Utilisation d'itérations chaotiques pour la génération de nombres pseudo-aléatoires

Christophe Guyeux

Institut FEMTO-ST, UMR 6174 CNRS
Département DISC – Équipe AND
Université de Franche-Comté

2 juillet 2013

Plan

1. Introduction

2. Chaos et machines à états finis

3. Applications aux PRNGs

4. Autres applications

Les réseaux envisagés

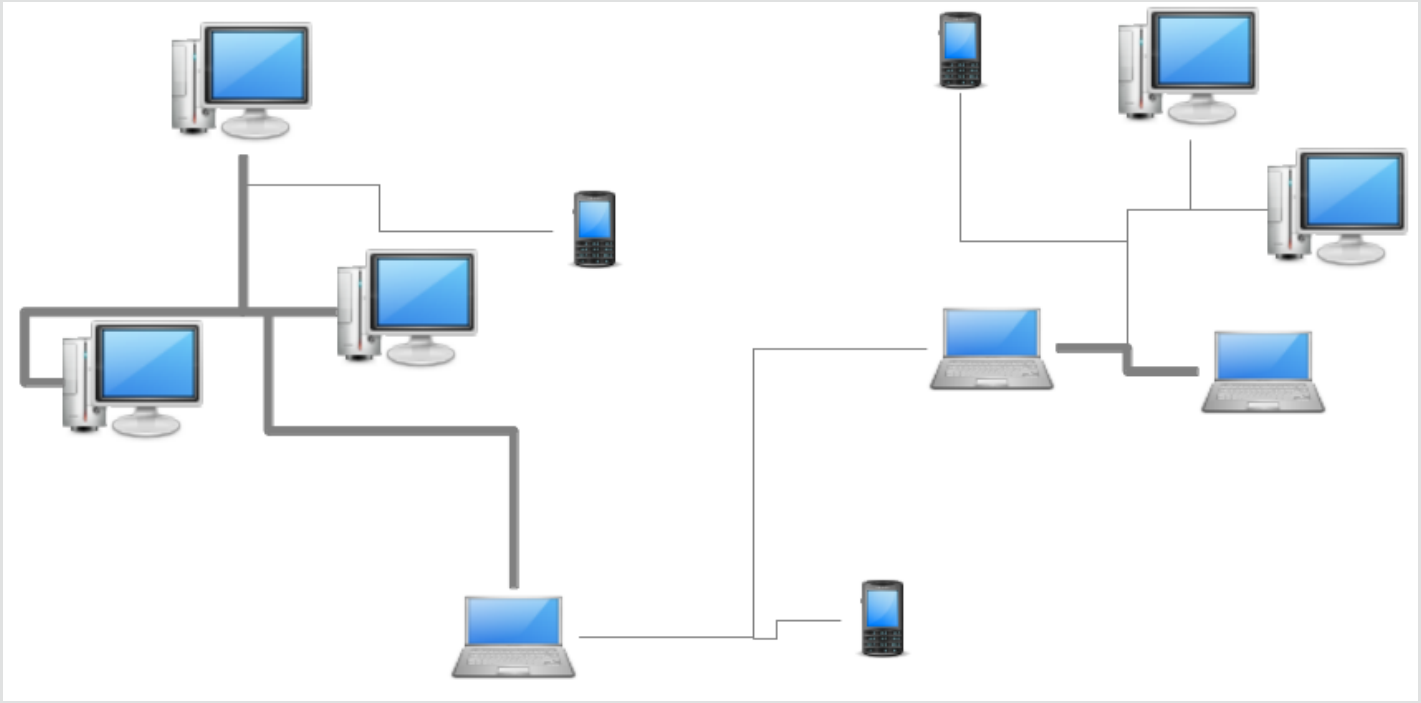


FIGURE : Architecture distribuée asynchrone

Les systèmes itératifs

Les systèmes itératifs en toute généralité

La formulation suivante englobe tous les modes d'itérations imaginables :

$$\begin{cases} x^0 \in \mathcal{X} \\ x^{n+1} = f^n(x^0, \ldots, x^n) \end{cases}$$

où $f^n : \mathcal{X}^{n+1} \to \mathcal{X}$

Les systèmes itératifs

Les systèmes itératifs en toute généralité

La formulation suivante englobe tous les modes d'itérations imaginables :

$$\begin{cases} x^0 \in \mathcal{X} \\ x^{n+1} = f^n(x^0, \ldots, x^n) \end{cases}$$

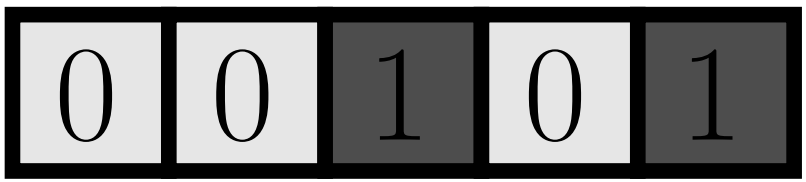
où $f^n : \mathcal{X}^{n+1} \to \mathcal{X}$

Différents modes d'itérations : séries, parallèles, chaotiques, asynchrones...

Mode d'itération séries

Etat du système

État initial (t=0)



Fonction de mise à jour

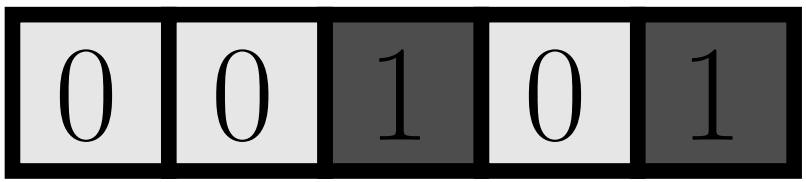
$$f : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

État initial (t=0)



Fonction de mise à jour

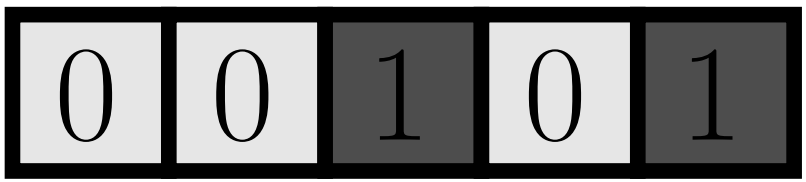
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

État initial (t=0)



Fonction de mise à jour

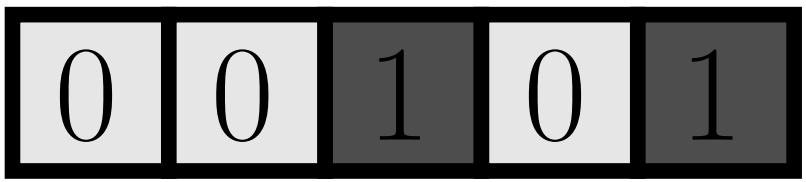
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

État initial (t=0)



Fonction de mise à jour

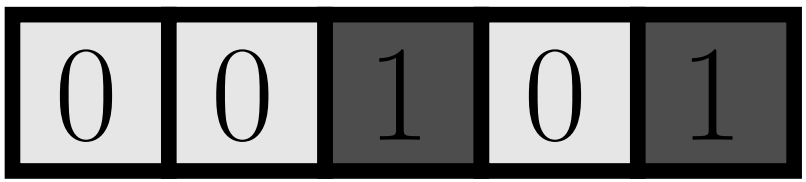
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

État initial (t=0)



Fonction de mise à jour

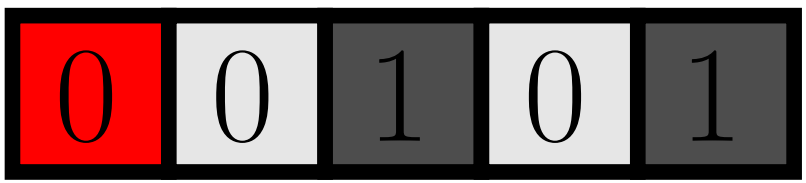
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

État initial (t=0)



Fonction de mise à jour

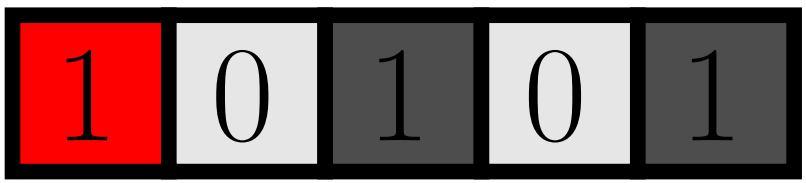
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 1)



Fonction de mise à jour

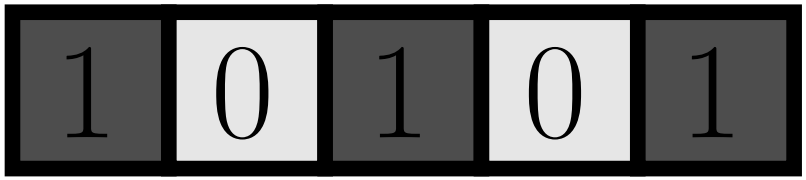
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 1)



Fonction de mise à jour

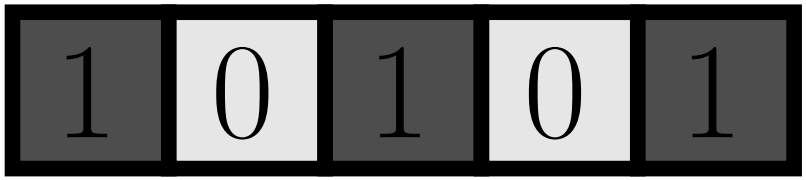
$$f : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1 . x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 1)



Fonction de mise à jour

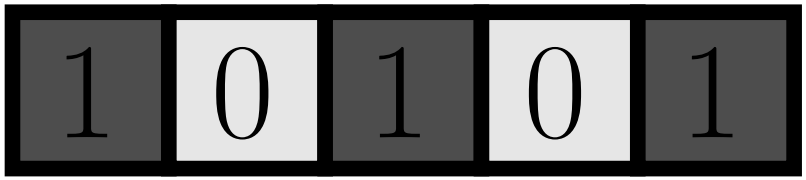
$$f : \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1 . x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 1)



Fonction de mise à jour

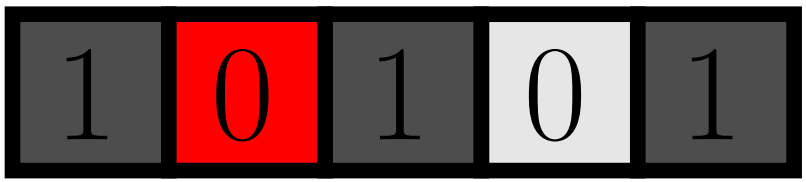
$$f : \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 1)



Fonction de mise à jour

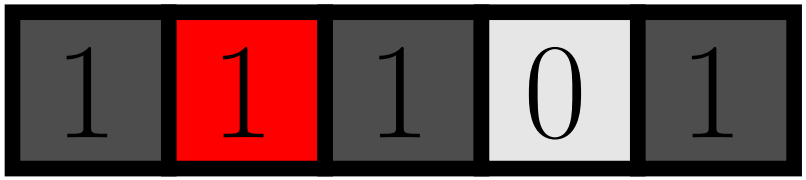
$$f : \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 2)



Fonction de mise à jour

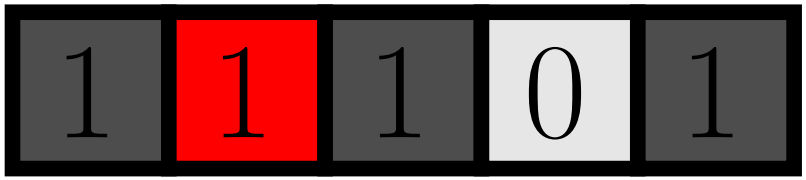
$$f : \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 2)



Fonction de mise à jour

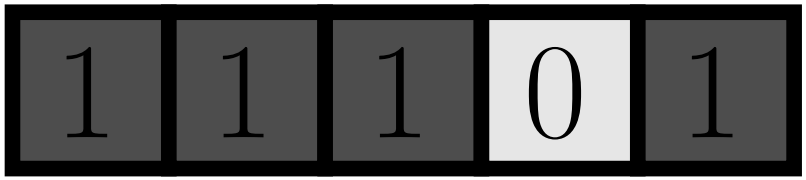
$$f : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1 . x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 2)



Fonction de mise à jour

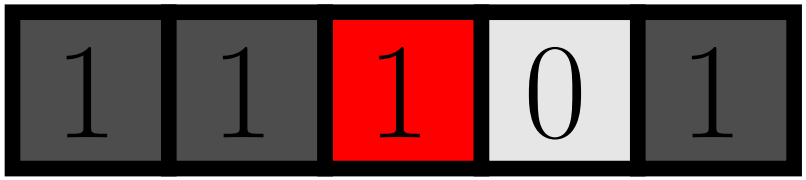
$$f : \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 2)



Fonction de mise à jour

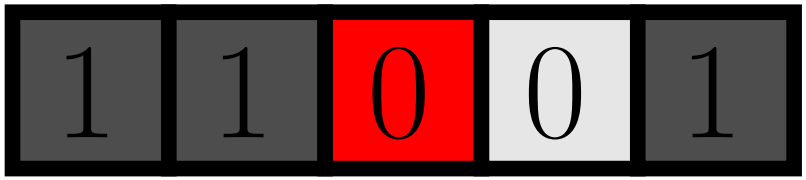
$$f : \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération séries

Etat du système

(t = 3)



Fonction de mise à jour

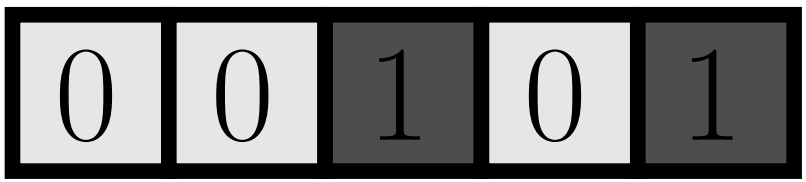
$$f : \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Cellule à mettre à jour : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1...

