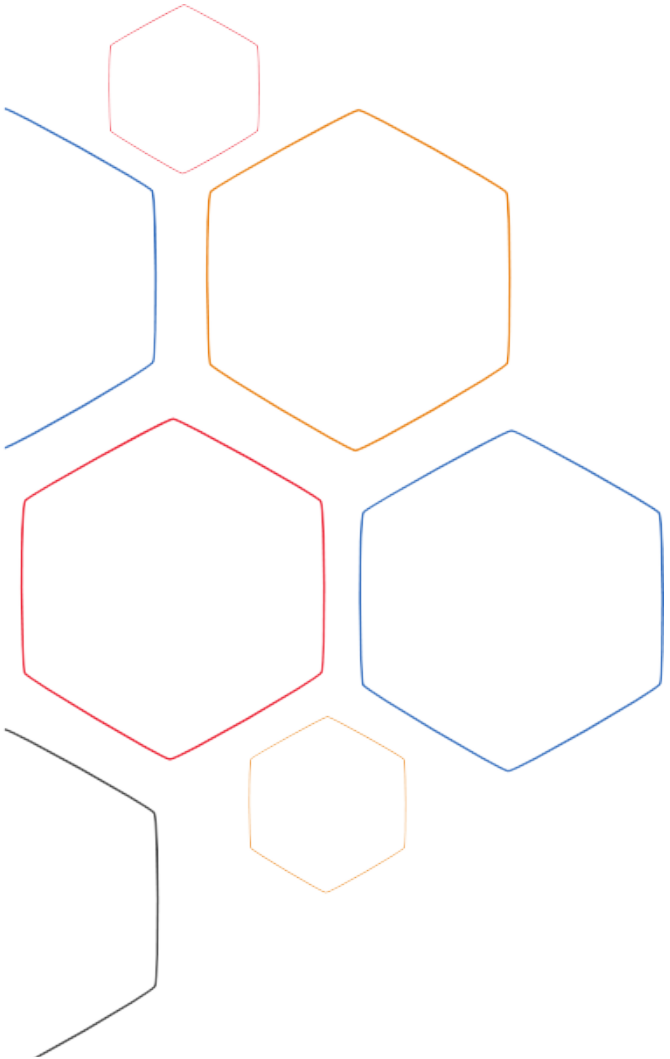
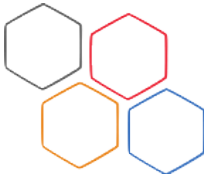


Cours de codes 1 sur 3

Eleonora Guerrini



A code: What for ?

- Recover faulty transmitted data
- Distributed Data Storage
- Conceive Fault Tolerant Algorithms

(ALGO DISTRIBUÉE)

A code: what is it ?

Definition

Un code correcteur est un ensemble de vecteurs (mots) et un couple d'algorithmes (Enc, Dec) qui gèrent la trasmission des mots sur un canal bruité.

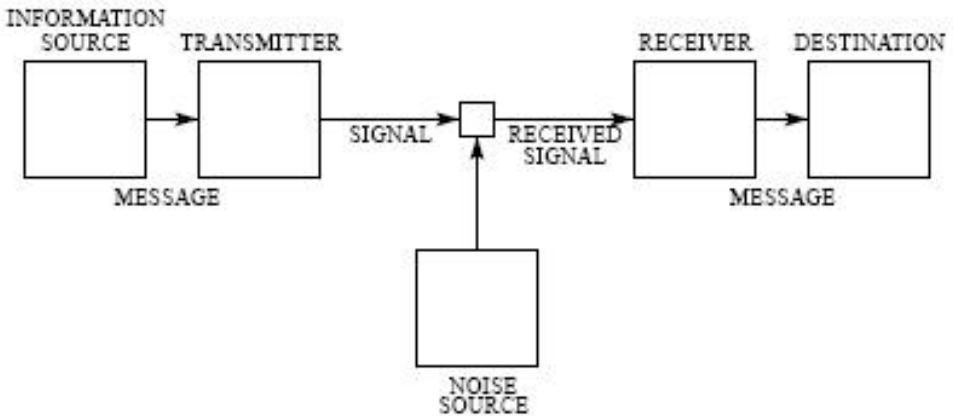
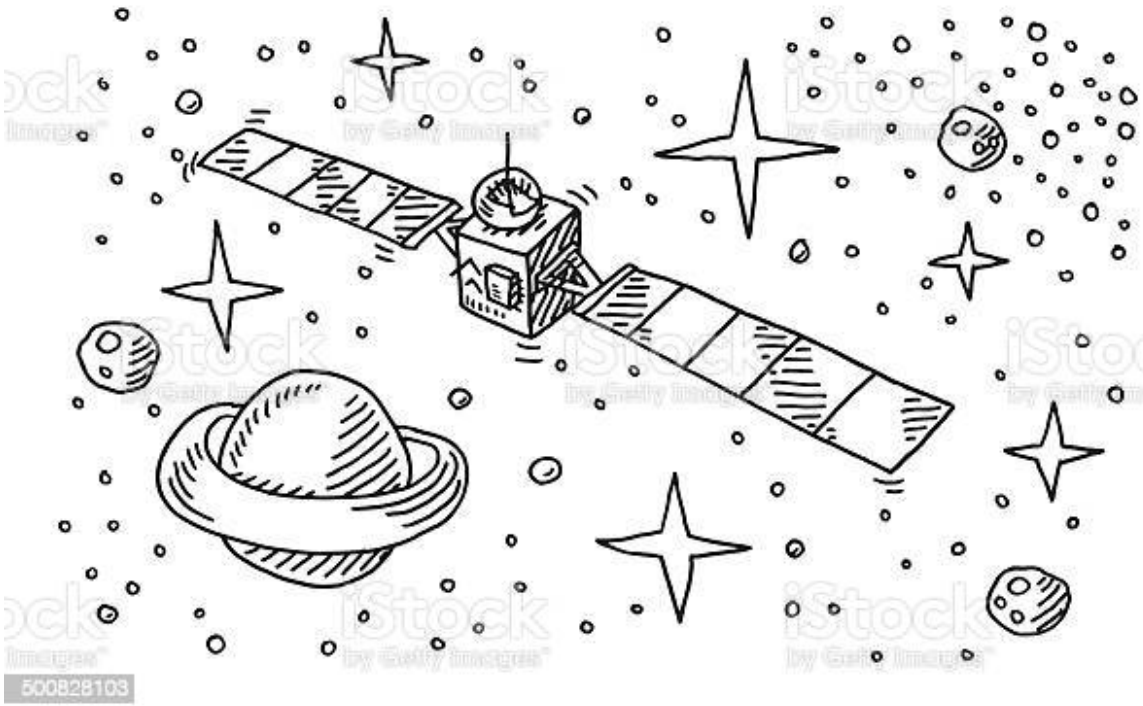


Fig. 1—Schematic diagram of a general communication system.

