Transmission d'images: Code de Golay - Code de Reed-Solomon

TIPE 2020-2021 : Enjeux sociétaux

Jeanne Domanski - 20086

Juin 2021

Table des matières

Transmission d'images

Définitions

Codes linéaires et cycliques

- Codes linéaires
- Codes cycliques
- Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

- Caractéristiques
- Encoder et décoder
- Etapes de l'algorithme

Application

Comparaison

- Taux de transmission
- Probabilité d'erreurs résiduelles

Transmission d'images

Définitions

Codes linéaires et cycliques

- Codes linéaires
- Codes cycliques
- Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

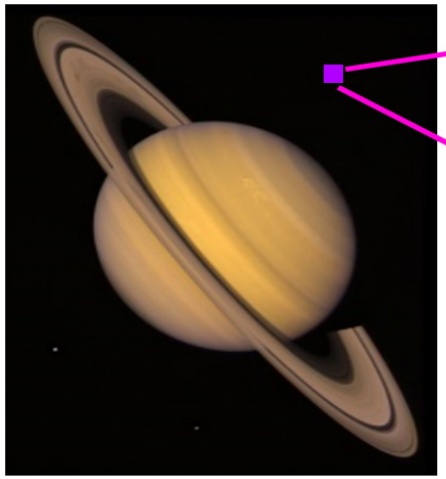
- Caractéristiques
- Encoder et décoder
- Etapes de l'algorithme

Application

Comparaison

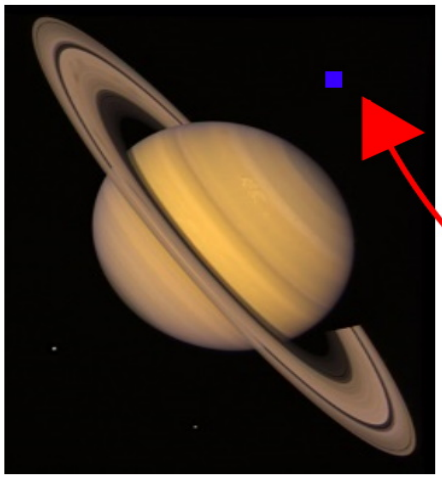
- Taux de transmission
- Probabilité d'erreurs résiduelles

Transmission d'images

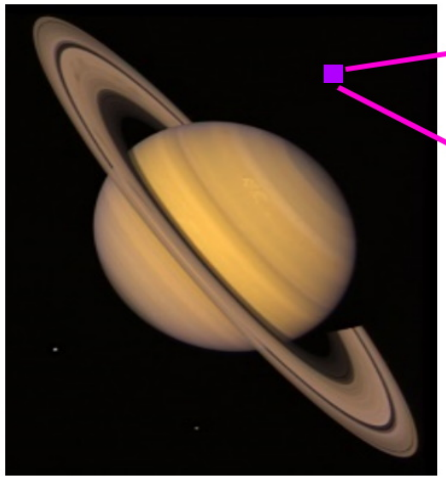


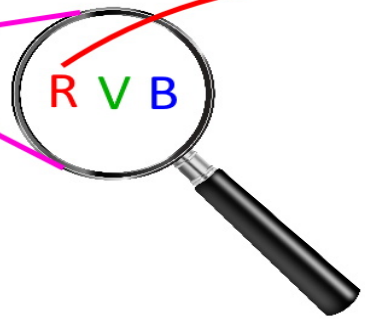
R = 10110010

R* = 00111010



Transmission d'images





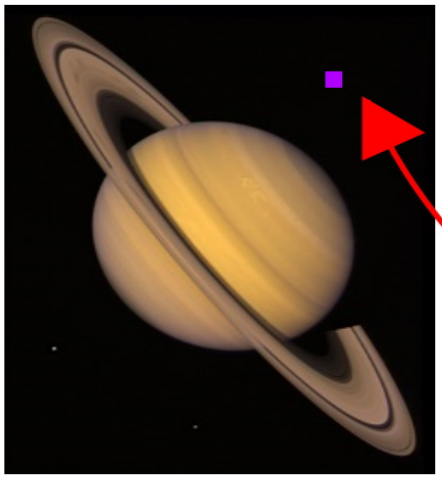
R = 10110010

Code

C = 01101011 10110010

C* = 01001111 10111000

Correction



R = 10110010

Définitions

Alphabet

$\mathbb{F}_q$ désigne un corps fini à q éléments, où q est une puissance d'un nombre premier

Poids de Hamming `[poids]`

Nombre de lettres non nulles d'un mot

Distance de Hamming `[distance_de_hamming]`

Nombre de lettres dont diffèrent deux mots, notée d_H

Distance minimale d'un code

Distance de Hamming minimale entre deux mots distincts du code, notée d

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Définitions

Erreur

Symbole erroné à l'intérieur d'un mot reçu

Effacement

Perte d'un symbole entier lors de la transmission, symbolisée par '*'

Correction

Restituer la valeur d'origine de chacun des symboles erronés d'un mot reçu

Capacité de correction

Plus grand nombre d'erreurs corrigibles par le code

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Codes linéaires

Définition

Code linéaire (n, k, d) sur $\mathbb{F}_q$: sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension k

Syndrome

Application linéaire ayant pour noyau le code, notée h ou S

Matrice de contrôle

Matrice canoniquement associée à h, notée H

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Codes cycliques

Définition

Si $(x_1, ..., x_n) \in C$ alors $(x_n, x_1, ..., x_{n-1}) \in C$

Polynôme générateur

Polynôme de $\mathbb{F}_q[X]$ unitaire diviseur de $X^n - 1$, noté g

Syndrome

Reste de la division euclidienne par g, du polynôme représentant le mot

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Tableau de déchiffrement

[tableau]
[syndrome_tab_dech]

Classe					Syndrome
0	c_1	c_2	c_3	...	$h(0) = 0$
u	$u + c_1$	$u + c_2$	$u + c_3$	...	$h(u)$
v	$v + c_1$	$v + c_2$	$v + c_3$	...	$h(v)$
...	...	...	...	...	...

