Master M2 Cryptis - Université de Limoges

Examen Codes et cryptographie - 21 fevrier 2020

Documents autorisés - durée 3h

Questions de cours:

a. Soit un code de Reed-Solomon $[10, 2, 9]$ sur le corps $GF(11)$ jusqu"a quelle distance peut on décoder avec un décodage classique ? Et avec l'algorithme de Sudan (justifier le calcul)?

b. Soit un code de Goppa $[2^m, 2^m - mt, 2t+1]$ qui peut décoder t erreurs, quelle est la densité de mots de l'espace qu'il peut décoder (ie le rapport des mots decodables de l'espace par le nombre de mots total de l'espace)? (justifier le calcul).

c. Soit un code de Reed-Solomon $[2^m - 1, k]$ sur $GF(2^m)$, pour $k = n - 2t$ et $n >> t$ calculer de maniere approchée la densité des mots décodés par le code. Comparer à la densité obtenue dans la question précédente.

d. Soit le corps $K = GF(q^m)$, on considère un code en metrique rang de longueur n et de dimension k. Rappeler le principe de la metrique rang, donner une borne inferieure equivalente à la borne de Singleton pour la metrique rang (attention la borne doit dependre de m,n et k, pas simplement de n et k).

e) Des codes codes linéaires binaires $[20, 9, 15]$ et $[16, 9, 9]$ peuvent-ils exister ? (justifier)

Partie I (Authentification par les codes : schéma de Veron):

Les questions sont indépendantes, les question 1,2,5,6 sont faciles, la 3 un peu moins et la 4 encore moins

On considère le schéma d'authentification de Veron, qui est une variation sur le schéma d'authentification de Stern vu en cours. Dans le cas de l'algo de Stern, la clé publique est un syndrome et la clé secrete un mot de petit poids associé, dans le cas de Veron, la clé publique x est un mot du code mG (G une matrice generatrice d'un code aleatoire $[n, k]$) bruité par une erreur e de poids w. Plus précisèment la clé publique est le triplet (G, x, w), avec G une matrice aleatoire $k \times n$, $x = mG + e$ et w le poids de e. La clé secrete est le couple décodé (m, e) (qu'on supposera unique pour un x fixé). On supposera dans la suite que C est un code de parametre $[n, n/2]$ (typiquement $n = 700$). Le protocole a pour but de montrer que le prouveur P connait le decodage (m, e) du mot bruité $mG + e$ au verifieur V. Dans la suite h est une fonction de hachage, on considere de

1. [Engagement] P choisit au hasard $u \in GF(2)^k$ et une permutation σ de $\{1, 2, \ldots, n\}$. P envoie à V les engagements c_1, c_2 and c_3 tels que:

$$c_1 = h(\sigma);\ c_2 = h(\sigma((u+m)G)); c_3 = h(\sigma(uG + x));$$

2. [Défi] V envoie $b \in \{0, 1, 2\}$ à P.

3. [Réponse] 3 cas :
   - si $b = 0$: P revele $(u + m)$ et σ.
   - si $b = 1$: P revele $\sigma((u + m)G)$ et $\sigma(e)$.
   - si $b = 2$: P revele u and σ.

4. [Verification Step] 3 cas :
   - si $b = 0$: V verifie que c_1, c_2 ont été calculé honnetement (ie qu'il est capable de reconstruire c_1 et c_2 et que cela correspond aux valeur de l'engagement).
   - si $b = 1$: V verifie que c_2, c_3 ont été calculé honnetement, et que le poids de $(\sigma(e)) = w$.
   - si $b = 2$: V verifie que c_1, c_3 ont été calculé honnetement.

Figure 1: Protocol of Veron

plus que la description la permutation σ équivaut à donner une 'graine' de 80 bits qui permet de reconstruire σ.

1) Montrer que le protocole fonctionne (montrer que si tout se passe normalement le verifieur peut effectivement verifier tous les cas), quand se sert-on de la clé publique ?

2) Montrer qu'un tricheur peut facilement anticiper n'importe quel choix de b pour le défi (ie choisir un engagement adequat qui lui permet de se faire passer pour P)

3) Montrer qu'un tricheur peut facilement anticiper 2 choix sur 3 de b (ie soit $b = 0$ ou 1, soit $b = 1$ ou 2, soit $b = 0$ ou 2). (Montrer au moins un des trois cas au choix $(0,1), (1,2)$ ou $(0,2)$). En déduire que la proba de triche est au moins 2/3.

4) **(question difficile)** Montrer que si un tricheur peut anticiper les 3 possibilités pour b, alors soit il est capable de trouver une collision pour la fonction de hachage h, soit il connait le secret m. (indice: l'idée est de dire que si un tricheur peut repondre à tout b, alors il est capable de construire des c_i de manière differente, et donc ou bien ces valeurs sont egales auquel cas on montre

qu'on connait le secret, ou bien on a trouvé une collision pour h). On en déduit que la proba de triche est exactement 2/3.

5) Calculer le cout moyen pour les communications (nbre de bits envoyés lors du protocole) pour l'execution de 1 tour dans le cas $n = 700, k = n/2$ et la taille du haché 160 bits. Si on veut une authentification avec proba de triche de 2^{-32}, combien de fois faut-il executer le protocole ? Quelle est alors le cout moyen des communications pour une telle authentification ?

