Codes de Reed-Solomon

- **Théorème**: Soient F un corps de cardinal $\geq n$ et $\alpha_1,\ldots,\alpha_n$ n éléments $\neq$ de F.
  Pour $1 \leq k \leq n$, $RS(n,k) = \{ [Q(\alpha_1),\ldots,Q(\alpha_n)] : Q(X) \in F[X], \deg(Q) \leq k\}$
  est un code de Reed-Solomon (n,k) sur F.
  Ce code est de dimension k+1 et de distance n-k (maximale):

  Code (n, k+1, n-k) sur F

- Problème des codes de Reed Solomon: $|V| \geq n$, donc $q=|V|$ grand
  - Pour un bloc de b bits: donc choisir $V=F(2^m)$ et n tels que : $(m \geq \log_2 n)$ et $(m.n \geq b)$.
    - Exemple: canal binaire, correction d'un taux d'erreur de 2% :
      - » b=100 => code (20, 16, 5) sur F(32) Rendement = 16/20 = 80%
      - » b=1000 => code (125, 85, 41) sur F(256) Rendement = 85/125 = 68%
      - » b=10000 => code(1000, 600, 401) sur F(1024) Rendement = 600/1000 = 60%

Décodage unique de Reed-Solomon par Berlekamp-Welch

- **Problème décodage avec $e \leq (d-1)/2 = (n-k)/2$ erreurs et $t=n-e \geq k+e+1$ valeurs correctes.**
  - Entrée: $y_1, \ldots, y_n \in F^n$ vérifiant $\exists P \in F[X]$ de degré k tel que $\#\{i / y_i = P(\alpha_i)\} \geq t$.
  - Sortie : le polynôme P unique tel que $\#\{i / y_i = P(\alpha_i)\} \geq t$.
    - *Preuve unicité*: Si P_1 et P_2 sont solutions: P_1-P_2 s'annulle en au moins n-2e $\geq$k+1valeurs. Or $\deg(P_1-P_2) \leq k$, d'où $P_1=P_2$.

- **Principe** :Soit $T=\{ i / y_i = P(\alpha_i) = L(\alpha_i) \}$; et $E(X)= \Pi_{i \notin T} (X-\alpha_i)$ le polynôme localisateur d'erreurs.
  - Soit L(X) le polynôme d'interpolation aux n points (α_i, y_i); degré(L(X)) $\leq$ n-1.
    Degré(E(X))$\leq$e et pour i=1..n : $E(\alpha_i).L(\alpha_i) = P(\alpha_i).E(\alpha_i)$.
    Posons $N(X)=P(X).E(X)$ qui est de degré k+e: on a donc pour i=1..n : $E(\alpha_i).L(\alpha_i) = N(\alpha_i)$
    ie un système de n équations à k+2.e+1$\leq$n inconnues (k coefs de E et k+e+1 de N);
    comme P est unique, ce système admet une solution unique.
- **Algorithme**
  - Étape 1: calculer le polynôme d'interpolation L(X) et former le système linéaire: $y_i.E(\alpha_i) - N(\alpha_i) = 0$ i=1..n
  - Étape 2: résoudre le système linéaire: on obtient les coefficients de E(X) et N(X).
  - Étape 3: retourner le polynôme N(X) / P(X).

- **Remarque**: il existe des algorithmes plus rapides en $O(n.\log^2 n)$.
  [Berlekamp-Massey, ou calcul de pgcd tronqué en utilisant un algorithme de pgcd rapide].

Décodage de Reed-Solomon par Berlekamp-Massey

- Codage = évaluation d'un polynôme en n = k+r points (FFT)
  - Redondance de r symboles
  - Mot de code $[y_0, \ldots, x_{k-1}, y_k, \ldots, y_{n-1}] = \Omega\,[x_0, \ldots, x_{k-1}, 0, \ldots, 0]$
- Distance « maximale » (au sens classique)
  - Avec 2r symboles de redondance, on corrige r erreurs
- Correction: on reçoit $z=[z_0, \ldots, z_{k-1}, z_k, \ldots, z_{n-1}] = y+e$ avec e de poids $\leq r$
  - $\Omega^{-1}z = \Omega^{-1}y + \Omega^{-1}e = x + \hat{e} = [x_0+\hat{e}_0, \ldots, x_{k-1}+\hat{e}_{k-1}, \mathbf{\hat{e}_k, \ldots, \hat{e}_{n-1}}]$
  - les *r* symboles de $\mathbf{\hat{e}_k, \ldots, \hat{e}_{n-1}}$ engendrent linéairement ê donc de calculer e

Codes concaténés

- Code concaténé: construit sur V avec q=|V| petit à partir de 2 codes:
  - « inner code » : C_{in} (n_{in}, k_{in}, d_{in}) avec Q mots sur le petit alphabet (V_{in} de taille q)
  - « outer code » : C_{out} (n_{out}, k_{out}, d_{out}) sur un grand alphabet (V_{out} de taille Q)

Code concaténé série

- **Codage:**

Mot source

k_{out} symboles de V_{out}, soit $k_{out}.k_{in}$ symboles de V_{in}

(codage C_{out})

⇓

n_{out} symboles de V_{out}, soit $n_{out}.k_{in}$ symboles de V_{in}

(codage C_{in})

⇓

Mot de code

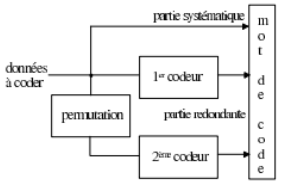
$n_{out}.n_{in}$ symboles de V_{in}

Donc: Code($n_{out}.n_{in}$, $k_{out}.k_{in}$, $d_{out}.d_{in}$) sur V_{in} , de rendement $R_{out}.R_{in}$.

