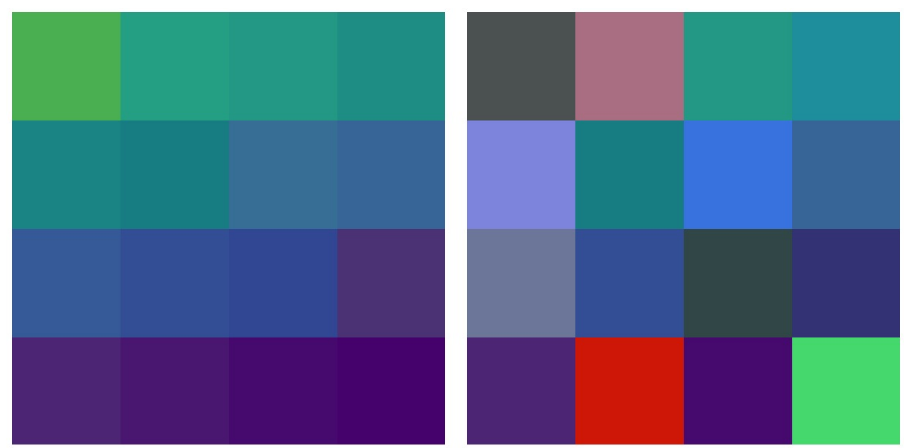
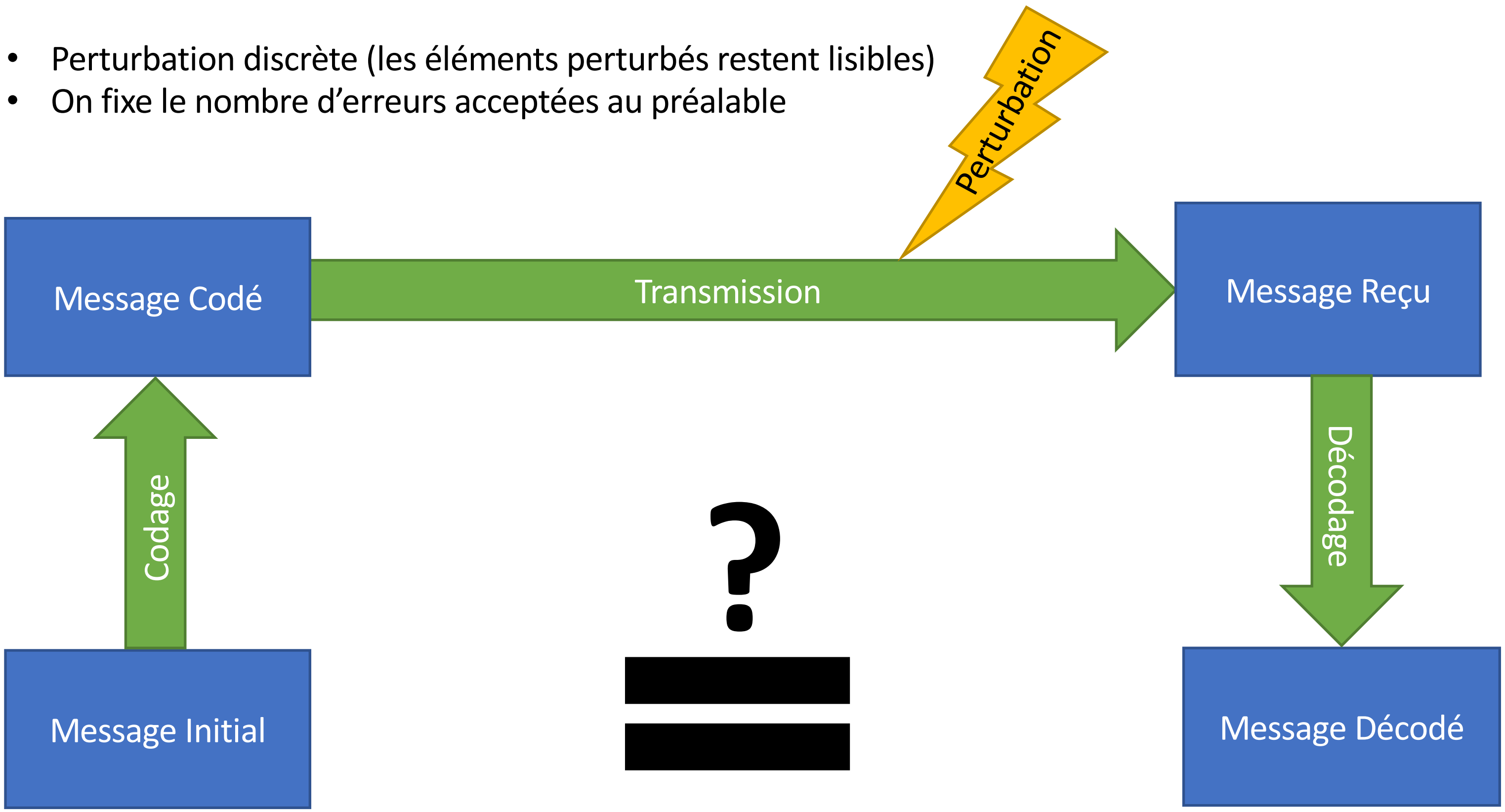
Transmission perturbée et codes correcteur d'erreur