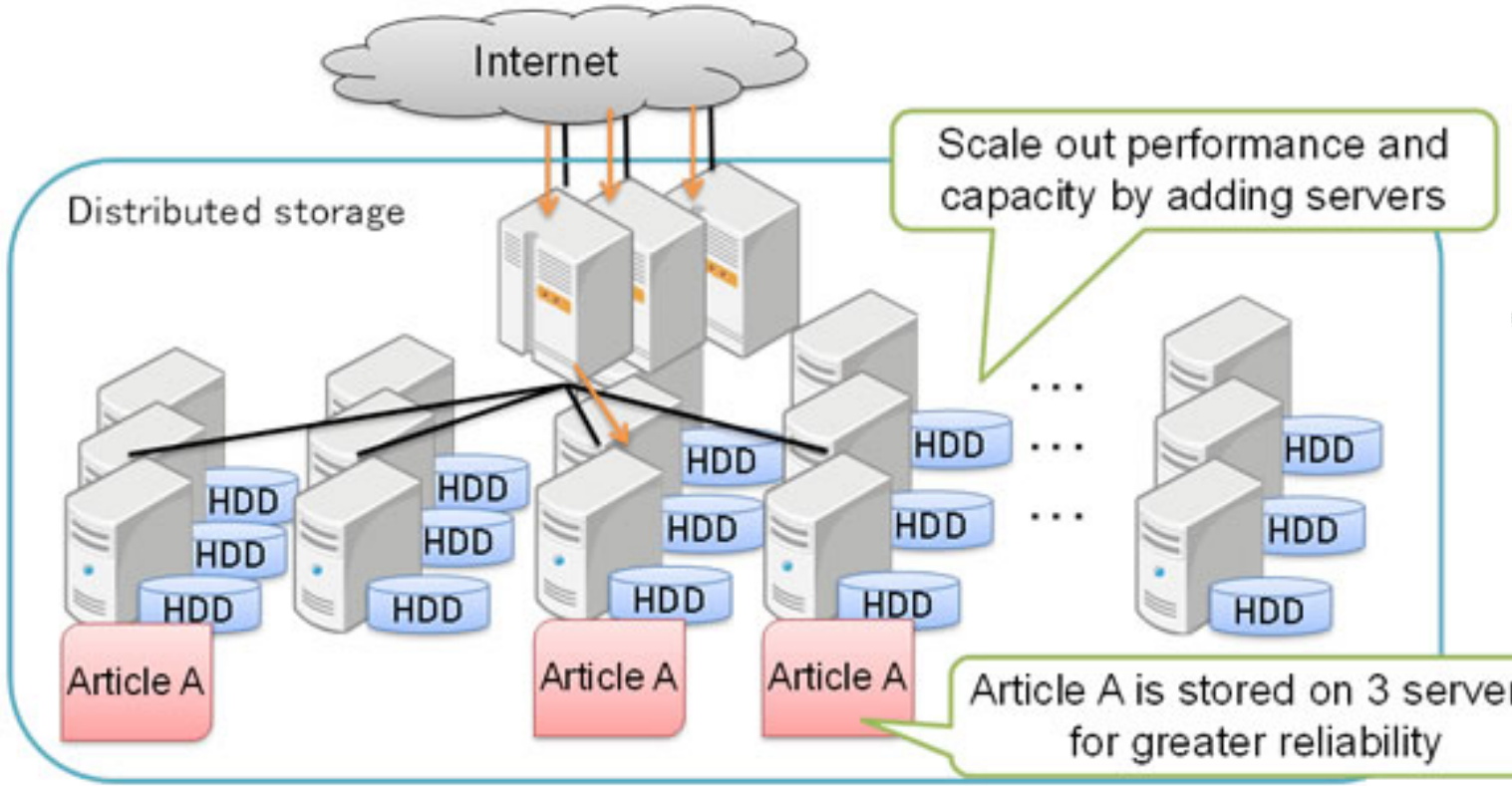
Codes correcteurs pour les transmissions



Dans les cas des Transmissions couteuses, il n'est pas possible de re-transmettre le message en cas de perturbation (ex. Satellites)

⊕ Besoin de corriger même si couteux

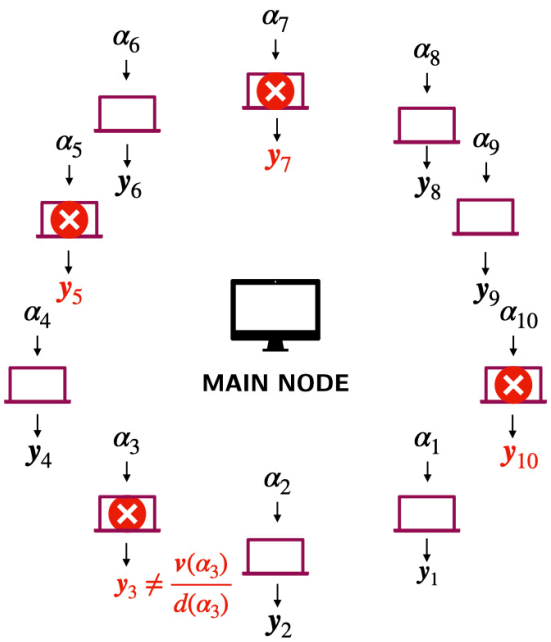
Codes correcteurs dans le stockage d'information



Impossible de re-demander le message endommagé

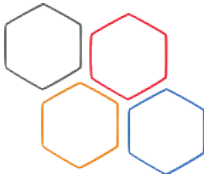
Si un serveur de stockage meurt, il faut avoir la possibilité de recuperer la donnée autrement

Codes correcteurs pour les algorithmes tolerants aux fautes



- cas algo distribuée
- Noeud defectueux
- Noeud malveillant

[ex : Produit matriciel en parallel]



Les questions fondamentales du cours 1

- Modèle: le code, le canal, le bruit
- Algorithmes: Encoder et decoder
- Parametres :Distance d'un code et Theorème de Shannon
- Exemples :code d'Hamming et decodage

Modèle de Shannon: le code, le canal, le bruit

Enc : Encodage $m \in (\mathbb{F}_2)^k \longrightarrow c \in (\mathbb{F}_2)^k$

Dec : Décodage Hamming 4 $\longrightarrow$ k'' $n = 7$

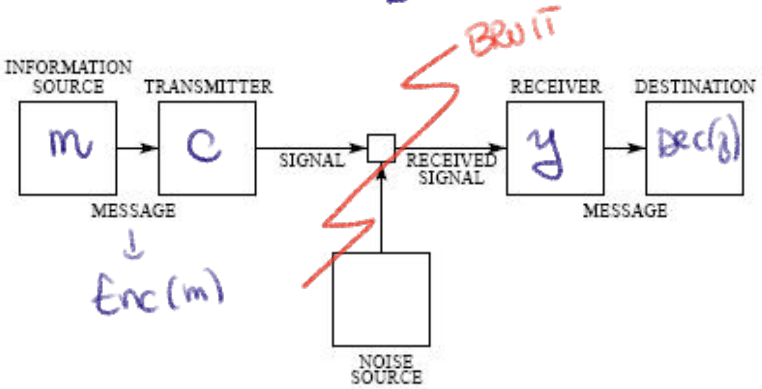


Fig. 1—Schematic diagram of a general communication system.