Autres concaténations possibles

(avec 2 codes ou plus)

- Concaténation parallèle (turbo-code)



Rendement = $R_1.R_2 / (1-(1-R_1)(1-R_2)) > R_1R_2$

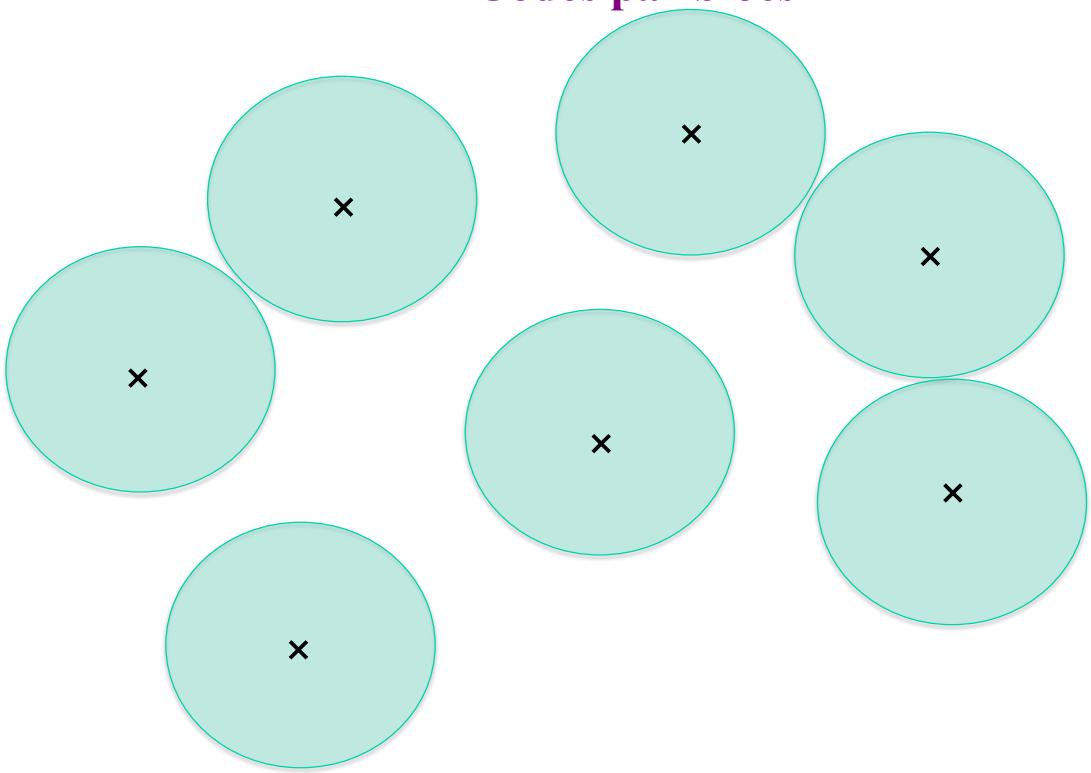
- Code produit, code croisé
- Concaténation mixte série et parallèle

Chapitre. Décodage en liste (List Decoding)

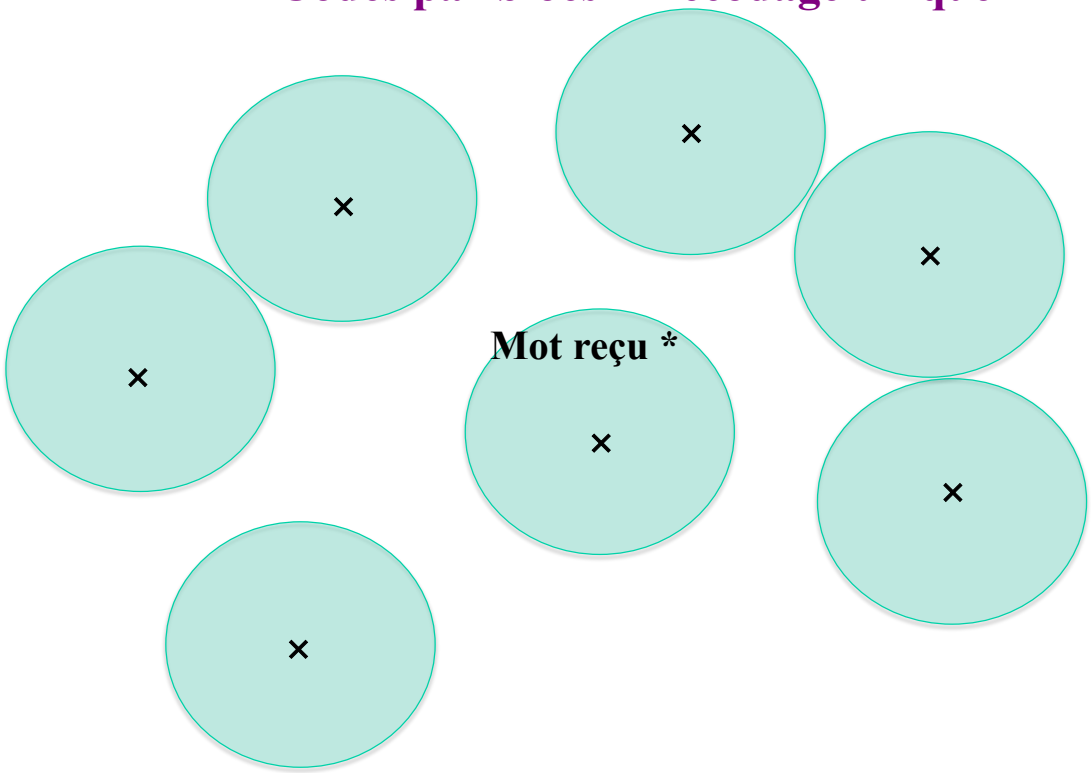
- Définitions: Décodage en liste et recouvrement en liste
- Bornes de Johnson
- Existence de codes linéaires décodables en liste
- Un algorithme de décodage en liste