Cadre général

- Perturbation discrète (les éléments perturbés restent lisibles)
- On fixe le nombre d'erreurs acceptées au préalable



La Redondance

Exemple de l'alphabet phonétique de l'OTAN :

A Alpha	**B** Bravo	**C** Charlie	**D** Delta	**E** Echo
F Foxtrot	**G** Golf	**H** Hotel	**I** India	**J** Juliette
K Kilo	**L** Lima	**M** Mike	**N** November	**O** Oscar
P Papa	**Q** Quebec	**R** Romeo	**S** Sierra	**T** Tango
U Uniform	**V** Victor	**W** Whisky	**X** X-ray	**Y** Yankee
		Z Zulu		

Le corps fini $\mathbb{F}_{16}$

Évariste Galois

$$\mathbb{F}_{16} = \frac{\mathbb{F}_2[X]}{(X^4 + X + 1)}$$

On note $\alpha = \overline{X}$ l'élément primitif du corps fini

$\mathbb{F}_{16}$	$\overline{0}$	$\overline{1}$	$\overline{X}$	$\overline{X^2}$	$\overline{X^3}$	$\overline{X+1}$	$\overline{X^2+X}$	$\overline{X^3+X^2}$	$\overline{X^3+X+1}$	$\overline{X^2+1}$
	/	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8

$\overline{X^3+X}$	$\overline{X^2+X+1}$	$\overline{X^3+X^2+X}$	$\overline{X^3+X^2+X+1}$	$\overline{X^3+X^2+1}$	$\overline{X^3+1}$
α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}

Paramètre globaux importants du code de Reed-Solomon

Paramètres choisis :

- Envoie d'un mot de 9 lettres de F_{16}
- Taux d'erreurs acceptées : majoré par 20% (jusqu'à 3 pour le message de 15 caractères, jusqu'à 2 pour le message de 9 caractères)

- Matrice d'encodage :

$$G = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^8 \\ & & \vdots & \\ 1 & \alpha^{15} & \cdots & \alpha^{14\times 8} \end{pmatrix} \in \mathcal{M}_{15,9}(\mathbb{F}_{16})$$

- Matrice de contrôle :

$$H = \begin{pmatrix} 1 & \alpha & \cdots & \alpha^{14} \\ 1 & \alpha^2 & \cdots & \alpha^{2\times 14} \\ & & \vdots & \\ 1 & \alpha^6 & \cdots & \alpha^{6\times 14} \end{pmatrix} \in \mathcal{M}_{6,15}(\mathbb{F}_{16})$$

Encodage Reed-Solomon:

U : Message initial

$\mathbb{F}_{16}{}^{9}$

Encodage
➔ Redondance

$$C = GU$$

$$G \in M_{15,9}(\mathbb{F}_{16})$$

C : Message codé

$\mathbb{F}_{16}{}^{15}$

Transmission

R = C+E : Message reçu

$\mathbb{F}_{16}{}^{15}$

Perturbation

$\mathbb{F}_{16}{}^{15}$

E : Erreurs

Décodage de Reed-Solomon par la méthode des polynômes

$\boxed{\text{Soit } Q(X,Y) = Q_0(X) + Y.Q_1(X) \text{ tel que } \forall i \in [\![1,15]\!], Q(\alpha^{i-1}, r_i) = 0}$

