UNIVERSITÉ
DE FRANCHE-COMTÉ

Département informatique
UFR ST

École doctorale SPIM

Le désordre des itérations chaotiques et leur utilité en sécurité informatique

THÈSE

présentée et soutenue publiquement le 13 décembre 2010

pour l'obtention du

Doctorat de l'Université de Franche-Comté
(spécialité informatique)

par

Christophe Guyeux

Composition du jury

Président :	Michel de Labachelerie
Rapporteurs :	Pascale Charpin, *Directrice de Recherche, INRIA-Rocquencourt* Éric Filiol, *Professeur, ESIEA - Laval* Pierre Spitéri, *Professeur Émérite, IRIT-ENSEEIHT*
Examinateurs :	Michel de Labachelerie, *Directeur de Recherche CNRS, Université de Franche-Comté* Laurent Larger, *Professeur, Université de Franche-Comté* Jean-Claude Miellou, *Professeur, Université de Franche-Comté* Congduc Pham, *Professeur, Université de Pau*
Directeur :	Jacques M. Bahi, *Professeur, Université de Franche-Comté*

Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC)

LIFC
Laboratoire d'Informatique de l'Université de Franche-Comté

Mis en page avec la classe thloria.

REMERCIEMENTS

Je souhaite avant toutes choses remercier mon directeur de thèse, le Professeur Jacques M. Bahi, pour son encadrement, sa disponibilité et son amitié. Il a su, malgré un emploi du temps bien chargé, toujours être présent à mes côtés, et me faire profiter de son expérience, son intelligence, et sa connaissance si fine des objets de ma recherche. Le travail que j'ai pu mener et ce document ne seraient pas ce qu'ils sont sans sa motivation et ses encouragements, sa patience, son recul, son regard critique, et la pertinence de ses conseils. Ce fut un grand plaisir de travailler avec lui, et j'espère pouvoir continuer à le faire longtemps encore.

Je tiens également à remercier Pascale Charpin, Éric Filiol, Pierre Spitéri, Michel de Labachelerie, Laurent Larger, Jean-Claude Miellou et Congduc Pham, qui ont bien voulu être membres de mon jury de thèse. Et plus particulièrement, merci à P. Charpin, É. Filiol et P. Spitéri qui m'ont fait l'honneur d'être les rapporteurs de cette thèse. Merci pour leurs suggestions et leurs conseils précieux, qui m'ont aidé à améliorer et clarifier ce mémoire.

Je tiens aussi à remercier tous les membres de l'équipe AND pour leur amitié et la bonne ambiance qu'ils contribuent à créer. Je remercie notamment Jean-François Couchot et Michel Salomon pour leurs relectures de qualité et leurs conseils.

Je ne remercierai jamais assez mon frère et mes parents, pour avoir toujours été présents, m'avoir toujours aidé et soutenu, et pas seulement durant mes études. Sans eux, leur gentillesse, leurs encouragements et leur dévouement, je n'en serais pas là. Malgré la distance, ils sont toujours présents à mes côtés.

Et surtout, merci à Nathalie, pour tout.

À Nathalie.

À mes parents et ma filleule,
en souvenir de mon grand-père.

Table des matières

Introduction générale

CHAPITRE 1

Introduction générale

Le sage s'adapte au monde tandis que le fou le transforme.

CONFUCIUS

Ce premier chapitre introduit les notations qui auront cours dans ce document et l'objet de notre étude, à savoir les systèmes itératifs. Une fois ce contexte établi, les réalisations de cette thèse sont détaillées. Notre objectif est de montrer qu'il est possible de développer des programmes au comportement chaotique, ne perdant pas ce dernier une fois implantés en machine. Nous avons appliqué notre approche, dans la mesure du possible, à des cas concrets issus du domaine de la sécurité informatique.

I. NOTATIONS ET CONTEXTE

1. Notations

Dans tout le document, pour prévenir tout conflit et pour éviter des écritures illisibles, nous avons pris pour convention les notations suivantes, utilisées habituellement en mathématiques discrètes :

- Le $n-$ième terme de la suite s sera noté s^n.
- La $i-$ième composante du vecteur v sera notée v_i.
- La $k-$ième composée de la fonction f sera notée $f^{(k)}$. Ainsi, $f^{(k)} = f \circ f \circ \ldots \circ f$, k fois.
- La dérivée de la fonction f sera notée f'.

Exemple 1 : Soit $u : \mathbb{N} \to \mathbb{R}^2$ une suite de $\mathbb{R}^2$. Alors u^0 désigne le premier terme de cette suite ; c'est un vecteur à deux composantes : u_1^0 et u_2^0.

D'autre part $\mathbb{B}$ désignera l'ensemble $\{0;1\}$ muni de ses lois usuelles d'algèbre de Boole (addition, multiplication et négation booléennes), et l'on notera de plus $\wedge$ le *et*, $\vee$ le *ou*, et $\oplus$ le *ou exclusif booléen*. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ et $\mathbb{R}$ sont les notations habituelles des ensembles respectifs suivants : entiers naturels, entiers relatifs, nombres rationnels et nombres réels. $(\mathcal{M}_{m,n}(\mathbb{A}), +, \times, .)$ désignera l'algèbre des matrices à m lignes et n colonnes sur l'algèbre $\mathbb{A}$. L'ensemble $\mathcal{X}^{\mathcal{Y}}$ est l'ensemble des applications de $\mathcal{Y}$ dans $\mathcal{X}$, et

donc $\mathcal{X}^{\mathbb{N}}$ désigne l'ensemble des suites à valeurs dans $\mathcal{X}$. Enfin, $[\![a;b]\!] = \{a, a+1, \ldots, b\}$ est l'ensemble des entiers compris entre a et b.

De plus, nous utiliserons la notation $\lfloor x \rfloor$ pour désigner la partie entière d'un réel x, c'est-à-dire le plus grand entier inférieur à x. $\lceil x \rceil$ désignera pour sa part le plus petit entier supérieur à x.

2. Définitions générales

a. Les systèmes itératifs

Posons la définition suivante :

DÉFINITION 1 (SYSTÈMES ITÉRATIFS) : On appelle *système itératif*, ou simplement *système*, tout couple $\Sigma = (\mathcal{X}, \mathcal{F})$ tel que :

- $\mathcal{X}$ est un ensemble, appelé *ensemble d'états*,
- $\mathcal{F} = (f^n)_{n\in\mathbb{N}^*}$ est une suite de fonctions, appelées *fonctions d'itérations*, telle que $\forall n \in \mathbb{N}^*$, $f^n : \mathcal{X}^n \to \mathcal{X}$. On dit alors que f^n est la fonction d'itération au temps n.

