Examen terminal de MAT239 (09/01/2015),

éléments de correction.

1 Partie I

Exercice 1

1. Partant de la matrice génératrice $G' = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$, on utilise l'algorithme de Gauss : on peut ajouter à la quatrième ligne la première, et échanger les deuxième et troisième ligne ; puis, on ajoute à la (nouvelle) troisième ligne la (nouvelle) quatrième, et enfin, on ajoute à la première ligne la (nouvelle) deuxième. D'où la matrice génératrice standard du code,
$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$.

2. La matrice de contrôle associée à G est $H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. Ses lignes sont non nulles et deux à deux distinctes, donc la distance minimale d du code est supérieure ou égale à 3. Mais la troisième ligne de G correspond à un mot du code à distance 3 de l'origine, donc $d = 3$. On en déduit que le code permet de détecter $d - 1 = 2$ erreurs, et d'en corriger $E(\frac{d-1}{2}) = 1$.

3. (a) Pour coder systématiquement les mots 1001 et 0011, on leur applique la matrice G. Ainsi, 1001 est codé en 1001001, et 0011, en 0011010.

   (b) En appliquant la matrice H à 0110110, on trouve le syndrôme 000. Donc 0110110 est un mot du code, et on le décode en 0110.

   Par contre, 1101111 a pour syndrôme 101 : ce n'est pas un mot du code. Pour le corriger, on établit une table de syndrômes, en calculant les syndrômes associés aux mots de poids croissant. Ainsi, 1000000 a pour syndrôme 110 ; 0100000 a pour syndrôme 011 ; 0010000 a pour syndrôme 101. C'est le syndrôme du mot à corriger, donc on s'arrête : à 1101111, on ajoute le vecteur d'erreurs 0010000 ; on obtient le mode du code 1111111, qu'on décode en 1111.

Exercice 2

On considère un code linéaire $C \subset \mathbb{F}_2^n$ donné par une application de codage (linéaire, injective) $\phi : \mathbb{F}_2^q \to \mathbb{F}_2^n$. On note d la distance minimale du code.

1. On définit l'application (linéaire) $\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ par

$$\forall w = w_1 \dots w_n \in \mathbb{F}_2^n, \quad \pi(w) = w_1 \dots w_{q-1} 0 \dots 0.$$

   (a) L'image $\pi(\mathbb{F}_2^n)$ de π est constituée des mots de $\mathbb{F}_2^n$ obtenus par concaténation de tous les mots de longueur $q-1$ avec un suffixe nul. Sa dimension est donc celle de $\mathbb{F}_2^{q-1}$, soit $q-1$.

   (b) Comme $\pi(C)$ est un sous-espace vectoriel de $\pi(\mathbb{F}_2^n)$, on en déduit que la dimension de $\pi(C)$ est inférieure ou égale à $q-1$. Ainsi, la dimension de $\pi(C)$ est strictement inférieure à celle de C, donc la restriction de π à C n'est pas injective, et il existe $w \in C \setminus \{0_n\}$ tel que $\pi(w) = 0_n$.

   (c) Ce mot w est un élément non nul de C, donc on a $p(w) \geq d$, par définition de d. De plus, comme $\pi(w) = 0$, les $q-1$ premiers bits de w sont nuls, d'où $p(w) \leq n - q + 1$. On a donc

$$d \leq p(w) \leq n - q + 1,$$

   qui implique $d \leq n - q + 1$, c'est-à-dire l'inégalité de Singleton,

$$d + q \leq n + 1.$$

2. (a) L'inégalité de Hamming pour le code C s'écrit, avec $t = E(\frac{d-1}{2})$,

$$\sum_{k=0}^{t} \binom{n}{k} \leq 2^{n-q}.$$

   (b) Si $n = 7$ et $q = 4$, comme $\binom{n}{0} = 1$ et $\binom{n}{1} = 7$, on a nécessairement $t \leq 1$, soit $\frac{d-1}{2} < 2$, et donc $d < 5$, ou $d \leq 4$.

   (c) Toujours avec $n = 7$ et $q = 4$, l'inégalité de Singleton donne aussi $d \leq 4$.

2 Partie II

Question de cours

Expliquer comment fonctionne le système de chiffrement RSA. Montrer pourquoi et grâce à quel théorème le déchiffrement est correct.

Problème

Le chiffrement de Goldwasser-Micali, proposé en 1982, est un des premiers cryptosystèmes à clé publique probabiliste et "probablement sûr". Son fonctionnement est le suivant :

— Génération de clés : Alice choisit deux grands nombres premiers p et q et calcule $n = pq$. Elle prend ensuite un élément $x \in (\mathbb{Z}/n\mathbb{Z})^\times$ qui n'est un résidu quadratique ni modulo p ni modulo q, c'est-à-dire tel que $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1$. La clé publique d'Alice est alors (x, n) et sa clé privée est (p, q).