Mode d'itération parallèle

Etat du système

État initial (t=0)



Fonction de mise à jour

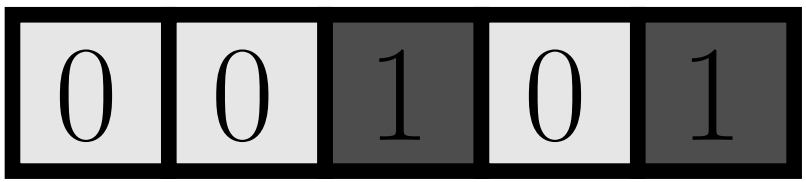
$$f : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1 . x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

État initial (t=0)



Fonction de mise à jour

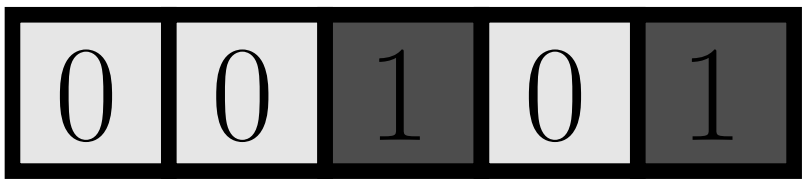
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

État initial (t=0)



Fonction de mise à jour

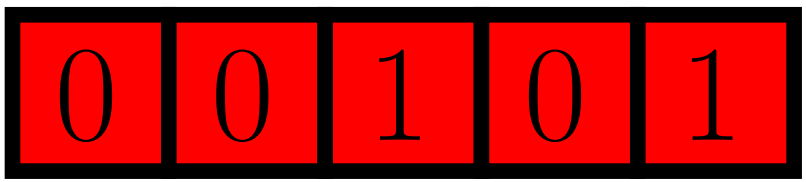
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

État initial (t=0)



Fonction de mise à jour

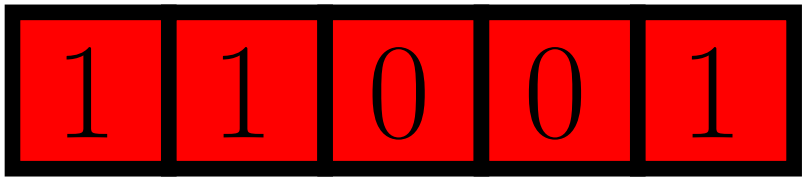
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

(t = 1)



Fonction de mise à jour

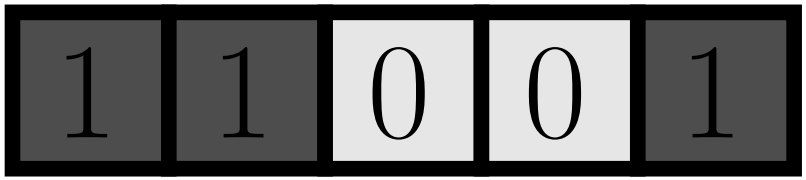
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

(t = 1)



Fonction de mise à jour

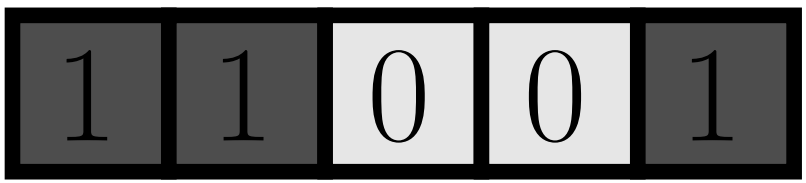
$$f : \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

(t = 1)



Fonction de mise à jour

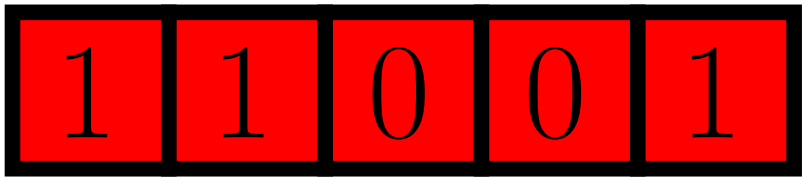
$$f : \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

(t = 1)



Fonction de mise à jour

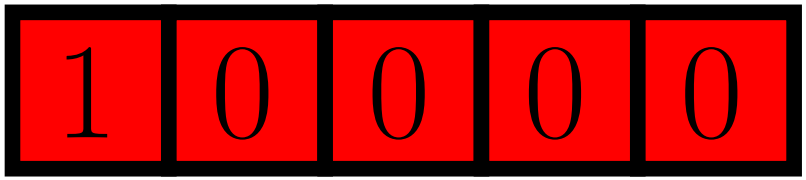
$$f : \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

(t = 2)



Fonction de mise à jour

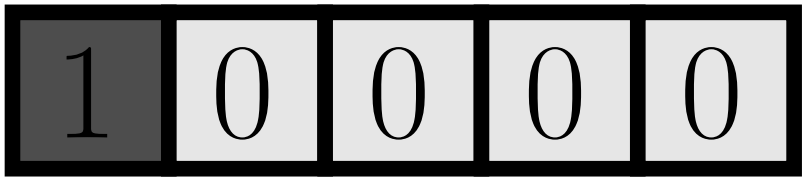
$$f : \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération parallèle

Etat du système

(t = 2)



Fonction de mise à jour

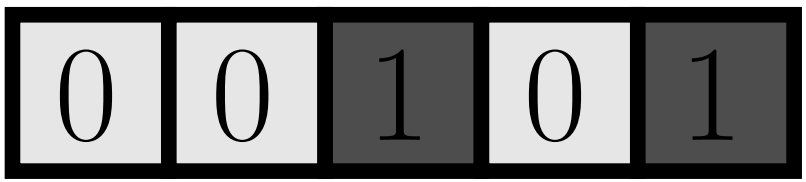
$$f : \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 12345, 12345, 12345...

Mode d'itération chaotique

Etat du système

État initial (t=0)



Fonction de mise à jour

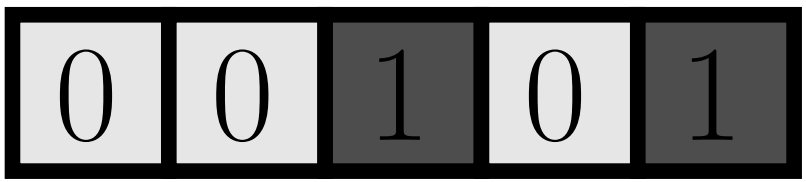
$$f : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1 . x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

État initial (t=0)



Fonction de mise à jour

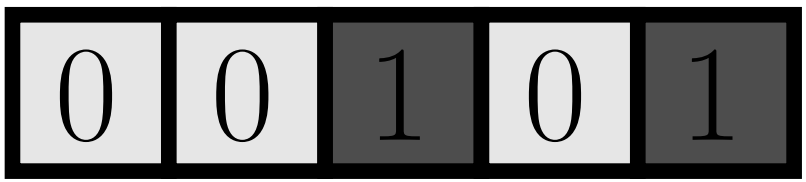
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

État initial (t=0)



Fonction de mise à jour

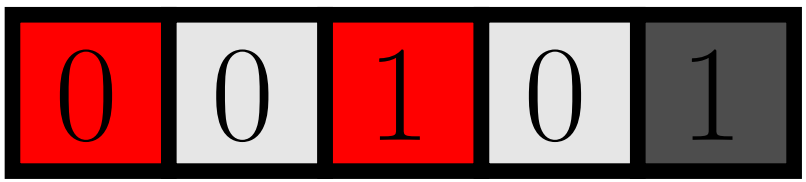
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

État initial (t=0)



Fonction de mise à jour

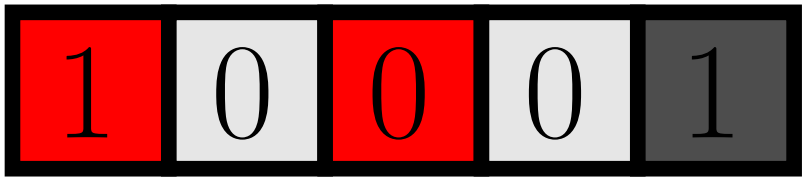
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 1)



Fonction de mise à jour

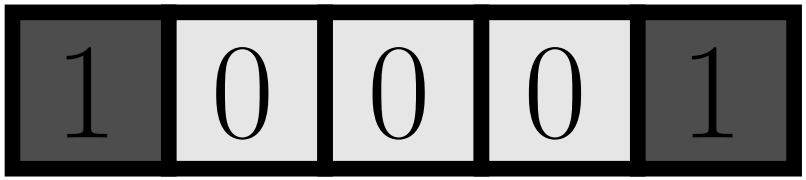
$$f : \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 1)



Fonction de mise à jour

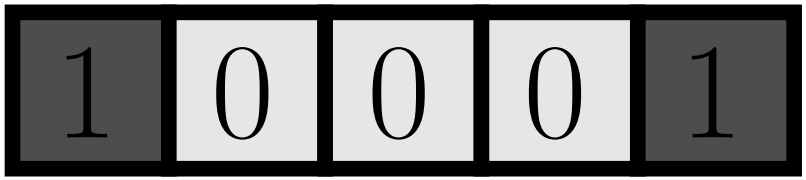
$$f : \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 1)



Fonction de mise à jour

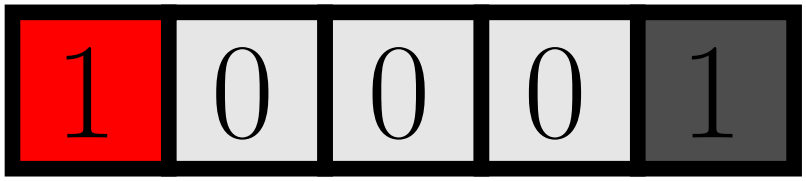
$$f : \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 1)



Fonction de mise à jour

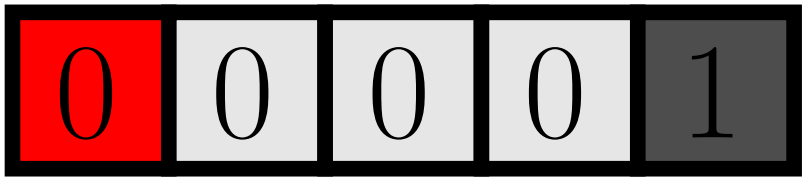
$$f : \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 2)



Fonction de mise à jour

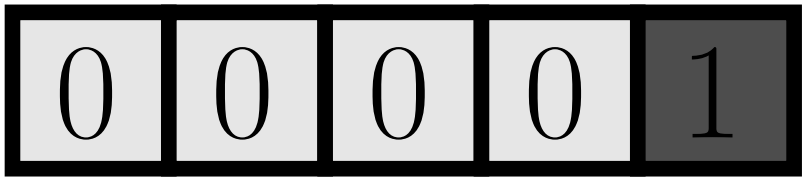
$$f : \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 2)



Fonction de mise à jour

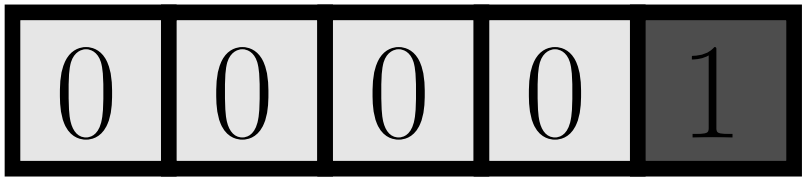
$$f : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 2)



Fonction de mise à jour

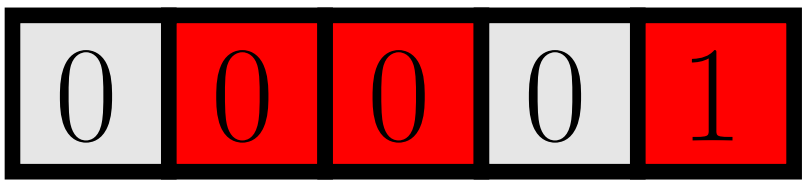
$$f : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 2)



Fonction de mise à jour

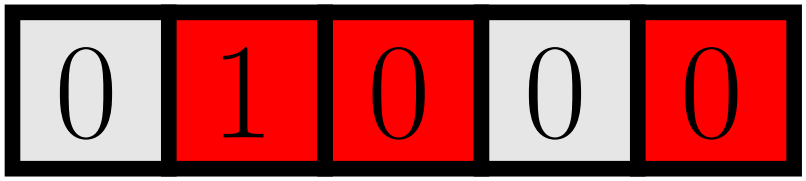
$$f : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 3)



Fonction de mise à jour

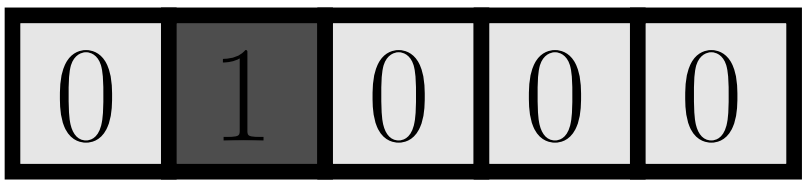
$$f : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Mode d'itération chaotique

Etat du système

(t = 3)



Fonction de mise à jour

$$f : \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \overline{x_1} + x_2 \\ \overline{x_2} \\ x_4 \\ x_1.x_3 \\ x_3 + \overline{x_5} \end{pmatrix}$$

Cellules à mettre à jour : 13, 1, 235, 12345, ...

Les « itérations chaotiques »

Définition (Itérations chaotiques)

Soient $f : \mathbb{B}^N \longrightarrow \mathbb{B}^N$ et $S \subset \mathcal{P}(\llbracket 1, N \rrbracket)^{\mathbb{N}}$. Les *itérations chaotiques* sont :

$$\begin{cases} x^0 \in \mathbb{B}^N \\ \forall n \in \mathbb{N}^*, \forall i \in \llbracket 1; N \rrbracket, x_i^n = \begin{cases} x_i^{n-1} & \text{si } i \notin S^n \\ f(x^{n-1})_i & \text{si } i \in S^n \end{cases} \end{cases}$$

Les « itérations chaotiques »

Définition (Itérations chaotiques)

Soient $f : \mathbb{B}^N \longrightarrow \mathbb{B}^N$ et $S \subset \mathcal{P}([\![1, N]\!])^{\mathbb{N}}$. Les *itérations chaotiques* sont :

$$\begin{cases} x^0 \in \mathbb{B}^N \\ \forall n \in \mathbb{N}^*, \forall i \in [\![1; N]\!], x_i^n = \begin{cases} x_i^{n-1} & \text{si } i \notin S^n \\ f(x^{n-1})_i & \text{si } i \in S^n \end{cases} \end{cases}$$

Itérations chaotiques et théorie du chaos : a priori, rien à voir.