On parle encore d'*itérations de la suite $\mathcal{F}$ sur $\mathcal{X}$*. ◊

Par soucis de concision, nous parlerons d'*itérations de $\mathcal{F}$* sur $\mathcal{X}$, étant entendu qu'ainsi calligraphiée, $\mathcal{F}$ désignera toujours une suite de fonctions.

REMARQUE. Les systèmes itératifs ont fortement à voir avec les « systèmes dynamiques discrets ». Ce terme désigne un certain nombre d'objets différents, suivant la branche des sciences que l'on considère. Ainsi, en mathématiques appliquées, ce terme possède plusieurs sens, qui ont été popularisés par Chazan et Miranker [CM69], Miellou [Mie75a], Bertsekas [BT88], Robert [Rob86], Bahi [Bah91, Bah98], *etc.* Ces définitions peuvent être vues comme des cas particuliers de ce qui précède. D'autres part, l'expression *systèmes dynamiques* existe aussi en topologie, et définit a priori un autre type d'objets. Nous rappellerons cela dans la partie II, avant de voir dans quelle mesure ces approches diffèrent. Pour éviter des ambiguïtés, nous avons préféré opter pour un terme et une définition nouveaux, afin de n'être pas mal interprété.

b. Configurations et itérations

Nous posons maintenant la notion de configuration d'un système.

DÉFINITION 2 (SYSTÈME À CONFIGURATION INITIALE) : On appelle *système* Σ *de configuration initiale* x^0, que l'on note $\Sigma(x^0)$, tout couple (Σ, x^0), où :

- $\Sigma = (\mathcal{X}, \mathcal{F})$ est un système itératif,
- $x^0 \in \mathcal{X}$.

On utilisera aussi la notation $\left(\mathcal{X}, \mathcal{F}, x^0\right)$, lorsque l'ensemble d'états et les fonctions d'itérations seront fournis. ◊

DÉFINITION 3 (CONFIGURATION D'UN SYSTÈME) : Soit $\left(\mathcal{X}, (f^n)_{n\in\mathbb{N}^*}, x^0\right)$ un système itératif de configuration initiale x^0. La *configuration du système au temps* $n \in \mathbb{N}^*$ est la valeur x^n de la suite $x = (x^n)_{n\in\mathbb{N}}$, de premier terme x^0, et définie par la relation de récurrence :

$$\forall n \in \mathbb{N}^*,\ x^n = f^n\left(x^0, \ldots, x^{n-1}\right).$$

On dit aussi que x^n est l'*état du système au temps* n. La suite $x = (x^n)_{n\in\mathbb{N}}$ est quant à elle appelée *suite de configurations*, ou simplement *configurations* du système. ◊

Remarque. Nous comprenons mieux le lien avec les systèmes « discrets » des mathématiques appliquées : l'état global est mesuré à des temps discrets. En d'autres termes, les configurations du système sont définies à l'aide d'une suite.

Définition 4 (Itérations d'un système) : Les itérations du système $\Sigma = (\mathcal{X}, \mathcal{F})$ sont, par définition, l'ensemble des systèmes suivants :

$$\left\{\Sigma(x) \;\middle|\; x \in \mathcal{X}\right\} = \left\{(\mathcal{X}, \mathcal{F}, x) \;\middle|\; x \in \mathcal{X}\right\}.$$

Définition 5 (Itérations (a)synchrones) : Les itérations de $\mathcal{F} = (f^n)_{n\in\mathbb{N}}$ sur $\mathcal{X}$ sont dites *synchrones* si $\forall x \in \mathcal{X}, \forall n \in \mathbb{N}^*$, l'application $(x_1, \dots, x_n) \in \mathcal{X}^n \mapsto f^{n+1}(x_1, \dots, x_n, x)$ est constante.

Dans le cas contraire, on parlera d'*itérations asynchrones*, ou *itérations à retard*. ◊

Remarque. En d'autres termes, dans le cas des itérations synchrones, seule la dernière composante importe pour f^n. Dans ce cas, on peut assimiler f^n à une fonction de $\mathcal{X}$ dans $\mathcal{X}$, ce que l'on fera fréquemment dans la suite de ce document.

Définition 6 (Cellule du système) : Soit $\Sigma = \left(\mathcal{X}, \mathcal{F}, x^0\right)$ un système à condition initiale, et $x = (x^n)_{n\in\mathbb{N}}$ sa suite de configurations. Supposons que $\mathcal{X}$ soit un produit cartésien de N ensembles. La valeur x_i^n est alors appelée *$i-$ième cellule du système au temps n*. ◊

c. Convergence et divergence

Nous nous intéresserons au comportement à terme de tels systèmes, ce qui nous conduit à poser les définitions suivantes :

Définition 7 (Convergence d'un système) : Soit $\Sigma(x^0) = \left(\mathcal{X}, \mathcal{F}, x^0\right)$ un système à condition initiale. On dira que $\Sigma(x)$ est *convergent* si la suite de configurations x^n l'est. Dans le cas contraire, $\Sigma(x^0)$ est dit *divergent*. ◊

Soient $x, y \in \mathcal{X}$. On peut très bien avoir $(\mathcal{X}, \mathcal{F}, x)$ convergent, et $(\mathcal{X}, \mathcal{F}, y)$ divergent. Cela nous incite à introduire la définition suivante :

Définition 8 (Convergence inconditionnelle) : Le système $\Sigma = (\mathcal{X}, \mathcal{F})$ est dit *inconditionnellement convergent* si $\forall x \in \mathcal{X}$, le système $(\mathcal{X}, \mathcal{F}, x)$ converge. Dans le cas contraire, Σ est dit *non convergent*. ◊

Définition 9 (Divergence inconditionnelle) : Le système $\Sigma = (\mathcal{X}, \mathcal{F})$ est dit *inconditionnellement divergent* si $\forall x \in \mathcal{X}$, le système $(\mathcal{X}, \mathcal{F}, x)$ diverge. ◊

II. Objectifs et réalisations

1. Démarche scientifique

Certains systèmes itératifs ont été étudiés par le passé en mathématiques appliquées, mais uniquement pour leur convergence. Il s'agissait jusqu'alors d'établir des conditions suffisantes de convergence [MS85], sous différentes classes d'hypothèses spécifiques : itérations synchrones ou asynchrones [CM69, Mie75a, Bau78], avec ou sans mémoire [Mie75b, Bau78], modèle de Bertsekas [BT88], *etc.* Ces cadres spécifiques ont été définis suivant les besoins exprimés par différents domaines de l'informatique,

tels que les grilles de calcul. Dans chacun de ces cadres, l'assurance de la convergence du système se traduit par l'assurance d'obtenir la solution au problème considéré. Les problèmes qui se posaient alors au mathématicien appliqué étaient, par exemple : « Peut-on être sûr de converger dans la situation qui m'intéresse, et si oui, à quelle vitesse ? », ou bien « Ne peut-on pas accélérer la convergence ? Comment, et sous quelles conditions ? » Ces problèmes sont très utiles dans de nombreux domaines, et très intéressants à étudier. De nombreux résultats ont été établis ces dernières décennies, la théorie est fort avancée, et a été appliquée à de nombreux domaines.