Bruit: Additif

y reçu $\exists \vec{e} \in (\mathbb{F}_2)^n$ tq

$y = c + e$

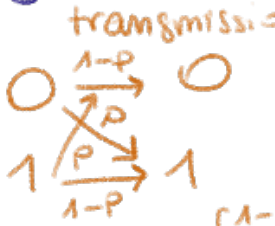
ex $c = (0,0,0,0,0,0,0)$

$y = (1,0,0,0,0,0,0)$

$e = (1,0,0,0,0,0,0)$

$Dec(y) = Enc(m)$

transmission



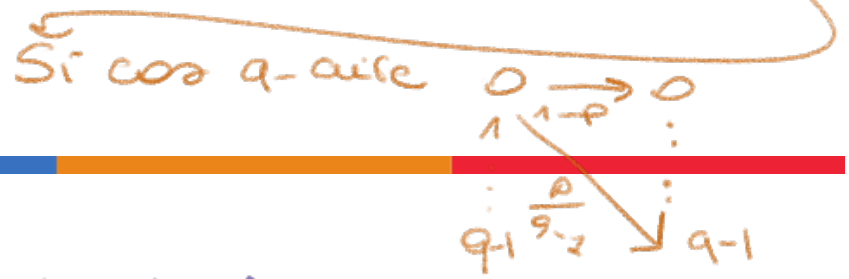
$\begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$

MATRICE de transition

CANAL :

- BSC (Binary Symmetric Channel)
  - Sans mémoire $P_r(y_i \neq c_i)$ est indep de $P_r(y_{i-1} \neq c_{i-1})$ autrement dit $Pr(y_i | c_i)$ est indep de $Pr(y_{i-1} \neq c_{i-1})$
  - P : Proba qu'il y a un flip de bit $0 < P < 1/2$ (sinon on inverse)
  - Symmetrique $P_r(c_i \text{ flip de } 0 \to 1) = P_r(c_i \text{ flip de } 1 \to 0)$

Encoding and Decoding



$Enc(m) : (\mathbb{F}_2)^k \longrightarrow (\mathbb{F}_2)^n$

$\{0,1\}^k \longrightarrow \{0,1\}^n$ $n > k$ redondance

- Injective
- $e \in (\mathbb{F}_2)^n$

↳ $Dec(m)$ et $Enc(m)$ on le choisit linéaire pour "efficacité"

$Enc(m)$ linéaire $\Rightarrow$ Algo d'algèbre linéaire pour encoder m

↳ C est un sous-esp. vect. de $(\mathbb{F}_2)^n$ de dim k

Theorème de Shannon

"GRAAL de la théorie de Codes correcteur

p loi de proba (binaire) $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$

Formulation "simple" du theore

- tout canal est toute CAPACITÉ (max inform. qu'on peut transmettre de façon fiable)

- $C_{sp} = 1 - H(p)$ (correction)
- R taux de transmission (redondance)
- $\frac{k}{n}$ rendement du code (rate)

dim esp mess → k ; longueur code → n

code Hamming [n=7, k=4 → 4/7 ? corrige 1 err.]

th Shannon

C_{sp} fixé $= 1 - H(p)$, p proba d'erreur du canal

$0 < p < 1/2$

$\varepsilon \leq 1/2 - p$

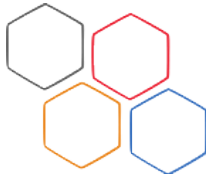
→ Pour $n \gg 0$

Pour envoyé et y reçu.

1. $\exists \delta > 0$ (Enc, Dec)

si $k \leq \lfloor (1 - H(p) + \varepsilon) \cdot n \rfloor$ alors

$- Pr(Dec(y_{reçu}) \neq 0) \leq 2^{-\delta n}$



2. Pour $k > \lceil (1-H(p)+\varepsilon)n \rceil$ pour tout (Enc, Dec) $\exists c \in \{0,1\}^n$ tq $P_n(Dec(y) \neq c) > 1/2$

Definition Code et Parametres

- Soit Σ un alphabet, $(\Sigma)^k$ l'espace des messages et k et n des entiers naturels tels que $k \leq n$.
- $C \subset (\Sigma)^n$:
- Linearité:
- On appelle k la dimension du code
- Rate:
- Distance (?) Hamming :

$Enc(m) : (\mathbb{F}_2)^k \longrightarrow (\mathbb{F}_2)^n$

• $Enc(0) = 0$ • $Enc(m_1 + m_2) = Enc(m_1) + Enc(m_2)$

$Enc((\mathbb{F}_2)^k)$ est esp. vect de $(\mathbb{F}_2)^n$

k/n

$x = (x_1 \dots x_n)$ $y = (y_1 \dots y_n)$

$d_H(x,y) = \sum (x_i + y_i)$

- $d_H(x,x) = 0$
- $d_H(x,y) = d_H(y,x)$
- $d_H(x,y) + d(y,z) \geq d_H(x,z)$

- Rapp. avec une base, G matrice base
- $Enc(m) = m \cdot G$

Hamming $k = 4$ $(\mathbb{F}_2)^4$

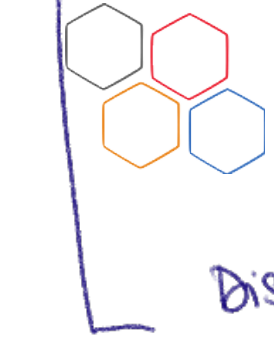
$mG = (1,0,1,1,0,1,0) \in (\mathbb{F}_2)^7$

mot de code de Hamming

$m = (1,0,1,1)$; $G = \begin{bmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&1 \\ 0&0&0&1&0&1&1 \end{bmatrix}$

$x = (0,0,1)$

$d_H(x,0) = 2$



y = (1,0,0)
z = (1,0,1)
(preuve vue)

• Correction Détection Effacements

Distance (d'Hamming) d'un code C

$$\min_{x_1 \neq x_2} \left(d(x_1, x_2) \mid x_1, x_2 \in C \right)$$

Quantifions les erreurs avec la distance et la définition du code

Detection et Correction: Modèle de hamming

$y = (y_1, \dots, y_n)$
$c = (c_1, \dots, c_n)$

Philosophie de décodage MLD
Maximun Likelihood Decoding (maximum de vraisemblance)

Problematique et Algorithmes: Rôle de la distance

Si y reçu, on veut $Dec(y) = c$

$P_r\{y \text{ reçu} \mid c \text{ envoyé}\} = \prod_{i=1}^{n} P_r\{y_i \text{ reçu} \mid c_i \text{ envoyé}\}$

$P_r(y \text{ reçu} \mid c \text{ envoyé})$ plus grande possible

si $y_i = c_i$: $n - d(y,c)$ fois
$y_i \neq x_i$: $d(y,c)$ fois

$n - d(y,c) = \#\{i \mid y_i = c_i\}$

$(1-p) \cdot p^{d(y,c)}$

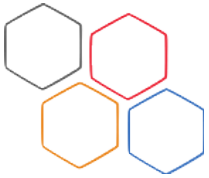
$c = 1\ 0\ 0$
$y = 0\ 1\ 0$

$p^{d(c,y)}$

Algorithme de decodage naif : MLD Algorithme

$(1-p)^n \cdot \left(\frac{p}{1-p}\right)^{d_H(y,c)}$

plus grand possible $\Rightarrow$ $d_H(y,c)$ plus petite possible



MLD decoder : bons et mauvais cotés

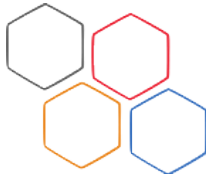
MLD Problème NP-hard

In : $y \in \mathbb{F}_2^n$, $\mathcal{C}$ code ($\mathcal{C} \in (\mathbb{F}_2)^n$ de dim k)