Les « itérations chaotiques »

Définition (Itérations chaotiques)

Soient $f : \mathbb{B}^N \longrightarrow \mathbb{B}^N$ et $S \subset \mathcal{P}(\llbracket 1, N \rrbracket)^{\mathbb{N}}$. Les *itérations chaotiques* sont :

$$\begin{cases} x^0 \in \mathbb{B}^N \\ \forall n \in \mathbb{N}^*, \forall i \in \llbracket 1; N \rrbracket, x_i^n = \begin{cases} x_i^{n-1} & \text{si } i \notin S^n \\ f(x^{n-1})_i & \text{si } i \in S^n \end{cases} \end{cases}$$

Itérations chaotiques et théorie du chaos : a priori, rien à voir.
Y a-t-il un lien ?

Les « itérations chaotiques »

Définition (Itérations chaotiques)

Soient $f : \mathbb{B}^N \longrightarrow \mathbb{B}^N$ et $S \subset \mathcal{P}(\llbracket 1, N \rrbracket)^{\mathbb{N}}$. Les *itérations chaotiques* sont :

$$\begin{cases} x^0 \in \mathbb{B}^N \\ \forall n \in \mathbb{N}^*, \forall i \in \llbracket 1; N \rrbracket, x_i^n = \begin{cases} x_i^{n-1} & \text{si } i \notin S^n \\ f(x^{n-1})_i & \text{si } i \in S^n \end{cases} \end{cases}$$

Itérations chaotiques et théorie du chaos : a priori, rien à voir.
Y a-t-il un lien ? Pour quoi faire ?

Quelques verrous

Liste de quelques verrous

- Chaos ?
- Systèmes itératifs et chaos ?
- Mathématiques vs informatique
- « Ceux qui font de la crypto par chaos n'ont rien compris » (Anonyme)

Notre approche

Ce que nous proposons

- Approche sérieuse du chaos (topologie)
- Expliquer à quoi il sert
- Le chaos peut être un plus, mais ne peut suffire :
  1. Prendre des algorithmes prouvés sûr
  2. Les traiter pour rajouter du chaos
  3. Sans perte de sécurité

Des propriétés topologiques

Le contexte historique

Les suites récurrentes

$u^0 \in \mathbb{R}, u^{n+1} = f(u^n)$, avec f continue

- Résolution d'équations (Newton)
- Discrétisation d'équations différentielles...

Convergence

- f monotone
- Applications contractantes
- Coppel : Pas de 2-cycle $\Rightarrow$ convergence

3-cycle implique chaos

Period Three Implies Chaos (Li et Yorke, 1975)

S'il y a un point de période 3, alors il y a un point de n'importe quelle période

3-cycle implique chaos

Period Three Implies Chaos (Li et Yorke, 1975)

S'il y a un point de période 3, alors il y a un point de n'importe quelle période

→ Désordre lié à la multiplicité des périodes

Non-convergence des IC

Théorème (Condition nécessaire de non-convergence)

Si les itérations chaotiques $\left(f,(x^0,S)\right)$ sont non convergentes, alors :

- soit f n'est pas contractante,
- soit S n'est pas pseudo-périodique (complète).

Non-convergence des IC

Théorème (Condition nécessaire de non-convergence)

Si les itérations chaotiques $\left(f,(x^0,S)\right)$ sont non convergentes, alors :

- soit f n'est pas contractante,
- soit S n'est pas pseudo-périodique (complète).

Quelle quantité de désordre ?

Présentation du problème

MATHS DISCRÈTES	TOPOLOGIE MATHÉMATIQUE
$f : \mathbb{B}^N \to \mathbb{B}^N$	$(\mathcal{X}, \tau)$ espace topologique $f : \mathcal{X} \to \mathcal{X}$ continue pour τ
$S \in \mathcal{S} = [\![1, N]\!]^{\mathbb{N}}$ $x^0 \in \mathbb{B}^{\mathbb{N}}$	$x^0 \in \mathcal{X}$
$x_i^{n+1} = \begin{cases} x_i^n & \text{si } i \neq S^n \\ f(x^n)_i & \text{si } i = S^n \end{cases}$	$\forall n \in \mathbb{N}, x^{n+1} = f(x^n)$

Modélisation des IC

Modélisation des IC en topologie

Soit $\mathcal{X} = [\![1; \mathrm{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathrm{N}}$, et $G_f(S, E) = (\sigma(S), F_f(i(S), E))$.
On modélise les itérations chaotiques unaires $\left(f, (S, x^0)\right)$ par le système dynamique discret :

$$\begin{cases} X^0 = (S, x^0) \in \mathcal{X}, \\ \forall k \in \mathbb{N}, X^{k+1} = G_f(X^k). \end{cases}$$

Modélisation des IC

Modélisation des IC en topologie

Soit $\mathcal{X} = [\![1;\mathrm{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathrm{N}}$, et $G_f(S,E) = (\sigma(S), F_f(i(S),E))$.
On modélise les itérations chaotiques unaires $\left(f,(S,x^0)\right)$ par le système dynamique discret :

$$\begin{cases} X^0 = (S,x^0) \in \mathcal{X}, \\ \forall k \in \mathbb{N}, X^{k+1} = G_f(X^k). \end{cases}$$

On peut donc étudier leur désordre topologique.

Métrique et continuité

Distance sur $\mathcal{X}$:

$$d((S,E);(\check{S};\check{E})) = d_e(E,\check{E}) + d_s(S,\check{S})$$

où $d_e(E,\check{E}) = \sum_{k=1}^{N} \delta(E_k,\check{E}_k)$, et $d_s(S,\check{S}) = \frac{9}{N}\sum_{k=1}^{\infty} \frac{|S^k - \check{S}^k|}{10^k}$.

Théorème

La fonction $G_f : (\mathcal{X},d) \to (\mathcal{X},d)$ est continue.

Étude de $(\mathcal{X}, d)$

Propriétés de $(\mathcal{X}, d)$

- $\mathcal{X}$ est infini indénombrable
- $(\mathcal{X}, d)$ est un espace métrique compact, complet et parfait

Étude de G_{f_0}

G_{f_0} est surjective, mais pas injective

$\Rightarrow (\mathcal{X}, G_{f_0})$ pas réversible.

Les approches Devaney et Knudsen

3 propriétés pour de l'imprévisibilité

1. *Indécomposabilité.* On ne doit pas pouvoir simplifier le système
   - Impossible de diviser pour régner
   - Des orbites doivent visiter tout l'espace
2. *Élément de régularité.*
   - Contrecarre l'effet précédent
   - Des points proches *peuvent* se comporter complètement différemment
3. *Sensibilité.* Des points proches *peuvent* finir éloignés

Exemple : définition de Devaney

1. *Transitivité :* Pour chaque couple d'ouverts non vides $A, B \subset \mathcal{X}$, il existe $k \in \mathbb{N}$ tel que $f^{(k)}(A) \cap B \neq \emptyset$
2. *Régularité :* Les points périodiques sont denses
3. *Sensibilité aux conditions initiales :* Il existe $\varepsilon > 0$ tel que $\forall x \in \mathcal{X}, \forall \delta > 0, \exists y \in \mathcal{X}, \exists n \in \mathbb{N}, d(x, y) < \delta$ et $d(f^{(n)}(x), f^{(n)}(y)) \geqslant \varepsilon$

Systèmes intrinsèquement compliqués

Définitions de l'indécomposabilité

- *Indécomposable* : pas la réunion de deux parties non vides, fermées et t.q. $f(A) \subset A$
- *Totalement transitive* : $\forall n \geqslant 1$, l'application composée $f^{(n)}$ est transitive.
- *Fortement transitif* : $\forall x, y \in \mathcal{X}, \forall r > 0, \exists z \in B(x, r), \exists n \in \mathbb{N}, f^{(n)}(z) = y$.
- *Topologiquement mélangeant* : pour toute paire d'ouverts disjoints et non vides U et V, il existe $n_0 \in \mathbb{N}$ tel que $\forall n \geqslant n_0, f^{(n)}(U) \cap V \neq \varnothing$.

Stabilité et expansivité

Définitions de la sensibilité

- $(\mathcal{X}, f)$ est *instable* si tous ses points le sont : $\forall x \in \mathcal{X}$, $\exists \varepsilon > 0, \forall \delta > 0, \exists y \in \mathcal{X}, \exists n \in \mathbb{N}, d(x,y) < \delta$ et $d(f^{(n)}(x), f^{(n)}(y)) \geqslant \varepsilon$
- $(\mathcal{X}, f)$ est *expansif* si $\exists \varepsilon > 0, \forall x \neq y, \exists n \in \mathbb{N}$, $d(f^{(n)}(x), f^{(n)}(y)) \geqslant \varepsilon$

Selon Li et Yorke

Définitions

Couple de Li-Yorke. (x, y) en est un quand :
$\limsup_{n\to+\infty} d(f^{(n)}(x), f^{(n)}(y)) > 0$ et
$\liminf_{n\to+\infty} d(f^{(n)}(x), f^{(n)}(y)) = 0$.

Ensemble brouillé. $B \subset \mathcal{X}$ en est un si tout couple de points distincts de B est de Li-Yorke.

Systèmes de Li-Yorke. $\mathcal{X}$ est compact et contient un ensemble brouillé indénombrable.

Approche entropie topologique

Entropie topologique

- $x, y \in \mathcal{X}$ sont ε*−séparés en temps* n s'il existe $k \leqslant n$ tel que $d\left(f^{(k)}(x), f^{(k)}(y)\right) > \varepsilon$.
- Les ensembles (n, ε)−séparé sont des ensembles de points qui seront tous ε−séparés en temps n
- $s_n(\varepsilon, Y)$: cardinal maximal d'un ensemble (n, ε)−séparé

$$h_{top}(\mathcal{X}, f) = \lim_{\varepsilon \to 0} \left[\limsup_{n \to +\infty} \frac{1}{n} \log s_n(\varepsilon, \mathcal{X}) \right]$$

Exposant de Lyapunov

L'exposant de Lyapunov

$$\lambda(x^0) = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| f' \left(x^{i-1} \right) \right|$$

Il doit être positif pour multiplier les erreurs

IC et propriété de Devaney

Théorème

G_{f_0} est régulier et transitif (Devaney).
Sa sensibilité est $\geqslant$ N – 1.

IC et propriété de Devaney

Théorème

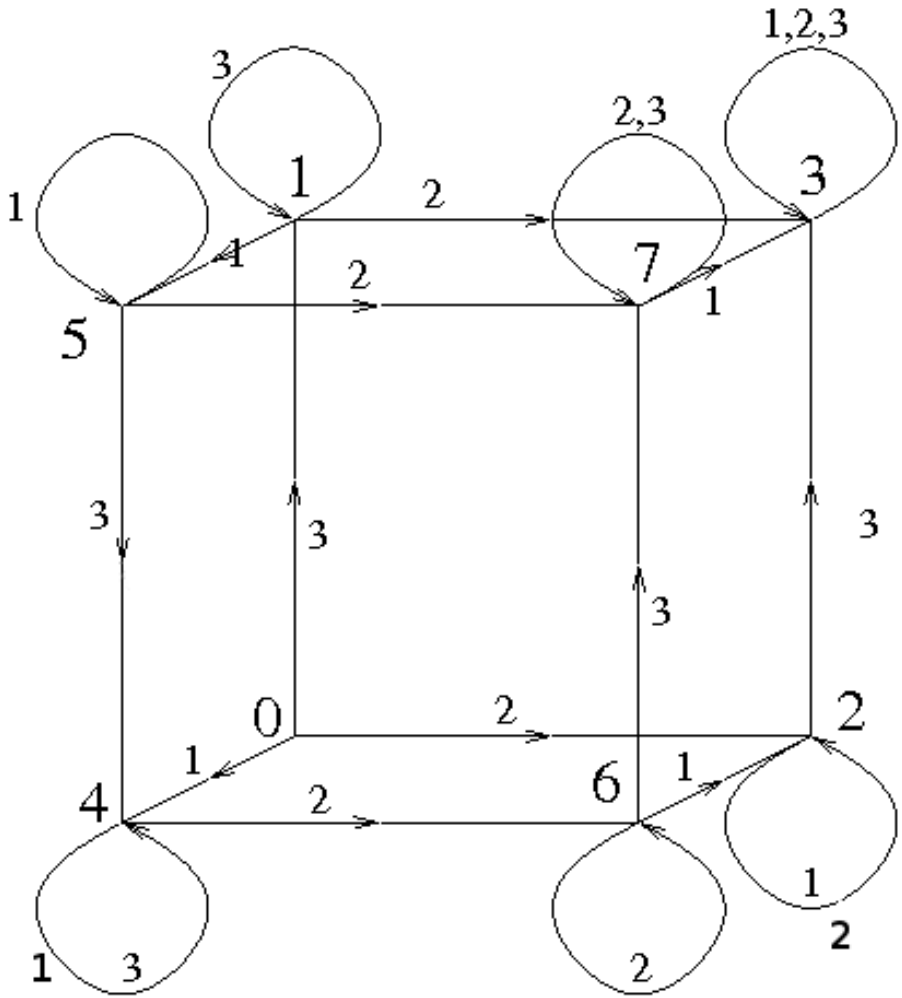
G_{f_0} est régulier et transitif (Devaney).
Sa sensibilité est $\geqslant$ N – 1.

Question

f_0 est-elle la seule fonction dont le système itératif vérifie la condition de Devaney ?

Pour y répondre, nous avons utilisé le graphe de tous les possibles par itérations chaotiques : le GTPIC.

Graphe de tous les possibles par IC



Nombre de fonctions imprévisibles

Caractérisation des IC imprévisibles selon Devaney

G_f vérifie l'hypothèse de Devaney ⇔ Son graphe des possibles est fortement connexe.

⇒ Il y a $\left(2^N\right)^{2^N}$ IC chaotiques.

Condition suffisante de chaoticité

Théorème

Soit f une fonction de $\mathbb{B}^n$ dans lui-même telle que :

1. Le graphe de connexion $G(f)$ n'a pas de cycle de longueur au moins 2 ;
2. Chaque arête de $G(f)$ ayant une boucle positive a aussi une boucle négative ;
3. Chaque arête de $G(f)$ est joignable à partir d'un noeud ayant une boucle négative.

Alors $\Gamma(f)$ est fortement connexe.