Tableau de déchiffrement : exemple de code $(5, 3)$

[decodage_tab_dech]

Classe				Syndrome
$(0,0,0,0,0)$	$(0,1,1,1,0)$	$(1,0,1,0,1)$	$(1,1,0,1,1)$	$(0,0,0)$
$(1,0,0,0,0)$	$(1,1,1,1,0)$	$(0,0,1,0,1)$	$(0,1,0,1,1)$	$(1,0,0)$
$(0,1,0,0,0)$	$(0,0,1,1,0)$	$(1,1,1,0,1)$	$(1,0,0,1,1)$	$(0,1,0)$
$(0,0,1,0,0)$	$(0,1,0,1,0)$	$(1,0,0,0,1)$	$(1,1,1,1,1)$	$(0,0,1)$
$(0,0,0,1,0)$	$(0,1,1,0,0)$	$(1,0,1,1,1)$	$(1,1,0,0,1)$	$(0,1,1)$
$(0,0,0,0,1)$	$(0,1,1,1,1)$	$(1,0,1,0,0)$	$(1,1,0,1,0)$	$(1,0,1)$
$(1,1,0,0,0)$	$(1,0,1,1,0)$	$(0,1,1,0,1)$	$(0,0,0,1,1)$	$(1,1,0)$
$(1,0,0,1,0)$	$(1,1,1,0,0)$	$(0,0,1,1,1)$	$(0,1,0,0,1)$	$(1,1,1)$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Mot reçu : $y = (1,0,1,1,1)$

Syndrome : $H^T \cdot y = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

Tableau de déchiffrement : exemple de code $(5, 3)$

[decodage_tab_dech]

Classe				Syndrome
$(0,0,0,0,0)$	$(0,1,1,1,0)$	$(1,0,1,0,1)$	$(1,1,0,1,1)$	$(0,0,0)$
$(1,0,0,0,0)$	$(1,1,1,1,0)$	$(0,0,1,0,1)$	$(0,1,0,1,1)$	$(1,0,0)$
$(0,1,0,0,0)$	$(0,0,1,1,0)$	$(1,1,1,0,1)$	$(1,0,0,1,1)$	$(0,1,0)$
$(0,0,1,0,0)$	$(0,1,0,1,0)$	$(1,0,0,0,1)$	$(1,1,1,1,1)$	$(0,0,1)$
$(0,0,0,1,0)$	$(0,1,1,0,0)$	$(1,0,1,1,1)$	$(1,1,0,0,1)$	$(0,1,1)$
$(0,0,0,0,1)$	$(0,1,1,1,1)$	$(1,0,1,0,0)$	$(1,1,0,1,0)$	$(1,0,1)$
$(1,1,0,0,0)$	$(1,0,1,1,0)$	$(0,1,1,0,1)$	$(0,0,0,1,1)$	$(1,1,0)$
$(1,0,0,1,0)$	$(1,1,1,0,0)$	$(0,0,1,1,1)$	$(0,1,0,0,1)$	$(1,1,1)$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Si $t = 1$

Mot erreur : $e = (0,0,0,1,0)$

Mot envoyé :

$x = y + e = (1,0,1,0,1)$

Mot reçu : $y = (1,0,1,1,1)$

Syndrome : $H^T \cdot y = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

Tableau de déchiffrement : exemple de code $(5, 3)$

[decodage_tab_dech]

Classe				Syndrome
$(0,0,0,0,0)$	$(0,1,1,1,0)$	$(1,0,1,0,1)$	$(1,1,0,1,1)$	$(0,0,0)$
$(1,0,0,0,0)$	$(1,1,1,1,0)$	$(0,0,1,0,1)$	$(0,1,0,1,1)$	$(1,0,0)$
$(0,1,0,0,0)$	$(0,0,1,1,0)$	$(1,1,1,0,1)$	$(1,0,0,1,1)$	$(0,1,0)$
$(0,0,1,0,0)$	$(0,1,0,1,0)$	$(1,0,0,0,1)$	$(1,1,1,1,1)$	$(0,0,1)$
$(0,0,0,1,0)$	$(0,1,1,0,0)$	$(1,0,1,1,1)$	$(1,1,0,0,1)$	$(0,1,1)$
$(0,0,0,0,1)$	$(0,1,1,1,1)$	$(1,0,1,0,0)$	$(1,1,0,1,0)$	$(1,0,1)$
$(1,1,0,0,0)$	$(1,0,1,1,0)$	$(0,1,1,0,1)$	$(0,0,0,1,1)$	$(1,1,0)$
$(1,0,0,1,0)$	$(1,1,1,0,0)$	$(0,0,1,1,1)$	$(0,1,0,0,1)$	$(1,1,1)$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Mot reçu : $y = (1,0,1,1,1)$

Syndrome : $H^T \cdot y = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

Si $t = 1$

Mot erreur : e $= (0,0,0,1,0)$

Mot envoyé :

$x = y + e = (1,0,1,0,1)$

Si $t = 2$

Mots erreurs : $(0,0,0,1,0)$ et $(0,1,1,0,0)$

Mots envoyés : $(1,0,1,0,1)$ et $(1,1,0,1,1)$

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Racines primitives de l'unité

Racine $p^{ème}$ de l'unité

$\alpha \in \mathbb{F}_q$ tel que $\alpha^p = 1$

Racine $p^{ème}$ primitive de l'unité

Racine $p^{ème}$ α telle que $\forall i \in [\![1, p-1]\!]$, $\alpha^i \neq 1$

Base de $\mathbb{F}_q$ où $q = x^p$ `[table_alpha]`

Soit $\alpha \in \mathbb{F}_q$ racine primitive, $(\alpha^0, ..., \alpha^{p-1})$ est une base de $\mathbb{F}_q$ en tant que $\mathbb{F}_x$-espace vectoriel

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Code de Reed-Solomon

Caractéristiques

On se place dans $\mathbb{F}_{2^p}$.

$$n = 2^p - 1 = k + 2 \times t$$

$R = [[0,1,1,1,1,0,0,0],[0,1,1,0,0,1,1,0],\ldots,[1,1,1,1,1,1,1,0]]$

- p : nombre de bits par symbole
- n : nombre de symboles transmis
- k : nombre de symboles d'information
- $d = 2 \times t + 1$: distance minimale
- t : nombre d'erreurs corrigibles
- $r = 2t$: nombre de symboles de redondance, nombre maximal d'effacements

$$2 \times t' + e' < d - 1 = 2 \times t$$

avec t' erreurs et e' effacements

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Code de Reed-Solomon

Encoder et décoder

Polynôme générateur [calcul_g_RS]

$$g(x) = \prod_{i=0}^{2t-1} (x - \alpha^i)$$

Encoder un mot A [RS_encoder]