Out : $c \in \mathcal{C}$ tq $d(y,c) = \min\{d(y,c) \mid c \in \mathcal{C}\}$

(même si code lin)

On se restraint à un cas plus precis où on met une borne sur ce qu'on peut corriger



Modèles de Décodage

ex si $\bar{c} = (1001100)$
$e = (1100000)$
$y = (0101100)$

$y = \bar{c} + e$

e : vecteur erreur
$\bar{c}$: mot envoyé

BDD Decoder — Bounded Decoding Distance

In : y reçu, $\mathcal{C}$ code, t borne sur les erreurs qu'on veut (peut-être) corriger

Out : e tq $d_H(c, y) \leq t$

si $\mathcal{C}$ est Code Hamming
si $t = 2$
Le décodeur nous donnera un e tq $d_H(c, y) \leq 2$
mais pas forcément $e = \bar{c}$ renvoyé

BMD Decoder

t est le "bon"
càd que il existe tjrs un seul mot de code possible en Output

FIN COURS 1-

Code de repetition et Code de parité

- Code de repetition :

Hamming $R = \frac{4}{7}$ corr = 1 err

Rendement $R = \frac{k}{n} =$

Repet : $\frac{3}{9}$ corr = 1 err. sur 3

$m \in (\mathbb{F}_2)^k \xrightarrow{Enc(m)} c = (m, m, m)$, $c \in (\mathbb{F}_2)^{3k}$ ⇝ $\begin{cases} y \neq c \text{ si err} \\ y = c \text{ si } \vec{e} = 0 \end{cases}$

ex : (100) $\xrightarrow{Enc(m)}$ (100 100 100) ⇝ (111 100 100) ← y

Peut detecter 2 erreurs (3 pas possib. si une erreur par bloc qui se repete)

ici je peux detecter 2 erreurs ; et dans ce cas speatique aussi les corriger par vote de majorité mais on générale on ne peut pas corriger 2 err :

- Codes de parité :

$m \in (\mathbb{F}_2)^k \xrightarrow{Enc(m)} c = (m, \oplus m_i) \in (\mathbb{F}_2)^{k+1}$

ex : (100) ⟼ c = (1001)

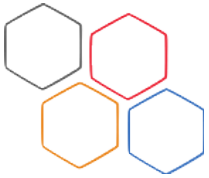
R : $\frac{k}{k+1}$ superbien

Detecte 1 erreur ? oui car si 1 erreur seulement la parité change.

y = (1011) mais corrige rien

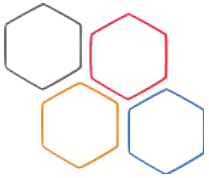
ex

(11010100)



Ex : Code de repetition et parity check

- dimension du code
- longueur du code
- rendement du code : ratio k/n



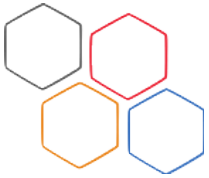
Décodeurs

Exo

- Code de repetition peut corriger 1 erreur
- Code de parité peut detecter un nombre impair d'erreurs (donc au pire coo 1)

3-repet (1) corrige 1 erreur, detecte 2 erreurs $R = \frac{k}{n} = \frac{1}{3}$

parité corrige 0, detecte 1 erreur $R = \frac{k}{k+1}$



Distance et Correction: Exo

Given a code C de longueur n et dimension k, les assertions suivantes sont équivalents

1. *C a distance minimale* $d \geq 2$,
2. *si* d *est impair,* C *peut corriger* $\left\lfloor \frac{(d-1)}{2} \right\rfloor$ *errors.*
3. C *peut detecter* $d-1$ *errors.*
4. C *peut corriger* $d-1$ *effacements*

(1) ⇒ (3) Si $y = c + \vec{e}$ où $d_H(c,y) = d$, comme $d(C) = d$ ie peut exister c_1 tq $c_1 = y$. Donc Detect(y) ⇒ Pas erreur (car y mot de C)

Si d-1 erreur c.a.d $d(y,c) \leq d-1$ donc y pas dans le code

(1) ⇐ (4) Si d-1 effacements ou plus ⇒ Unicité de c qui complete $y_i = \begin{cases} ? \\ c_i \end{cases}$ Si d effacements il peut y avoir 2 mots de code qui corresp.

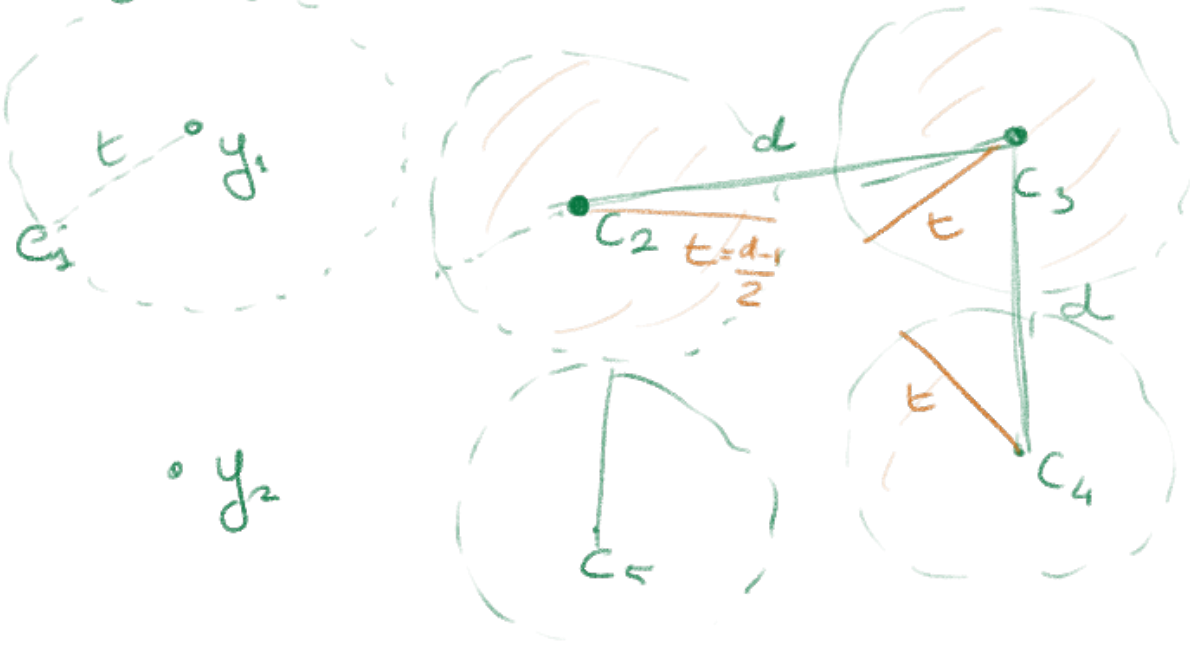
⇐ C peut corriger d-1 eff.

Preuve : ① ⇒ ② ① ⇒ ③ ① ⇒ ④ et après ② ⇒ ①

On fait que les bouts intéressants

car ① ⇒ ②.