- Référence:
  - Venkatesan Guruswami, “Algorithmic results in list decoding”, in Fundations and trends in theoretical computer science vol. 2, n.2, 2006, pages 107-195.

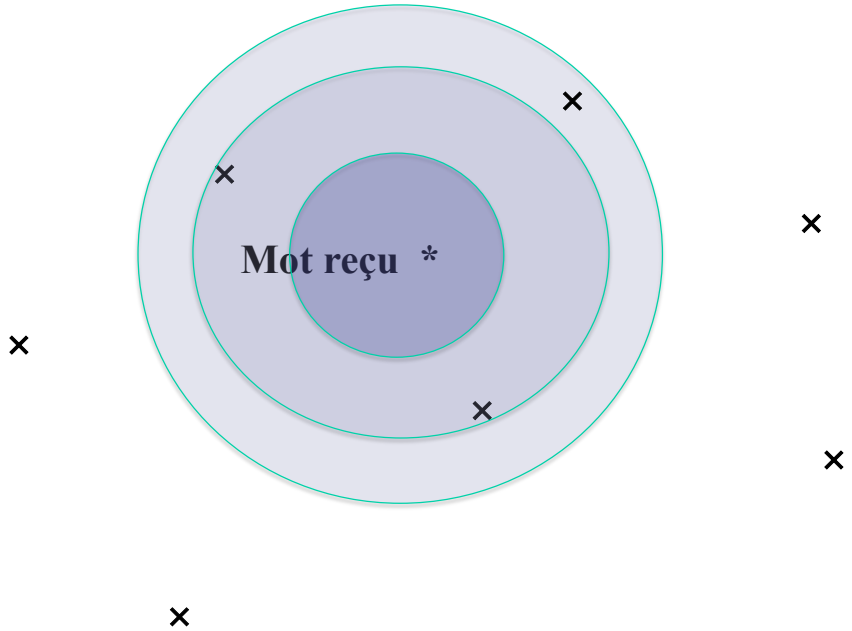
Codes par blocs



Codes par blocs – Décodage unique



Codes par blocs – décodage par liste



Décodage en liste (List-decoding)

- **Problème du décodage en liste** d'un code C (n,k) sur V jusqu'à une fraction d'erreur p (ou rayon p.n) :
  - entrée : $y \in V^n$
  - sortie : la liste des mots de C à distance $\leq$ p.n de x, ie $\{ c \in C : d_H(c,y) \leq p.n \}$

- **Définition**: **(p,L) – décodabilité en liste :**
  pour $0<p<1$ et un entier L, un code $C \subset V^n$ est dit **(p,L)-décodable**
  (ou décodable jusqu'à une faction p d'erreurs avec une liste de taille L)
  ssi pour tout $y \in V^n$: $\text{Card}\{ c \in C : d_H(c,y) \leq p.n \} \leq L$.

  - La famille de codes $(C_n)_{n \in N}$ est (p,λ)-décodable ssi $\forall n$: C_n est $(p, \lambda(n))$-décodable avec $\lambda: N^* \rightarrow N^*$.
    - Cas particulier: si λ est constante égale à L, la famille est dite (p-L)-décodable.

Recouvrement en liste (List-recovering)

- **Définition**: (list-recovering): pour $0<p<1$ et un entier $u \leq L$,
  un code $C \subset V^n$ est dit ***(p, u, L)*-recouvrable** ssi:
  pour toute séquence $S_1, S_2, \ldots, S_n$ de n sous-ensembles de V avec $|S_i| \leq u$,
  $\text{Card}\{ c=[c_1, \ldots, c_n] \in C$ tels que « $\#\{i: c_i \in S_i\} \geq (1-p).n$ » $\} \leq L$.
  - La valeur « *u* » est appelée *taille de la liste d'entrée*.

- Remarque 1: (p, L)-décodable == (p, 1, L)-recouvrable.

- Remarque 2: (0,u,L)-recouvrement de liste dans le cas non bruité (p=0):
  - Entrée: u possibilités pour c_1, u possibilités pour c_2, …, u possibilités pour c_n
  - Sortie: les au plus L mots de code correspondant à ces possibilités.