$C = A \times X^{2t} + B$

avec B tel que $A \times X^{2t} = g \times Q + B$

Code de Reed-Solomon

Etapes de l'algorithme

Soit R le mot reçu et C son mot de code associé

Etapes de l'algorithme

1. déterminer le domaine d'effacements I_o
2. déterminer le polynôme localisateur d'effacement σ_o
3. remplacer les '*' de R par des symboles de $\mathbb{F}_q$
4. calcul du syndrome S
5. calcul du polynome modifié du syndrome S_o
6. trouver les polynomes σ_1 et ω
7. méthode *time-domain* : obtenir E et l'existence de C
8. calcul de C

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Code de Reed-Solomon

Domaine et polynôme d'effacement

Domaine d'effacement [calcul_I_0]

$$I_o = \{i | R_i = *\}$$

$$t_o = |I_o|$$

Polynôme localisateur d'effacement [calcul_sigma_0]

$$\sigma_o = \prod_{i \in I_o} (1 - \alpha^i x)$$

[remplacer]

Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Code de Reed-Solomon

Syndrome

Syndrome [S]

$$S_j = \sum_{i=0}^{n-1} R'_i \alpha^{ij} \text{ pour } j \in [\![0, r-1]\!]$$

$$S(x) = S_0 + S_1 x + ... + S_{r-1} x^{r-1}$$

Syndrome modifié [S_0]

$$S_o(x) = S(x)\sigma_o(x) \mod x^r$$

Code de Reed-Solomon

Problème$(x^r, S_o(x), \mu, \nu)$

$$\mu = \lfloor (r - t_0)/2 \rfloor$$

$$\nu = \lceil (r + t_0)/2 \rceil - 1$$

Problème$(x^r, S_o(x), \mu, \nu)$ [Euclide_Pracines]

$\deg(\sigma_1) \leqslant \mu$, $\deg(\omega) \leqslant \nu$

$$\sigma_1(x) S_o(x) \equiv \omega(x) \mod x^r$$

Polynôme localisateur d'erreur/effacement

$$\sigma(x) = \sigma_o(x)\sigma_1(x)$$

Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Code de Reed-Solomon

Méthode *time-domain* [time_domain]

$$C = (C_0, C_1, ..., C_{n-1})$$

$$C_i = R'_i - E_i \text{ [fin]}$$

Théorème

Il existe un mot du code C à une distance d'au plus r de R si on vérifie :

$$\deg(\omega(x)) < t_o \quad (1)$$

$$\sigma_1(0) \neq 0 \quad (2)$$

$$\sigma(x) | (1 - x^n) \quad (3)$$

Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Application à la transmission d'images

code Reed-Solomon $(255, 239, 17)$

Figure – Image de Saturne prise par la sonde *Voyager 2*

[RS_appliquer]

Source : *Larger version of Saturne* NASA
https://voyager.jpl.nasa.gov/galleries/images-voyager-took/saturn/

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Application à la transmission d'images

code Reed-Solomon $(255, 239, 17)$

Figure – Image de Saturne prise par la sonde *Voyager 2*

[RS_appliquer]

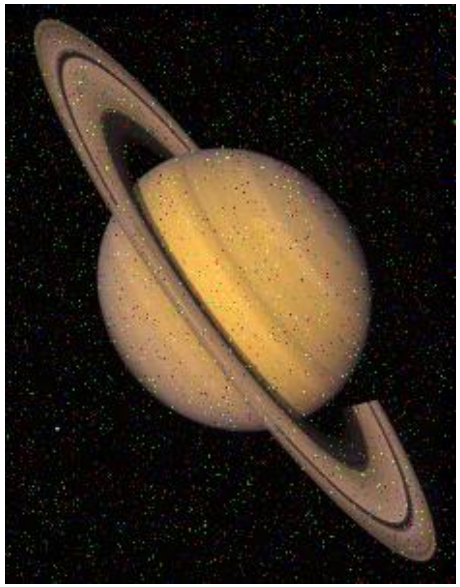


Figure – Image entachée de 20 000 erreurs aléatoires

[RS_entacher_im_codee]

Source : *Larger version of Saturne* NASA
https://voyager.jpl.nasa.gov/galleries/images-voyager-took/saturn/

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Application à la transmission d'images

code Reed-Solomon $(255, 239, 17)$

Figure – Image de Saturne prise par la sonde *Voyager 2*

`[RS_appliquer]`

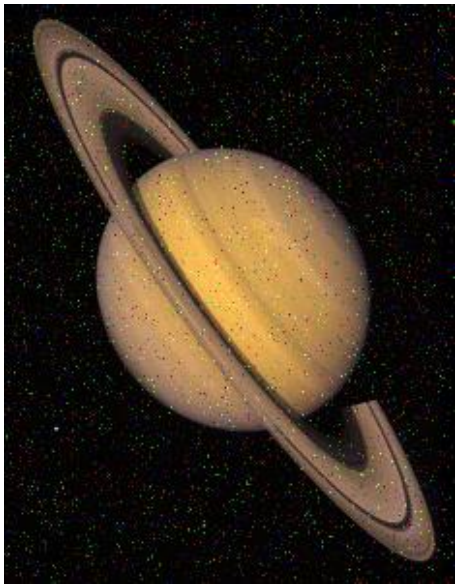


Figure – Image entachée de 20 000 erreurs aléatoires

`[RS_entacher_im_codee]`

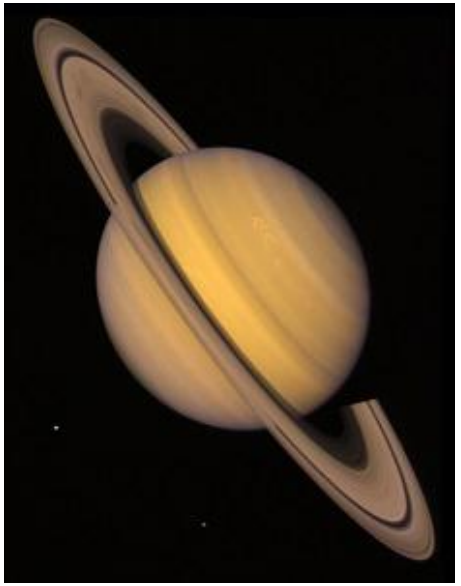


Figure – Image corrigée par le code de Reed-Solomon

`[RS_decode_im]`

Source : *Larger version of Saturne* NASA
`https://voyager.jpl.nasa.gov/galleries/images-voyager-took/saturn/`

