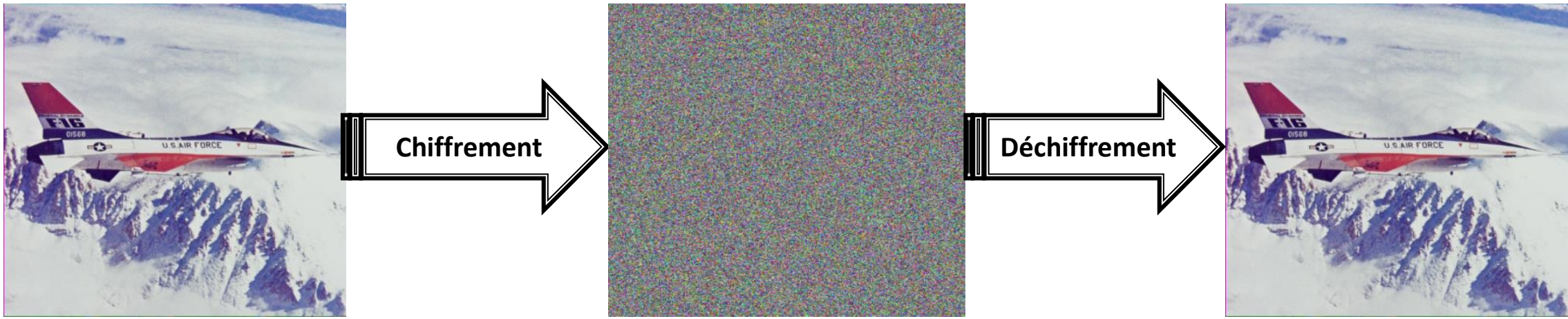
Chapitre 5: Chiffrement d'images

Chiffrement d'images

- Le chiffrement d'image est défini comme le processus de conversion de l'image claire sous une forme non reconnaissable afin de la protéger contre tout accès non autorisé. Dans le sens contraire, le déchiffrement est l'opération inverse permettant de restituer l'image claire à partir de l'image chiffrée



Chiffrement d'images

La cryptographie moderne se base sur deux principes fondamentaux

Principe de Kirchhoff

La sécurité d'un crypto système doit reposer seulement sur le secret de la clé de chiffrement et non sur l'algorithme.

Principes de Shannon

Un crypto-système doit respecter les conditions de confusion et de diffusion :

- **Confusion :** Création d'une forte relation entre l'information en clair et la clé de chiffrement.
- **Diffusion :** Création d'une forte relation entre l'information chiffrée et l'information claire suivante

Techniques de chainage

Techniques de permutation et de substitution des pixels

Chiffrement d'images

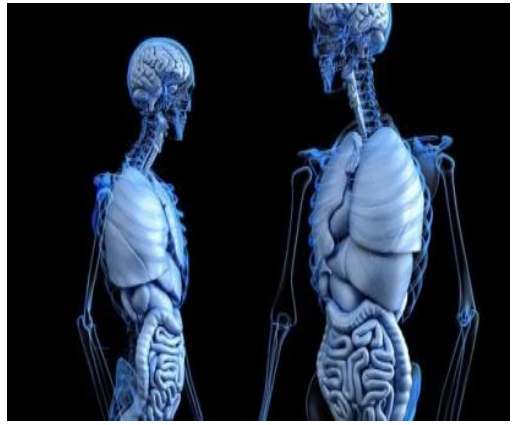
Caractéristiques des images

Une forte corrélation

existence d'une relation linéaire entre les valeurs des pixels voisins

Une redondance élevée

Existence de plusieurs pixels voisins de même intensité lumineuse dans la même image.



Faiblesse des Algorithmes standard (DES, AES, RSA...)

Les systèmes traditionnels ne sont pas adéquats au chiffrement de ce type de données. (Utilisation d'un chiffrement par blocs indépendants)
→ (Exposition aux attaques statistiques)

La solution la plus efficace est d'utiliser une technologie basée sur le chaos en utilisant des cartes chaotiques.

Cartes chaotiques

Une carte chaotique est un système **dynamique, non linéaire**, **non périodique**, **déterministe** et **imprévisible** à long terme. Il permet de générer un signal:

- **Déterministe**
- **Non périodique.**
- **Très sensible aux conditions initiales et aux paramètres de contrôle qui forment la clé secrète.**