Nous avons choisi pour notre part de nous intéresser au problème inverse, à savoir la recherche de la divergence, du désordre, de l'imprévisibilité – en un mot, du chaos. Nous nous sommes demandés si les systèmes itératifs définis au chapitre précédent ne pouvaient pas se comporter de telle manière, et s'il pouvait y avoir un quelconque intérêt à agir de la sorte.

Il nous a semblé que tel était le cas dans certaines branches de l'informatique, notamment dans la sécurité informatique. Il nous est en effet apparu qu'un programme s'exécutant sur une machine pouvait être vu comme un système :

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

dans lequel $\mathcal{X}$ représente la mémoire de la machine, x^0 les données que l'on y met. La machine réalise dans cette mémoire une succession d'opérations au rythme d'une horloge interne, dépendant éventuellement des valeurs précédemment calculées, l'action de la machine n'étant pas forcément la même à chaque coup d'horloge, ou cycle (ce point de vue sera formalisé au chapitre 18). Nous sommes donc parti de l'hypothèse qu'il peut y avoir un intérêt à utiliser des systèmes itératifs qui soient facilement programmables, et dont le comportement ne soit pas prévisible. Encore faut-il en existe ; pour en trouver, notre démarche a été la suivante.

Il nous a d'abord semblé clair qu'un tel système ne pouvait pas converger inconditionnellement. Nous avons donc traduit, par contraposition, les conditions suffisantes de convergence présents dans les livres de mathématiques, en conditions nécessaires de divergence pour les systèmes itératifs. Notre objectif étant *in fine* de concevoir des programmes, nous nous sommes restreint aux itérations sur $\mathcal{X}$ de la forme $\mathbb{B}^{\mathbb{N}}$. De plus, nous avons privilégié la simplicité pour initier cette recherche et, pour cette raison, nous nous sommes jusqu'à présent exclusivement concentrés sur les itérations synchrones sur $\mathbb{B}^{\mathbb{N}}$. Ces itérations sont plus faciles à étudier mathématiquement, à programmer réellement, et il sera toujours temps de passer à des situations moins élémentaires si nous n'obtenons pas satisfaction.

Ce cadre étant établi, nous avons pu ainsi obtenir des conditions nécessaires de divergence, qui nous ont permis de trouver quelques exemples canoniques de systèmes non convergents, le plus emblématique étant les itérations chaotiques avec la négation vectorielle. La partie I de ce document fait état de cette recherche.

Seulement, une multitude de systèmes non convergents sont très largement prévisibles. À ce stade, nous savions ce que nous ne voulions pas (la convergence), mais nous ignorions ce que nous recherchions : il nous fallait préciser ce que pouvait bien vouloir dire « imprévisible » et « désordonné » pour de tels systèmes dynamiques. Le nom des objets retenus, les itérations chaotiques, nous ont mis sur la voie. La partie II fait l'objet de cette précision : elle a pour cadre la théorie mathématique du chaos, qui offre diverses définitions topologiques correspondant à ce que l'on entend habituellement par imprévisibilité, désordre et chaos, pour les systèmes dynamiques. Notons dès à présent que l'adjectif chaotique a historiquement été choisi par les mathématiciens appliqués, pour signifier que ces itérations pouvaient avoir un comportement irrégulier, mais jusqu'à présent cet adjectif n'avait aucun lien avec la théorie du même nom, et le rapprochement n'avait jamais été fait.

Dès lors, il nous fallait étudier les itérations chaotiques dans le cadre de la théorie mathématique du chaos. Le problème que nous avons rencontré est que les systèmes itératifs des mathématiques appliquées ne sont pas les systèmes dynamiques de la topologie. Il nous fallut donc reformuler les itérations chaotiques sous une nouvelle forme, compatible avec les objets d'étude de la théorie du chaos. D'autre part, qui dit étude topologique, dit structure topologique, et $\mathcal{X}$ en était démuni. Nous avons donc dû réécrire les itérations chaotiques, munir $\mathcal{X}$ d'une topologie pertinente, et vérifier que ces itérations chaotiques étaient bien continues pour cette dernière. L'étude du chaos des itérations chaotiques était alors possible, et les résultats ont dépassé nos attentes. Ils sont reproduits dans la partie III.

L'étude des propriétés de désordre, de non convergence et d'irrégularité des systèmes itératifs tels qu'ils sont utilisés en mathématiques appliquées, réalisée avec les outils de ces mathématiques appliquées et sous l'angle de la théorie mathématique du chaos, constitue le premier apport théorique de ce travail de recherche.

La question de l'utilisation de ces systèmes itératifs chaotiques s'est alors posée, l'enjeu consistant à ne rien perdre des propriétés de chaos, une fois sur machine. Il nous a semblé que deux raisons pouvaient induire qu'une fonction chaotique perde ses propriétés lors de son implémentation : d'une part, le fait qu'une machine ne manipule que des nombres (dits *nombres machines*) qui sont des approximations des réels, et d'autre part le problème de la finitude de la machine. Le premier problème n'était pas à considérer, vu que nous itérions sur $\mathbb{B}^{\mathbb{N}}$. Quant au second, nous l'avons solutionné en considérant que la machine ne travaille pas en vase clos, mais peut lire de nouvelles données à chaque itération. Ce faisant, nous n'avons plus une machine avec un nombre fini d'états, qui boucle forcément indéfiniment, mais un programme informatique aux devenirs infinis.

En d'autres termes, nous avons été capables d'écrire des programmes qui se comportent réellement de manière chaotique, au sens mathématique le plus rigoureux qui soit. Il ne s'agit pas, comme il en a été question jusqu'à présent, d'un « chaos discret », d'une approximation du chaos, ou d'une fonction chaotique que l'on programme ensuite sans trop se poser de question. Mais d'un programme dont on peut prouver mathématiquement qu'il se comporte de manière chaotique, aux sens topologiques forts du terme. Ce point, qui constitue l'apport pratique majeur de ce travail de recherche, est notamment détaillé au chapitre 18, et dans les parties IV et V.