Comparaison avec le code de Golay

Taux de transmission

Code détecteur correcteur d'erreurs	**Taux de transmission :** k/n *(en %)*
Code de Golay $(23, 12, 7)$	$52,17$
Code de Golay $(24, 12, 8)$	$50,00$
Code de Reed – Solomon $(255, 239, 17)$	$93,72$

Figure – Taux de transmission des codes de Golay et de Reed-Solomon [6]

Comparaison avec le code de Golay

Probabilité d'erreurs résiduelles

Probabilité d'avoir i des n lettres fausses `[P_i]`

$$P_i = \binom{n}{i} p^i (1-p)^{n-i}$$

Expression finale `[epsilon]`

$$\varepsilon = 1 - \sum_{i=0}^{t} P_i$$

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Comparaison avec le code de Golay

Probabilité d'erreurs résiduelles

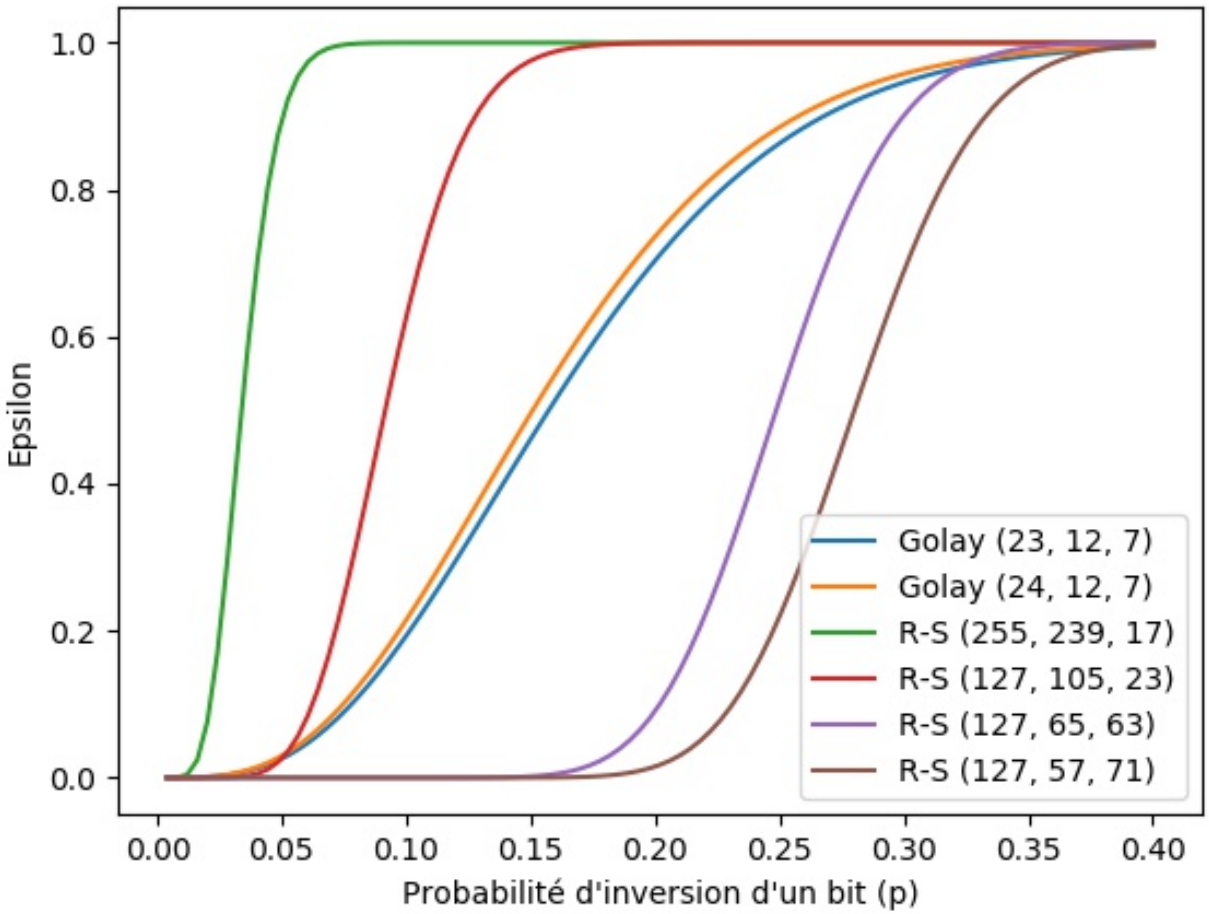


Figure – Probabilités d'erreurs résiduelles des codes de Golay et de Reed-Solomon `[compare]`