- **Théorème**: Le code concaténé-série d'un *outer code* (p_1, u, L)-recouvrable et d'un *inner code* (p_2, u)-décodable est $(p_1.p_2, L)$-décodable en liste.

Borne de Johnson et existence de codes (p,L) décodables en liste

- **Notations:** C code (n, k, d) sur V avec |V|=q
  - $\mathbf{B_q(c, r)}$ [ou **B(c, r)**]: boule de Hamming de V^n centre $c \in V^n$ et de rayon r).
  - **B**=$C \cap B_q(c, r)$ et **(B 2)** = tous les sous-ensembles de cardinal 2 de B.
- **Quel est le nombre de mots de code dans $\mathbf{B_q(x, r)}$?**
  - Si $r \leq (d-1)/2$: $|B| \leq 1$, donc le rayon de décodage pour une liste de taille 1 est (d-1)/2.
  - **Cas général**: *Borne de Johnson*: soient $a=E_{x \in B}[\, d_H(c,x)]$ et $b=E_{\{x,y\} \in (B\ 2)}[\, d_H(x,y)]$, et soit $D= (1- a.q.(n.q - n)^{-1})^2 - (1 - b.q.(n.q - n)^{-1})$ alors si D>0:
    $$|B| \leq (1 - b.q.(n.q - n)^{-1}) / D$$
- **Corollaire** :
  - soient $\delta \in]0,1[$ tel que $d = n.(1 - q^{-1})(1-\delta)$ et $\gamma \in]0,1[$ et $r = n.(1 - q^{-1})(1-\gamma)$: si $\gamma^2 > \delta$, alors pour tout $c \in V^n$: $|\, B_q(c, r) \cap C\,| \leq (1 - \delta) / (\gamma^2 - \delta)$
  - *Preuve*: soient δ' et γ' tels que $E_{x \in B}[\, d_H(c,x)]=n.(1 - q^{-1})(1-\gamma')$ et $E_{\{x,y\} \in (B\ 2)}[\, d_H(x,y)]= n.(1 - q^{-1})(1-\delta')$. La borne de Johnson s'écrit: $|B| \leq \delta' (\gamma'^2 - \delta')$. Or $\delta' \leq \delta < \gamma^2 \leq \gamma'^2$, d'où le corollaire.
- **Cas particulier:** si q grand et si $r \leq n-(n^2-nd+1)^{1/2}$: $|\, B_q(c, r) \cap C\,| \leq n^2$.
  Application: si taux erreur $p < 1- (1-d)^{1/2}$: le nbre de mots de code à distance $\leq p.n$ est $n^{O(1)}$.
- Donc, si p tend vers 1, on peut faire un décodage par liste (mais pas un décodage unique).

Existence de codes (p,L) décodables en liste

- Soit taux d'erreur et $\mathbf{H(p) = H_q(p)} = p.\log_q(q-1) - p.\log_q p - (1-p).\log_q(1-p)$ entropie q-aire
- Remarque: $n.H_q(p)$ est proche de $\log_q |B_q(0, p.n\ )|$:
  $$\log_q(\, n.H_q(p) - o(n)\ ) \leq \log_q(|B_q(0, p.n\ )|) \leq n.H_q(p).$$

- **Théorème**: Pour q, $L \geq 2$ et $\forall p \in]0,1-q^{-1}[$, $\exists$ famille de codes q-aires qui sont (p, L) décodables en liste et de rendement $R \geq 1- H_q(p) - 1/L$.
  - *Preuve*: basée sur un codage aléatoire de longueur n grande. On construit un code aléatoire de longueur n à partir de M mots tirés aléatoirement (non nécessairement distincts): on fixe M pour que le code soit (p,L)-décodable avec une bonne probabilité.
  - Prob{L+1 mots fixés $\in B_q(0, p.n\ )$ } = $(|B_q(0, p.n\ )| / q^n)^{L+1} \leq q^{-n.(L+1).(1-H(p))}$.
    Donc Prob(échec)=Prob{L+1 mots de code $\in$ même boule de rayon p.n} $\leq C(M, L+1).q^n.q^{-n.(L+1).(1-H(p))}$.
    Soit $\rho = 1 - H_q(p) - (L+1)^{-1}$; en posant $M=q^{\rho n}$ on a : Prob(échec)$< 1/(L+1)! < 1/3$.
    Parmi les M mots, il y a au moins M/2 mots distincts avec prob > ½ (coupon collector).
    Le code formé par ces $M/2 \geq q^{\rho n}/2 \geq q^{n.(1-H(p)-1/L)}$ mots est (p-L)-decodable avec probabibilité $\geq 2/3$.
    Donc il existe nécessairement un code (p,L)-décodable avec M/2, donc un rendement $R \geq 1- H_q(p) - 1/L$.

- **Intérêt**: preuve non constructive, mais le rendement optimal pour le décodage en liste jusqu'à un rayon p est $1-H_q(p)$, la capacité du canal q-aire.