Etude topologique

Etude topologique des ICs

- $\forall f \in \mathcal{C}$, $Per\,(G_f)$ est infini dénombrable, G_f est fortement transitive, est chaotique selon Knudsen,
- $\left(\mathcal{X}, G_{f_0}\right)$ est topologiquement mélangeant, expansif (constante 1), est chaotique selon Li-Yorke, a une entropie topologique infinie, un exposant de Lyapunov de $ln(\mathrm{N})$
- Indécomposabilité, instabilité, chaos de Wiggins, de la multiplicité des périodes...

Une semi-conjugaison topologique

Une semi-conjugaison topologique

IC G_{f_0} sur $\mathcal{X}$ = IC g sur $\mathbb{R}$:

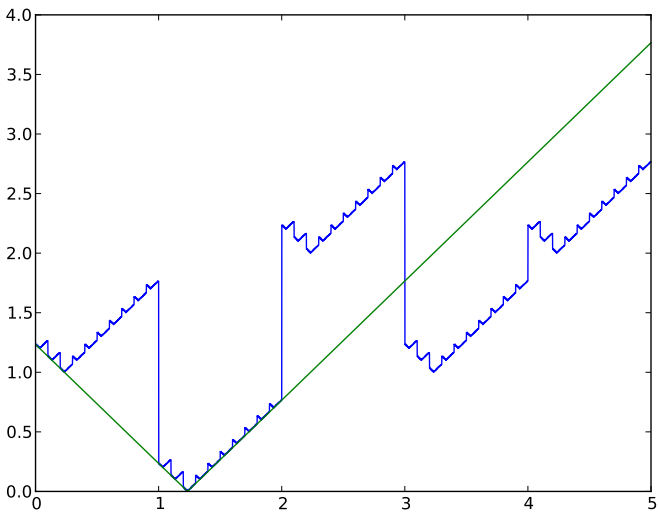
$$\begin{array}{ccc} \left(\mathcal{S}_{10} \times \mathbb{B}^{10}, d \right) & \xrightarrow{G_{f_0}} & \left(\mathcal{S}_{10} \times \mathbb{B}^{10}, d \right) \\ \Big\downarrow \varphi & & \Big\downarrow \varphi \\ \left(\left[0, 2^{10} \right[, D \right) & \xrightarrow[g]{} & \left(\left[0, 2^{10} \right[, D \right) \end{array}$$

1. Prendre la première décimale d de $x \in \left[0, 2^{10} \right[$
2. Nier le bit numéro d de $E(x)$
3. Supprimer d

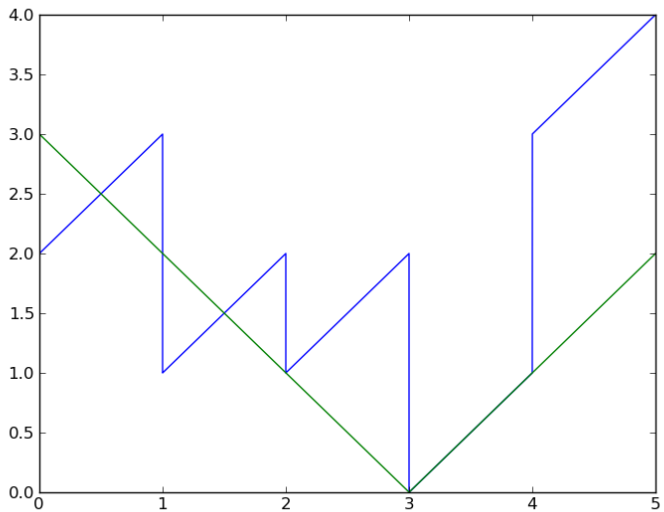
Comparaison des distances

Comparaison de distances

D est plus fine que la distance euclidienne.



(a) Application $x \rightarrow dist(x; 1,234)$.



(b) Application $x \rightarrow dist(x; 3)$.

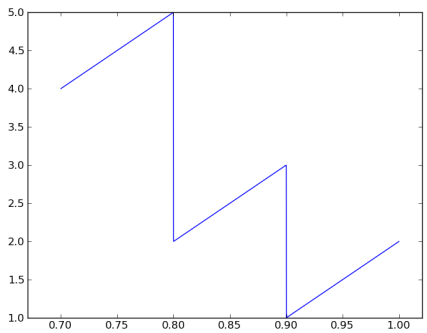
Étude des ICs sur $\mathbb{R}$

Analyse des itérations chaotiques réelles

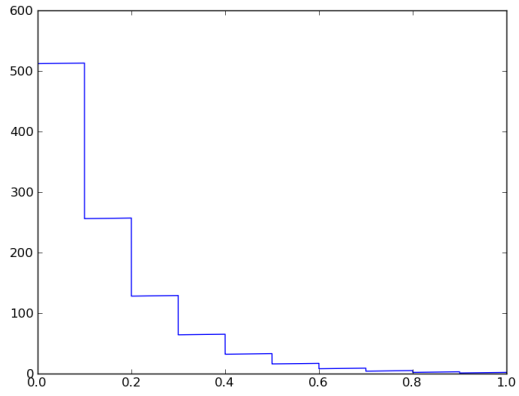
Les itérations chaotiques g définies sur $\mathbb{R}$ sont :

- Infiniment dérivables sur $\left[0, 2^{10}\right[$, sauf aux 10241 points de l'ensemble I défini par $\left\{ \frac{n}{10} \;/\; n \in [\![0; 2^{10} \times 10]\!] \right\}$.
- Affine, de pente 10, sur chaque sous-intervalle.

Les itérations chaotiques G_{f_0} sur $\mathbb{R}$



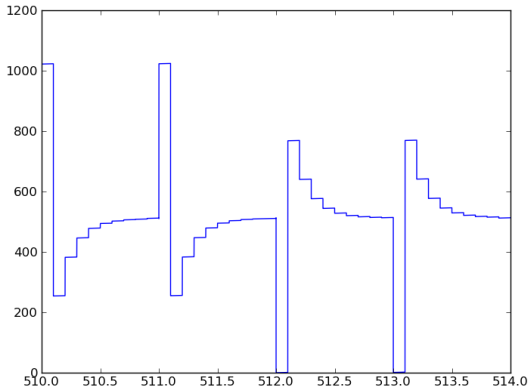
(c) Sur (0,7 ; 1).



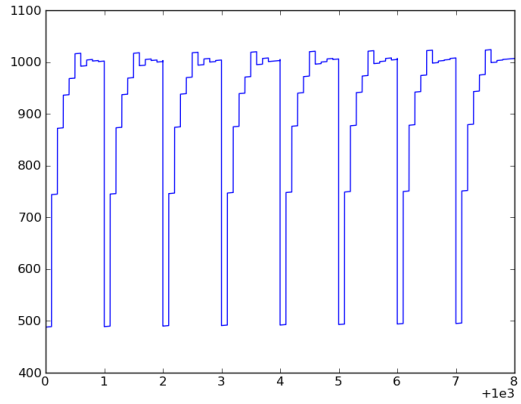
(d) Sur (0 ; 1).

FIGURE : Les itérations chaotiques.

Les itérations chaotiques G_{f_0} sur $\mathbb{R}$



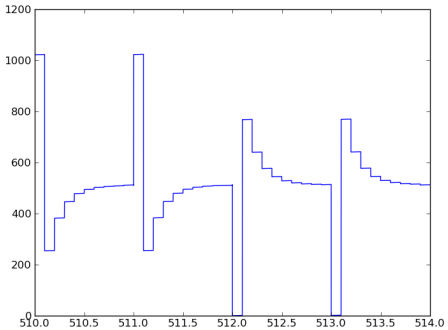
(a) Sur (510 ; 514).



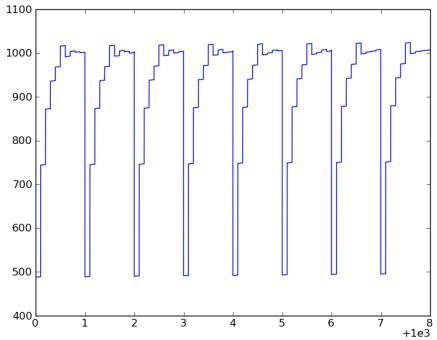
(b) Sur (1000 ; 1008).

FIGURE : Les itérations chaotiques.

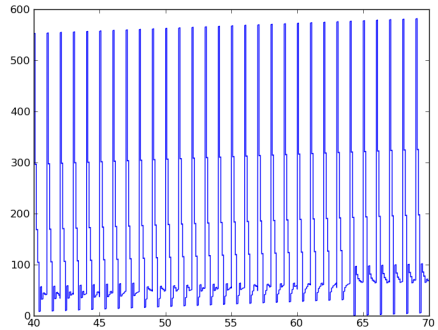
Les itérations chaotiques sur $\mathbb{R}$



(a) Sur (510 ; 514).



(b) Sur (1000 ; 1008).



(c) Sur (40 ; 70).

FIGURE : Les itérations chaotiques.

Chaos des IC G_{f_0} sur $\mathbb{R}$

Chaos de Devaney sur $\mathbb{R}$

Les IC sur $\mathbb{R}$ sont chaotiques selon Devaney, quand $\mathbb{R}$ a sa topologie usuelle.

Exposant de Lyapunov

$$\forall x^0 \in \mathcal{L}, \lambda(x^0) = \lim_{n\to+\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| g' \left(x^{i-1} \right) \right| = \ln(10).$$

Topologie des programmes

Premières questions

Le chaos dans mon PC ?

Le désordre, l'imprévisibilité (vrai, sans perte) sont-ils possibles sur un ordinateur ?

- Il n'y a pas de réels sur mon PC
- Toute machine ayant un nombre fini d'états finit par entrer dans un cycle.

Introduction

Deux cas de figure

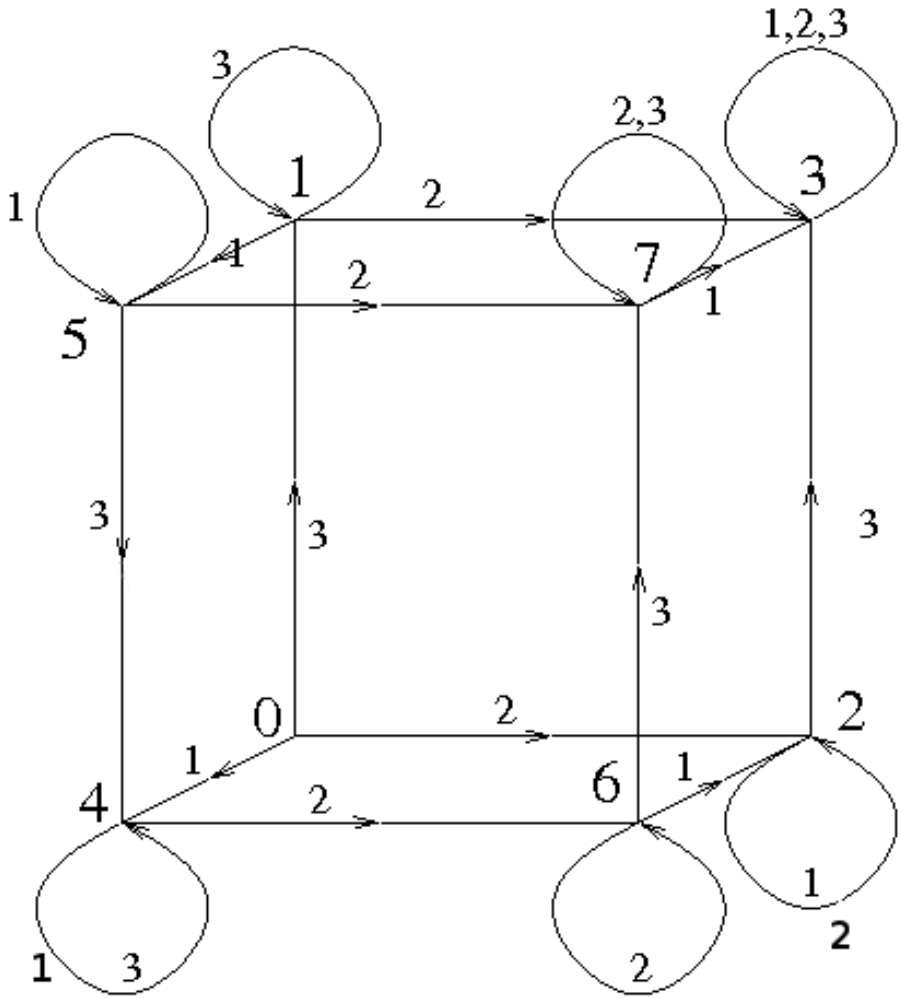
- En vase clos :
  - 4 Go de mémoire $\Rightarrow 2^{4000000000}$ états possibles...
  - Lemme de filature/lemme fantôme
- $\mathcal{X} = \mathbb{B}^{\mathrm{N}} \times \mathcal{P}\left(\llbracket 1; \mathrm{N} \rrbracket\right)^{\mathbb{N}}$:
  - Pas de réels, que des entiers bornés par N
  - On peut utiliser le média à chaque itérée

Introduction

Deux questions

- Peut-on construire des automates chaotiques ?
- Peut-on évaluer si un programme est chaotique ?

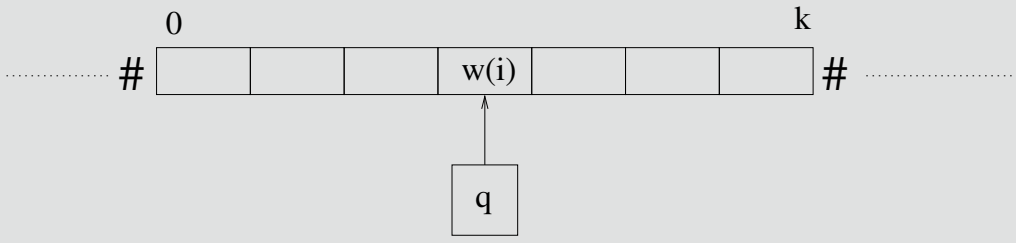
Une machine de Moore chaotique



Le chaos d'un programme

Machines de Turing et systèmes itératifs

Soit (w, i, q) la configuration actuelle de la machine de Turing



- $w = \sharp^{-\omega} w(0) \dots w(k) \sharp^{\omega}$ est la bande de lecture,
- i est la position de la tête de lecture,
- q décrit l'état de la machine,
- et δ est sa fonction de transition.