Comme nous l'avons signalé ci-dessus, dans notre recherche d'applications à notre théorie, il nous est de prime abord apparu que ces systèmes itératifs chaotiques pouvaient notamment apporter un certain « quelque chose » à la sécurité informatique. Supposons par exemple qu'un programme ne soit pas quelconque, mais qu'il manipule des données sensibles. Nous pensons qu'alors il pourrait y avoir des situations dans lesquelles ce programme ne devrait pas être trop prévisible. Un adversaire ayant accès, ne serait-ce qu'un instant, à certains calculs de la machine sur laquelle s'exécute ce programme, ne devrait pas être en mesure d'exploiter cette connaissance – en déterminant, par exemple, quel pourrait en être le rendu final. De tels exploits ne sont pas si improbables, notamment dans les attaques par canal auxiliaire. Ces *attaques* font partie d'une vaste famille de techniques cryptanalytiques, qui exploitent des propriétés inattendues d'un algorithme de cryptographie lors de son implémentation logicielle ou matérielle. Ils tirent leur raison d'être du fait qu'une sécurité « mathématique » d'un problème ne garantit pas forcément une sécurité de fait lors de l'utilisation pratique d'un programme informatique le mettant en œuvre : ces attaques exploitent des failles logicielles ou matérielles dans l'utilisation d'un algorithme, pourtant prouvé sûr [1] en théorie. Donnons-en quelques exemples :

1. Les auteurs de [AcKKS07] sont parvenus en 2006 à acquérir, en quelques millièmes de secondes, 508 bits sur 512 d'une clé cryptée par RSA. Ils se sont appuyés sur une faille

1. Dans la suite du document, le terme « sûr(e) » fera référence à la notion de sécurité, et non à celle de sûreté informatique (qui stipule que rien de mal ne peut arriver)

de sécurité dans la technologie des processeurs superscalaires : dans certaines conditions, certains processeurs peuvent laisser transparaître de l'information visiblement exploitable.

2. Les auteurs de [PBA10] ont récemment réussi à émettre une signature (RSA 1024) corrompue en influant sur la tension appliquée au processeur : ils ont ainsi généré une erreur matérielle par cycle d'horloge, ce qui a permis de récupérer les bits de la clé privée un par un.
3. En 1995, Paul Kocher a présenté une attaque contre RSA permettant d'obtenir la clé de déchiffrement lorsque l'attaquant possède certaines informations sur les documents chiffrés, et qu'il est de plus capable de mesurer les temps de déchiffrement de plusieurs cryptogrammes [Koc95].
4. D'autres exemples d'attaque par chronométrage peuvent être trouvés dans [BB03]. Dans cet article, Boneh et Brumley ont retrouvé la factorisation RSA sur une connexion, en utilisant les informations que laissent filtrer certaines optimisations appliquées au théorème des restes chinois.

De telles attaques, se développant fortement ces dernières années [QSLC02], ne révèlent fréquemment qu'une partie du secret. Les questions sont alors : « Comment se prémunir contre ce genre d'attaques, et comment réduire les dégâts si elles réussissent en partie ? » La solution consistant à construire, au niveau logiciel ou matériel, des contre-mesures adaptées à chaque nouvelle attaque, n'est pas pleinement satisfaisante, et a tendance depuis peu à laisser la place à des approches plus formelles et universelles [DP08]. Or, dans toutes ces attaques par canaux auxiliaires, il y a un intérêt à observer la machine, car on peut tirer quelque chose de son comportement, et espérer que ce quelque chose soit exploitable. Nous pensons qu'une approche visant à rendre les programmes imprévisibles pourrait avoir son intérêt, et mériterait au moins d'être étudiée. Cette approche ne serait pas un remplacement, mais un complément aux propriétés communément requises pour ce genre d'objets (sécurité sémantique, calculatoire, *etc.*)

Dans les faits, ce point de vue s'est traduit par une nouvelle définition de sécurité, complémentaire aux existantes. Cette dernière requiert d'un programme qu'il soit chaotique, et que l'observation ponctuelle de son action ne serve à rien. Peu importe son déterminisme, et notre connaissance de son fonctionnement (*i.e.*, peu importe si l'on a accès à son code), en l'absence d'une clé, l'observation du travail de la machine ne mène à rien. Cette sécurité, nous l'avons formulée au chapitre 20 de la partie IV, dans un cadre particulier, à savoir la dissimulation d'informations (data hiding), car elle nous est initialement apparue dans ce contexte. Nous avons montré que notre notion complémentaire de sécurité est « prête à l'emploi » et utile, en comparant deux algorithmes de dissimulation donnés : le nôtre, et le plus sûr du marché. Nous avons pu les comparer pour certaines classes d'attaques qui jusqu'à présent ne pouvait pas être étudiées, par manque d'outil adéquat. Nous avons montré que notre algorithme était le meilleur dans certaines de ces situations, prouvant conséquemment que notre approche était utile. Ces différentes avancées sont l'objet de la partie IV.

Nous considérons cela comme notre second apport théorique. Jusqu'alors, la théorie du chaos était utilisée en informatique pour construire, avec une réussite relative, des programmes se voulant sûrs. Nous estimons que cette théorie peut servir non pas à la construction, mais à l'évaluation d'un certain niveau de sécurité.

Nous n'avons pas voulu nous restreindre à la dissimulation d'informations, et nous avons souhaité proposer d'autres applications dans le domaine de la sécurité informatique. Les fonctions de hachage nous ont semblé particulièrement adaptées à cet objectif pour diverses raisons, notamment parce qu'il est aisé d'en construire à partir des itérations chaotiques, et parce que les propriétés requises pour de telles fonctions nous évoquent d'autres propriétés définies dans la théorie du chaos. Nous avons proposé dans la partie V un algorithme de hachage au comportement chaotique, et nous avons initié son étude.

Enfin, un dernier domaine nous a semblé intéressant à étudier, à savoir les réseaux de capteurs. Ces derniers se modélisent directement à l'aide de systèmes itératifs, et leur utilisation dans certains contextes soulève de nombreux problèmes de sécurité. En réalisant l'état de l'art, nous nous sommes aperçus qu'un problème, dit *agrégation sécurisée des données au sein des réseaux de capteurs sans fils*, n'avait pas encore de solution satisfaisante, mais qu'un cryptosystème que nous avions croisé dans nos lectures pouvait résoudre ce problème. Nous avons donc proposé de l'utiliser, expliqué comment le faire, et mené des expérimentations, permettant ainsi d'apporter une solution à cette agrégation sécurisée. Nous avons ensuite poursuivis nos recherches dans ce domaine, en adaptant nos réflexions concernant la dissimulation de données aux problèmes de sécurité et d'agrégation de données authentifiées dans les réseaux de capteurs. La partie VI contient notre contribution dans ce domaine.