Cartes chaotiques

Les cartes unidimensionnelles

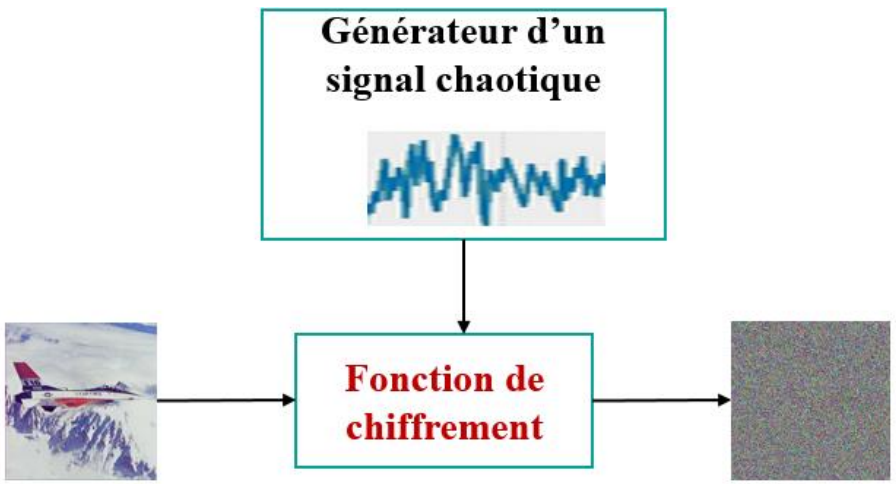
Carte chaotique	Expression	Paramètres
Logistique	$X_{n+1} = \mu X_n(1 - X_n)$	$\mu \in [3{,}75\ ,4]$: Paramètre de contrôle. $X_0 \in]0{,}5\ 1[$: Valeur initiale.
Tente	$X_{n+1} = \begin{cases} rX_n & si\ \ X_n < 0.5 \\ r\ (1 - X_n) & si\ X_n \geq 0.5 \end{cases}$	$r \in]0\ ,2]$: Paramètre de contrôle. $X_0 \in]0\ ;1]$: Valeur initiale.
PWLCM	$X_{n+1} = \begin{cases} \dfrac{X_n}{p}\ si\ 0 \leq X_n \leq p \\ \dfrac{X_n - p}{0.5 - p}\ si\ p \leq X_n \leq 0.5 \\ f(1 - X_n, p) ailleurs \end{cases}$	$p \in]0\ ,0.5[$: Paramètre de contrôle. $X \in]0\ ,1[$: Valeur initiale.
Chebeshev	$\mathrm{X}_{n+1} = F_C(r_1, \mathrm{X}_n) = \cos\left(r_1 arcos(\mathrm{X}_n)\right)$	$r_1 \in \mathbb{N}$: Paramètre de contrôle. X_0 : Valeur initiale.
Sinus	$X_{n+1} = F_S(r_2, X_n) = r_2 sin\left(\pi(X_n)\right)$	$r_2 \in]0{,}1[$: Paramètre de contrôle. X_0 : Valeur initiale.

Cartes chaotiques

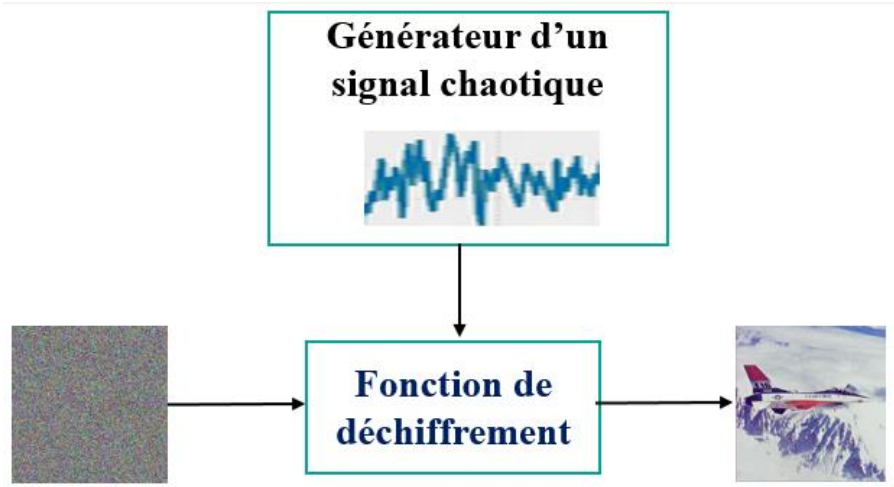
Les cartes bidimensionnelles

Carte chaotique	Expression	Paramètres
La carte logistique 2D	$\begin{cases} x_{n+1} = \mu_1 x_n(1-x_n) + \mu_2 y_n^2 \\ y_{n+1} = \mu_3 y_n(1-y_n) + \mu_4 x_n^2 \end{cases}$	Avec $\begin{cases} x_0 \; et \; y_0 \in]0,1] \\ \mu_1 \in [2.75;\ 3.4] \\ \mu_2 \in [0{,}15\ ;0{,}21] \\ \mu_3 \in [2.75;\ 3.45] \\ \mu_4 \in [0{,}13; 0{,}15] \end{cases}$
La carte Hénon	$\begin{cases} x_{n+1} = 1 - a x_n^2 + y_n \\ y_{n+1} = b x_n \end{cases}$	Avec $\begin{cases} x_0 \; et \; y_0 \in]0{,}0.5] \\ a = 1.4 \\ b = 0.3 \end{cases}$

Chiffrement chaotique d'images



Chiffrement de données basé sur le chaos



Déchiffrement de données basé sur le chaos

Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Etapes de l'algorithme**

1. **Génération des clés de chiffrement**
   - Développement de séquences chaotiques en utilisant la carte logistique
   - Développement de vecteurs pseudo-aléatoires
2. **Préparation de l'image à chiffré**
   - Vectorisation de l'image originale
   - XOR avec le vecteur chaotique
3. **Confusion**
   - Permutation P-Box
   - Table de substitution S-Box
4. **Diffusion**
   - Chainage: Mode Cipher Block Chaining (CBC)

Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Génération de la clé de chiffrement**

❑ Développement de séquences chaotiques en utilisant la carte logistique:

La carte logistique est une suite simple, dont la récurrence est non linéaire, donnée par la relation suivante : $\boldsymbol{X_{n+1} = \mu X_n\,(1 - X_n)}$. Le comportement chaotique de cette carte est obtenu si:

$$\mu \epsilon [3{,}57\,;\,4] \;\; et \; X_0 \in]0{,}5\,;\,1[$$

- $\boldsymbol{X_n}$ présente la variable dynamique prenant des valeurs dans l'intervalle]0, 1[
- $\boldsymbol{\mu}$ est le paramètre de contrôle du système prenant des valeurs dans l'intervalle]0, 4]

```
import numpy as np
```

[7]:
```
def logistic_function(x,mu):
    return mu*x*(1-x)
```

[8]:
```
def logistic_map_gen(n,mu=3.755,xk=0.79878796):
    logistic_map=np.zeros(n)
    logistic_map[0]=xk
    for i in range(1,n):
        xk=logistic_function(xk,mu)
        logistic_map[i]=xk
    return logistic_map
```

