science, 2005.
[Can] Canteaut A., Attaques de cryptosystèmes à mots de poids faible et construction de fonctions t-résilientes , thèse, Université Paris VI, 1996.
[CGG] Cayrel P.-L., Gaborit P., Girault M., "**Identity-based identification and signature schemes using correcting codes",** *International Workshop on Coding and Cryptography, WCC 2007.*
[Fin] Finiasz M., Nouvelles constructions utilisant des codes correcteurs d'erreurs en cryptographie à clé publique , thèse, Ecole Polytechnique, 2004.
[PW] Papini O., Wolfmann J., Algèbre Discrète et Codes Correcteurs, Springer-Verlag, 1995.

[Retour]