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

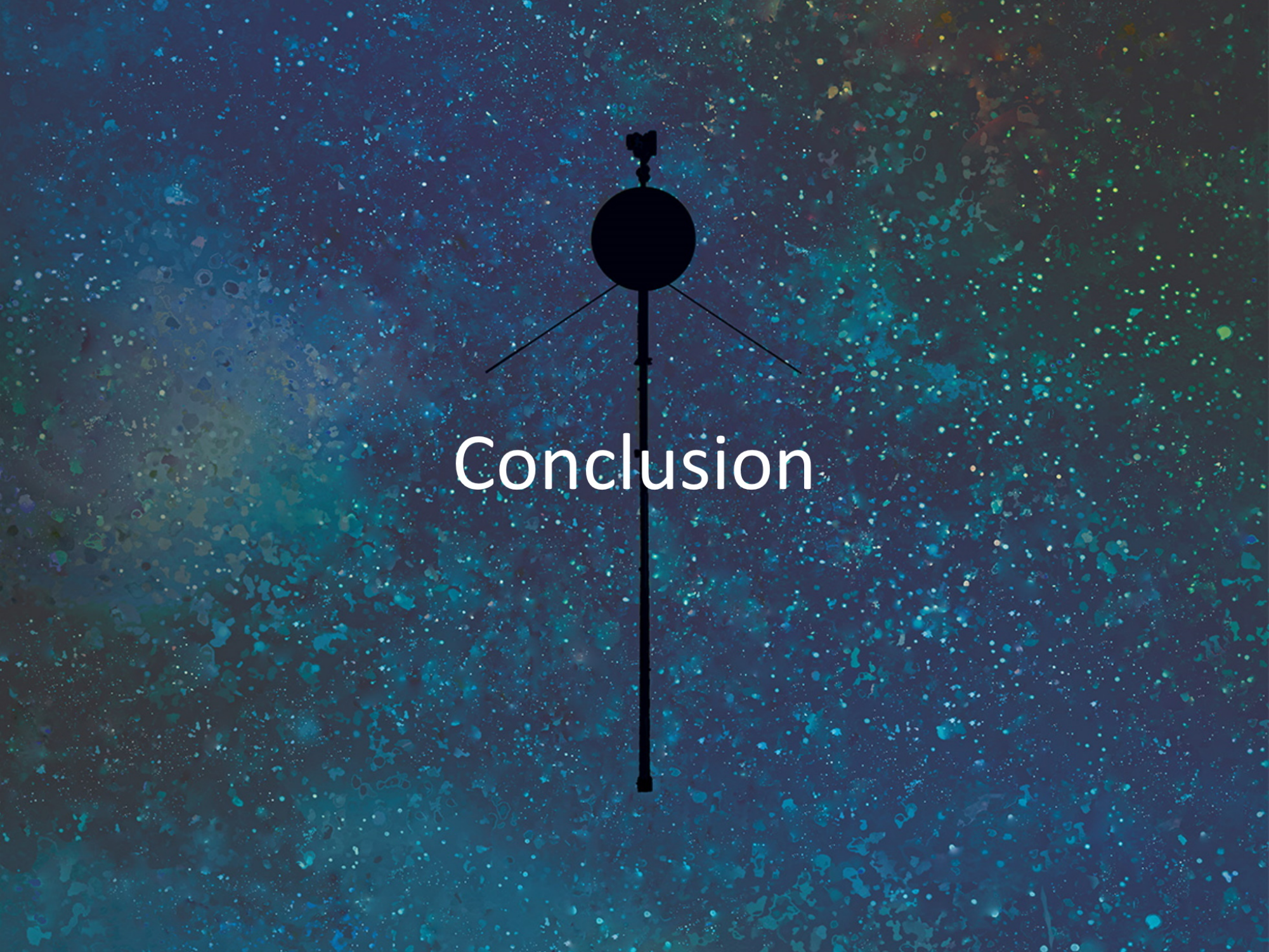
Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles


Conclusion

Annexes

1. Bibliographie
2. Tableau de déchiffrement
3. Manipulation des puissances de α
4. Méthode *time-domain*
5. Transmission d'images
   - Appliquer le code à une image
   - Entacher d'erreurs
   - Retrouver l'image représentée par un tableau
   - Décoder une image entachée d'erreurs
6. Erreurs résiduelles
   - Pourcentage d'erreurs résiduelles
   - Probabilités d'erreurs résiduelles du tableau de déchiffrement
7. Application à une autre image

Bibliographie

[1] Odile PAPINI, Jacques WOLFMANN : *Algèbre discrète et codes correcteurs* : Springer-Verlag (1994)

[2] Pierre WASSEF : Cours d'arithmétique à l'université Pierre et Marie Curie : https://perso.univ-rennes1.fr/christophe.mourougane/enseignements/2007-8/arithmetique.pdf

[3] Pierre MEUNIER : *Algèbre : Cryptologie - Codes linéaires correcteurs d'erreurs* : Cépaduès Edition (2016)

[4] R. J. MCELIECE : *The Decoding of Reed-Solomon Codes* : https://ntrs.nasa.gov/citations/19890010973 (1988)

[5] T. K. TRUONG, J. K. HOLMES, I. S. REED, X. YIN : *A simplified procedure for decoding the (23,12) and (24,12) Golay codes* : https://ntrs.nasa.gov/citations/19890010085 (1989)

[6] Abbas DANDACHE, Thierry VALLINO, Fabrice MONTEIRO, Jean-Pierre DELAHAYE : *Code Reed-Solomon (127, k, d) avec effacements : simulation et conception sur réseaux de circuits programmables (FPGA)* : http://documents.irevues.inist.fr/bitstream/handle/2042/2118/005.PDF%20TEXTE.pdf?sequence=1 (1998)

[7] François DUBOIS : Cours sur les codes et automates finis au CNAM : https://www.imo.universite-paris-saclay.fr/~fdubois/cours/codes-automates/codauto-cours-06-18mars2020.pdf (2020)

Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

- Codes linéaires
- Codes cycliques
- Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

- Caractéristiques
- Encoder et décoder
- Etapes de l'algorithme

Application

Comparaison

- Taux de transmission
- Probabilité d'erreurs résiduelles

Tableau de déchiffrement

Listing 1 – Création du tableau

```
1 def tableau(H):
2     """Renvoie le tableau de déchiffrement de H (et les syndromes
       )"""
3     c, s = debut_tableau(H) # initialise avec les classes de 0 à
       n
4     classe = c.copy()
5     synd = s.copy()
6     mot1 = classe[1][0] # premier mot de la classe 1
7     for i in range(1, len(H[0])): # ajoute 1 à la i_ième colonne
       du mot
8         mot = mot1.copy()
9         mot[i] = somme_binaire(mot[i], 1)
10        if syndrome(H, mot) not in synd: # on regarde si le h(mot
       ) est déjà dans synd
11            c1 = classe[1].copy() # récupère la classe 1
12            liste = [mot] # initialise avec mot
13            for j in range(1, len(c1)): # ajoute 1 à la i_ème
       colonne de chaque mot de classe 1
14                elt = c1[j].copy()
15                elt[i] =  somme_binaire(elt[i], 1)
16                liste.append(elt.copy())
17            classe = classe + [liste]
18            synd.append(syndrome(H, mot))
19    return classe, synd
```


Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Tableau de déchiffrement

Listing 2 – Premières classes du tableau

```
1 def classe_0(H):
2     """Renvoie la première ligne du tableau : classe de syndrome
       0"""
3     liste = [[0 for _ in range(len(H[0]))]] # liste initialisée
       avec le mot nul
4     G = mat_generatrice(H)
5     for i in range(len(G)): # ajout des lignes de G
6         liste.append(G[i])
7     for i  in range(len(G)): # ajout des sommes 2 à 2 de lignes
       de G pas encore dans la liste
8         for j in range(len(G)):
9             if j != i and somme_liste(G[i], G[j]) not in liste:
10                liste.append(somme_liste(G[i], G[j]))
11    return liste
```

```
1 def classe_j(H, j):
2     """Renvoie la j_ieme classe du tableau pour 1 <= j <= n"""
3     liste = classe_0(H).copy() # récupère la classe 0
4     for i in range(len(liste)): # ajoute 1 à la (j-1)_ième
       colonne de chaque ligne
5         liste[i][j - 1] = somme_binaire(1, liste[i][j - 1])
6     return liste
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Tableau de déchiffrement

Listing 3 – Décodage par tableau de déchiffrement

```
1 def decodage_tab_dech(y, H, e):
2     """Corrige le mot y pour un code de matrice H
      avec au plus e erreurs avec le tableau de dé
      chiffrement"""
3     s = syndrome(H, y)
4     classe, synd = tableau(H)
5     for i in range(len(synd)):
6     # on cherche la ligne correspondant au
      syndrome de y
7         if s == synd[i]:
8             i_s = i
9     lm_env = [] # = liste des mots envoyés
      possibles
10    for i in range(classe[i_s]):
11        if poids(classe[i_s][i]) <= e: # si le
      poids du mot erreur est inférieur à e
12            lm_env.append(somme_liste(y, classe[
      i_s][i])) # on ajoute le mot erreur au mot reç
      u y
13    return lm_env
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Manipulation des puissances de α

Listing 4 – Puissances de α (extrait de `[table_alpha]`)

```
    else:
    # sinon, il faut la calculer et l'ajouter dans le tableau
        if i <= p - 1:
            Pa = [0 for _ in range(p)]
            Pa[i] = 1
            tab[i] = Pa.copy()
            return Pa
        elif i == p:
            tab[p] = f[:-1].copy()
            return f[:-1]
        else:
        # si la puissance est de degré supérieur à p
            P = [0] + puissances_alpha(f, i - 1, tab)
            # on écrit alpha**i = alpha * alpha**(i - 1)
            if degre(P) == p:
            # on recopie les puissances de degré inférieur à p
                Q = somme_listes(P[:p], puissances_alpha(f, p,
    tab))
            else:
                Q = P[:p]
                for j in range(p, degre(P)):
                    if P[j] != 0:
                        aj = puissances_alpha(f, j, tab)
                        # on exprime les puissances supérieures à
     p en fonction des puissances strictement inférieures à p
                        Q = somme_listes(Q, aj)
                        # on les ajoute aux puissances déjà
    calculées
            tab[i] = Q.copy()
            return Q
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Manipulation des puissances de α

Listing 5 – Evaluer un polynôme en α^i

```
1 def evalue_alpha(U, i):
2     """Evalue U en alpha**i"""
3     if i == 0:
4         return [sum(U) % 2]
5     else:
6         P = [0 for _ in range(len(U) * i)]
7         for j in range(len(U)):
8             if U[j] != 0:
9                 P[j * i] += 1
10        E = P[:11].copy()
11        compteur = 11
12        while len(P) > compteur:
13            if P[compteur] != 0:
14                E = somme_listes(E, table_alpha(g,
      compteur))
15            compteur += 1
16        return E
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Manipulation des puissances de α

Listing 6 – Evaluer un polynôme dont les coefficents sont des polynômes en α en α^i

```
1 def evalue_alpha_Pracines(U, i):
2     """Evalue U polynôme dont les coefficients dépendent de alpha
        en alpha**i"""
3     if i == 0:
4     # on évalue le polynôme en 1
5         L = []
6         for j in range(len(U)):
7         # on somme les coefficients
8             L = somme_binaire(L, U[j])
9         return L
10    else:
11        E = []
12        for j in range(len(U)):
13            if enleve_zero(U[j]) != []:
14                # on multiplie U[j] par alpha**(i*j)
15                U_j = prod_puiss_alpha(U[j], i * j)
16                # on l'ajoute à ceux déjà calculés
17                E = somme_listes(E, U_j)
18        return E
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Méthode *time-domain*

Listing 7 – Méthode *time-domain*

```
1 def time_domain():
2     """renvoie le polynome E et decode (= bool existence de C)"""
3     if(sigma_0 == [[0 for _ in range(p)]] or len(omega)>= t_0 +
       len(sigma_1)):
4         decode = false
5     else:
6         count = 0
7         E = []
8         for i in range(n):
9             sigma_p = derive_Pracines(sigma,1)
10            c = evalue_alpha_Pracines(sigma_p, -i) # sigma_prime(
       a**-i)
11            if evalue_alpha_Pracines(sigma, -i) == [] and c != 0:
12                count += 1
13                a = prod_scal(-1, inverse(table_alpha(f,i*(L-1)))
       ) #-a**(-i(L-1))
14                b = evalue_alpha_Pracines(omega,-i) # omega(a**-i
       )
15                E_i = prod(a, prod(b, inverse(c))) #a * b/c
16            else:
17                E_i = 0
18            E.append(E_i)
19        if count == len(sigma):
20            decode = True
21        else:
22            decode = False
23    return decode, E
24
25 decode, E = time_domain()
```


Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Appliquer le code à une image

Listing 8 – Appliquer le code de Reed-Solomon à une image

```
1 def RS_appliquer(image):
2     """Encode chaque mot : groupe de 239 couleurs (1/3 de pixel)
       de l'image avec le code de Reed_Solomon"""
3     A = time.perf_counter()
4     L_paquets, L_reste = RS_separer(image)
5     l, q = len(L_paquets), len(L_paquets[0])
6     Lp_encode, Lr_encode = [[] for _ in range(l)], []
7     for i in range(l):
8         for j in range(q):
9         # on encode chaque élément de L_paquets[i]
10            Lp_encode[i].append(RS_encoder(L_paquets[i][j]))
11    for k in range(len(L_reste)):
12    # on encode chaque élément de L_reste
13        Lr_encode.append(RS_encoder(L_reste[k]))
14    B = time.perf_counter()
15    print(B-A)
16    return Lp_encode, Lr_encode
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Entacher d'erreurs un mot codé