[9]:
```
x=logistic_map_gen(10)
x
```

[9]:
```
array([0.79878796, 0.60352521, 0.8985059 , 0.34242988, 0.84551958,
       0.49046391, 0.93840853, 0.21703134, 0.6380824 , 0.86715435])
```

Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Génération de la clé de chiffrement**

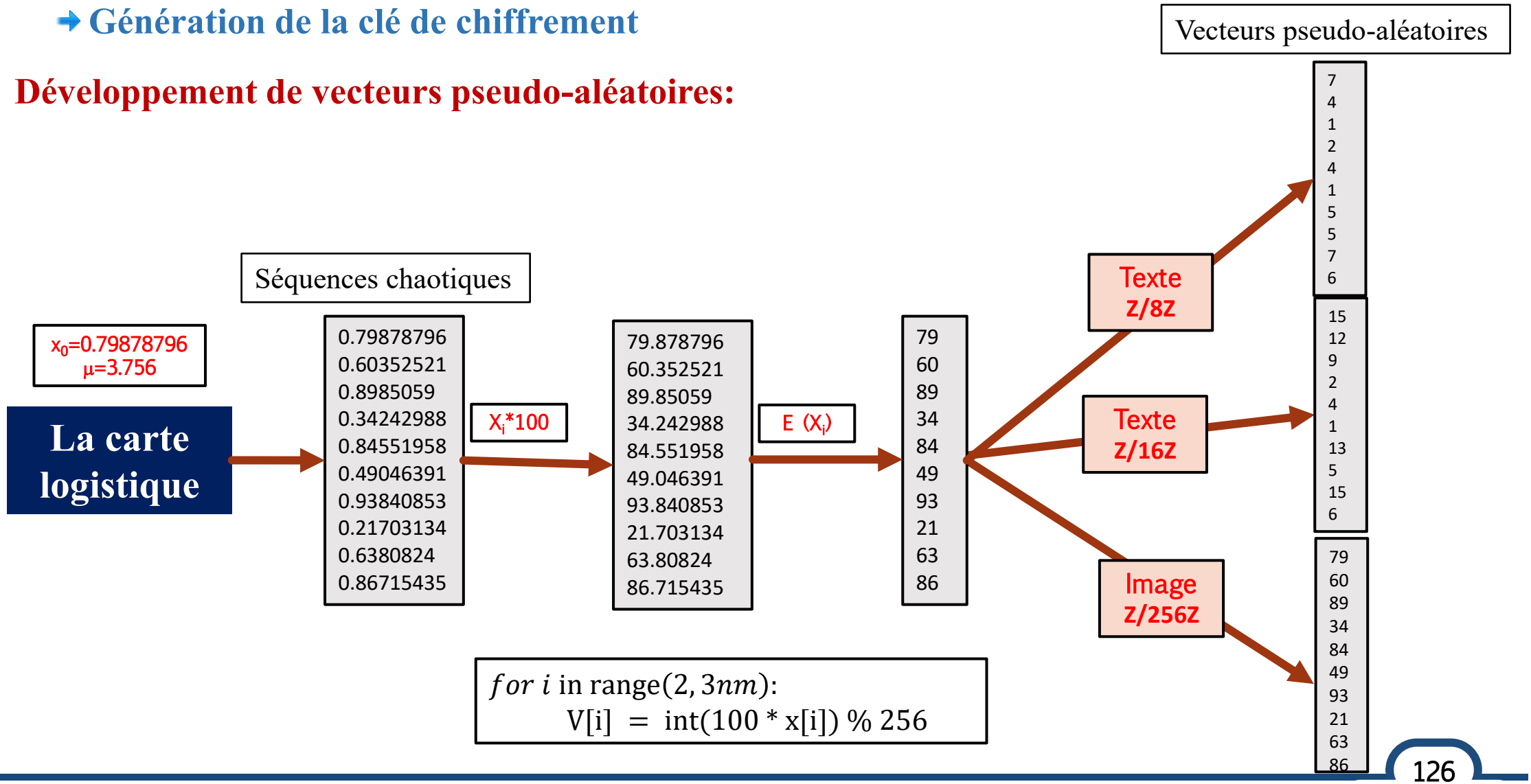
❑ **Développement de séquences chaotiques en utilisant la carte logistique:**

	μ=3.756	μ=3.756	μ=3.757	μ=3.758	μ=3.759	μ=3.760
x_0=0.79878796	0.79878796	0.79878796	0.79878796	0.79878796	0.79878796	0.79878796,
x_1	0.60352521	0.60368594	0.60384666	0.60400739	0.60416811	0.60432884,
x_2	0.8985059	0.8986201	0.89873403	0.8988477	0.89896111	0.89907426,
x_3	0.34242988	0.34217918	0.34192899	0.34167929	0.3414301	0.34118142,
x_4	0.84551958	0.84544777	0.84537593	0.84530405	0.84523215	0.84516023,
x_5	0.49046391	0.49078089	0.49109802	0.49141529	0.49173268	0.49205019,
x_6	0.93840853	0.93868077	0.93895228	0.93922305	0.93949308	0.93976237,
x_7	0.21703134	0.21619229	0.21535462	0.21451835	0.21368349	0.21285006,
x_8	0.6380824	0.63646615	0.63484662	0.63322386	0.63159792	0.62996886,
x_9	0.86715435	0.86905197	0.87093418	0.87280078	0.87465159	0.87648644
x_0=0.79878797	0.79878797	0.79878797	0.79878797,	0.79878797	0.79878797	0.79878797
x_1	0.60368591	0.60368591	0.60384664	0.60400736	0.60416809	0.60432882
x_2	0.89862011	0.89862011	0.89873404	0.89884772	0.89896112	0.89907427
x_3	0.34217913	0.34217913	0.34192893	0.34167924	0.34143005	0.34118136
x_4	0.84544771	0.84544771	0.84537586	0.84530399	0.84523209	0.84516017
x_5	0.49078105	0.49078105	0.49109818	0.49141545	0.49173285	0.49205036
x_6	0.93868078	0.93868078	0.93895229	0.93922306	0.93949309	0.93976238
x_7	0.21619225	0.21619225	0.21535458	0.21451832	0.21368346	0.21285002
x_8	0.63646607	0.63646607	0.63484654	0.63322378	0.63159785	0.62996879
x_9	0.86905205	0.86905205	0.87093425	0.87280085	0.87465166	0.8764865

Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Génération de la clé de chiffrement**

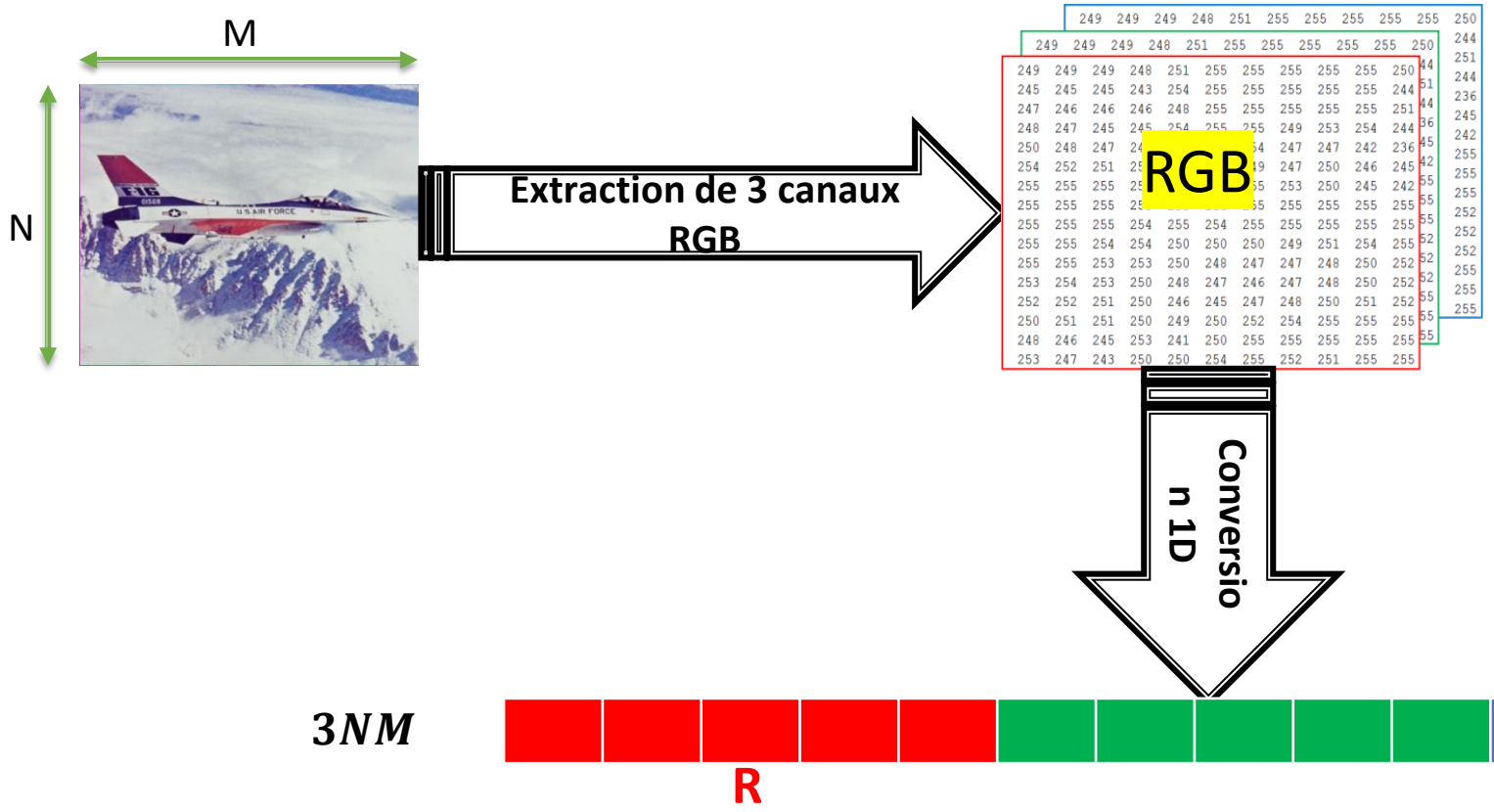
❑ **Développement de vecteurs pseudo-aléatoires:**



Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - ➔ **Vectoriser l'image**

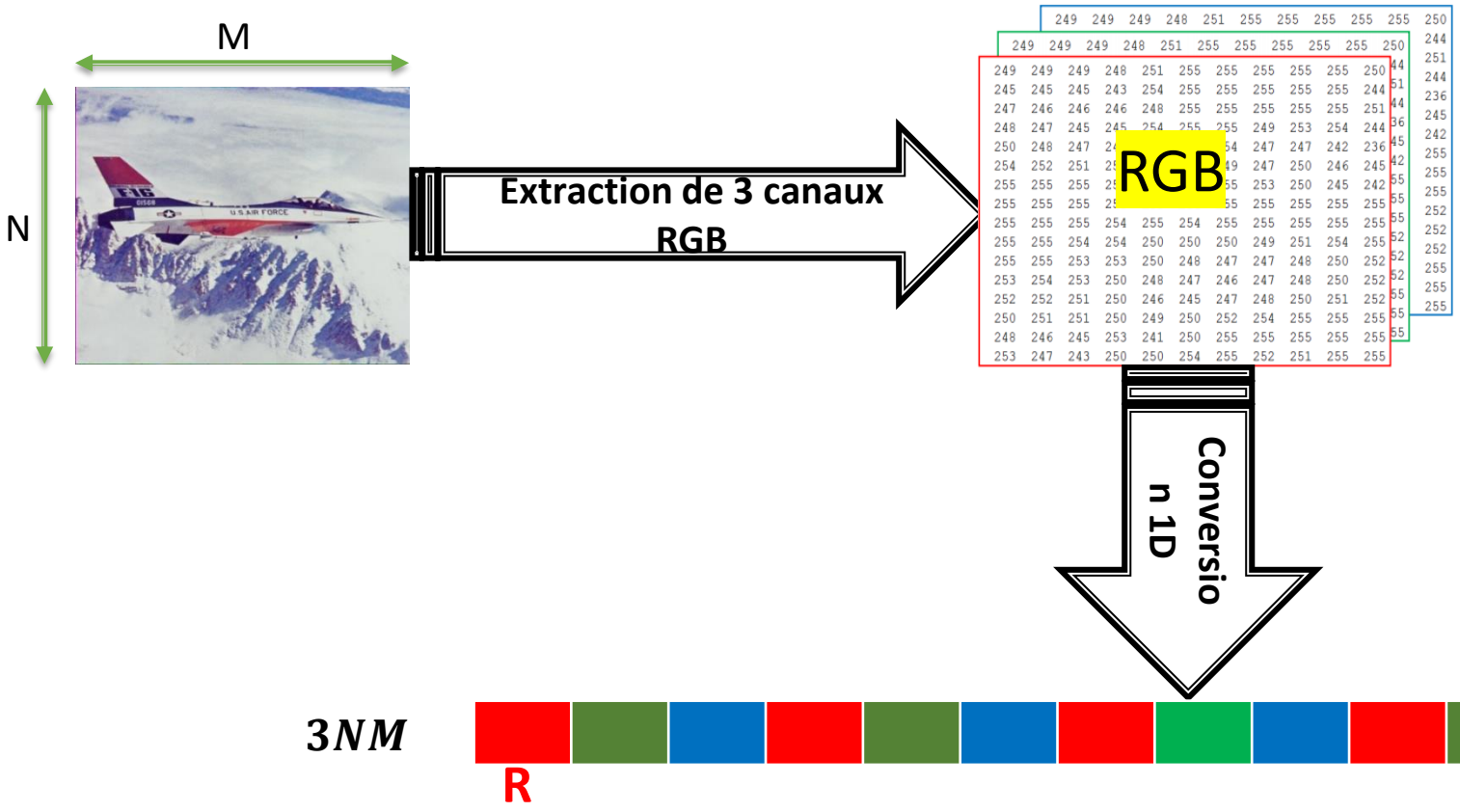
❑ **Lecture en largeur:**



Chiffrement d'images

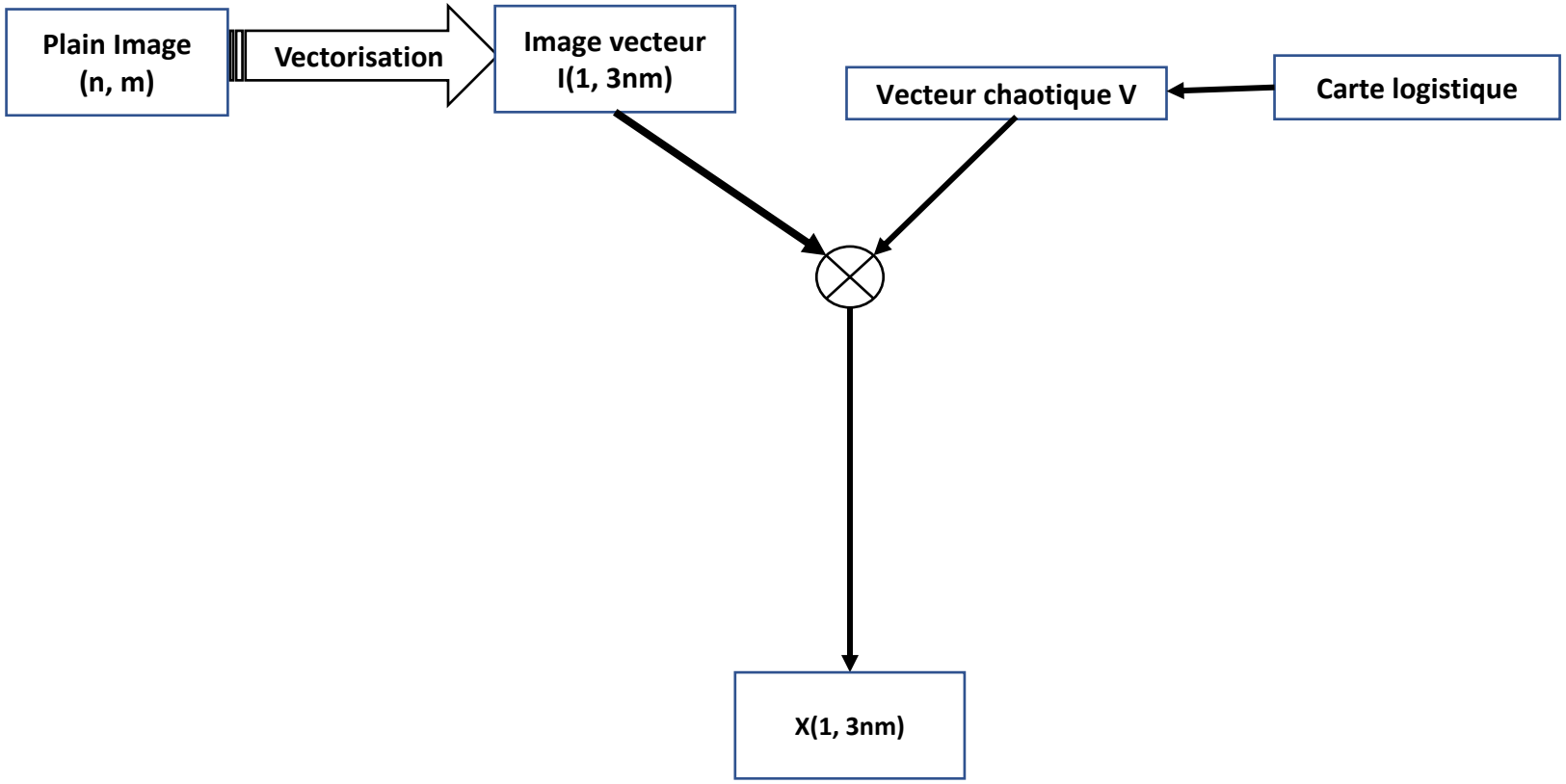
- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Vectoriser l'image**