$Q \in \mathbb{F}_{16}[X,Y]$

$U(X) = u_0 + u_1 X + \cdots + u_8 X^8$ $\qquad \forall i \in [\![1,15]\!], U(\alpha^{i-1}) = c_i$

$P = Q(X, U(X)) = 0 \Rightarrow P = 0 \text{ i.e. } U = -\frac{Q_0}{Q_1}$

Décodage de Reed-Solomon par la méthode des syndromes

On appelle **Syndrome** le vecteur S :

$$S = HR = H(C + E) = HE$$

- Si S est nul alors le message est correct donc C = R
- Sinon :
  - **Q_1 étant localisateur d'erreurs**, on trouve les indices de présence d'erreurs
  - On extrait de HE = S le système réduit :

$$\begin{pmatrix} \alpha^{i_1-1} & \alpha^{i_2-1} & \alpha^{i_3-1} \\ \alpha^{2\times(i_1-1)} & \alpha^{2\times(i_2-1)} & \alpha^{2\times(i_3-1)} \\ \alpha^{3\times(i_1-1)} & \alpha^{3\times(i_2-1)} & \alpha^{3\times(i_3-1)} \end{pmatrix} \begin{pmatrix} e_{i_1} \\ e_{i_2} \\ e_{i_3} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix}$$

  - On retrouve C car C = R - E
- Ayant C on retrouve U en inversant G à gauche

Implémentation en Python

Modules Utilisés :

- **Pyfinite** :
  - **ffield** : Corps fini F_{16}
  - **GenericMatrix** : Matrices à coefficient dans F_{16}
- **Matplotlib** : Traitement et affichage de l'image

Fonctions créées dans les fichiers python

Fonctions de l'algorithme de Reed-Solomon

- encode (3 lignes)
- decode.polynomes (29 lignes)
- decode.syndromes (48 lignes)
- erreur (7 lignes)
- mesure_temps (9 lignes)

Fonction du Corps

- puissance (4 lignes)
- deg (5 lignes)
- simple (7 lignes)
- div_euclid (20 lignes)

Fonctions de traitement d'une image

- hexa (4 lignes)
- unhexa (8 lignes)
- convert_t_l (11 lignes)
- convert_l_t (6 lignes)
- cut (7 lignes)
- uncut (10 lignes)
- modification_tableau_rs (8 lignes)
- modification_tableau (6 lignes)

Au total

4 fichiers python
385 lignes de code

Préparation de l'image

R<256
V<256
B<256

Hexa(R) : R_1, R_2 < 16

On code chaque Composante comme deux éléments de $\mathbb{F}_{16}$

On concatène ces doublets dans une liste R (taille = 22365)

R = [6, 1, 5, 14, 5, 11, 5, 7, 5, 3, 4, 14, 4, 9, 4, 7, 4, 1, 4, 3, 4, 6, 4, 1, 4, 5, 4, 9, 4, 7,...]

On découpe cette liste en sous-liste de taille 9 pour l'envoi par Reed-Solomon

R = [[6, 1, 5, 14, 5, 11, 5, 7, 5], [3, 4, 14, 4, 9, 4, 7, 4, 1], [4, 3, 4, 6, 4, 1, 4, 5, 4], [9, 4, 7, ...]]

387 x
260

Exemple de traitement d’une composante d’un pixel

R1 = [6, 1, 5, 14, 5, 11, 5, 7, 5]

encode

[5, 5, 9, 4, 1, 3, 6, 13, 1, 10, 11, 5, 0, 11, 12]

2 erreurs

3 erreurs

[3, 5, 9, 6, 13, 3, 6, 13, 1, 10, 11, 5, 0, 11, 12]