2. Plan détaillé de cette thèse

Partie I : Approche « mathématiques discrètes de la divergence ». On présente dans cette partie quelques exemples classiques d'itérations sur un système, dans le cas particulier où l'ensemble des états est fini. Nous commençons par faire le point sur le comportement asymptotique des itérations parallèles, et nous constatons que ces dernières sont trop prévisibles pour nos besoins. Nous nous concentrons ensuite sur les itérations dites « chaotiques », et nous remarquons qu'elles ont un meilleur potentiel, qu'elles rendent possibles des comportements non convergents très riches. Nous verrons quelles sont les conditions nécessaires pour que de tels comportements apparaissent, avant d'en tirer un exemple canonique : la fonction dite de « négation vectorielle ». Les outils utilisés dans cette partie sont principalement issus des mathématiques discrètes et appliquées : théorie des graphes, calcul matriciel, résultat de convergence sous hypothèse de contraction, *etc.*

Partie II : Introduction à la Théorie du Chaos. De la partie précédente, nous avons tiré certaines conditions nécessaires de non convergence pour des itérations chaotiques, qui ne se terminent alors ni par un cycle, ni par un point fixe. Cela ne veut pas forcément dire pour autant que leur comportement n'est pas prévisible. Pour affirmer un tel résultat, il nous faut des outils mesurant le caractère prévisible ou chaotique de tels systèmes. C'est le rôle de la théorie mathématique du chaos, et l'objet de cette partie est justement de la rappeler. Après quelques rappels de topologie, on donne diverses définitions de la notion de chaos : telle qu'elle est définie par Devaney, par Li et Yorke, par Knudsen, *etc.* On en rappelle quelques variantes, et l'on fournit aussi d'autres outils permettant de mesurer le caractère erratique d'un système itératif : expansivité, entropie topologique, exposant de Lyapunov, *etc.* Les raisons d'une telle prolifération de notions de chaos seront expliquées dans ce chapitre.

Partie III : Apport théorique. Nous reprenons nos itérations chaotiques de la partie I, et nous prouvons qu'elles portent bien leur nom. Elles sont effectivement du chaos selon Devaney, Li et Yorke... Toutes les notions rappelées dans la précédente section sont étudiées, et l'on s'aperçoit que le chaos de ces itérations est très prononcé. Cela est vrai pour la négation vectorielle, mais pas pour une fonction d'itération quelconque : on en profitera donc pour caractériser et dénombrer ces fonctions au comportement chaotiques. Pour cela, nous avons commencé par réécrire les itérations chaotiques dans le cadre de la théorie mathématique du chaos, notamment les modéliser sur un espace topologique, que l'on étudiera en détail. Cette partie contient aussi une discussion concernant l'impact de la modélisation, et du choix de la topologie qui a été la nôtre. Nous verrons notamment comment se ramener à des résultats sur la droite réelle. La partie se termine par une discussion concernant le problème de l'implantation des systèmes chaotiques en machine. Nous verrons notamment qu'il est possible de concevoir des programmes réellement chaotiques.

Partie IV : Applications à la science de l'information dissimulée. Après avoir rappelé en quoi consistent les techniques de dissimulation de l'information (tatouage et stéganographie), on en dresse un état de l'art, principalement centré sur les notions de robustesse et de sécurité. Nous introduisons ensuite notre propre notion de sécurité, dite *chaos-sécurité*, nous en expliquons son intérêt et en quoi elle nous semble pertinente. Nous comparerons cette nouvelle notion à celles déjà existantes, et nous établirons que l'étude de chaos-sécurité est toujours faisable. Pour illustrer notre propos, nous mènerons l'étude de chaos-sécurité de deux algorithmes : l'étalement de spectre, et le dhCI, une méthode de notre cru que l'on prouvera stégo-sûre et raisonnablement robuste. Nous tirerons les conséquences de cette comparaison, prouvant ainsi que le dhCI permet de contrer plus d'attaques que l'étalement de spectre.

Partie V : Application aux fonctions de hachage. Nous présenterons dans cette partie la manière d'utiliser les itérations chaotiques pour en faire des fonctions de hachage. Nous évoquerons en quoi l'approche chaos nous semble pertinente dans ce domaine. Nous proposerons ensuite notre propre fonction de hachage, et en détaillerons ses propriétés, telles qu'elles peuvent être déduites de notre étude théorique précédemment menée.

Partie VI : Application aux réseaux de capteurs. Nous nous intéresserons enfin au problème particulier de l'agrégation sécurisée des données au sein des réseaux de capteurs sans fil. Ces réseaux ne sont rien d'autre que des systèmes itératifs. Les problèmes de sécurité étant sensibles dans ces derniers, nous y avons vu là une possible application à notre théorie. Nous expliquerons dans cette partie en quoi consiste les réseaux de capteurs sans fil, et ce que signifie l'agrégation sécurisée des données au sein de ces réseaux. Nous rappellerons quelles ont été les approches jusqu'à présent, et l'on introduit deux solutions originales. La première sera basée sur un cryptosystème *presque* complètement homomorphe sur courbes elliptiques. La seconde utilisera des techniques de dissimulation d'information pour garantir l'authenticité des données, même après agrégation. Nous présenterons enfin les résultats de nos simulations.

III. Publications issues de cette thèse

1. Revue internationale

1. Jacques M. Bahi and Christophe Guyeux. Hash Functions Using Chaotic Iterations. Journal of Algorithms & Computational Technology, 4(2) :167–181, 2010. [GB10].

2. Actes de conférences internationales sélectives

2. Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. A novel pseudo-random generator based on discrete chaotic iterations. In Internet 2009, pages 71–76, Cannes, France, August 2009. IEEE. [BGW09].
3. Jacques M. Bahi and Christophe Guyeux. A new chaos-based watermarking algorithm. In SECRYPT 2010, International conference on security and cryptography, pages ***–***, Athens, Greece, 2010. [BG10b].
4. Jacques M. Bahi and Christophe Guyeux. Topological chaos and chaotic iterations, application to hash functions. In WCCI'10, IEEE World Congress on Computational Intelligence, pages ***–***, Barcelona, Spain, July 2010. IEEE. Best paper [BG10c].
5. Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul. Efficient and robust secure aggregation of encrypted data in sensor networks. In SENSORCOMM 2010, The Fourth

International Conference on Sensor Technologies and Applications, pages 472–477, Venice, Italy, July 2010. IEEE. [BGM10a].

6. Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul. Secure data aggregation in wireless sensor networks. Homomorphism versus watermarking approach. In ADHOC-NETS 2010, 2nd Int. Conf. on Ad Hoc Networks, volume * of Lecture Notes in ICST, pages ***–***, Victoria, Canada, August 2010. To appear in the Springer LNISCT series. [BGM10b].
7. Jacques M. Bahi and Christophe Guyeux. An improved watermarking algorithm for internet applications. In INTERNET'10. The 2nd Int. Conf. on Evolving Internet, pages 119–124, Valencia, Spain, September 2010. IEEE seccion ESPANIA. [BG10a].
8. Qianxue Wang, Jacques M. Bahi, Christophe Guyeux, and Xaole Fang. Randomness quality of CI chaotic generators. Application to internet security. In INTERNET'10. The 2nd Int. Conf. on Evolving Internet, pages 125–130, Valencia, Spain, September 2010. IEEE seccion ESPANIA. Best paper. [WBGF10].
9. Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. A pseudo random numbers generator based on chaotic iterations. Application to watermarking. In WISM 2010, Int. Conf. on Web Information Systems and Mining, pages 202–211, Sanya, China, October 2010. LNCS series. [BGW10b].
10. Christophe Guyeux, Nicolas Friot, and Jacques M. Bahi. Chaotic iterations versus Spread-spectrum : chaos and stego security. In IIH-MSP'10, 6-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Darmstadt, Allemagne, pages 208–211, Octobre 2010. [GFB10].
11. Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. Improving random number generators by chaotic iterations. Application in data hiding. In ICCASM 2010, Int. Conf. on Computer Application and System Modeling, volume 13, pages 643–647, Taiyuan, China, October 2010. IEEE. [BGW10a].

3. Divers

12. Jacques M. Bahi, Abdallah Makhoul, and Christophe Guyeux. Efficient and Robust Secure Aggregation of Encrypted Data in Sensor Networks for critical applications. Dans RESSACS, Journée thématique PHC/ResCom sur RESeaux de capteurS et Applications Critiques de Surveillance, Bayonne, France, Juin 2010. Note : Communication orale.
13. Jacques M. Bahi, Jean-François Couchot, Olivier Grasset, and Christophe Guyeux. Discrete Dynamical Systems : Necessary Divergence Conditions for Synchronous Iterations. Rapport de recherche RR2010-04, LIFC.

Première partie

Approches mathématiques appliquées de la divergence

INTRODUCTION À L'APPROCHE MATHÉMATIQUE DISCRÈTE

Nous souhaitons réussir à faire fonctionner un ordinateur de manière imprévisible. Vue que sa mémoire est un vecteur de bits et que la réalisation de ses calculs est dictée par une horloge interne, il nous a semblé que le cadre d'étude le plus naturel était les mathématiques discrètes, que l'objet d'étude le plus pertinent était les systèmes itératifs sur $\mathbb{B}^{\mathsf{N}}$ et que la première contrainte était d'éviter la convergence inconditionnelle.

De plus, nous souhaitions que les programmes correspondant à ces systèmes itératifs soient exploitables en pratique (donc rapides), et que les systèmes associés puissent être étudiés en profondeur. En d'autres termes, nous recherchions des systèmes itératifs sur $\mathbb{B}^{\mathsf{N}}$ « simples », mais au comportement imprévisible. Les plus simples de ces systèmes itératifs sont les itérations séries et parallèles. Elles ont de ce fait été le point de départ de notre recherche.

CHAPITRE 2

Quelques exemples d'itérations élémentaires

> Commençons par me faire un magasin d'idées, vraies ou fausses, mais nettes, en attendant que ma tête en soit assez fournie pour pouvoir les comparer et choisir.
>
> *Les confessions*
> JEAN-JACQUES ROUSSEAU

On présente dans ce chapitre quelques exemples classiques d'itérations sur un système, *dans le cas particulier ou l'ensemble des états est fini*. Ce faisant, on restreint grandement le cadre général des systèmes itératifs, mais l'on se rapproche du cadre spécifique, tant théorique que pratique, de nos travaux de recherche. On en profite pour introduire des représentations graphiques pour les itérations, permettant une illustration plus aisée du propos.

Ces modèles sont parmi les plus simples qui soient. Ils nous permettront quand même de mieux saisir le fonctionnement des systèmes itératifs en mathématiques appliquées. Le lecteur voulant approfondir le sujet pourra se reporter au livre de François Robert [Rob86], à la thèse de Jacques M. Bahi [Bah91] ou à son HDR [Bah98].

Dans tout le chapitre, $\mathcal{X}$ désignera un ensemble fini d'états de la forme $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_{\mathsf{N}}$, où $\mathsf{N} \in \mathbb{N}^*$ est éventuellement égal à 1. On va introduire le mode d'itérations parallèles, et ses dérivées : série, et séries-parallèles. Ces modes seront pour nous l'occasion de présenter divers outils utiles à leur compréhension.

Ce chapitre doit beaucoup au cours de Jean-François Couchot, que j'ai utilisé avec son aimable autorisation. Le lecteur voulant approfondir le sujet pourra se rapporter au livre de François Robert [Rob86].

I. LES ITÉRATIONS PARALLÈLES

Parmi les itérations les plus simples qui soient se trouvent les itérations parallèles, définies notamment dans [Rob86]. Nous les reformulons ici à partir de notre définition 1 des systèmes itératifs :

DÉFINITION I.1 (ITÉRATIONS PARALLÈLES) : On appelle *itérations parallèles* de f sur $\mathcal{X}$ toute itération synchrone $\Sigma = (\mathcal{X}, \mathcal{F})$ telle que la suite $\mathcal{F}$ est constante égale à $f : \mathcal{X} \to \mathcal{X}$. On parle encore d'itérations parallèles $(\mathcal{X}, f)$. ◊

Les configurations de $\Sigma(x^0)$ s'écrivent alors :

$$x^{n+1} = f(x^n).$$

En d'autres termes, à chaque itérée, on réactualise *toutes* les cellules du système, en utilisant toujours la même fonction $f : \mathcal{X} \to \mathcal{X}$, et uniquement la dernière configuration du système.

Exemple I.1 : Soit $\mathcal{X} = \mathbb{B}^3$, et $f : (x_1, x_2, x_3) \in \mathcal{X} \mapsto (x_1\overline{x_2} + x_3, x_1 + \overline{x_3}, x_2x_3)$.

$\mathcal{X}$ contient 8 états, de $(0,0,0)$ à $(1,1,1)$, que l'on numérote naturellement de 0 à 7 suivant l'écriture de ces chiffres en base 2. Les itérations parallèles de f sur $\mathcal{X}$ peuvent se décrire par la *table de transition* suivante, dont le sens est immédiat :

	x			$f(x)$		
0	0	0	0	0	1	0
1	0	0	1	1	0	0
2	0	1	0	0	1	0
3	0	1	1	1	0	1
4	1	0	0	1	1	0
5	1	0	1	1	1	0
6	1	1	0	0	1	0
7	1	1	1	1	1	1

NOTATION I.1. On utilisera parfois l'acronyme IP pour « itérations parallèles ».

II. INTRODUCTION DE QUELQUES OUTILS

La présentation des itérations parallèles est prétexte à l'exposé de quelques outils, telle la table de transition de l'exemple I.1. Ces outils permettent de mieux comprendre le comportement de certaines itérations.