② Dire que C corrige t erreurs : $\exists$ unique $c \in C$ tq $d_H(y,c) \leq t$ pour y reçu et t erreurs faites. Décodage déterministe



Comme $d(C) = d$ alors tout couple de mots de code c_1, c_2 est tq $d_H(c_1,c_2) \geq d$

Si $t = \frac{d-1}{2}$ si $t = \frac{d-1}{2}$

$\mathcal{B}_t(c_1) \cap \mathcal{B}_t(c_2) = \emptyset$

donc si y a t erreurs

$\Rightarrow$

$\exists \bar{c}$ tq $d_H(\bar{c}, y) \leq t \Rightarrow$

$\exists ! \bar{c}$ tq $y \in \mathcal{B}_t(\bar{c})$

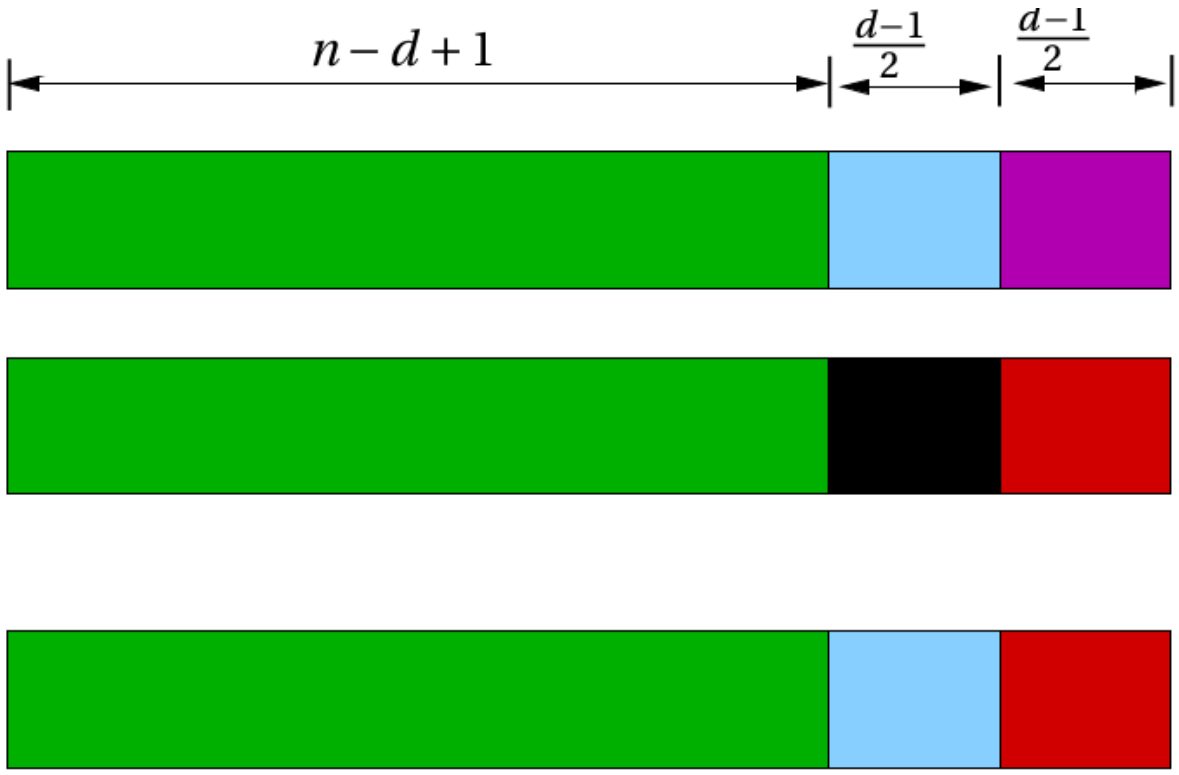
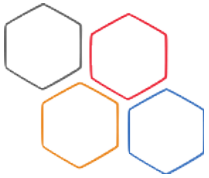


Figure 1.3: Bad example for unique decoding.



Qu'est-ce qu'on peut esperer comme rate et distance

(A) Rate (rendement) : $\frac{k}{n}$ plus grand possible

(B) t plus grand possible

(C) Corriger et detecter efficacement

(D) Calculer la distance efficacement

Qu'est-ce que c'est un bon code ?

(A) + (B) + (C) mais (A) vs (B) vs (C)

Hamming (A) et (C) mais pas du tout (B)

Code Repet (B) et (C) mais pas du tout (A)

On definit un "bon code" celui qui a les meilleurs parametres possible

⇓

Comment les paramètres peuvent se comporter.

Borne (d'Hamming)

$m \to \text{Enc}(m)$

$m \in (\mathbb{F}_2)^k \longrightarrow c \in (\mathbb{F}_2)^n$

Boule d'empilement de sphère

Pour un code $C \Rightarrow (n, k, d)$ binaire

ou $t = \frac{d-1}{2}$

$$2^k \sum_{i=0}^{t} \binom{n}{i} \leq 2^n$$

Si $\mathcal{B}_t(c) = \{ y \in \mathbb{F}_2^n \mid d_H(c, y) \leq t \}$, alors

$\mathcal{B}_t(c_1) \cap \mathcal{B}_t(c_2) = \emptyset \quad \forall c_1, c_2 \in C$

Donc $\bigcup_{c \in C} \mathcal{B}_t(c) = \bigsqcup_{c \in C} \mathcal{B}_t(c)$ (# c'est à dire toutes les intersection $\emptyset$)

$$\left| \bigsqcup_{c \in C} \mathcal{B}_t(c) \right| = \text{nb. boules} \times \underbrace{\text{Volume d'une boule}}_{\sum_{i=0}^{t} \binom{n}{i}} \leq \underbrace{2^n}_{\substack{\text{mots possible} \\ \text{dans l'espace} \\ (\mathbb{F}_2)^n}}$$

2^k ← nb mots de codes

$\cdot$ invariante par c choisi, donc ici $c = \vec{0}$

Pour Hamming $7+1=8$ on a égalité $n=7, k=4, t=1$

Les codes qui atteignent la borne d'empil-de sphères (c.à.d =) s'appellent PARFAITS (Hamming est donc un code parfait)

→ Conséquence : $\forall y$ reçu $\exists$ toujours un unique c tq $d_H(y,c)=1$ ou 0

(D) Comment calculer d'"efficacement" ? la distance d'un code.

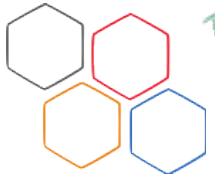
Lemme Si C linéaire alors $\boxed{d(C)=\min_{c\neq 0} w(c)}$

où $w(c) \overset{\text{def}}{=} |\{i \mid c_i \neq 0\}|$. on appelle $w(c)$ le Poids de c

ex : $(10011) \rightarrow w(c)=3$

⌂ → Preuve ? Il faut montrer que $\min_{\substack{c_1\neq c_2 \\ c_1,c_2\in C}} d_H(c_1,c_2) = \min_{\substack{c\neq 0 \\ c\in C}} w(c)$

$c_1(100)$; $c_2=(110)$ $w(c_2)=|\{0,1,0\}|$



Pour c_1, c_2 $d(c_1, c_2) = w(c_1 - c_2)$ car C est linéaire donc $c_1 - c_2 \in C$

Matrice generatrice et de parité

$$\min_{c_1 \neq c_2} \{ d(c_1, c_2) \} = \min_{\substack{c \in C \\ c \neq 0}} \{ w(c) \} + \forall w(\bar{c}), c \in C$$

$d(\bar{c}, 0) = w(\bar{c})$.