Existence de codes *linéaires* (p,L) décodables en liste

- **Théorème**: Pour q puissance d'un entier premier, $\forall p \in]0,1-q^{-1}[$ et $\forall L \geq 2$, $\exists$ famille de codes q-aires **linéaires** qui sont (p, L) décodables en liste et de rendement $R \geq 1- H_q(p) - 1/\log_q(L+1)$.
  - *Preuve*: non constructive, en tirant au hasard un code linéaire (ie des mots linéairement dépendants).

- **Implication pour de grands alphabets :**
  On a $H_q(p) = p.\log_q(q-1) - p.\log_q p - (1-p).\log_q(1-p)$
  $= p - p.\log_q(q/(q-1)) + H_2(p)/\log_2 q.$

  Donc, pour q grand, et pour L grand, il existe des codes linéaires décodables en liste de rendement proche de $1 - H_q(p) \simeq 1 - p$.
  - Autrement dit: décodage en liste avec rendement R jusqu'à une fraction d'erreur (1-R)
  - De plus: la capacité de canal est $1 - H_q(p) \geq 1 - p - 1//\log_2 q$: donc on peut, avec q suffisamment grand, obtenir un code de rendement 1-p-ε pour décoder en liste jusqu'à une fraction p d'erreurs.

Décodage de codes de Reed-Solomon

- Code RS(n , k+1, distance = n-k) sur F, supposé de cardinal $q \geq n$.
  - évaluation d'un polynôme de degré k aux n abscisses: $\alpha_1, \ldots, \alpha_n$ de F

- **Problème jouet 1 : Décodage de 2 mots mélangés avec $n \geq 2k+1$:**
  - Entrée: 2 mots $a_1, \ldots, a_n$ et $b_1, \ldots, b_n \in F^n$ vérifiant $\exists P_1, P_2 \in F[X]$ de degré k tels que
    $\forall i$: $\{ a_i = P_1(\alpha_i)$ et $b_i = P_2(\alpha_i) \}$ ou $\{ b_i = P_1(\alpha_i)$ et $a_i = P_2(\alpha_i) \}$.
  - Sortie : les coefficients de P_1 et P_2.
- Algorithme:
  - On a: $P_1(\alpha_i) + P_2(\alpha_i) = a_i + b_i$ et $P_1(\alpha_i).P_2(\alpha_i) = a_i.b_i$
  - Par interpolation, on peut donc calculer $S=P_1+P_2$ et $P=P_1.P_2$ (de degrés k et 2k).
    Soit alors $Q(X, Y) = (Y - P_1(X)).(Y - P_2(X)) = Y^2 - S(X).Y + P(X)$.
  - La factorisation de Q en facteurs irréductibles (Q de degré 2 en Y) donne P_1 et P_2.

Décodage de codes de Reed-Solomon

- **Problème jouet 2 : Décodage d'un mot mélangé avec $n \geq 6k+1$:**
  - Entrée: un mot $y_1, \ldots, y_n \in F^n$ vérifiant $\exists P_1, P_2 \in F[X]$ de degré k tels que
    $\forall i : \; y_i = P_1(\alpha_i)$ ou $y_i = P_2(\alpha_i)$
    et de plus pour j= 1, 2: $\#\{i / y_i = P_j(\alpha_i)\} \geq n/3$ [ie au moins 2k valeurs pour P_1 et P_2]
  - Sortie : les coefficients de P_1 et P_2.
    - Remarque:
      - Soit $Q(X, Y) = (Y - P_1(X)).(Y - P_2(X))$: on a alors $\forall i : \; Q(\alpha_i, y_i) = 0$.
      - Mais ces équations ne définissent pas Q de manière unique !
- **Algorithme**: On cherche Q de la forme $Q(X,Y) = Y^2 - (\Sigma_{j=0}^{k} q_{1,j}.X^j).Y + (\Sigma_{j=0}^{2k} q_{2,j}.X^j)$: les 6k+1 équations $Q(\alpha_i, y_i) = 0$ donnent un système linéaire en les 3k+2 inconnues $q_{\_,\_}$. Soit Q(X,Y) la solution (unique) de ce système.
  - *Lemme*: $Q(X,P_1(X)) = 0$ autrement dit $(Y - P_1(X))$ est un facteur de Q [de même P_2].
    - Preuve: $R(X)= Q(X,P_1(X))$: R est de degré $2k < n/3$.
      Soit $I=\{i \text{ tq } y_i = P_1(\alpha_i)\}$: $\forall i \in I \; R(\alpha_i)=0$; donc R a au moins n/3 racines. D'où R=0.
  - Donc la solution Q vérifie $Q(X, Y) = (Y - P_1(X)).(Y - P_2(X))$: la factorisation de Q par rapport à Y donne alors P_1 et P_2.