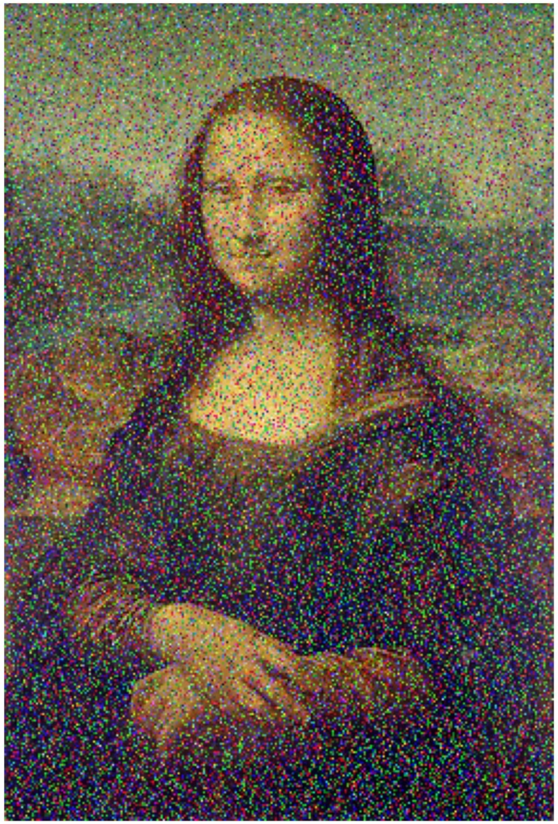
décode

[6, 1, 5, 14, 7, 11, 5, 14, 5]

[6, 1, 5, 14, 5, 11, 5, 7, 5]

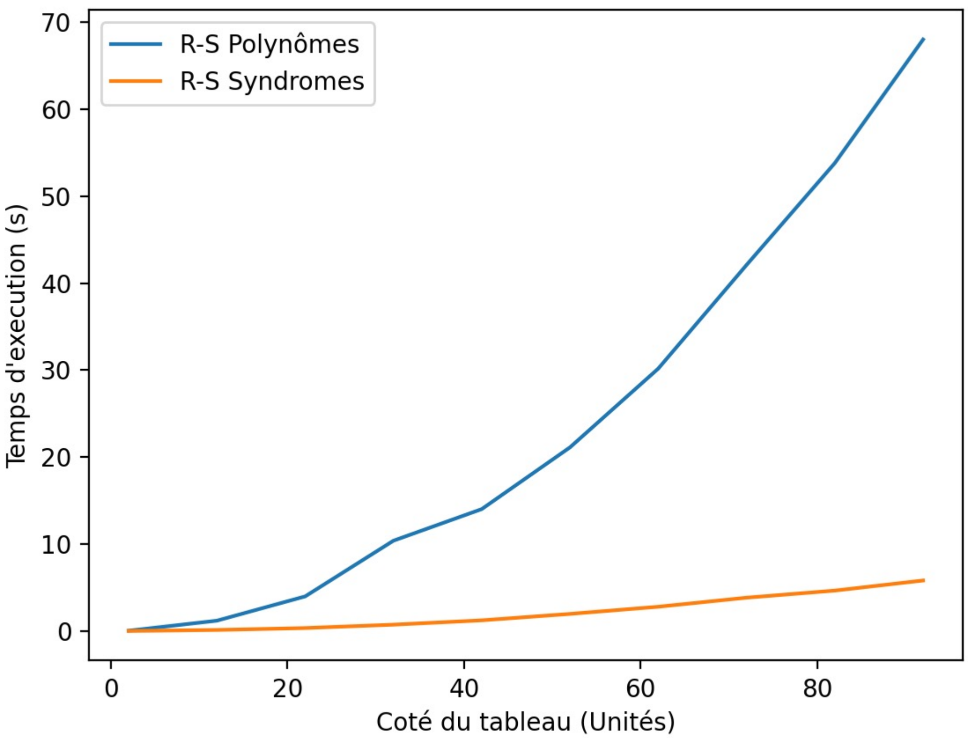
On concatène les messages reçu puis on reconstitue les composantes R, V, B de chaque Pixel

Sans Reed-Solomon



Avec Reed-Solomon

Complexité des algorithmes de décodage



Annexe: Démonstration de Q1 localisateur d'erreur

Pour chaque i entre 0 et 15:

$$\left\{ \begin{array}{lcl} Q_0(\alpha^{i-1}) + r_i Q_1(\alpha^{i-1}) & = & 0 \\ Q_0(\alpha^{i-1}) + c_i Q_1(\alpha^{i-1}) & = & 0 \end{array} \right.$$

Par définition de Q
Car P est nul

Donc $$(r_i - c_i) Q_1(\alpha^{i-1}) = 0$$

Si $r_i \neq c_i$ alors il y une erreur, et alors : $Q_1(\alpha^{i-1}) = 0$