Avantage de la linéarité :

- $d(C) = \min_{\substack{c \neq 0 \\ c \in C}} w(c)$
- C s.esp.vect de $(\mathbb{F}_q)^n$ de dim k

$q \in 2$

- $Enc(m) : (\mathbb{F}_2)^k \longrightarrow (\mathbb{F}_2)^n$ linéaire

$\Downarrow$

$\exists$ matrice $G \in (\mathbb{F}_2)^{n \times k}$ tq

$c = Enc(m) = G \cdot m^T$

$m = (m_1, \dots, m_k)$

$$\underbrace{\begin{bmatrix} & \\ & \end{bmatrix}}_{n \times k} \begin{Vmatrix} m^T \\ \\ \end{Vmatrix}_{k \times 1} = c \in (\mathbb{F}_2)^n$$

Décodage d'un code d'Hamming

Code d'HAMMING $n=7,\ k=4,\ q=2$

$m=(x_1,x_2,x_3,x_4) \xrightarrow{Enc(m)} (x_1,x_2,x_3,x_4,x_2+x_3+x_4,\ x_1+x_3+x_4,\ x_1+x_2+x_4)$

C_H = code d'Hamming (C_H est linéaire)

$0 \in C_H$

si $c_1, c_2 \in C_H \rightarrow c_1+c_2 \in C_H$,

$Enc(m_1)+Enc(m_2) = Enc(m_1+m_2)$ ok.

G matrice génératrice

$$G=\begin{bmatrix}1&0&0&0&1&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{bmatrix}$$

Encodage

H matrice de parité (matrice gén. du noyau)

$$\begin{bmatrix}0&0&0&1&1&1&1\\0&1&1&0&0&1&1\\1&0&1&0&1&0&1\end{bmatrix}$$

Propn de H

$H\vec{y}^T = \vec{0} \iff y \in C$

$H\vec{y}^T \in (\mathbb{F}_q)^n$

Detect In: $y \in (\mathbb{F}_2)^n$ Out: Bool (vrai si erreurs) → if $H\vec{y}^T \neq 0$ Bool := true

Correction d'erreur

Exemple de Décodage

Est-ce que l'on peut aussi décoder avec H ?

Si $\vec{y}$ reçu et $\vec{e}$ erreur $\vec{y} = c + \vec{e}$ où $c \in C$

$$Hy = H(c+e) = \underbrace{Hc}_{=0} + \underbrace{He}_{\text{synchome}}$$

vecteur de $(\mathbb{F}_2)^{n-k}$ qui contient l'info. des erreurs –

Ex avec ℓ_H :

$c = (1\,0\,0\,0\,0\,1\,1)$ $\quad Hc^T = 0 \rightsquigarrow$ pas d'erreurs.

$$\begin{bmatrix} 0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1 \end{bmatrix} \begin{bmatrix} 1\\0\\0\\0\\0\\1\\1 \end{bmatrix} = \vec{0}$$

$c = (1\,0\,1\,1\,0\,1\,0) \not\in C \rightsquigarrow y = (0\,0\,1\,1\,0\,1\,0)$

Bruit

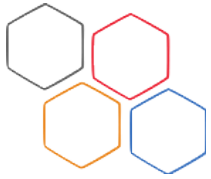
$$Hy^T = \begin{bmatrix} 0\\0\\1 \end{bmatrix} \rightsquigarrow \text{écriture binaire de la pos. d'err.}$$

$(0\,0\,1) \rightarrow$ erreur à l'indice 1

synchome

$e = (1\,0\,1\,1\,0\,1\,0)$

– x –



Idée
Si on prend une matrice "quelconque" comme G,
est-ce que j'obtiens un bon code ?

→ ① Optimale par une borne
② Decodage efficace NON

[ex Hamming ① + ②]

① C'est oui grace à la borne Gilbert-Varshamov

Construction Code : Fix n, d, $\mathbb{F}_q$
Output $\mathcal{C}_G \in (\mathbb{F}_q)^n$ tq $d(\mathcal{C}_G) \geq d$

Greedy $\mathcal{C}_G$

1. $\mathcal{C}_G \leftarrow \emptyset$
2. while $\exists\, v \in (\mathbb{F}_q)^n$ tq $d(v,c) \geq d \;\forall c \in \mathcal{C}_G$

$\mathcal{C}_G = \mathcal{C}_G \cup \{v\}$

3. Output $\mathcal{C}_G$

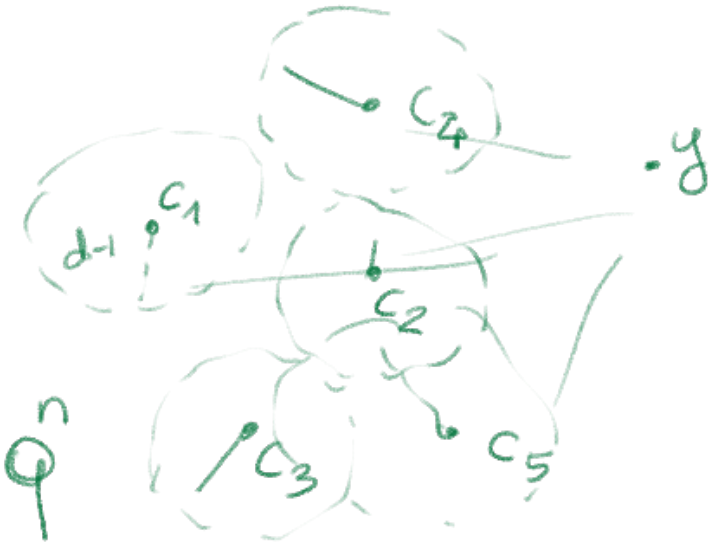
GreedyConst $(n,d) \to \mathcal{C}_G$

① Est-ce que $d(\mathcal{C}_G) = d$?

② Est-ce que le nombre de mots dans $\mathcal{C}_G$ est le plus grand possible (si $\mathcal{C}_1$ code quelconque tq a param (n,d), alors nb de mots dans $\mathcal{C}_1 \leq$ nb mots dans $\mathcal{C}_G$

→ Trivial (par construction)

→ L'algo s'arrête quand $\nexists v \in (\mathbb{F}_q)^n, v \notin \mathcal{C}_G$ tq $d(v,c) \geq d \ \forall c \in \mathcal{C}_G$



② Après l'algo $\left| \bigcup_{c \in \mathcal{C}_G} B(c, d-1) \right| = q^n$

car si $\exists y \in (\mathbb{F}_q)^n$ tq $y \notin \bigcup_{c \in \mathcal{C}_G} B(c,d-1) \Rightarrow d(y,c) \geq d \ \forall c \in \mathcal{C}_G \Rightarrow$ y pouvait être choisi dans Greedy

$$\sum_{c \in \mathcal{C}_G} |B(c,d-1)| \geq \left| \bigcup_{c \in \mathcal{C}_G} B(c,d-1) \right| = q^n$$

Borne Gilbert-Varshamov

$$q^k \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i}$$

$q^k \cdot Vol_q(d-1,n) \geq q^n \Rightarrow$ $q^k \geq \frac{q^n}{Vol_q(d-1,n)}$

Attention: différence entre borne Hamming et Borne Gilbert-Varsh.

q^k : nb. de mots du code

$q^k \leq$ "$\ldots Vol_q(\frac{d-1}{2}, n)\ldots$" $q^k \geq \ldots Vol_q(d-1,n)$

Interprétation Asymptotique –

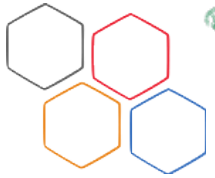
- $\frac{k}{n} = R \quad \delta = \frac{d}{n} \quad 1 = \frac{n}{n} \rightsquigarrow d = \delta n$

$$q^k \geq \frac{q^n}{Vol_q(\delta n, n)} \rightsquigarrow q^k \geq \frac{q^n}{q^{nH_q(\delta)}} \rightsquigarrow R \geq 1 - H_q(\delta)$$

Pour n, q et $\delta \in (0,1)$ $\exists$ famille de codes tq $R \geq 1 - H_q(\delta)$

(?) Est-ce que la borne de Gilbert-Varshamov peut être atteinte ?

théorème : Un code linéaire aléatoire atteint la Borne de Gilbert-Varshamov



Preuve : Soit $w(c)$ = poids du mot c.