Décodage en liste de codes de Reed-Solomon

- **Problème reconstruction polynomiale avec ≤e erreurs, t=n-e:**
  - Entrée: $y_1, \ldots, y_n \in F^n$ vérifiant $\exists P \in F[X]$ de degré k tel que $\#\{i / y_i = P(\alpha_i)\} \geq t$.
  - Sortie : la liste de tous les polynômes P tels que que $\#\{i / y_i = P(\alpha_i)\} \geq t$.
- Généralisation lemme précédent: soit $Q(X,Y) = \Sigma_{j,j} q_{i,j}.X^i.Y^j$.
  - Déf: Degré (1,k)-pondéré : $wdeg_{1,k}(Q(X, Y)) = \text{Max}\{ i+kj \text{ tq } q_{i,j} \neq 0\}$
  - **Lemme**: Soit $Q(X,Y)\neq 0$ tel que $wdeg_{1,k}(Q) < t$ et $\forall i : \; Q(\alpha_i, y_i) = 0$. Soit P(X) de degré k tel que $\#\{i / y_i = P(\alpha_i)\} \geq t$. Alors Q(X, P(X))=0, ie (Y-P(X)) facteur de Q(X,Y).
    - *Preuve*: $R(X)=Q(X, P(X))$ est de degré $\text{Max}\{i+kj \text{ tq } q_{i,j} \neq 0\} < t$ et s'annulle en ≥ t valeurs α_i; d'où R=0.
  - Donc: il suffit de montrer l'existence de Q, puis de résoudre un système linéaire.
- **Théorème**: Si $(D+2)(D+1)>2.k.n$ alors $\exists Q(X,Y)\neq 0$ tel que $wdeg_{1,k}(Q)\leq D$ et $\forall i \; Q(\alpha_i, y_i)=0$.
  - *Preuve*: Comme $wdeg_{1,k}(Q)\leq D$, on a $Q(X,Y) = = \Sigma_{j=0}^{D/k} \Sigma_{i=0}^{D-jk} q_{i,j}.X^i.Y^j$. (car degré ≤ D/k en y et ≤D en X).
    Les n équations $Q(\alpha_i, y_i)=0$ donnent un système linéaire à n équations et m inconnues, les $q_{i,j}$.
    Si m>n, alors on a (au moins) une solution non nulle. Or $m = \Sigma_{j=0}^{D/k} \Sigma_{i=0}^{D-jk} 1 = \Sigma_{j=0}^{D/k} (D - jk + 1) \geq (D+1)(D+2) / (2.k)$ qed.
- **Algorithme**: Soit t tel que $t^2 > 2.k.n$ le nombre de valeurs correctes et soit $D=[(2.k.n)^{1/2}]$.
  - Étape 1: résoudre le système linéaire pour calculer $Q(X,Y)\neq 0$ 0 tel que $wdeg_{1,k}(Q)\leq D$ et $\forall i \; Q(\alpha_i, y_i)=0$.
  - Étape 2: (factorisation / calcul de racines en Y de Q(X,Y) pour trouver tous les polynômes P(X) de degré k tels que Q(X, P(X))=0; pour chacun vérifier que $\#\{i / y_i = P(\alpha_i)\} \geq t$.
  - Coût: système lin=$O((n+D)^3)$. De plus, le nombre de facteurs P(X) possibles de Q(X,Y) est $\leq D/k \leq \sqrt{(2n/k)}$.

Conclusion Décodage en liste de codes de Reed-Solomon

- Si le nombre de valeurs correctes t vérifie $t > \sqrt{(2.n.k)}$, l'algorithme est de coût polynomial et retourne une liste de candidats de taille $\leq \sqrt{(2.n/k)}$.
  - Donc, avec $k = \Omega(n)$, une liste de taille $O(1)$

- Le décodage en liste d'un code Reed-Solomon(n, k+1, n-k) de rendement $R = (k+1)/n$:
  - Peut être fait et garanti jusqu'à une fraction d'erreurs $= 1 - \sqrt{(2.R)}$ et retourne moins de $\sqrt{(2/R)}$ candidats.

  - En particulier: avec un rendement faible, on peut corriger jusqu'à presque 100% d'erreurs.
    - Utile en cryptography : exemple pour construire des "prédicats à sens unique" àpartir de fonctions à sens unique, utilisés dans les générateurs pseudo-aléatoires cryptographiquement non-prédictibles (CSPRNG).

- Extension au décodage unique: si #erreurs $\leq (d-1)/2 < (n-k)/2$, alors on a $t > (n+k)/2$; on interpole $Q(X,Y) \neq 0$ en le cherchant sous la forme :
  $Q(X,Y) = A(X).Y + B(X)$ avec degré(A) $< (n-k-1)/2$ et degré(B) $< (n+k-1)/2$.
  Le polynôme corrigé est alors $-B(X)/A(X)$.
  - *Preuve*: Soit f (X) la solution unique et e(X) le polynôme localisateur d'erreur.
    Le choix $A(X)=e(X)$ et $B(X)=-f(X)/e(X)$ donne un polynôme Q(X,Y) qui convient et $wdeg_{1,k}(Q) \leq (n-k-1)/2 + k < t$.

V Exemples de codes utilisés dans des matériels

CDROM / DVDROM (1/3)

- Codes par bloc cycliques
- Exemple : CD Audio :
  - Données = octets
  - K=trame = 24 octets codés sur 32 octets sur le CD
  - CIRC = Code de Reed-Solomon entrelacé croisé
    - Code (32, 24) = entrelacement de 2 codes cycliques
  - Base:
    - code de Reed-Solomon (255, 251, 5)
    - Code C1(28,24, 5)
    - Code C2(32,28,5)
  - Codes en détail
  - Description de l'entrelacement

CDROM : code CIRC (2/3)

Le code C_1

Il s'agit d'un code (28,24,5) sur le corps à 256 éléments.