Le chaos d'un programme

Machines de Turing et systèmes itératifs

On définit f par :

- Si $\delta(q; w(i)) = (q'; a; \rightarrow)$, alors $f(w(0)\dots w(k); i; q) = (w(0)\dots w(i-1)\ a\ w(i+1)\dots w(k); i+1; q')$
- Si $\delta(q; w(i)) = (q'; a; \leftarrow)$, alors $f(w(0)\dots w(k); i; q) = (w(0)\dots w(i-1)\ a\ w(i+1)\dots w(k); i-1; q')$

La machine peut être écrite sous la forme $x^{n+1} = f(x^n)$

A quoi ça sert ?

Un programme chaotique, pour quoi faire ?

- Se placer dans de bonnes conditions lors de conception de nouveaux algorithmes
- Renforcer les attaques (virus chaotique)
- Simuler numériquement des processus chaotiques
- Renforcer la sécurité
- Battre l'intelligence artificielle

Les applications visées

- Génération de nombres pseudo-aléatoires
- Fonctions de hachage
- Tatouage numérique et dissimulation d'information
- Réseaux de capteurs sans fil : vidéosurveillance, secure scheduling...

Applications

Générateurs pseudo-aléatoires

Chaos et aléas

Motivations

Non transitivité, non régularité, non sensibilité, mauvaise entropie, non complexité...

Chaos et aléas

Motivations

Non transitivité, non régularité, non sensibilité, mauvaise entropie, non complexité...

$\Rightarrow$ Batteries de tests statistiques échouées

Notre PRNG

Le PRNG $CI_f(PRNG_1, PRNG_2)$

Paramètres : Une fonction $f : \mathbb{B}^N \to \mathbb{B}^N$, et deux PRNGs :

- $S \in [\![1, N]\!]^{\mathbb{N}}$
- et $m \in S^{\mathbb{N}}, S \subset \mathbb{N}$

Graine : Les graines de S et m, et $E \in \mathbb{B}^N$

PRNG : $\left(G_f(E, S)^{m^i}\right)_{i \in \mathbb{N}}$

Le Old CI_{f_0}(logistic,logistic)

m (logistic map) : 2
S (logistic map) :

État interne du système x :

$$\begin{matrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{matrix} \qquad (1)$$

Sortie :

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2
S (logistic map) : 1

État interne du système x :

$$\begin{aligned} \mathbf{0} &\rightarrow \mathbf{1} \\ \mathbf{0} &\rightarrow 0 \\ \mathbf{0} &\rightarrow 0 \\ \mathbf{0} &\rightarrow 0 \end{aligned} \qquad (1)$$

Sortie :

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2
S (logistic map) : 1 3

État interne du système x :

$$\begin{aligned} &\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \\ &\mathbf{0} \rightarrow 0 \rightarrow 0 \\ &\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \\ &\mathbf{0} \rightarrow 0 \rightarrow 0 \end{aligned} \qquad (1)$$

Sortie :

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2
S (logistic map) : 1 3

État interne du système x :

$$\begin{aligned} &\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \\ &\mathbf{0} \rightarrow 0 \rightarrow 0 \\ &\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \\ &\mathbf{0} \rightarrow 0 \rightarrow 0 \end{aligned} \qquad (1)$$

Sortie : 1 0 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1
S (logistic map) : 1 3

État interne du système x :

$$\begin{aligned} &\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \\ &\mathbf{0} \rightarrow 0 \rightarrow 0 \\ &\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \\ &\mathbf{0} \rightarrow 0 \rightarrow 0 \end{aligned} \qquad (1)$$

Sortie : 1 0 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1
S (logistic map) : 1 3 2

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \tag{1}$$

Sortie : 1 0 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1
S (logistic map) : 1 3 2

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1 4
S (logistic map) : 1 3 2

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1 4
S (logistic map) : 1 3 2 1

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1 4
S (logistic map) : 1 3 2 1 1

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow \mathbf{1} \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1 4
S (logistic map) : 1 3 2 1 1 2

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) : 2 1 4
S (logistic map) : 1 3 2 1 1 2 1

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow \mathbf{0} \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow 0 \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) :	2			1	4			...
S (logistic map) :	1	3		2	1	1	2	1

État interne du système x :

$$
\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow \mathbf{0} \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow 0 \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0
\end{aligned}
\qquad (1)
$$

Sortie : 1 0 1 0 1 1 1 0 0 0 1 0

Le Old Cl_{f_0}(logistic,logistic)

m (logistic map) :	2			1	4			...
S (logistic map) :	1	3		2	1	1	2	1 ...

État interne du système x :

$$\begin{aligned}
&\mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow \mathbf{1} \rightarrow 1 \rightarrow \mathbf{0}\ldots \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow \mathbf{0} \rightarrow 0\ldots \\
&\mathbf{0} \rightarrow 0 \rightarrow \mathbf{1} \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1\ldots \\
&\mathbf{0} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0\ldots
\end{aligned} \qquad (1)$$

Sortie : 1 0 1 0 1 1 1 0 0 0 1 0 ...

Choix de l'ensemble $\mathcal{M}$

x^{i+1}	$\mathcal{M}$:	{1}	{2}	{3}	{4}	{5}	{6}	{7}	{8}	{9}	{10}	{11}	{12}	{13}
0			√		√		√		√		√		√	
1		√		√		√		√		√		√		√
2		√		√		√		√		√		√		√
3			√		√		√		√		√		√	
4		√		√		√		√		√		√		√
5			√		√		√		√		√		√	
6			√		√		√		√		√		√	
7				√		√		√		√		√		√
8		√		√		√		√		√		√		√
9			√		√		√		√		√		√	
10			√		√		√		√		√		√	
11				√		√		√		√		√		√
12			√		√		√		√		√		√	
13				√		√		√		√		√		√
14				√		√		√		√		√		√
15					√		√		√		√		√	

FIGURE : Nombre d'itérations entre deux sorties

Variantes

Quelques variantes du CI PRNG

- *New* $CI_f(PRNG_1, PRNG_2)$: éviter de changer deux fois de suite un même bit entre deux outputs
  - Ne plus compter le nombre d'itérées entre deux outputs
  - Mais le nombre de bits à changer
- Utiliser des tables précalculées
- *XorCIPRNG* : $S^{n+1} = S^n \oplus PRNG^n$
- etc.

Nouvelle version de $CI_f(PRNG_1, PRNG_2)$

m	0	4					2		2		
k	0	4				+1	2		2	+1	
b		1	4	2	$\underline{2}$	3	3	4	1	$\underline{1}$	4
d	r	r $\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\\0\\1\end{pmatrix}$		$\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$	r $\begin{pmatrix}0\\0\\1\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\1\\1\end{pmatrix}$	r $\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$		$\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$
S		1	4	2		3	3	4	1		4
x^0	x^0					x^4		x^6			x^8
0	0	$\xrightarrow{1}$ 1				1		1	$\xrightarrow{1}$ 0		0
1	1			$\xrightarrow{2}$ 0		0		0			0
0	0				$\xrightarrow{3}$ 1	1	$\xrightarrow{3}$ 0	0			0
0	0		$\xrightarrow{4}$ 1			1		$\xrightarrow{4}$ 0 0		$\xrightarrow{4}$ 1	1

Sortie en bits : $x_1^0 x_2^0 x_3^0 x_4^0 x_1^4 x_2^4 x_3^4 x_4^4 x_1^6 x_2^6 \ldots = 0100101110000001\ldots$
Sortie en entiers : $x^0, x^4, x^6, x^8 \ldots = 4, 11, 8, 1\ldots$

TABLE 3.4 – Exemple d'une génération avec le New CI(XORshift,XORshift) PRNG

FIGURE : Le NEW CI PRNG

Nouvelle version de $CI_f(PRNG_1, PRNG_2)$

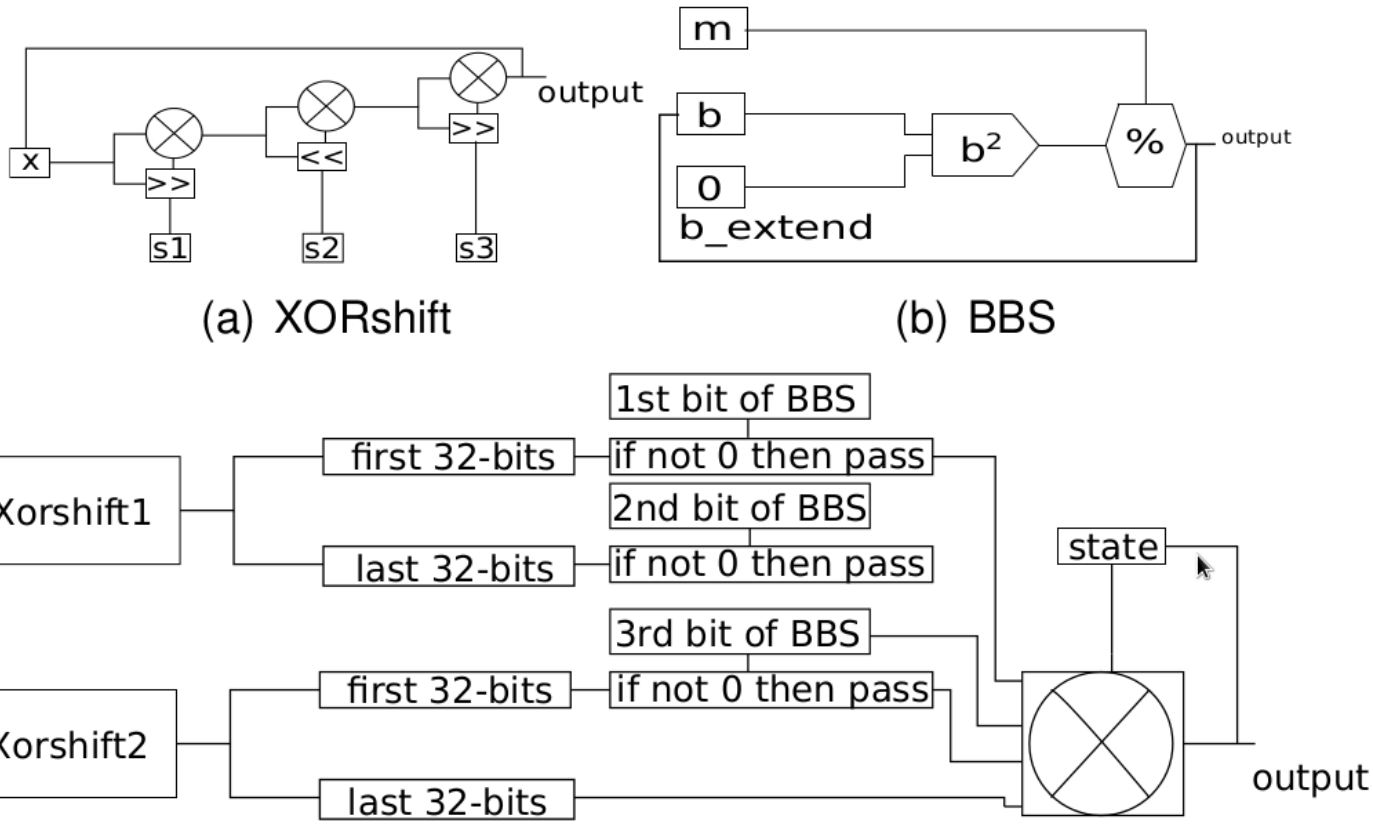


FIGURE : Version FPGA

Notre générateur GPU

Algorithm 4: Main kernel for the chaotic iterations based PRNG GPU efficient version

```
Input: InternalVarXorLikeArray: array with internal
       variables of 1 xor-like PRNGs in global
       memory;
NumThreads: Number of threads;
array_comb1, array_comb2: Arrays containing
combinations of size combination_size;
Output: NewNb: array containing random numbers
        in global memory
if threadId is concerned then
    retrieve data from
    InternalVarXorLikeArray[threadId] in local
    variables including shared memory and x;
    offset = threadIdx%combination_size;
    o1 = threadIdx-offset+array_comb1[offset];
    o2 = threadIdx-offset+array_comb2[offset];
    for i=1 to n do
        t=xor-like();
        t=t^shmem[o1]^shmem[o2];
        shared_mem[threadId]=t;
        x = x^t;
        store the new PRNG in
        NewNb[NumThreads*threadId+i];
    store internal variables in
    InternalVarXorLikeArray[threadId];
```

Algorithm 5: main kernel for the BBS based PRNG GPU

```
Input: InternalVarBBSArray: array with internal
       variables of the 8 BBS in global memory;
NumThreads: Number of threads;
array_comb: 2D Arrays containing 16 combinations
(in first dimension) of size combination_size (in
second dimension);
array_shift[4]={0,1,3,7};
Output: NewNb: array containing random numbers
        in global memory
if threadId is concerned then
    retrieve data from
    InternalVarBBSArray[threadId] in local variables
    including shared memory and x;
    we consider that bbs1 ... bbs8 represent the
    internal states of the 8 BBS numbers;
    offset = threadIdx%combination_size;
    o1 =
    threadIdx-offset+array_comb[bbs1&7][offset];
    o2 =
    threadIdx-offset+array_comb[8+bbs2&7][offset];
    for i=1 to n do
        t<<=4;
        t|=BBS1(bbs1)&15;
        ...;
        t<<=4;
        t|=BBS8(bbs8)&15;
        // two new shifts
        shift=BBS3(bbs3)&3;
        t<<=shift;
        t|=BBS1(bbs1)&array_shift[shift];
        shift=BBS7(bbs7)&3;
        t<<=shift;
        t|=BBS2(bbs2)&array_shift[shift];
        t=t^shmem[o1]^shmem[o2];
        shared_mem[threadId]=t;
        x = x^t;
        store the new PRNG in
        NewNb[NumThreads*threadId+i];
    store internal variables in
    InternalVarXorLikeArray[threadId] using a
    rotation;
```

Résultats

Premiers résultats

1. Générateur chaotique dès que le GTPIC de G_f est fortement connexe
2. Toutes les autres propriétés de chaos
3. Sortie uniforme si la matrice d'adjacence réduite du GTPIC est doublement stochastique
4. Les résultats aux tests statistiques sont meilleurs (DieHARD, NIST, TestU01)

NIST pour les PRNG en entrée

TABLE 5.1 – NIST SP 800-22 test results ($\mathbb{P}_T$)