1. Graphe d'itérations

Lorsque $\mathcal{X}$ est fini et que sa taille est raisonnable, les itérations parallèles de f sur $\mathcal{X}$ peuvent être représentées par un *graphe d'itérations*. Commençons tout d'abord par quelques rappels.

a. Rappels

DÉFINITION I.2 (GRAPHE ORIENTÉ, SOMMETS, ARCS) : Un *graphe orienté* est un couple (S, A), où :

- S est un ensemble fini dont les éléments sont appelés *sommets*, ou *nœuds*.
- A est un ensemble fini de paires (ordonnées) de sommets. Les éléments de A sont appelés *arcs*.

Lorsque $a = (s_1, s_2) \in A$, on dira que l'arc a va de s_1 à s_2. On dit aussi que s_1 est l'extrémité initiale et s_2 l'extrémité finale de a. ◊

Quand on passe d'un sommet à un autre sommet par une suite d'arcs, on parle de chemin :

DÉFINITION I.3 (CHEMIN) : Un *chemin* conduisant du sommet a au sommet b est une suite de la forme $(v_0, e_1, v_1, e_2, v_2, ..., e_k, v_k)$ où les v_i sont des sommets, $v_0 = a$ et $v_k = b$, et les e_i sont des arcs tels que e_i va de v_{i-1} à v_i. ◊

Un graphe orienté peut se représenter par une matrice d'incidence.

DÉFINITION I.4 (MATRICE D'INCIDENCE) : La *matrice d'incidence* d'un graphe orienté à n sommets est une matrice booléenne avec n lignes et n colonnes, telle que l'élément b_{ij}, situé au croisement entre la i^{eme} ligne et la j^{eme} colonne, est défini par :

- $b_{ij} = 1$ s'il existe un arc orienté de j vers i,
- $b_{ij} = 0$ sinon. ◊

b. Les graphes d'itérations

On peut alors définir ce qu'est un graphe d'itérations :

DÉFINITION I.5 (GRAPHE D'ITÉRATIONS) : Le *graphe d'itérations* des itérations parallèles $(\mathcal{X}, f)$, est un graphe orienté tel que :

- chaque sommet représente une configuration $x \in \mathcal{X}$.
- une transition d'une configuration $x \in \mathcal{X}$ vers une configuration $x' = f(x) \in \mathcal{X}$ est représentée par une flèche (arc orienté) du nœud représentant x vers le nœud représentant x'. ◊

Exemple I.2 : Les transitions des itérations parallèles de l'exemple I.1 sont représentées par le graphe d'itérations de la figure 2.1(b). En effet, par exemple, l'image de $(0, 0, 1)$ par f est $(1, 0, 0)$. Donc, en base 10, f envoie le sommet 1 sur le sommet 4.

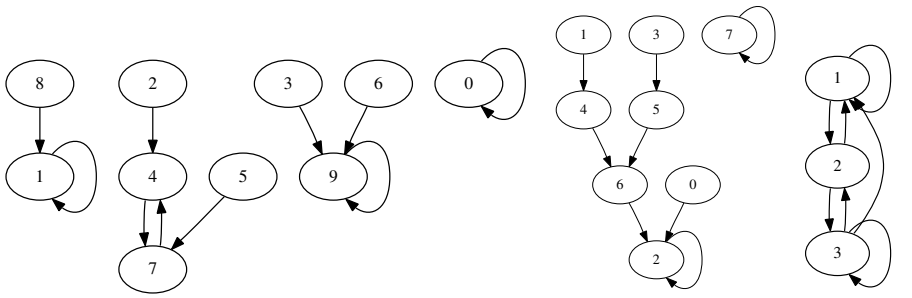
Exemple I.3 : On considère un système à 10 configurations : $\mathcal{X} = \{0, \ldots, 9\}$. Pour chaque configuration $x \in \mathcal{X}$, $f(x)$ est définie par : « on calcule x^2, dont on additionne les chiffres, en recommençant si nécessaire pour obtenir un nombre x' dans $\mathcal{X}$ ». Le graphe d'itérations de ce système est donné à la figure 2.1(a).

2. Graphe de connexion

a. Définition et premier exemple

Dans le cas des itérations parallèles de f sur $\mathcal{X}$, f envoie un élément $(x_1, \ldots, x_N)$ de $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_N$ vers un autre élément de $\mathcal{X}$. Son image $f(x_1, \ldots, x_N)$ est à valeurs dans ce produit, et est de ce fait un multiplet :

NOTATION I.2. On notera $f(x_1, \ldots, x_N) = (f_1(x_1, \ldots, x_N), \ldots, f_N(x_1, \ldots, x_N))$, en relevant bien que cette notation n'a plus rien à voir, dans ce contexte, avec le $n-$ième terme de la suite de fonction $\mathcal{F}$.



(a) Graphe d'itérations de l'exemple I.3. (b) Graphe d'itérations de l'exemple I.1. (c) Connexions de I.4.

FIGURE 2.1 – Graphes d'itérations et de connexion.

Pour chaque cellule du système Σ soumis aux itérations de f, l'ensemble des cellules qui vont effectivement lui apporter de l'information va être donné par un graphe, nommé *graphe de connexion* du système :

DÉFINITION I.6 (GRAPHE DE CONNEXION) : On appelle *graphe de connexion* des itérations parallèles de $f = (f_1, \ldots, f_{\mathsf{N}})$ sur $\mathcal{X}$, le graphe orienté :

- ayant N sommets,
- tel qu'il existe un arc du sommet i vers le sommet j si, et seulement si il existe $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{\mathsf{N}}) \in \mathcal{X}^{\mathsf{N}-1}$, tel que l'application

$$x \in \mathcal{X}_i \to f_j(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_{\mathsf{N}})$$

ne soit pas constante.

Le graphe de connexion de f est noté $G(f)$. ◊

Ce graphe de connexion est fréquemment représenté par sa *matrice d'incidence* :

Exemple I.4 : Soit $\mathsf{N} = 3$, $\mathcal{X} = \mathbb{B}^3$, et la fonction f de l'exemple I.1, que l'on peut écrire :

$$\begin{aligned} f_1(x_1, x_2, x_3) &= x_1\overline{x_2} + x_3 \\ f_2(x_1, x_2, x_3) &= x_1 + \overline{x_3} \\ f_3(x_1, x_2, x_3) &= x_2 x_3. \end{aligned}$$

Son graphe de connexion est donné à la figure 2.1(c). Il se comprend de la manière suivante : la cellule 1 du système reçoit de l'information des cellules 1, 2, et 3. La cellule 2 reçoit de l'information exclusivement des cellules 1 et 3. Enfin, la cellule numéro 3 reçoit de l'information de la cellule 2 et d'elle-même.

La matrice d'incidence de ce graphe de connexion est la suivante : $B(f) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$.