Listing 9 – Entacher d'erreurs un mot codé

```
1 def RS_entacher_mot(mot, e):
2     """Entache de e erreurs le mot de code"""
3     #on fixe le nombre d'erreurs e0 et d'effacements t0, tq 2*t0
       + e0 = e où e<2t
4     t0 = random.randrange(0,e//2)
5     e0 = e - 2*t0
6     ind_me0 = [i for i in range(len(mot))] #on liste tous les
       indices du mot
7     #erreurs
8     ind_e0 = random.sample(ind_me0, e0) #on sélectionne les
       indices des symboles erronés
9     #effacements
10    ind_mt0 = []
11    for i in ind_me0: #on liste les indices restants du mot
12        if i not in ind_e0:
13            ind_mt0.append(i)
14    ind_t0 = random.sample(ind_mt0, t0)  #on sélectionne les
       indices des symboles effacés
15    for i in ind_e0:
16        #on entache d'erreurs e0 symboles en falsifiant jusqu'à p
        bits
17        e2 = random.randrange(1,p)
18        ind_se2 = [j for j in range(len(mot[i]))] #on liste les
       indices du symbole
19        ind_be2 = random.sample(ind_se2, e2) #on sélectionne les
       indices des bits erronés du symbole
20        for j in ind_be2:
21            #on falsifie e2 bits du symbole
22            mot[i][j] = (mot[i][j] + 1) % 2
23    for i in ind_t0:
24        #on efface t0 symboles en les remplaçant par '*'
25        mot[i] = '*'
26    return mot
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

- Codes linéaires
- Codes cycliques
- Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

- Caractéristiques
- Encoder et décoder
- Etapes de l'algorithme

Application

Comparaison

- Taux de transmission
- Probabilité d'erreurs résiduelles

Entacher d'erreurs une image codée

Listing 10 – Entacher d'erreurs une image codée (extrait de [RS_entacher_im_codee])

```
1     assert 2*t * (n_Lp + n_Lr) >= nb #on peut introduire au plus
       2*t * (n_Lp + n_Lr) erreurs
2     ind_tot = [i for i in range(n_Lr + n_Lp)]
3     #de 0 à n_Lr, on aura les mots de Lr_encode
4     # de n_Lr jusqu'à n_Lr + n_Lp, on aura les mots de Lp_encode
5     ind_m = [] #indices des mots déjà entachés
6     while compt < nb:
7         #on entache des mots qui n'ont pas été entaché jusqu'à ce
        que nb erreurs aient été introduites
8         ind_e = random.sample(ind_tot, 1) #on choisit un mot à
       entacher au hasard
9         ind_e = ind_e[0]
10        if ind_e not in ind_m: #si ce mot n'est pas déjà entaché
11            ind_m.append(ind_e)
12            e = random.randrange(1,2*t)
13            compt += e
14            if ind_e < n_Lr:
15                Lr_encode[ind_e] = RS_entacher_mot(Lr_encode[
       ind_e],e)
16            else:
17                ind_e -= n_Lr
18                ligne = ind_e // col_tot
19                colonne = ind_e % col_tot
20                Lp_encode[ligne][colonne] = RS_entacher_mot(
       Lp_encode[ligne][colonne],e)
21    return (Lp_encode, Lr_encode)
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

Codes linéaires

Codes cycliques

Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

Caractéristiques

Encoder et décoder

Etapes de l'algorithme

Application

Comparaison

Taux de transmission

Probabilité d'erreurs résiduelles

Décoder une image entachée d'erreurs

Corriger les mots de l'image codée

Listing 11 – Décoder une image : Corriger les mots

```
1 def RS_decode_im(L_code, decode):
2     Lp_encode, Lr_encode = L_code
3     #on corrige les mots
4     for mot in Lr_encode:
5         mot = decode(mot)
6     for ligne in Lp_encode:
7         for mot in ligne:
8             mot = decode(mot)
9     k = len(Lr_encode[-1]) #longueur d'un mot encodé
10    val_rest = enleve_zero_Pracines(k) #on enlève les zeros qu'on
        avait rajouté pour voir les 'vrais' valeurs restantes
11    nb_octet_Lr = (len(Lr_encode) - 1) * k + k - val_rest #on
       calcule combien d'octets on a dans Lr_encode
12    ligne_tot = len(Lp_encode) #nombre de lignes
13    lg_val_rest = Lr // ligne_tot #longueur des valeurs restantes
        pour chaque ligne
14    nb_octet_ligne_Lp = len(Lp_encode[0]) * k
15    octet_ligne = nb_octet_ligne_Lp + nb_octet_Lr // ligne_tot #
       nombre d'octet par ligne
16    colonne_tot = octet_ligne // 3 #nombre de pixels par ligne ie
        nombre de colonnes
17    im_e = Image.new('RGB', (ligne_tot,colonne_tot), (255, 255,
       255))
```

Décoder une image entachée d'erreurs

Récupérer les octets des listes de mots

Listing 12 – Décoder une image : Récupérer les octets

```
1     #on reconstitue l'image
2     #on decompose Lr en liste d'octets
3     Lr_octet = []
4     for mot in Lr_decode:
5         for octet in mot:
6             Lr_octet.append(octet)
7     #on decompose les lignes de Lp en octets
8     Lp_ligne_octet = []
9     for ligne in Lp_encode:
10        Lp_ligne = []
11        for mot in ligne:
12            for octet in mot:
13                Lp_ligne.append(octet)
14        Lp_ligne_octet.append(Lp_ligne)
```


Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Décoder une image entachée d'erreurs