Test name	Logistic	XORshift	ISAAC
Frequency (Monobit) Test	0.53414	0.14532	0.67868
Frequency Test within a Block	0.00275	0.45593	0.10252
Runs Test	0.00001	0.21330	0.69931
Longest Run of Ones in a Block Test	0.08051	0.28966	0.43727
Binary Matrix Rank Test	0.67868	0.00000	0.89776
Discrete Fourier Transform (Spectral) Test	0.57490	0.00535	0.51412
Non-overlapping Template Matching Test*	0.28468	0.50365	0.55515
Overlapping Template Matching Test	0.10879	0.86769	0.63711
Universal Statistical Test	0.02054	0.27570	0.69931
Linear Complexity Test	0.79813	0.92407	0.03756
Serial Test* (m=10)	0.41542	0.75792	0.32681
Approximate Entropy Test (m=10)	0.02054	0.41902	0.30412
Cumulative Sums (Cusum) Test*	0.60617	0.81154	0.36786
Random Excursions Test*	0.53342	0.41923	0.50711
Random Excursions Variant Test*	0.28507	0.52833	0.40930
Success	15/15	14/15	15/15

FIGURE : Le NIST pour 3 PRNG

NIST pour le Old CI

TABLE 5.5 – NIST SP 800-22 test results ($\mathbb{P}_T$) for Old CI algorithms (N = 4 and k = 13)

Test name	Old CI			
	Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
Frequency (Monobit) Test	0.85138	0.59554	0.40119	0.33453
Frequency Test within a Block	0.38382	0.55442	0.89776	0.71974
Runs Test	0.31908	0.45593	0.31908	0.38382
Longest Run of Ones in a Block Test	0.13728	0.01671	0.08558	0.67868
Binary Matrix Rank Test	0.69931	0.61630	0.47498	0.79813
Discrete Fourier Transform (Spectral) Test	0.12962	0.00019	0.77918	0.67868
Non-overlapping Template Matching Test*	0.48473	0.53225	0.53568	0.51258
Overlapping Template Matching Test	0.47498	0.33453	0.36691	0.07571
Universal Statistical Test	0.09657	0.03292	0.26224	0.85138
Linear Complexity Test	0.41902	0.40119	0.61715	0.21330
Serial Test* (m=10)	0.53427	0.01339	0.33453	0.76102
Approximate Entropy Test (m=10)	0.99146	0.13728	0.53414	0.22482
Cumulative Sums (Cusum) Test*	0.75530	0.04646	0.31915	0.47658
Random Excursions Test*	0.65406	0.50362	0.50804	0.46305
Random Excursions Variant Test*	0.55388	0.34777	0.48400	0.54863
Success	15/15	15/15	15/15	15/15

FIGURE : Résultats du Old CI PRNG

NIST pour le New CI

TABLE 5.10 – NIST SP 800-22 test results ($\mathbb{P}_T$) for new CI algorithms

Test name	New CI		
	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
Frequency (Monobit) Test	0.47498	0.88317	0.83430
Frequency Test within a Block	0.89776	0.40119	0.33453
Runs Test	0.81653	0.31908	0.00576
Longest Run of Ones in a Block Test	0.79813	0.06688	0.47498
Binary Matrix Rank Test	0.26224	0.88317	0.69931
Discrete Fourier Transform (Spectral) Test	0.00716	0.33453	0.59559
Non-overlapping Template Matching Test*	0.44991	0.46467	0.51446
Overlapping Template Matching Test	0.51412	0.69931	0.88317
Universal Statistical Test	0.67868	0.24928	0.06282
Linear Complexity Test	0.65793	0.65793	0.94630
Serial Test* (m=10)	0.42534	0.90619	0.44137
Approximate Entropy Test (m=10)	0.63719	0.22482	0.13728
Cumulative Sums (Cusum) Test*	0.27968	0.84065	0.14139
Random Excursions Test*	0.28740	0.30075	0.34625
Random Excursions Variant Test*	0.48668	0.34294	0.55048
Success	15/15	15/15	15/15

FIGURE : Résultats du New CI PRNG (Nist)

DieHard pour les PRNG en entrée

TABLE 5.2 – Results of DieHARD battery of tests

No.	Test name	Logistic	XORshift	ISAAC
1	Overlapping Sum	Pass	Pass	Pass
2	Runs Up 1	Pass	Pass	Pass
	Runs Down 1	Pass	Pass	Pass
	Runs Up 2	Pass	Pass	Pass
	Runs Down 2	Pass	Pass	Pass
3	3D Spheres	Pass	Pass	Pass
4	Parking Lot	Pass	Pass	Pass
5	Birthday Spacing	Pass	Pass	Pass
6	Count the ones 1	Pass	Fail	Pass
7	Binary Rank 6×8	Pass	Pass	Pass
8	Binary Rank 31×31	Pass	Fail	Pass
9	Binary Rank 32×32	Pass	Fail	Pass
10	Count the ones 2	Pass	Pass	Pass
11	Bit Stream	Pass	Pass	Pass
12	Craps Wins	Pass	Pass	Pass
	Throws	Pass	Pass	Pass
13	Minimum Distance	Pass	Pass	Pass
14	Overlapping Perm.	Pass	Pass	Pass
15	Squeeze	Pass	Pass	Pass
16	OPSO	Pass	Pass	Pass
17	OQSO	Pass	Pass	Pass
18	DNA	Pass	Pass	Pass
	Number of tests passed	18	15	18

FIGURE : DieHard pour 3 PRNG

DieHard pour le Old CI

TABLE 5.6 – Results of DieHARD battery of tests for Old CI algorithms (N = 4)

No.	Test name	Old CI			
		Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
1	Overlapping Sum	Pass	Pass	Pass	Pass
2	Runs Up 1	Pass	Pass	Pass	Pass
	Runs Down 1	Pass	Pass	Pass	Pass
	Runs Up 2	Pass	Pass	Pass	Pass
	Runs Down 2	Pass	Pass	Pass	Pass
3	3D Spheres	Pass	Pass	Pass	Pass
4	Parking Lot	Pass	Pass	Pass	Pass
5	Birthday Spacing	Pass	Pass	Pass	Pass
6	Count the ones 1	Pass	Pass	Pass	Pass
7	Binary Rank 6 × 8	Pass	Pass	Pass	Pass
8	Binary Rank 31 × 31	Pass	Pass	Pass	Pass
9	Binary Rank 32 × 32	Pass	Pass	Pass	Pass
10	Count the ones 2	Pass	Pass	Pass	Pass
11	Bit Stream	Pass	Pass	Pass	Pass
12	Craps Wins	Pass	Pass	Pass	Pass
	Throws	Pass	Pass	Pass	Pass
13	Minimum Distance	Pass	Pass	Pass	Pass
14	Overlapping Perm.	Pass	Pass	Pass	Pass
15	Squeeze	Pass	Pass	Pass	Pass
16	OPSO	Pass	Pass	Pass	Pass
17	OQSO	Pass	Pass	Pass	Pass
18	DNA	Pass	Pass	Pass	Pass
	Number of tests passed	18	18	18	18

FIGURE : Résultats du Old CI PRNG

DieHard pour le New CI

TABLE 5.11 – Results of DieHARD battery of tests for new CI algorithms (N = 32)

No.	Test name	New CI			
		Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
1	Overlapping Sum	Pass	Pass	Pass	Pass
2	Runs Up 1	Pass	Pass	Pass	Pass
	Runs Down 1	Pass	Pass	Pass	Pass
	Runs Up 2	Pass	Pass	Pass	Pass
	Runs Down 2	Pass	Pass	Pass	Pass
3	3D Spheres	Pass	Pass	Pass	Pass
4	Parking Lot	Pass	Pass	Pass	Pass
5	Birthday Spacing	Pass	Pass	Pass	Pass
6	Count the ones 1	Pass	Pass	Pass	Pass
7	Binary Rank 6×8	Pass	Pass	Pass	Pass
8	Binary Rank 31×31	Pass	Pass	Pass	Pass
9	Binary Rank 32×32	Pass	Pass	Pass	Pass
10	Count the ones 2	Pass	Pass	Pass	Pass
11	Bit Stream	Pass	Pass	Pass	Pass
12	Craps Wins	Pass	Pass	Pass	Pass
	Throws	Pass	Pass	Pass	Pass
13	Minimum Distance	Pass	Pass	Pass	Pass
14	Overlapping Perm.	Pass	Pass	Pass	Pass
15	Squeeze	Pass	Pass	Pass	Pass
16	OPSO	Pass	Pass	Pass	Pass
17	OQSO	Pass	Pass	Pass	Pass
18	DNA	Pass	Pass	Pass	Pass
	Number of tests passed	18	18	18	18

FIGURE : Résultats du New CI PRNG (DieHard)

TestU01 pour les PRNG en entrée

TABLE 5.4 – TestU01 Statistical Test

Test name	Battery	Number of tests (516)	Logistic	XORshift	ISAAC
Rabbit	32×10^9 bits	38	21	14	0
Alphabit	32×10^9 bits	17	16	9	0
Pseudo DieHARD	Standard	126	0	2	0
FIPS_140_2	Standard	16	0	0	0
SmallCrush	Standard	15	4	5	0
Crush	Standard	144	95	57	0
Big Crush	Standard	160	125	55	0
Number of failures			261	146	0

FIGURE : TestU01 pour 3 PRNG

TestU01 pour le Old CI

TABLE 5.8 – TestU01 Statistical Test for old CI algorithms (N = 4)

Test name			Old CI			
			Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
Rabbit	32×10^9 bits	38	7	2	0	0
Alphabit	32×10^9 bits	17	3	0	0	0
Pseudo DieHARD	Standard	126	0	0	0	0
FIPS_140_2	Standard	16	0	0	0	0
SmallCrush	Standard	15	2	0	0	0
Crush	Standard	144	47	4	0	0
Big Crush	Standard	160	79	3	0	0
Number of failures		518	138	9	0	0

FIGURE : Résultats du Old CI PRNG

TestU01 pour le New CI

TABLE 5.13 – TestU01 Statistical Battery for the New CI algorithms (N = 32)

Test name			New CI		
			Logistic + Logistic	ISAAC + XORshift	ISAAC + ISAAC
Rabbit	32×10^9 bits	38	0	0	0
Alphabit	32×10^9 bits	17	0	0	0
Pseudo DieHARD	Standard	126	0	0	0
FIPS_140_2	Standard	16	0	0	0
SmallCrush	Standard	15	0	0	0
Crush	Standard	144	0	0	0
Big Crush	Standard	160	0	0	0
Number of failures			0	0	0

FIGURE : Résultats du New CI PRNG (TestU01)

Comparaison avec le BBS

PRNGs	XOR	BBS	CI1 XOR XOR	CI1 BBS XOR	CI2 XOR XOR	CI2 BBS XOR	CI3 XOR XOR	CI3 BBS XOR	CI4 XOR XOR	CI4 BBS XOR
NIST	14/15	2/15	15/15	15/15	15/15	15/15	15/15	8/15	15/15	15/15
DieHARD	15/18	2/18	15/15	15/15	15/15	15/15	15/15	8/18	15/15	15/15
CP	5/5	4/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
TesuU01	370/516	212/516	411/516	375/516	516/516	422/516	15/15	351/516	516/516	516/516

Résultats

Table I: NIST and Diehard test suite passing rate the for PRNGs without CI

Types of PRNGs	Linear PRNGs		Lagged PRNGs				ICG PRNGs	Mixed PRNGs		
PRNG / *Tests*	LCG	MRG	AWC	SWB	SWC	GFSR	INV	LCG2	LCG3	MRG2
NIST	11/15	14/15	**15/15**	**15/15**	14/15	14/15	14/15	14/15	14/15	14/15
Diehard	16/18	16/18	15/18	16/18	**18/18**	16/18	16/18	16/18	16/18	16/18

Table II: NIST and Diehard test suite passing rate the for PRNGs without CI

Types of PRNGs	Linear PRNGs		Lagged PRNGs				ICG PRNGs	Mixed PRNGs		
Single CIPRNG / *Tests*	LCG	MRG	AWC	SWB	SWC	GFSR	INV	LCG2	LCG3	MRG2
Old CIPRNG										
NIST	**15/15** *	**15/15** *	**15/15**	**15/15**	**15/15** *	**15/15** *	**15/15** *	**15/15** *	**15/15** *	**15/15**
Diehard	**18/18** *	**18/18** *	**18/18** *	**18/18** *	**18/18**	**18/18** *	**18/18** *	**18/18** *	**18/18** *	**18/18** *
New CIPRNG										
NIST	**15/15** *	**15/15** *	**15/15**	**15/15**	**15/15** *	**15/15** *	**15/15** *	**15/15** *	**15/15** *	**15/15**
Diehard	**18/18** *	**18/18** *	**18/18** *	**18/18** *	**18/18**	**18/18** *	**18/18** *	**18/18** *	**18/18** *	**18/18** *
Xor CIPRNG										
NIST	14/15*	**15/15** *	**15/15**	**15/15**	14/15	**15/15** *	14/15	**15/15** *	**15/15** *	**15/15**
Diehard	16/18	16/18	17/18*	**18/18** *	**18/18**	**18/18** *	16/18	16/18	16/18	16/18

