Feuille TD 6 - Codage de canal - Entropie - Compression

Construction d'un code de Reed-Solomon adapté

1. $p_8 = 1 - 0.999^8 = 0.00797$

Rappel.

2.

a. *Comme on envoie des octets, on choisit* $\mathbb{F}_{256}$*, qui a* $q = 256 = 2^8$ *éléments, comme corps de base. Un code de Reed-Solomon impose alors de choisir* $n = q - 1 = 255$*.*

b. *Pour corriger au moins* $0.0079n = 2.03$ *erreurs, il faut avoir pouvoir corriger* 3 *erreurs, donc choisir un code de distance* $\delta \geq 2 * 3 + 1 = 7$*. Avec un code de Reed-Solomon, le polynôme générateur est de degré* $r = \delta - 1 = 6$*.*

c. *Le nombre maximal d'erreurs détectées est 6.*

d. *Soit* $g = \sum_{i=0}^{6} g_i X^i$ *le polynôme générateur. La matrice génératrice a 248 lignes et 255 colonnes. Elle s'écrit sous la forme*

$$\begin{bmatrix} g_0 & g_1 & \dots & g_6 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_6 & \dots & 0 \\ & & & \dots & & & \end{bmatrix}$$

e. *On choisit donc* $P = 1 + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^8$ *pour implémenter* $\mathbb{F}_{256} = \mathbb{F}_2[\alpha]/P$*. Soit* $X = \sum_{i=0}^{7} x_i \alpha^i$ *avec* $x_i \in \{0, 1\}$ *; X est représenté par l'octet* $(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$*.*
Soit $Y = \sum_{i=0}^{7} y_i \alpha^i$ *avec* $y_i \in \{0, 1\}$ *un autre élément du corps.*
$X + Y$ *est alors représenté par l'octet* $(x_0 + y_0, \dots x_7 + y_7)$*.*
Pour $X.Y$*, on calcule le produit de polynômes* $X.Y$ $\bmod P$*; les coefficients de ce polynôme permettent d'obtenir le codage de* XY *sous forme d'octet.*

f. *Comme P est primitif,* α *un élément générateur de* $\mathbb{F}_{256}^{\star}$ *(une racine primitive 255-ième de l'unité). Il suffit donc de choisir comme polynôme générateur* $g = \prod_{i=1..6}(X - \alpha^i)$ $\bmod P = X^6 + \sum_{i=0}^{5} g_i X^i$*, où chaque* g_i *est un élément de* $\mathbb{F}_{256}$ *représenté par un octet (cf question précédente).*

3. *La capacité du canal binaire symétrique de probabilité d'erreur* $q = 0.001$ *est* $C = 1 + q \log_2 q + (1 - q) \log_2 (1 - q) = 0.98859$*.*
Le rendement du code ne dépasse pas la capacité du canal (deuxième théorème de Shannon).

Sécurisation des communications

4. *Le problème est que chaque séquence de dé reçue par le casino puisse être identifiée et associée à un croupier. On utilise donc une signature par le croupier de la séquence lors de la transmission.*
On peut utiliser une signature avec une clef secrète (construite entre le casino et le croupier).

Mais, pour que les joueurs puissent verifier que la séquence émise est bien celle jouée, il est préférable d'utiliser un système à clef publique. Chaque croupier a une clef privée et une clef publique (allouées par le casino par exemple). Le croupier chiffre la séquence avec sa clef privée: ainsi le serveur du casino et les joueurs puvent bien verifier, avec la clef publique du croupier, qu'il a transmis les bonnes informations.
Pour éviter qu'un jet de dé chiffré par un vrai croupier soit bufferisé par un faux-croupier puis retransmis plus tard par le faux-croupier (qui veut se faire passer pour le vrai ;-), chaque émission est estampillée avec un numéro de lancer (incrémenté de 1 par le croupier) et une date éventuellement.
Enfin, pour minimiser le coût de chiffrement, le croupier peut limiter le chiffrement au résumé de la séquence (résumé MD5 ou SHA-1 par exemple).

En conclusion : Chaque croupier c a une clef publique. Il numérote chaque séquence qu'il envoie dans un en-tete (date + numéro de séquence); il obtient un message m clair.
Il calcule le résumé $r = SHA1(m)$ de m. Il chiffre $r' = D_c(r)$ avec sa clef privée et transmet sur le canal le message (m, r').
Les joueurs et le casino peuvent vérifier que le message a bien été émis par le croupier: il leur suffit de vérifier que $SHA1(m)$ et $E_c(r')$ sont égaux.

Codage des lancers

On suppose les dés du casino non pipés. On cherche à coder les séquences de tirages sur le canal binaire :

1. $H = \log_2(1/6) = 2.58$

2. *On code les 6 faces par 3 bits: I=000, II=001, III=010, IV=011, V=100, VI=101. Les messages 110 et 111 sont inutilisés.*

3. $l = 3$. *La longueur est toujours supérieure à l'entropie qui est une borne inférieure pour tout codage binaire.*

4. *L'entropie est une borne inférieure. On peut faire un peu mieux en faisant un codage de Huffman. On code alors V et VI sur 2 bits: V=10 et VI=11*
   Le codage est de longueur$=4.3/6 + 2.2/6 = 2.67$

5. *Il est optimal; mais on peut encore améliorer en codant plusieurs lancers successifs (extension de source). Ainsi si l'on code 5 jeters successifs, on a* $6^5 = 7776$ *possibilités. Un code par bloc de même taille peut être réalisé avec* $\log_2 7776 = 12.92$ *soit 13 bits; ce code est de longueur moyenne par lancer:* $l = 13/5 = 2.6$. *Un code de Huffman de ces 5 jeters ne peut être que meilleur (optimalité de l'arbre de Huffmann).*
   Asymptotquement, on tend vers $H = 2.58$. *Par exemple, avec 22 lancers, on a* $\log_2(6^{21}) = 56.8$*; donc 22 lancers sont codables par un code de meme longueur = 57 bits, pour une longueur moyenne de code par lancer de dé = 57/22=2.59*