❑ **Lecture en profondeur:**



Chiffrement d'images

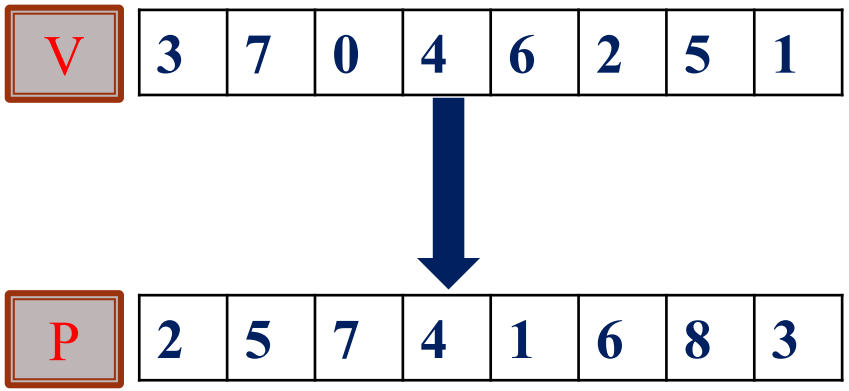
- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **XOR avec un vecteur chaotique**



Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Construction d'une permutation P-Box**

- **Exemple 1: : P-Box de taille 8 (Z/8Z):**




```
# Extraire les 8 premiers éléments
P1= V[:8]
# Trier les 8 premiers éléments
P = np.argsort(P1)
```

Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - ➔ **Construction d'une matrice de substitution S-Box**

❑ **Exemple : Chiffrement avec S-Box de taille 8x8 (Z/8Z):**

P	2	7	5	0	3	6	4	1

S-box 8x8

	0	1	2	3	4	5	6	7
0	2	7	5	0	3	6	4	1
1	3	6	4	1	2	7	5	0
2	5	0	3	6	4	1	2	7
3	1	2	7	5	0	3	6	4
4	6	4	1	2	7	5	0	3
5	0	3	6	4	1	2	7	5
6	4	1	2	7	5	0	3	6
7	7	5	0	3	6	4	1	2

Image Originale

6	4	5	2
3	5	4	1
2	4	2	5
6	3	4	6

Vecteur original	Clé	Vecteur chiffré
6	4	0
4	0	3
5	1	7
2	4	1
3	3	5
5	2	1
4	1	2
1	5	3
2	7	0
4	4	7
2	2	3
5	3	3
6	2	2
3	3	5
4	5	1
6	7	1

Image Chiffrée

0	3	7	1
5	1	2	3
0	7	3	3
2	5	1	1

Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Construction d'une matrice de substitution S-Box**

S-Box de taille 256x256 (Z/256Z):

Valeur du pixel original

Clé de chiffrement K	0	1	2	3	4	…	253	254	255
0									
2									
3									
4									
5									
…									
253									
254									
255									

- La première ligne de notre S-box est initialisée par les éléments de la permutation P.
- Les autres lignes sont générées par un décalage de i de la ligne précédente:

```
import numpy as np
P1 = V[:256]
P = np.argsort(P1)
ST = np.zeros((256, 256), dtype=np.uint8)
ST[0] = P
for i in range(1, 256):          # i = 1 .. 255
    for j in range(256):         # j = 0 .. 255
        ST[i, j] = ST[i-1, (j + i) % 256]
```