Résultats

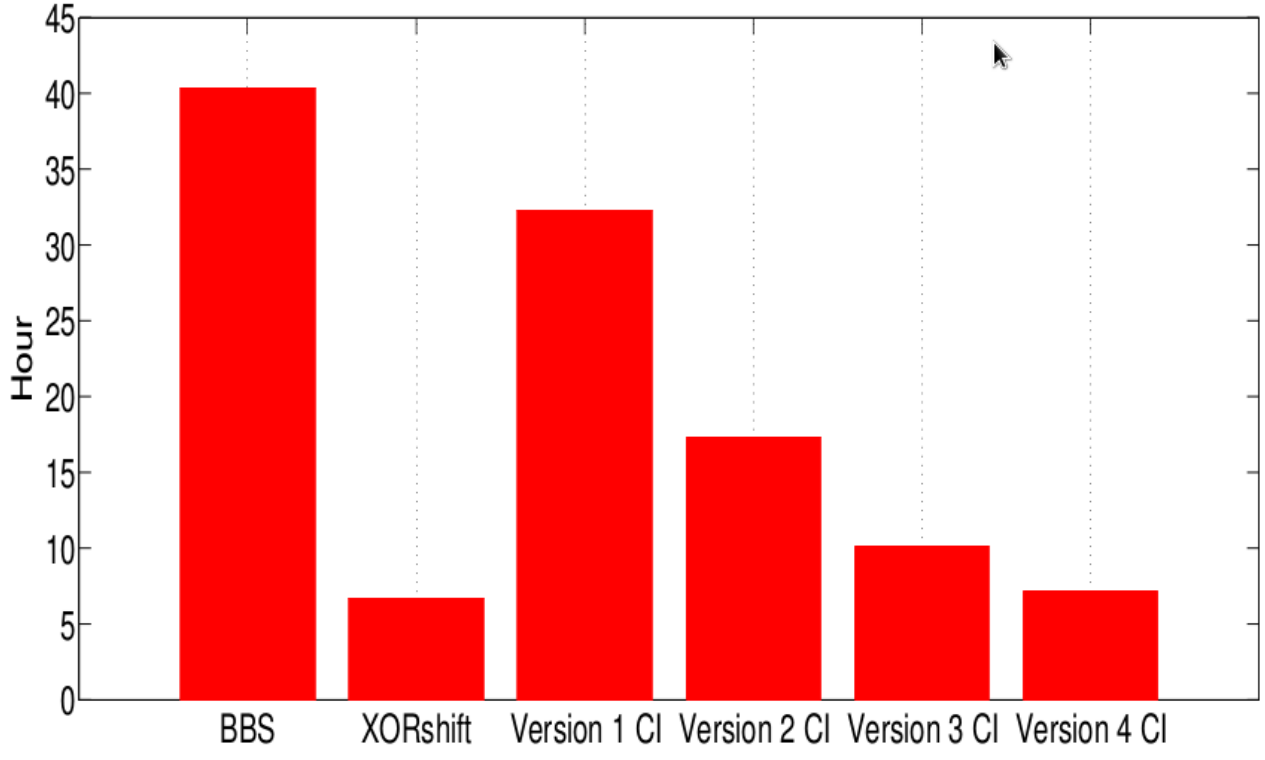


FIGURE : Perte de vitesse

Les PRNG cryptographiquement sûrs

Générateur G *cryptographiquement sûr*

Soit $\mathcal{D} : \mathbb{B}^M \longrightarrow \mathbb{B}$ un algorithme probabiliste fonctionnant en temps T. Soit $\varepsilon > 0$. $\mathcal{D}$ est une (T, ε)-attaque de distinction sur le générateur G si

$$\left| Pr[\mathcal{D}(G(k)) = 1 \mid k \in_R \{0,1\}^\ell] - Pr[\mathcal{D}(s) = 1 \mid s \in_R \mathbb{B}^M] \right| \geqslant \varepsilon,$$

où la probabilité est prise sur le lancer de pièces interne de $\mathcal{D}$, et la notation "$\in_R$" signifie que le choix de l'élément est aléatoire et uniforme sur l'ensemble considéré.

Les PRNG cryptographiquement sûrs

Générateur G *cryptographiquement sûr*

Soit $\mathcal{D} : \mathbb{B}^M \longrightarrow \mathbb{B}$ un algorithme probabiliste fonctionnant en temps T. Soit $\varepsilon > 0$. $\mathcal{D}$ est une (T, ε)-attaque de distinction sur le générateur G si

$$\left| Pr[\mathcal{D}(G(k)) = 1 \mid k \in_R \{0,1\}^\ell] - Pr[\mathcal{D}(s) = 1 \mid s \in_R \mathbb{B}^M] \right| \geqslant \varepsilon,$$

où la probabilité est prise sur le lancer de pièces interne de $\mathcal{D}$, et la notation "$\in_R$" signifie que le choix de l'élément est aléatoire et uniforme sur l'ensemble considéré.

Sécurité du CIPRNG

Si le premier PRNG en entrée est sûr, alors notre PRNG l'est aussi.

Derniers Résultats

Nos derniers résultats

1. Implantation sur GPU $\Rightarrow$ 20 milliards de nombres (32 bits) par seconde sur un PC
2. Utilisation de BBS $\Rightarrow$ 1 milliards de nombres sûrs par seconde
3. Version chaotique du cryptosystème asymétrique probabiliste de Blum-Goldwasser
4. Mixage avec dispositif optique (Larger, OPTO)

Le générateur mixé

Côté DISC

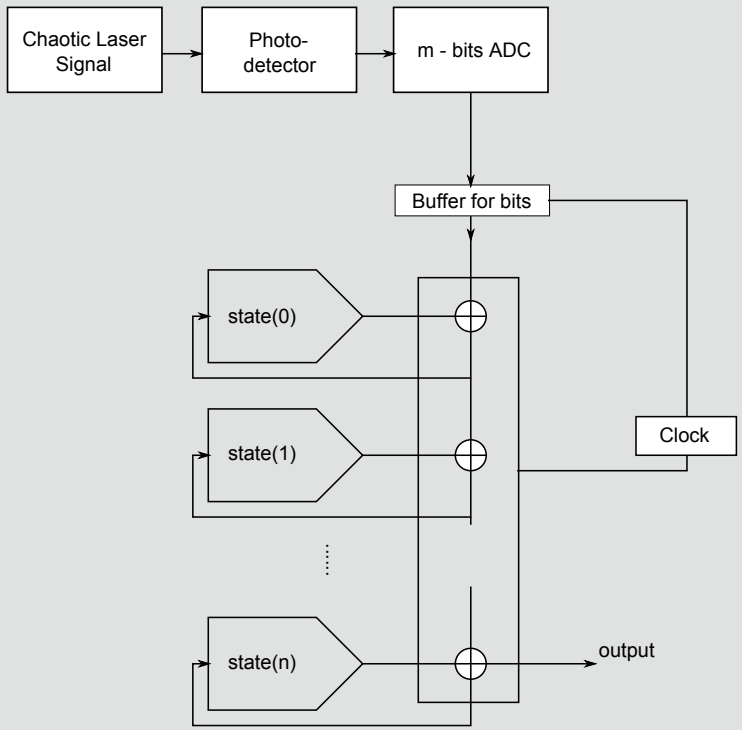


FIGURE : Premier PRNG mixé réalisé

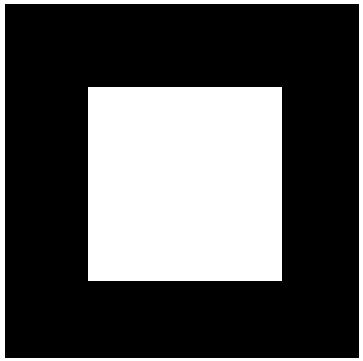
Le générateur mixé

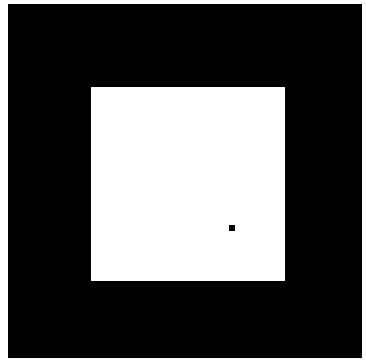
Tests *n*	1	10	20	30	40
NIST suite	0/15	14/15	15/15	15/15	15/15
DieHARD suite	1/18	11/18	14/18	18/18	18/18

D’autres applications en cryptologie

Fonctions de hachage

Les fonctions de hachage





```
34A5C1B3DFFCC8902F7B248C3ABEFE2C9C9538E5104D117B399C999F74CF1CAD
5E67725CAA6B7B7434BE57F5F30F2D3D57056FA960B69052453CBC62D9267896
```

Hachage par itérations chaotiques

Des fonctions de hachage à partir d'IC

- *État initial :* valeur hachée du média
- *Stratégie :* bits du media regroupés 9 par 9, PRNG cryptographiquement sûr, etc.
- *Condensé :* dernier état de *N* IC.

Pourquoi le chaos ?

Lien hachage/chaos

Rapidité	Complexité linéaire
Résistance à la préimage	Lyapunov, Devaney, Entropie
Effet avalanche	Expansivité, sensibilités
Uniforme répartition, diffusion	Transitivités
Confusion	Sensibilité aux CI

Pourquoi le chaos ?

Lien hachage/chaos

Rapidité	Complexité linéaire
Résistance à la préimage	Lyapunov, Devaney, Entropie
Effet avalanche	Expansivité, sensibilités
Uniforme répartition, diffusion	Transitivités
Confusion	Sensibilité aux CI

$\Rightarrow$ Si la fonction de hachage initiale est résistante aux collisions/à la seconde préimage, alors notre fonction l'est aussi

D’autres applications en cryptologie

Dissimulation d’information

Introduction

Stéganographie (Cachin 2004)

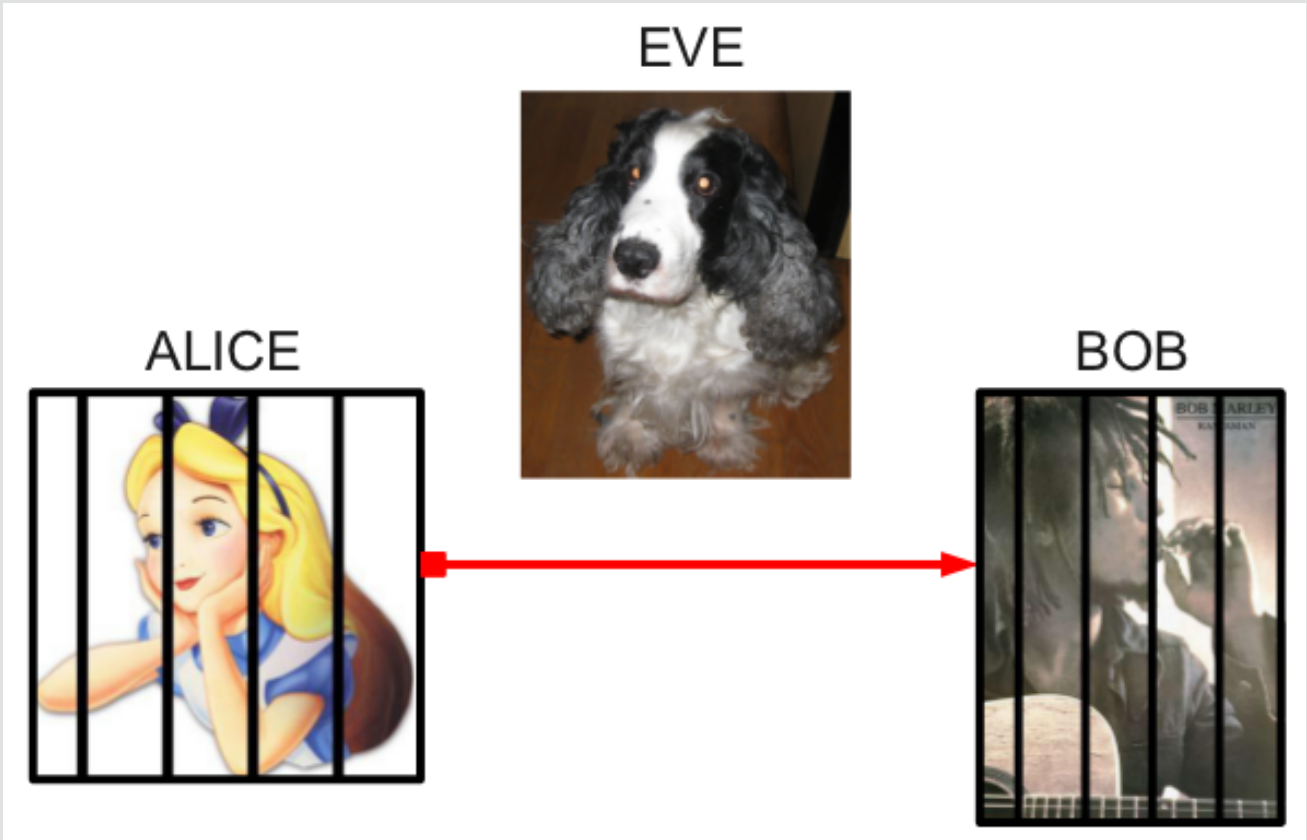
L'art et la science de communiquer de telle sorte que la présence d'un message ne peut pas être détectée.

Tatouage numérique (Furon 2005)

L'art de cacher des métadonnées dans du contenu numérique de manière robuste.

Problème du prisonnier de Simmons

Le problème du prisonnier de Simmons



Impact du contexte (Kalker, 2001, Furon, 2005)

Classes d'attaque

Attaques	Images d'origines	Images tatouées	Messages cachés
Objet tatoué seul (WOA)		×	
Message connu		×	×
Original connu	×	×	
Message constant			×
Original estimé	× (Estimé)		

Impact du contexte (Kalker, 2001, Furon, 2005)

Classes d'attaque

Attaques	Images d'origines	Images tatouées	Messages cachés
Objet tatoué seul (WOA)		×	
Message connu		×	×
Original connu	×	×	
Message constant			×
Original estimé	× (Estimé)		

STEGO-SECURITE (WOA) : $\forall K_1 \in \mathbb{K}, p(Y|K_1) = p(X)$.

Problèmes de l'approche existante

Les problèmes de l'approche existante

- Restreint :
  - Uniquement dans le WOA,
  - Pour le prisonnier de Simmons
  - Avec une unique clé.
- Existence de modèles probabilistes
  $\Rightarrow$ Nouvelles catégories d'attaques.
- Difficile à étudier.

Les techniques d'étalement de spectre

Étalement de spectre (naturel)

L'étalement de spectre, c'est $y = x + w$, où :

- x : vecteur hôte,
- y : vecteur tatoué,
- w : vecteur filigrane, obtenu à partir du message caché m comme suit :

$$w = \sum_{i=0}^{N_c-1} -\left(1 + \eta\,(-1)^{m^i} \frac{< x, u^i >}{| < x, u^i > |}\right) \frac{< x, u^i >}{\|u^i\|^2}\, u^i$$

Approche complémentaire

Notre approche

1. Écrire le schéma ainsi : $x^{n+1} = f(x^n)$.
2. Mesurer son imprévisibilité (topologique).

Approche complémentaire

Notre approche

1. Écrire le schéma ainsi : $x^{n+1} = f(x^n)$.
2. Mesurer son imprévisibilité (topologique).

Avantage : universalité

1. Étude toujours possible (Schéma $\Rightarrow$ Turing $\Rightarrow$ SDD).
2. Plus petite construction math (topologie/tribu).
3. Non restreinte au WOA, au prisonnier de Simmons, à l'utilisation d'une clé.
4. Plus varié (objectifs visés = propriétés topologiques).