Reconstituer les pixels

Listing 13 – Décoder une image : Reconstituer les pixels

```
1     #on rassemble en pixels les octets de chaque ligne
2     for i in range(ligne_tot):
3         #pour chaque ligne de pixel de l'image
4         j = 0
5         while j + 2 < nb_octet_ligne_Lp:
6             pixel = (bin_to_dec(Lp_ligne_octet[j]), bin_to_dec(
      Lp_ligne_octet[j + 1]), bin_to_dec(Lp_ligne_octet[j + 2]))
7             im_e.putpixel((i,j),pixel)
8             j += 3
9         l = 0 #valeur de départ du parcours de Lr_octet
10        if j + 3 == nb_octet_ligne_Lp: #il reste 1/3 de pixel
11            pixel = (bin_to_dec(Lp_ligne_octet[j + 2]),
      bin_to_dec(Lr_octet[0]), bin_to_dec(Lr_octet[1]))
12            im_e.putpixel((i,j),pixel)
13            l = 2
14        elif j + 4 == nb_octet_ligne_Lp: #il reste 2/3 de pixel
15            pixel = (bin_to_dec(Lp_ligne_octet[j + 2]),
      bin_to_dec(Lp_ligne_octet[j + 3]), bin_to_dec(Lr_octet[0]))
16            im_e.putpixel((i,j),pixel)
17            l = 1
18        while l + 2 < nb_octet_Lr: #on parcourt tous les octets
      de la liste Lr
19            pixel = (bin_to_dec(Lr_octet[j + 3 + l]), bin_to_dec(
      Lr_octet[j + 3 + l + 1]), bin_to_dec(Lr_octet[j + 3 + l + 2])
      )
20            im_e.putpixel((i,j + 3 + l),pixel)
21            l += 3
22    im_e.show()
```


Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles

Pourcentage d'erreurs résiduelles

Listing 14 – Nombre d'erreurs résiduelles

```
1 def erreurs_residuelles(im, im_c):
2     """Compte le nombre d'erreurs residuelles dans l'image corrig
      ée en la comparant avec l'image initiale"""
3     l, p = im.size
4     nb = 0
5     for i in range(l):
6         for j in range(p):
7             pix = im.getpixel((i, j))
8             pix_c = im_c.getpixel((i, j))
9             nb += distance_de_Hamming(pixel_to_list(pix),
       pixel_to_list(pix_c))
10    return nb
```

Listing 15 – Pourcentage d'erreurs résiduelles

```
1 def pourcentage_erreurs_residuelles(im, im_c):
2     """Calcule le poucentage d'erreurs residuelles dans l'image
       corrigée en la comparant avec l'image initiale"""
3     l, p = im.size
4     nb_erreurs = erreurs_residuelles(im, im_c)
5     nb_demi_pixels = l * p
6     nb_composantes = nb_demi_pixels * 24
7     return (nb_erreurs / nb_composantes) * 100
```


Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images

Définitions

Codes linéaires et cycliques

- Codes linéaires
- Codes cycliques
- Tableau de déchiffrement

Racines primitives

Code de Reed-Solomon

- Caractéristiques
- Encoder et décoder
- Etapes de l'algorithme

Application

Comparaison

- Taux de transmission
- Probabilité d'erreurs résiduelles

Probabilités d'erreurs résiduelles du tableau de déchiffrement

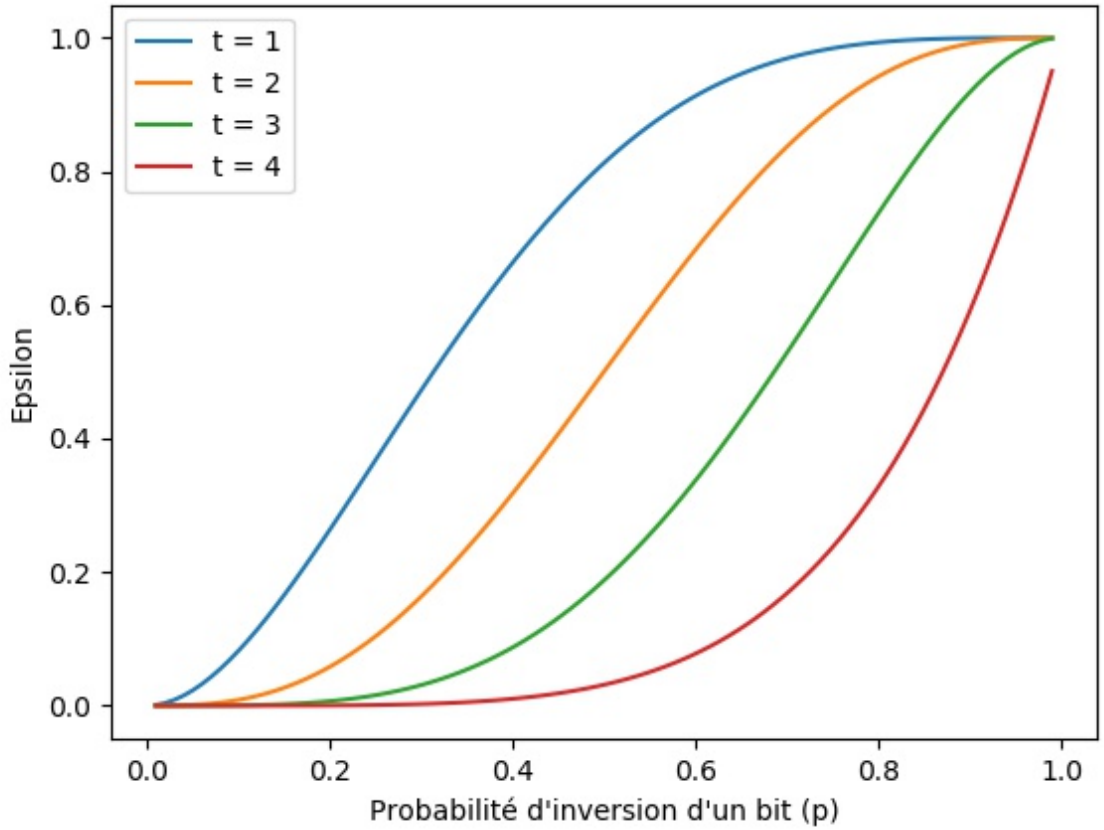


Figure – Probabilités d'erreurs résiduelles du tableau de déchiffrement pour différentes capacités de correction `[trace_tab]`

Application à une autre image

Figure – Image de Jupiter prise par la sonde *Voyager 2*

Figure – Image entachée de 10 000 erreurs aléatoires

Source :
Jupiter and two moons NASA
`https://voyager.jpl.nasa.gov/galleries/images-voyager-took/jupiter/`

Figure – Image obtenue après correction par le code de Golay

Transmission d'images : Code de Golay - Code de Reed-Solomon

Jeanne Domanski

Transmission d'images
Définitions
Codes linéaires et cycliques
Codes linéaires
Codes cycliques
Tableau de déchiffrement
Racines primitives
Code de Reed-Solomon
Caractéristiques
Encoder et décoder
Etapes de l'algorithme
Application
Comparaison
Taux de transmission
Probabilité d'erreurs résiduelles