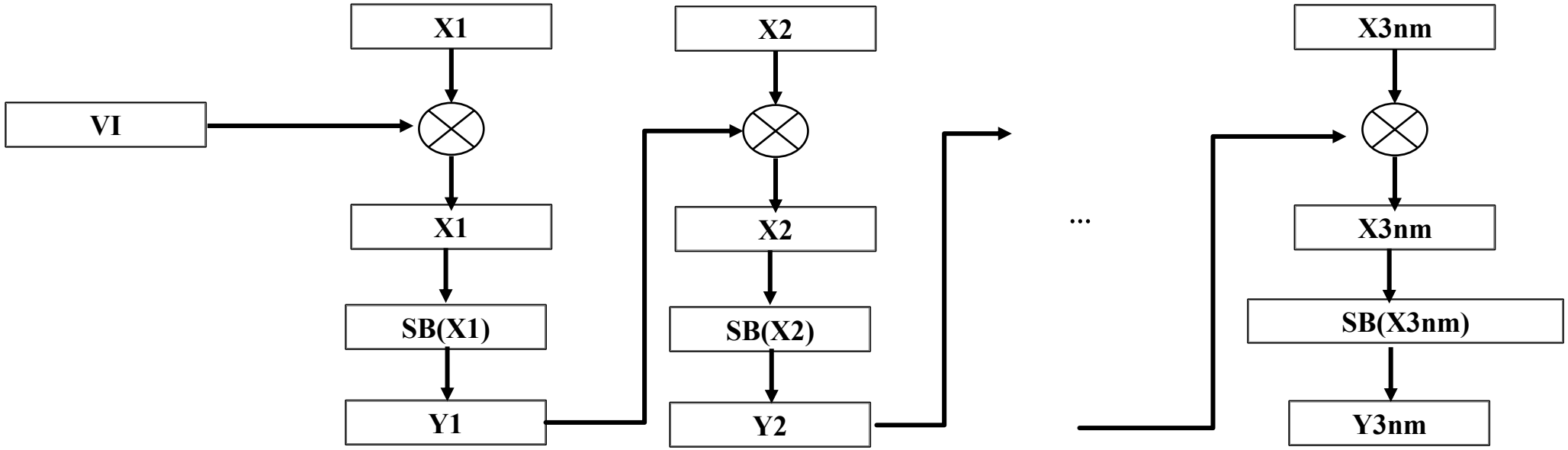
Formule de chiffrement

$$C(X(i)) = ST(V(i), X(i))$$

Chiffrement d'images

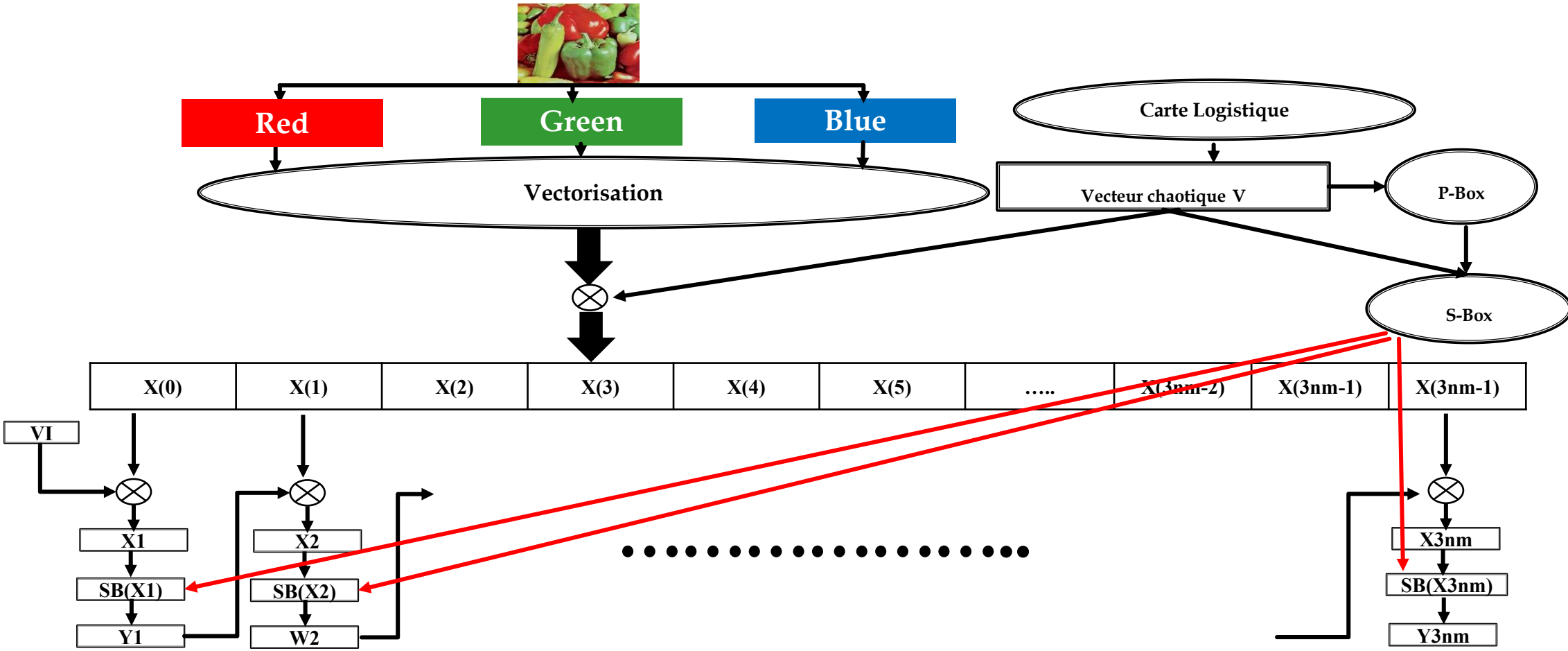
- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Diffusion**

❑ **Chainage (CBC):**



Chiffrement d'images

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Schéma de l'algorithme**



Chiffrement d'images

Chiffrement d'images utilisant la carte logistique, une permutation et une substitution

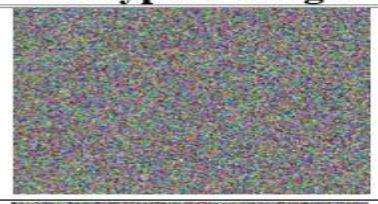
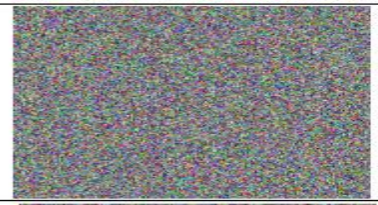
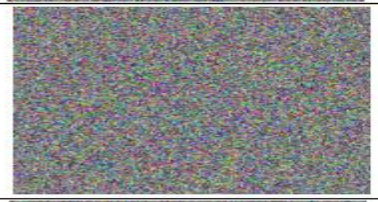
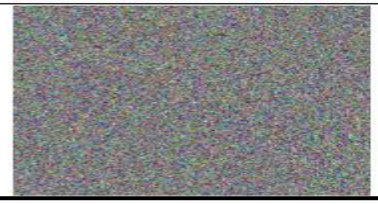
➔ Résultats

- **Test Visuel**
- **Robustesse vis à vis des attaques par force brute**
  - ✓ **Analyse de l'espace clé:** Taille de l'espace clé largement supérieure à 2^{100} bits;
- **Robustesse vis à vis des attaques statistiques:**
  - ✓ **Analyse de l'histogramme** : Uniformité de l'histogramme
  - ✓ **Analyse de corrélation**: coefficients de corrélation ≈ **0**
  - ✓ **Analyse de l'entropie**: au voisinage de la valeur **8.**
- **Robustesse envers les attaques différentielles**
  - ✓ Sensibilité à des légers modifications
    - **NPCR(**Taux de changement de nombre de pixels): $\mathbf{NPCR_{espéré}}$ **≈99,6093 %**
    - **UACI (**Changement moyen de l'intensité unifiée): $\mathbf{UACI_{espéré}}$ **≈ 33,3435%**

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Résultats**

✓ **Test Visuel**

Le teste visuel permet de visualiser si les images chiffrées contiennent des informations sur l'image originale et si les images déchiffrées sont identiques aux originales.