Si x est un mot de 24 octets le mot de code de C_1 qui lui correspond est le mot de 28 octets égal à $(x, x\,R_1{}^t)$, où la matrice de dimension (4,24) R_1 est définie par

$$R_1 = \begin{bmatrix} a^6 & a^{192} & a^{142} & a^{159} & a^{99} & a^{88} & a^{104} & a^{144} & a^{55} & a^{180} & a^{174} & a^{101} & a^{111} & a^{118} & a^{169} & a^{107} & a^{132} & a^{25} & a^{167} & a^{239} & a^{168} & a^{188} & a^{11} \\ a^{45} & a^{108} & a^{248} & a^{131} & a^{64} & a^{221} & a^{100} & a^{235} & a^{147} & a^{45} & a^{198} & a^{21} & a^{228} & a^{186} & a^{231} & a^{56} & a^{68} & a^{81} & a^{46} & a^{32} & a^{60} & a^{225} & a^{13} \\ a^{50} & a^{52} & a^{59} & a^{132} & a^{186} & a^{81} & a^{128} & a^{126} & a^{133} & a^{32} & a^{213} & a^{195} & a^{43} & a^{198} & a^{194} & a^{13} & a^{167} & a^{167} & a^{252} & a^{61} & a^{3} & a^{12} & a^{6} \\ a^{42} & a^{136} & a^{153} & a^{93} & a^{82} & a^{98} & a^{138} & a^{49} & a^{174} & a^{168} & a^{95} & a^{105} & a^{112} & a^{163} & a^{101} & a^{126} & a^{19} & a^{161} & a^{233} & a^{162} & a^{182} & a^{105} & a^{3} \end{bmatrix}$$

Le code C_2

Il s'agit d'un code (32,28,5) sur le corps à 256 éléments.

Si x est un mot de 28 octets le mot de code de C_2 qui lui correspond est le mot de 32 octets égal à $(x, x\,R_2{}^{tr})$, où la matrice $(4,28) R_2$ est :

$$R_2 = (\,R_1 \mid R'\,) \text{ où } R' = \begin{bmatrix} a^{232} & a^{98} & a^{54} & a^{174} \\ a^{167} & a^{211} & a^{180} & a^{143} \\ a^{24} & a^{41} & a^{188} & a^{164} \\ a^{92} & a^{48} & a^{168} & a^{67} \end{bmatrix}$$

CDROM : code CIRC (3/3)

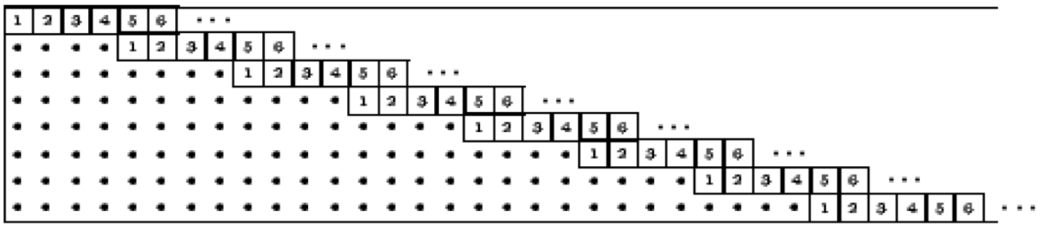


Figure 1: Table d'entrelacement à retard 4 de profondeur 8

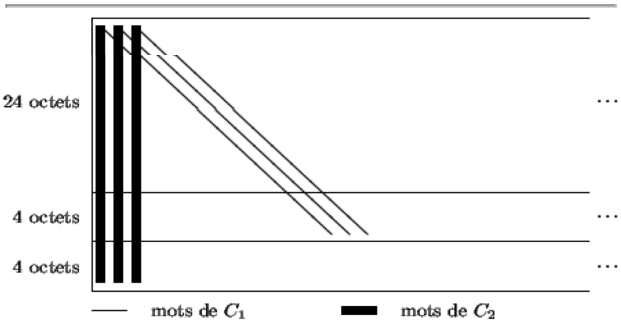


Figure 2: Schéma de codage

Satellites : ex. Voyager

- Photos de Saturne et Jupiter (1977)
  - Données peu critiques (images 800x800 - 8 bits)
  - Données critiques: GSE (General Science&Engineering)
    - Mesures
    - Contrôle
- Données GSE codées par Golay(24,12) sur F_2 (6-correcteur)
- Autres données: code convolutif

GSM

- Signal parole: par tranche de 20 ms
- Codec: Numérisation : 260 bits
  = 50 très critiques + 132 critiques + 78 complémentaires
- Codage de canal : 456 bits
  - » 50 bits : CRC (X3+x+1)
  - » 182+3 : codage convolutif : x2 : 378 bits
  - » +78= 4 bits controles :
- Entrelacement (diagonal : sur plusieurs trames de 456 bits)
- Chiffrement / modulation

Autres codes et applications

Codes Goppa(n,k,d) : définis par une courbe f(x,y) sur un GF

n=nombre de points de f et d ≥ n-deg(f); ex: $x^3y+y^3+x=0$ ➔ Goppa(24,3,20)

Pour l'instant difficiles à rendre efficaces mais très grande distance minimale

Turbo codes [C. Berrou, A. Glavieux] : codes de convolution

fonctionnent seulement sur un petit alphabet (n et k ≤ 8)