6) Ecrire le cout des communications pour le protocole de Veron en fonction de $n, k = n/2$ pour un haché de taille 160. Faire pareil pour Stern, montrer que le protocole de Veron permet de gagner un peu sur le cout des communications.

Partie II: Une fausse bonne idee

un cryptographe du dimanche pense à un nouveau schéma de signature à base de codes.

Soit n un entier (a priori grand de l'ordre de 5000), soit $a = (a_1, a_2, ..., a_n)$ un mot random de F_2^n, on considère la matrice $x \times n$ A obtenue par n shifts de a vers la droite. (A est cyclique). On considère la matrice $n \times 2n$ $H = (IA)$.

Soit (x, y) de longueur $2n$ avec x et y des mots de petit poids w de F_2^n. On considere les matrices circulantes $n \times n$ X et Y obtenue comme A à partir de x et y. Soit S=X+YA.

Pour le schéma on note: clé privée: (x,y) Clé publique S.

Signature: pour un message M et pour h(.) une fonction de hachage qui rend des mots random de petits poids t et de longueur n, on appelle $Sign(M) = (h(M)^t)(XY)$.

Verification: le verifieur verifie que $H.(Sign(M)^t) = F.h(M)$ et que le poids de Sign(M) est plus petit que $2wt$.

1) Pour une matrice $n \times 2n$ H, et pour b random de F_2^n, rappeler pour quel poids de z il est en moyenne facile de trouver un z tel que $H.z^t = b$

2) Expliquer pourquoi le schema de signature fonctionne, d'apres 1) quelle contrainte a-t-on sur le poids de Sign(M) ? Si w et t sont $\ll$ n, quel est en general le poids de la signature ??

3) Quelle est la taille des parametres du schema ? (clé publique, clé secrete) ? en utilisant des propri`etés de cyclicité peut on les reduire ?

4) on veut maintenant attaquer le schema. Montrer qu'a partir de la signature il ets tres facile de retrouver la cl secrete.

5) Comme le schema precedent est tres facile a casser, notre cryptographe du dimanche decide de considerer la variation suivante: memes clés.

Signature: on prend u dans F_2^{2n} random de poids W (avec W de taille 'moyenne')

SIgn(M)= le couple $(Z = u + h(M, U)(XY), U = H.u^t)$

Verification $H.(Sign(M)^t) = U + (h(M, U)^t)(XY)$ et Poids (Sign(M)) plus petit que W+2wt

a) Montrer que la signature fonctionne.

b) Quelles sont les contraintes sur W, w, t pour que le probleme reste dur ?

c) Montrer que le schema peut s'attaquer facilement à partir de plusieurs signatures obtenues (penser à des attaques statistiques pour retrouver chacunes positions de la clé secrete).

d) Montrer qu'en passant par les codes MDPC on peut attaquer encore plus rapidement.

6) Peut on imaginer le meme type de schema en metrique rang ???

Partie III- Métrique rang

0) Donner la définition de la métrique rang pour un code sur $GF(q^m)$. Qu'est-ce que le support d'un mot en metrique rang ?

1) En métrique de Hamming, on utilise souvent des permutations pour masquer la structure d'un code, pourquoi ? Quel est l'équivalent de cette notion de permutation en métrique rang ? et pourquoi ?

2) Codes LRPC. On considère une matrice LRPC (Low Rank Parity Check code) $H(h_{ij})$, $(n-k)\times n$ sur $K = GF(q^m)$ (pour fixer les idees q=2, m=40, k=n/2, n=40 par exemple), ou tous les h_{ij} appartiennent à un meme sous-espace vectoriel F de K de base $\{F_1, ..., F_d\}$ de dimension d sur $GF(q)$. Soit G la matrice generatrice associée à la matrice duale H.

Soit maintenant le mot reu $y = mG + e$, pour m le message et e une erreur de poids r et de support E enegndré par $\{E_1, ..., E_r\}$. On cherche à décoder y (en supposant $r \sim d << m, n$).

a) Montrer que pour décoder y il suffit de résoudre le problème:

$$H.e^t = H.y^t$$

.

b) On appelle $s(s_1, ..., s_{n-k})$ le syndrome $H.y^t$. Montrer que l'espace S engendré par les s_i (sur $GF(q)$ est au plus de dimension rd. En supposant que $rd << n-k$ et m, quelle est a priori la structure (très simple) de S ? (justifiez).

c) En suposant que S soit exactement l'espace produit $< E.F >$ de dimension rd. Quelle est la dimension de l'espace $S_i = F_i^{-1}.S$? Montrer que $E \subset S_i$.

d) En se basant sur c) expliquer comment retrouver E (avec une forte probabilité). En déduire un algorithme de décodage de e.

e) En supposant que tout se passe bien pour les divers probabilités rencontrées, quelle est la condition necessaire sur $n-k, r$ et d pour le decodage puisse marcher?

f) Quelle est la distance maximale à laquelle on peut decoder, en fonction de r,d et $n-k$. Dans le cas $d=2$, qu'obtient-on ? Comparer à la distance de decodage d'un code Gabidulin ? Quel est néanmoins à votre avis l'avantage des codes de Gabidulin sur les codes LRPC ?