Il nous faut montrer qu'il existe une matrice $k \times n$, où le rang est $k = (1 - H_q(\delta) - \varepsilon)n$, t.q

$\forall m \in (\mathbb{F}_q)^{*} \rightarrow w(mG) \geq d$. (Code a distance d)

$\Pr[w(mG) < d] = \dfrac{\text{Vol}_q(d-1, n)}{q^k \text{ x tout mots } \in (\mathbb{F}_q)^n}$ ← mots de poids $\leq d-1$

$d = \delta n$

$$\Pr[\exists m \neq 0 \text{ t.q } w(mG) < d] \leq \frac{q^{nH_q(\delta)}}{q^k} \leq q^{-k} \cdot q^{-\varepsilon n} \times (q^k - 1)$$

car $k \leq (1 - H_q(\delta) - \varepsilon)n$

$$\leq q^{-\varepsilon \cdot n}$$

Il nous reste à prouver que G a rang plein.

Dire que G a rang plein équivaut à montrer que $\nexists\ m \neq 0$ tq $mG = 0$. Dans notre cas, il suffit de prouver que

$Pr[\exists m \neq 0 \mid w(mG) = 0] \approx 0$. On réutilise la preuve précédente →

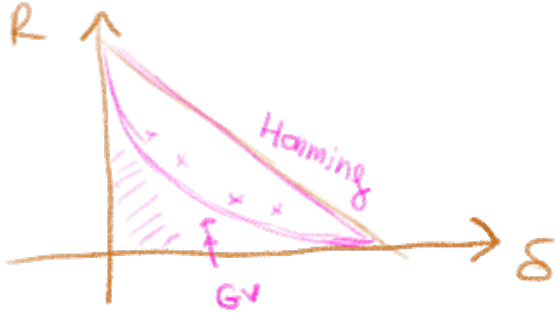
$$q^{-\varepsilon n} \geq Pr[\exists m \neq 0 \text{ tq } w(mG) < d] \geq Pr[\exists m \neq 0 \text{ tq } w(mG) = 0]$$

de façon effective.

- Est-ce qu'on peut construire des codes qui atteignent (optimaux) pour la borne G.V. ? → Problème ouvert

$\delta = d/n$

$R = k/n$



Codes de Reed-Solomon

(A) + optimaux pour une borne (borne de Singleton)
↳ MDS (maximum distance separable)

(B) + Encodage et décodage efficace (très !)

(C) + il existe une infinité de paramètres tq un code de RS existe

(A) Borne de Singleton

Pour un code à paramètres (n, k, d) [Version code linéaire]

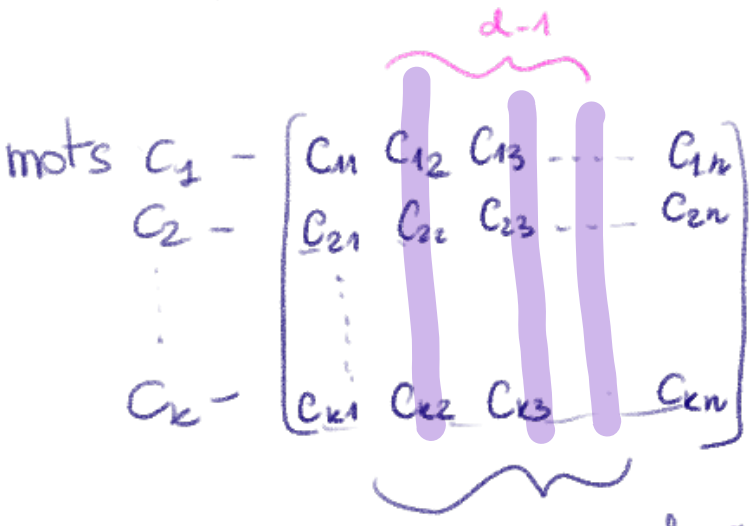
$$d \leq n - k + 1$$

version Asymptotique

$$\delta \leq 1 - R + \frac{d(\cdot)}{N}$$

nb. d'erreurs à corriger $t = \frac{d-1}{2} \leq \frac{n-k}{2}$

Preuve : C linéaire $\rightsquigarrow$ G matrice génératrice $\in (\mathbb{F}_q)^{k \times n}$ $(\forall c \in C,\ c = mG)$



On poinçonner le code (cad on enlève des coordonnées) dans $d-1$ coordonnées

[Quand on poinçonne on enlève les lignes qui sont égales.]

je supprime $d-1$ coordonnées. Il reste dans la matrice $n-(d-1)$ $= n-d+1$ colonnes.

C' le code poinçonné, $C' = C_{\substack{n'=n-d+1 \\ d' \geq d \\ k'=k}}$

dans C', on a pas de perte de mots (dimension) car $d(c_i, c_j) \geq d \Rightarrow d(c'_i, c'_j) \geq 1$,

Donc $k \leq n-d+1$. ⊠ $\Rightarrow$ $d \leq n-k+1$.

Si le code n'est pas linéaire, même preuve mais avec la liste de tous les mots et non pas la base de la matrice.

- Si un code est tel que $d = n-k+1$ il est dit Maximum Distance Separable } MDS

Si $q=2$ (codes binaires) → seulements MDS bandes ($n-k=0$ répétition)

↕ ? Conjecture MDS

$q-1=n$ ok

Si $q \geq n$ (ok MDS, on peut construire les RS)

On retient : Pour construire un MDS, aujourd'hui, il faut un $\mathbb{F}_q$ avec $q \geqslant n-1$.