Pertinence de la notion

Liens sécurité/chaos

Sensibilité	Attaque de l'original estimé Bonne authentification tatouage fragile
Transitivité	Zoom inutile. Zeroing attack Authentification meilleure
Régularité + transitivité	Attaques actives
Expansivité + transitivité forte	Attaque de l'original connu Attaque du message connu

Étude approfondie

Propriétés qualitatives

Les techniques d'étalement de spectre sont chaotiques (Devaney) et :

- Fortement transitives.
- Topologiquement mélangeantes.

Propriétés quantitatives

- Sensibles aux conditions initiales, et leur constante est $\geqslant \frac{N_b}{2}$.
- *Ne sont pas expansives*.

Question

Question

- Ne peut-on faire mieux ?
- *i.e.* Stégo-sûr, Chaos-sûr et expansif.

Question

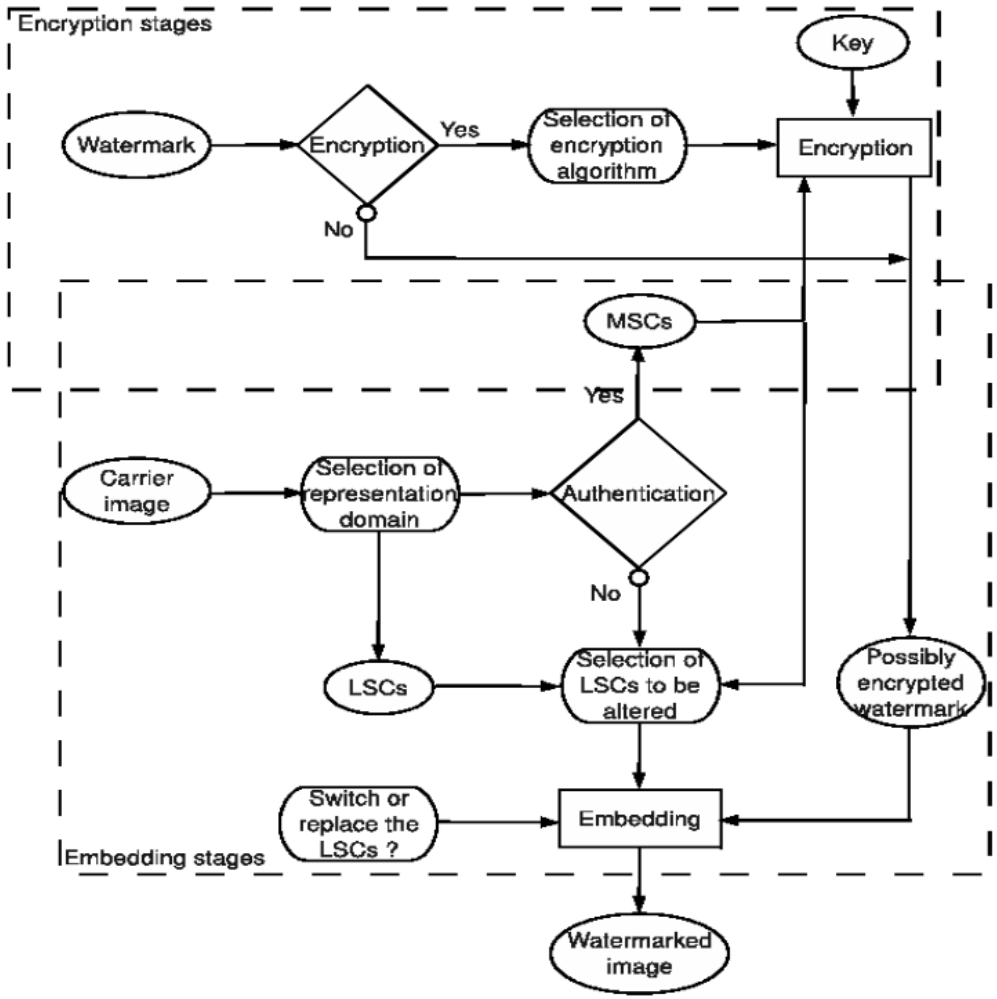
Question

- Ne peut-on faire mieux ?
- *i.e.* Stégo-sûr, Chaos-sûr et expansif.

Réponse

Si, en utilisant les IC (si elles sont stégo-sûres)

Le dhCI


Encryption stages
Key
Watermark
Encryption
Yes
Selection of encryption algorithm
Encryption
No
MSCs
Yes
Carrier image
Selection of representation domain
Authentication
No
LSCs
Selection of LSCs to be altered
Possibly encrypted watermark
Switch or replace the LSCs ?
Embedding
Embedding stages
Watermarked image

Variantes

Variantes de l'algorithme

- Négation ou substitution
- Tatouage dans les domaines spatial et fréquentiels (ondelettes, FFT, DCT)
- Codes de Reed-Solomon
- Robustesse : compression JPEG, rotation, redimension, zeroing attack, flou...

Sécurité du dhCI

Théorème (chaos-sécurité)

Le dhCI est chaos-sûr, et :

- *qualitatif :* Transitivité forte, mélange topologique, chaos selon Li-Yorke et Knudsen.
- *quantitatif :* Sensibilité de N – 1, expansivité de 1, exposant de Lyapunov de $ln(N)$, entropie topologique ∞.

Théorème (stégo-sécurité)

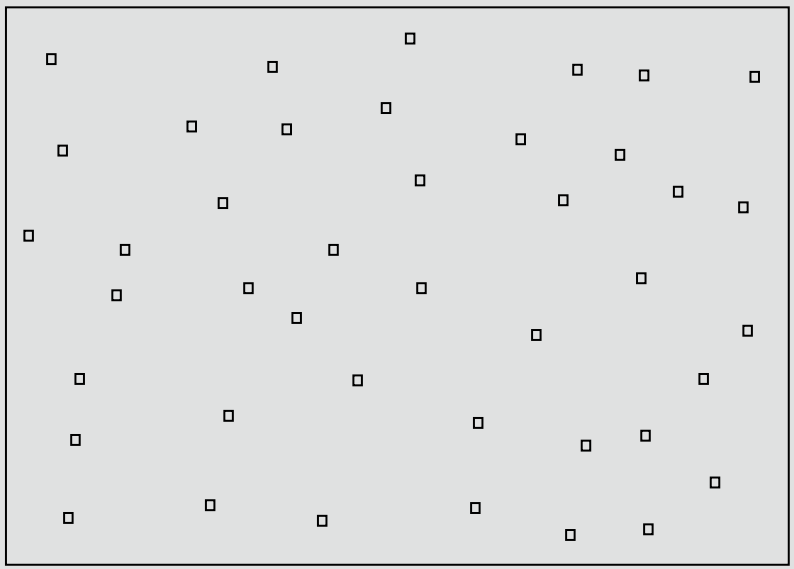
Cet algorithme est, pour certaines stratégies, stégo-sûr.

D’autres applications en cryptologie

Application aux réseaux de capteurs

Surveillance par itérations chaotiques

Zone surveillée



Objectifs

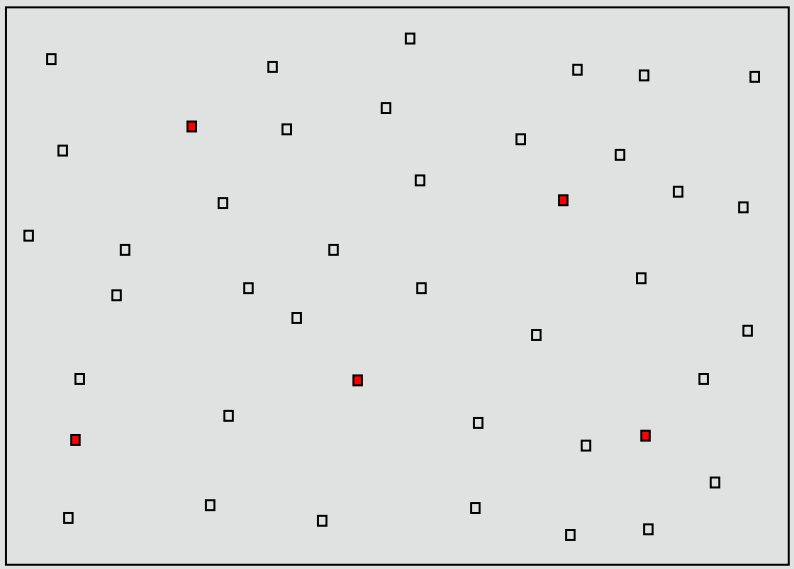
- Surveiller une région
- Capteurs vidéo sur batterie
- Alarme si intrusion

Problématique

- Sécurité (attaques malicieuses)
- Économie d'énergie

Surveillance par itérations chaotiques

Zone surveillée



Objectifs

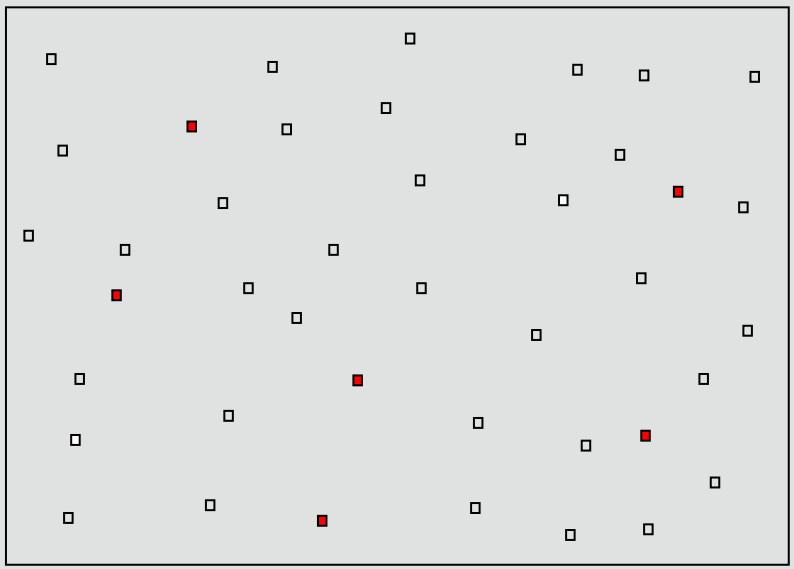
- Surveiller une région
- Capteurs vidéo sur batterie
- Alarme si intrusion

Problématique

- Sécurité (attaques malicieuses)
- Économie d'énergie

Surveillance par itérations chaotiques

Zone surveillée



Objectifs

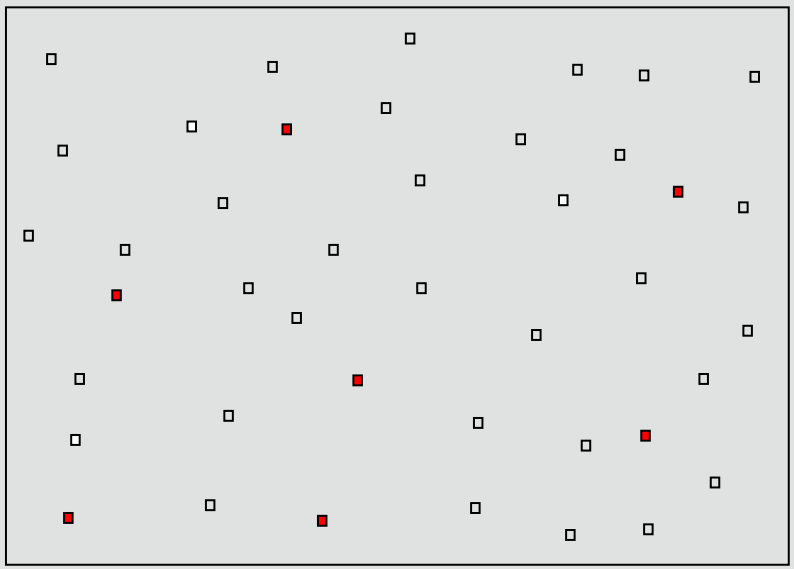
- Surveiller une région
- Capteurs vidéo sur batterie
- Alarme si intrusion

Problématique

- Sécurité (attaques malicieuses)
- Économie d'énergie

Surveillance par itérations chaotiques

Zone surveillée



Objectifs

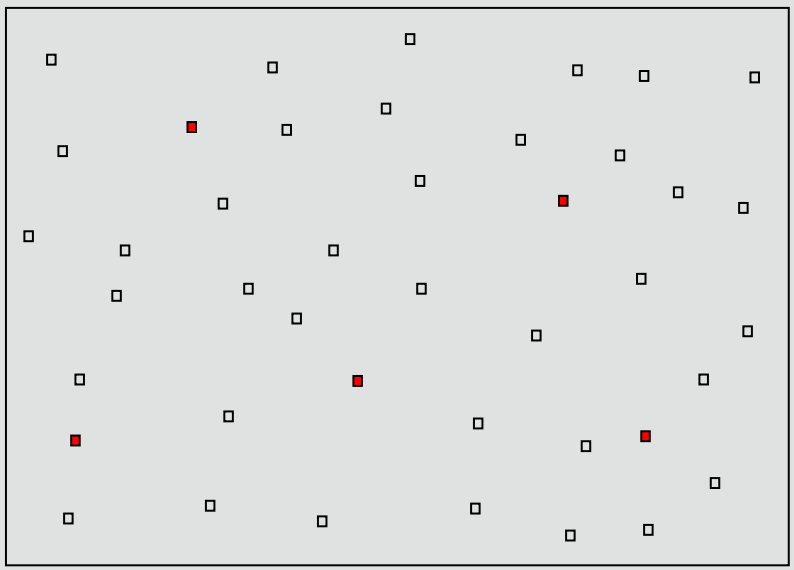
- Surveiller une région
- Capteurs vidéo sur batterie
- Alarme si intrusion

Problématique

- Sécurité (attaques malicieuses)
- Économie d'énergie

Surveillance par itérations chaotiques

Zone surveillée



Principe

- Les nœuds ne surveillent pas tous
- Nœuds à réveiller choisis par IC
- Stratégie obtenue du milieu :
  - Hash de la vidéo = graine
  - Termes obtenus du PRNG

Évaluation de la technique

Intérêt de l'approche par itérations chaotiques

Attaques malicieuses	Chaos
Complexité	Linéaire
Couverture	Transitivités
Énergie	Propriétés statistiques

Merci pour votre attention