— Chiffrement : pour envoyer un message, Bob le transcrit en une suite de bits $b_0 \dots b_N$ et chiffre chaque bit indépendamment (autrement dit, l'espace des messages clairs est $\{0, 1\}$). Pour chiffrer un bit $b \in \{0, 1\}$, Bob choisit au hasard un élément $y \in (\mathbb{Z}/n\mathbb{Z})^\times$ et transmet le chiffré $c = y^2 x^b \mod n$.

— Déchiffrement : pour trouver le message clair b' correspondant à un chiffré reçu $c \in (\mathbb{Z}/n\mathbb{Z})^\times$, Alice calcule le symbole de Legendre $\left(\frac{c}{p}\right)$; s'il vaut 1 alors $b' = 0$, sinon $b' = 1$.

1. $\left(\frac{5}{13}\right) = 5^{(13-1)/2} = 5^2 \times 5^4 = -1 \times 1 = 1 \mod 13$
   $\left(\frac{5}{17}\right) = 5^{(17-1)/2} = 5^2 \times 5^4 \times 5^8 = 8 \times 13 \times -1 = -1 \mod 17$
   Si x est choisi aléatoirement, alors il y a une chance sur deux que ce ne soit pas un carré modulo p (idem modulo q) donc une chance sur quatre que ce soit un choix convenable pour le chiffrement de Goldwasser-Michali.
2. Si $b = 0$ alors $c = y^2 \mod n$ donc $\left(\frac{c}{p}\right) = 1$ et $b' = 0$. Si $b = 1$ alors $c = y^2 x \mod n$ donc $\left(\frac{c}{p}\right) = \left(\frac{x}{p}\right) = -1$ et $b' = 1$. Dans tous les cas $b = b'$.
3. On calcule $\left(\frac{c}{p}\right) = \left(\frac{184}{13}\right) = \left(\frac{2}{13}\right) = 2^6 = 2^2 \times 2^4 = 4 \times 3 = -1 \mod 13$. Le message clair est donc $b = 1$.
4. Si l'on n'utilise pas d'aléa, le chiffrement devient déterministe ; en particulier le bit 0 (resp. le bit 1) est chiffré toujours de la même manière. Le message clair est alors soit égal au chiffré soit à la négation logique du chiffré. De la même façon, si Bob utilise plusieurs fois une même valeur de y pour communiquer avec Alice, le message chiffré avec cette valeur aura les mêmes propriétés que celui que l'on obtient sans utiliser d'aléa.
5. On suppose $c = c_1 c_2 z^2 \mod n$. Par conséquent, $\left(\frac{c}{p}\right) = \left(\frac{c_1}{p}\right)\left(\frac{c_2}{p}\right) = 1$ si et seulement si $b_1 = b_2$ donc si et seulement si $b_1 + b_2 = 0 \mod 2$.
6. Pour simplifier la génération des clefs, on se propose dans cette question de modifier le protocole en demandant juste que l'élément $x \in (\mathbb{Z}/n\mathbb{Z})^\times$ vérifie $\left(\frac{x}{p}\right) = -1$.
   (a) Un entier x tiré aléatoirement a une chance sur deux de satisfaire cette propriété.
   (b) C'est la même démonstration que pour la question 2.
   (c) Si $\left(\frac{x}{q}\right) = 1$, alors $\left(\frac{x}{n}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right) = -1$. Donc si Charlie voit c, il peut à partir du calcul de $\left(\frac{c}{n}\right)$ déduire facilement $\left(\frac{c}{p}\right)$ et donc déchiffrer c.
7. Problème de résiduosité quadratique : étant donnés x et $n = pq$, déterminer si x est un carré modulo n. Si Charlie sait résoudre ce problème, alors il sait déterminer si c est un carré modulo n. Comme les carrés modulo n sont des carrés modulo p (et modulo q), il peut déterminer la valeur de $\left(\frac{c}{p}\right)$ et donc déchiffrer c.
8. Réciproquement, si Charlie est capable de retrouver le message clair correspondant à n'importe quel chiffré c alors il sait calculer $\left(\frac{c}{p}\right)$, donc il sait calculer $\left(\frac{c}{q}\right)$. En effet tout message crypté c obtenu par le chiffrement de Goldwasser-Michali satisfait $\left(\frac{c}{n}\right) = 1$ donc $\left(\frac{c}{p}\right) = \left(\frac{c}{q}\right)$. Enfin comme c est un carré modulo n si et seulement si c'est un carré modulo p et modulo q, il sait résoudre le problème de résiduosité quadratique.
9. Les messages clairs sont en fait des bits donc de taille 1, alors que les chiffrés sont de la taille de n donc beaucoup plus gros (puisque n doit être choisi suffisamment gros pour être difficile à factoriser). Même si ce chiffrement paraît très "gourmand" puisqu'il nécessite un facteur d'expension en $\log_2(n)$ sur la taille des chiffrés, il est néanmoins intéressant pour ces propriétés d'homomorphie additive (cf question 5) : on peut en effet additionner des chiffrés et obtenir directement le chiffré de la somme sans connaître les messages clairs correspondants, ainsi on peut déléguer certains calculs à un serveur distant où seraient stockées les données chiffrées sans que celui-ci ait besoin d'une clef de déchiffrement.