(B) Révision sur les polynomes sur $\mathbb{F}_q[X]$

① $\mathbb{F}_q[X]$: Polynômes à coeff en $\mathbb{F}_q$

$\mathbb{F}_q[X]_{<k}$: Polynômes à coeff en $\mathbb{F}_q$ et degrée borné par $k-1$

↳ Esp-vectoriel à dim k . : Base $\{1, X, X^2, \ldots, X^{k-1}\}$

② Déf Racine d'un polynôme

Soit $f \in \mathbb{F}_q[X]$, $\deg(f) = D$

$$f = \sum_{i=0}^{D} a_i X^i$$

On dit que 2 est racine de f si

- ① $f(2) = 0$
- ② $(X-2) \mid f(X)$ ssi $f(X) = 0 \mod (X-2)$
- ③ $f(X) = (X-2) \cdot f_1(X)$ où $\deg(f_1(X)) < D$

② ⇔ ③

On fait la div. euclidienne

① $f(X) = (X-2) \cdot P_1 + r(X)$ avec $\deg r < 1$

$f(2) = (2-2) \cdot P_1(2) + r \Rightarrow f(2) = r$

$f(2) = 0 = 0$ donc $r = 0$ ⇒ ②

③

Degree Mantra

⊙ $f(x) \in \mathbb{F}_q[x]$, $f \neq 0$, $\deg(f) = D \Rightarrow f$ a au plus D racines

Preuve si 2 racine $\rightarrow$ $f = (x-2) \cdot f_1$; $\deg(f_1) = D-1$

Récursivement si $D = 1 \rightarrow f = (x-2)$

$f = ax + b$ $\Rightarrow \exists a^{-1}$ $\Rightarrow x + b \cdot a^{-1}$ $c = -2$

Si f n'a pas pas de racine en $\mathbb{F}_q$ $k \Rightarrow$ ok

Conséquence :

Si $f(x) \in \mathbb{F}_q[x]$, $\deg(f) \leq D$ et f a plus de $D+1$ racines

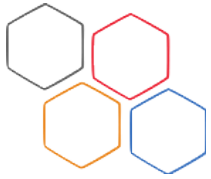
$\Rightarrow f \equiv 0$

Def codes RS $\quad [q \geq n]$

In : $\mathbb{F}_q[x]_{<k}$, $A = \{a_1, \ldots, a_n\}$ distincts $a_i \in \mathbb{F}_q$

Message : $p(x) \in \mathbb{F}_q[x]_{<k}$ $\xrightarrow{\text{Enc}}$ $(p(a_1), p(a_2), \ldots, p(a_n))$

$\text{Enc}(m) = (\text{Eval}_{p(a_1)}, \ldots, \text{Eval}_{p(a_n)})$



ex : $k=3$, $q=7$

Messages : $1, 2, \dots 6$ les constantes ; $x+1, 2x+3, \dots$ les pol. de deg 1 ; $x^2+3x+1, \dots$ les pol. de deg 2

$n = 7$ → $A = \{0, 1, \dots, 6\}$

$Enc(x^2+3x) = (0, 4, 3, \dots)$ (0 = Eval(0), 4 = Eval(1), 3 = Eval(2))

Enc : $P_{deg<k} \to$ vect des coeff (m) $\to mG = (Eval_p(\alpha_1) \dots)$

G : matrice Vandermonde $(\alpha_1 \dots \alpha_n)$

① Les codes de RS sont MDS :

C code de RS, alors $\boxed{d = n-k+1}$

C a dimension k, on veut montrer que distance $\geq n-k+1$. Si $m_1, m_2 \in \mathbb{F}_q[X]_{<k}$

$d(Enc(m_1), Enc(m_2)) \geq n-k+1$

$c_1 = (m_1(\alpha_1), m_1(\alpha_2), \dots, m_1(\alpha_n))$

$c_2 = (m_2(\alpha_1), m_2(\alpha_2), \dots, m_2(\alpha_n))$

$\#\{i \mid m_1(\alpha_i) = m_2(\alpha_i)\} \leq k-1$

→ (pour $m_1 - m_2$ a au plus $k-1$ racines)

$d(c_1, c_2) = \#\{i \mid m_1(\alpha_i) \neq m_2(\alpha_i)\} = n - \#\{i \mid m_1(\alpha_i) = m_2(\alpha_i)\}$

$d \geq n - k + 1$.

On peut corriger t erreurs, où : $\frac{d-1}{2} = \frac{n-k}{2} = t$

In: Si pas d'erreurs
$x_1 \dots x_n \in \mathbb{F}$
$(y_1, \dots, y_n)$ vecteur reçu

m ? ⇝ Eval-Interp : Out $L \in \mathbb{F}_q[X]$
$L(x_i) = y_i$
ce Polyn. est le polynome de Lagrange $Log(x)$

Eval-Interp

$\forall i,\ L_i(x_k) = \begin{cases} 1 & \text{si } i = k \\ 0 & \text{si } i \neq k \end{cases}$ $\quad L_i = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(X - x_j)}{(x_i - x_j)}$

ex $L_1(x_1) = \prod_{\substack{j=1 \\ j \neq 1}}^{n} \frac{(x_1 - x_j)}{(x_1 - x_j)} = 1$

$L_1(x_2) = \prod_{\substack{j=1 \\ j \neq 1}}^{n} \frac{(x_2 - x_j)}{(x_1 - x_j)} = 0$

$Log_{(x_i, y_i)}(X) = \sum_{i=1}^{n} y_i L_i(X)$

$Log(x_1) = \sum_{i=1}^{n} y_i \cdot L_i(x_1) = \underbrace{L_1(x_1)}_{1} \cdot y_1 + \underbrace{\sum_{i=2} \dots}_{0}$

$\exists$ (à une const près) 1 unique poly de deg $\leq n-1$ qui interpole n points_

$\Rightarrow$ $Log_{(x_i, y_i)}(x)$ Poly. Lagrange_

En cas d'erreurs : Equation clé

Pour $c = Eval(m(\alpha_1), \_ m(\alpha_n))$ envoyé, on reçoit

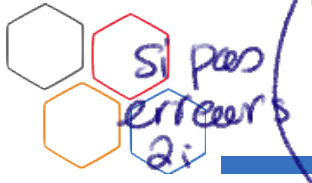
$y = (y_1, \_\_\_, y_n)$ tq $\begin{cases} y_i = m(\alpha_i) \\ y_i \in \mathbb{F}_q \text{ si erreur. } y \neq m(\alpha_i) \end{cases}$

LOCATEUR D'ERREURS

$y_i = m(\alpha_i)$ vrai si pas err_ je définis $\Lambda = \prod_{i \in E} (x - \alpha_i)$

$E = \{i \mid y_i \neq m_i(\alpha_i)\}$

$\Lambda y_i = \Lambda m$ vrai si evaluée sur $(\alpha_1 \ldots \alpha_n)$



$\Lambda(Q_i)\, y_i = \Lambda(Q_i)\, m(Q_i)$ ok

Si erreurs en Q_i : $\underbrace{\Lambda(Q_i)}_{=0} y_i = \underbrace{\Lambda(Q_i)}_{=0} m(Q_i)$

$$\left\{ \begin{array}{l} \Lambda(Q_i)\, y_i = \Lambda(Q_i)\, m(Q_i) \\ \vdots \end{array} \right. \quad \text{n-equations}$$

($\Lambda(Q_i)$: inconnue ; $\Lambda(Q_i)\, m(Q_i)$: inconnue, inconnues)

⚠ pas bricolé

je cherche φ, ψ tq :

$$\varphi\, y_i = \psi \quad \text{mod } (x - Q_i)$$

c.à.d $\varphi(Q_i)\, y_i = \psi(Q_i)$ [avec $\deg(\varphi) \leq \deg(\Lambda) = t$
avec $\deg(\psi) \leq t + k - 1$

inconnues $\Rightarrow$ les coeff de ψ et de φ

$2t + k - 1$

equ n

si $n \geq 2t + k - 1$ sol_unique $(RS)n = 2t+k-1$

et comme je sais que $\varphi = \Lambda$, $\psi = \Lambda m$ est une solution

$\Longrightarrow$ dans le cas d'unicité forcément $\varphi = \Lambda$ et $\psi = \Lambda m$

$m = \psi / \varphi$