Décodage est complexe mais la correction est très grande

Cryptographie : système McEliece : code linéaire 2^{500} mots

Problème à sens unique : trouver un mot de poids minimal

Facile si la matrice génératrice normalisée est connue

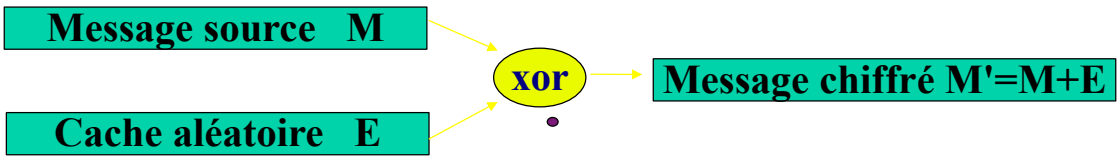
Difficile (énumérer tous les mots !) si la matrice est mélangée

Clef publique : matrice génératrice permutée

Clef privée : matrice génératrice normalisée

Codes et cryptographie

- Masquage d'information par ajout d'erreur



- Clé privée :
  - générateur de suite aléatoire (clé jetable)
- Clé publique :
  - génération du cache aléatoire par une méthode publique
  - Seul le destinataire peut supprimer le cache
- Problème: connaissant M'=X+E, trouver X :

  => *correction d'erreurs*

Problèmes difficiles en théorie des codes linéaires

- **Décodage de code linéaire** (dans F_2):
  - Entrée: H matrice (n-k, n) ; s vecteur de F_2^{n-k}; w entier
  - Sortie : e vecteur de F_2^n tel que $e.H^t=s$ et $w(e) < w$
- *Problème de décision associé*: "**Poids d'un coset**" NP-complet
  - Il existe e de F_2^n tel que $e.H^t=s$ et $w(e) < w$
- Rem: Décodage borné : idem mais $w(e) <t$ avec $d(H)=2t+1$
- **Mot de poids donné**:
  - Entrée: H matrice (n-k, n) ; w entier positif
  - Sortie : c vecteur de F_2^n de poids w tel que $c.H^t=0$
- *Problème de décision associé*: "**Poids d'un code**" NP-complet
  - Il existe c de F_2^n de poids w tel que $c.H^t=0$
- Rem: Mot de poids minimal : idem mais w(c) minimum

Cryptosystème de Mac Eliece

- Clé secrète Alice :
  - *G* matrice génératrice d'un code Goppa(*n*,*k*,d) t-correcteur
    Proposé: Goppa(1024, 524, 101) taille=n.k
  - S matrice (k,k) inversible
  - P matrice (n,n) permutation
  - N = SGP
- Clé publique Alice : la matrice N (k,n) du code (n,k) t-correcteur
- Chiffrement : Soit *m* un message source de *k* chiffres
  - On choisit aléatoirement *E* vecteur de poids t
  - On envoie $m' = m.N + e$
- Déchiffrement: On reçoit *m'* et on le décode :
  - On calcule: $u = m'.P^{-1}$
  - Puis v = correction de t erreurs dans u (code G)
  - Enfin $m = u.S^{-1}$

- Code original proposé en 1978 : k=524 n=1024 t=50
- Goppa(1024, 524, 101)
  => Taille clé publique = 524 kbits = 67 ko
  - Secret : construction du code:
    Reed-Solomon généralisé qui permet le **décodage rapide** par Berlekamp-Massey
- Paramètres suggérés: k=1608 n=2048 t=81

Cryptosystème de Niederreiter

- Clé publique : H matrice de contrôle sous forme systématique d'un code C(n,k,d) - taille clé = (n-k).k
- Clé secrète : un algorithme de décodage rapide du code C
- Chiffrement :
  - m est mis sous forme de vecteur de C (ds F_2^n)de poids (d-1)/2
  - On envoie $m' = m.H$
- Déchiffrement
  - m' est un syndrome: on le décode avec l'algorithme rapide
- Sécurité: équivalent à Mac Eliece
- Taille de clef : avec Goppa(1024, 524, 101) => 33.5 ko

Comparaison Mac Eliece / RSA

[Berger - Canteaut - Sendrier 2003]

	Mc Eliece code binaire $[1024, 524, 101]$	Niederreiter code binaire $[1024, 524, 101]$	RSA 512 bits $e = 17$	RSA 1024 bits $e = 17$
Taille des clés publiques	67072 octets	32750 octets	128 octets	256 octets
Taux de transmission	$51,17\%$	$55,20\%$	100%	100%
Nb. d'opérations binaires du chiffrement par bit d'information	514	49	2560	5120
Nb. d'opérations binaires du déchiffrement par bit d'information	3575	5343	393000	1572000

	Niederreiter $n = 1024$ $t = 50$	Niederreiter $n = 2048$ $t = 40$	RSA 512 bits $e = 17$	RSA 1024 bits $e = 17$
Taille des clés publiques	32750 octets	88440 octets	128 octets	256 octets
Taux de transmission	$55,2\%$	$60,5\%$	100%	100%
Chiffrement	49	55	2560	5120
Déchiffrement	5343	8029	393000	1572
Cryptanalyse	2^{64}	2^{102}	2^{70}	2^{100}