Name	Original Image	Encrypted images	Decrypted images
Female (256x256)			
Couple (256×256)			
House (256×256)			
Splash (512×512)			

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Résultats**

✓ **Analyse de l'espace clé**

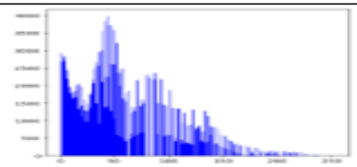
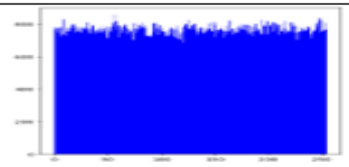
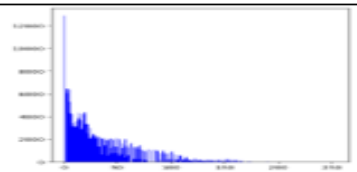
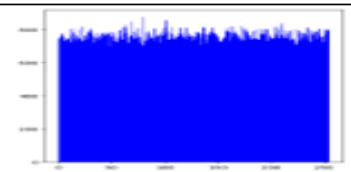
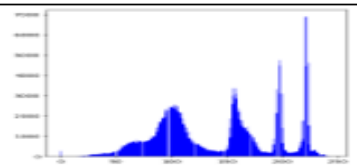
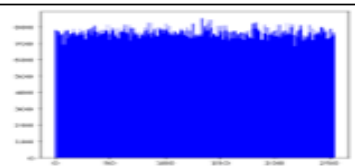
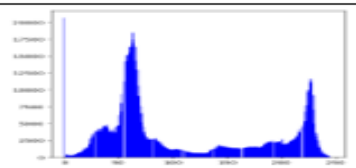
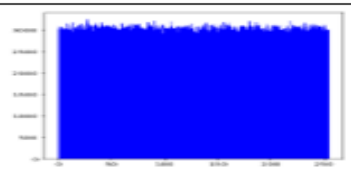
L'espace clé total de notre système est composée d'une condition initiale u_0 et d'un paramètre de contrôle μ_0 (sont des réelles de 64 bits)

$$\approx 2^{128} \gg 2^{100}$$

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Résultats**

✓ **Attaques statistiques**

Analyse d'histogramme: Un histogramme est une représentation graphique de la distribution des pixels dans une image. Pour éviter une attaque statistique, les images chiffrées doivent avoir un histogramme **uniforme**.

Image	Original image Histogram	Encrypted image Histogram
Female		
Couple		
House		
Splash		

- **Chiffrement d'images utilisant la carte logistique, une permutation et une substitution**
  - **Résultats**

✓ **Attaques statistiques**

Analyse de l'entropie: L'entropie est une mesure du caractère aléatoire utilisée pour décrire la texture d'une image simple. Sa valeur est proche de **8** pour les images chiffrées. Mathématiquement, elle est définie comme suit :

$$H(X) = -\sum_{i=0}^{n-1} Pr(X_i) \times log_2\left(\frac{1}{Pr(X_i)}\right)$$

Avec X désigne l'image de test, Xi est la valeur du pixel et Pr(Xi) représente la probabilité de Xi

Image	Image chiffrée
Female	7.9991
House	7.9991
Couple	7.9990
Splash	7.9997

≈8

Chiffrement d'images utilisant la carte logistique, une permutation et une substitution

➔ Résultats

✓ **Attaques statistiques**

Analyse de la corrélation: La corrélation détermine l'indépendance des pixels adjacents. Elle est définie comme suit :

$$C_{x\,y} = \frac{\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i - E(x)\right)^2 \times \sum_{i=1}^{N}\left(y_i - E(y)\right)^2}}$$

Avec E(x) est la moyenne des valeurs de pixels de l'image.

Image	Horizontal	Vertical	Diagonale
Female	-0.0008	0.0007	-0.0002
House	0.0002	-0.0026	0.0009
Couple	0.0014	0.0015	-0.0061
Splash	-0.0005	0.0008	-0.0001

≈0

Chiffrement d'images utilisant la carte logistique, une permutation et une substitution

➔ Résultats

✓ **Attaques différentielles**

- NPCR (Number of Pixels Change Rate) : le taux de pixels différents entre les deux images chiffrées qui différent d'un pixel. (**NPCR > 99.60 %**)
- UACI (Unified Average Changing Intensity) : la différence de l'intensité moyenne. (**UACI > 33,34 %**)

$$NPCR = \frac{\sum_1^w \sum_1^h D_{i\,j}}{w * h} \times 100\ (\%)$$

D(i, j) est donné comme suit :

$$D_{i\,j} = \begin{cases} 1 & si \quad C1_{i\,j} \neq C2_{i\,j} \\ 0 & sinon \end{cases}$$

$$UACI = \frac{1}{w \times h} \frac{\sum_{i\,j} |C1_{i\,j} - C2_{i\,j}|}{255} \times 100\ \ (\%)$$

C1 représente l'image cryptée de l'image originale et C2 l'image cryptée de l'image originale modifiée.

Name	NPCR %	UACI %
Female	99.75	33.44
House	99.76	33.47
Couple	99.73	33.52
Splash	99.73	33.48

NPCR > 99,6% **UACI > 33,34%**