Remarquons que nous avons introduit une notation dans l'exemple précédent :

NOTATION I.3. $B(f)$ désignera la matrice d'incidence du graphe de connexion de f. On emploiera simplement l'expression « matrice d'incidence de f » pour $B(f)$.

b. Particularités du graphe de connexion

Le graphe de connexion donne de premiers renseignements sur la « complexité » d'itérations fixées :

DÉFINITION I.7 (CONNEXION TOTALE) : Le graphe de connexion est dit *totalement connecté* si chaque cellule reçoit de l'information de toutes les cellules. ◊

REMARQUE. La matrice d'incidence correspondante est alors « pleine » de 1. En termes issus de la théorie des graphes, nous dirons que $G(f)$ est complet.

De manière équivalente,

DÉFINITION I.8 (CONNEXION FAIBLE) : Un graphe de connexion est *faiblement connecté* si chaque cellule ne dépend que d'un « petit nombre » de cellules. ◊

Cela se représente par une matrice d'incidence très *creuse* : beaucoup de 0 pour peu de 1.

III. AUTRES MODES ÉLÉMENTAIRES D'ITÉRATIONS

Nous rappelons ici la définition d'autres modes élémentaires classiques d'itérations sur un ensemble fini d'états. Les représentations graphiques introduites dans la précédente section pourront encore être utilisées avec ces modes.

1. Itérations séries

a. Définition

Dans les itérations parallèles, toutes les cellules évoluent simultanément. On présente ici un autre mode opératoire élémentaire, dit *série* : à partir d'une configuration x^n, on fait d'abord évoluer la première cellule, puis la deuxième (en fonction du résultat de l'opération précédente), *etc.* Une fois arrivé à la dernière cellule, la nouvelle configuration obtenue sera x^{n+1}. D'où :

DÉFINITION I.9 (MODE SÉRIE) : Soit $f : \mathcal{X} \to \mathcal{X}$, où $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_\mathsf{N}$ est un ensemble fini d'états. Les *itérations séries* de f sont la suite de configurations $x \in \mathcal{X}^{\mathbb{N}}$ dont le premier terme x^0 est donné, et dont le terme $n+1$ se calcule ainsi :

$$\left\{\begin{array}{rcl} x_1^{n+1} & = & f_1(x_1^n, x_2^n, \ldots, x_\mathsf{N}^n) \\ x_2^{n+1} & = & f_2(x_1^{n+1}, x_2^n, \ldots, x_\mathsf{N}^n) \\ & \vdots & \\ x_\mathsf{N}^{n+1} & = & f_\mathsf{N}(x_1^{n+1}, x_2^{n+1}, \ldots, x_{\mathsf{N}-1}^{n+1}, x_\mathsf{N}^n) \end{array}\right.$$

b. Lien série/parallèle

Les itérations séries peuvent être vues comme des itérations parallèles [Rob86], de la manière suivante :

PROPOSITION I.1 : *Les itérations séries sont en fait des méthodes d'itérations parallèles pour un opérateur noté $G : \mathcal{X} \to \mathcal{X}$ associé à f, et défini par :*

$$G = F_N \circ \ldots \circ F_2 \circ F_1.$$

où l'on a posé, $\forall i \in [\![1, N]\!]$:

$$\begin{array}{rccc} F_i : & \mathcal{X} & \longrightarrow & \mathcal{X} \\ & (x_1, \ldots, x_N) & \longmapsto & (x_1, \ldots, x_{i-1}, f_i(x_1, \ldots, x_N), x_{i+1}, \ldots, x_N). \end{array}$$

PREUVE : *En effet, pour $x = (x_1, x_2, \ldots, x_N)$ de $\mathcal{X}$*

$$\begin{array}{rcl} g_1(x_1, x_2, \ldots, x_i, \ldots, x_N) & = & f_1(x_1, x_2, \ldots, x_i, \ldots, x_N) \in \mathcal{X}_1 \\ g_2(x_1, x_2, \ldots, x_i, \ldots, x_N) & = & f_2(g_1(x), x_2, \ldots, x_i, \ldots, x_N) \in \mathcal{X}_2 \\ & \vdots & \\ g_i(x_1, x_2, \ldots, x_i, \ldots, x_N) & = & f_i(g_1(x), g_2(x), \ldots, g_{i-1}(x), x_i, \ldots, x_N) \in \mathcal{X}_i \\ & \vdots & \\ g_N(x_1, x_2, \ldots, x_i, \ldots, x_N) & = & f_N(g_1(x), g_2(x), \ldots, g_i(x), \ldots, g_{N-1}(x), x_N) \in \mathcal{X}_N \end{array}$$

□

Les itérations séries sont donc des itérations parallèles, et l'on peut utiliser les graphes d'itérations et de connexion pour les représenter.

Exemple I.5 : On reprend l'exemple I.1. Dans cette situation, on en déduit $G : \mathbb{B}^3 \to \mathbb{B}^3$ définie par :

$$\begin{array}{rcccccl} g_1(x_1, x_2, x_3) & = & f_1(x_1, x_2, x_3) & = & x_1\overline{x_2} + x_3 & & \\ g_2(x_1, x_2, x_3) & = & f_2(g_1(x), x_2, x_3) & = & x_1\overline{x_2} + x_3 + \overline{x_3} & = & 1 \\ g_3(x_1, x_2, x_3) & = & f_3(g_1(x), g_2(x), x_3) & = & 1.x_3 & = & x_3 \end{array}$$

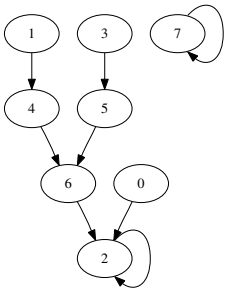
Pour comparer f et G, on donne leur matrice d'incidence :

$$B(f) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad B(G) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

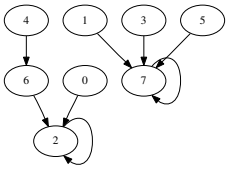
et leur table de transition :

	x			$f(x)$			$G(x)$		
0	0	0	0	0	1	0	0	1	0
1	0	0	1	1	0	0	1	1	1
2	0	1	0	0	1	0	0	1	0
3	0	1	1	1	0	1	1	1	1
4	1	0	0	1	1	0	1	1	0
5	1	0	1	1	1	0	1	1	1
6	1	1	0	0	1	0	0	1	0
7	1	1	1	1	1	1	1	1	1

Les graphes d'itérations correspondants sont donnés à la figure 2.2.



Graphe d'itérations de f



Graphe d'itérations de G

FIGURE 2.2 – Graphes d'itérations parallèles et séries

2. Itérations séries-parallèles

Entre les itérations parallèles et les itérations séries, il existe une famille d'itérations intermédiaires, dites *séries-parallèles* [Rob86].

Intuitivement, se donner des itérations séries-parallèles, c'est :

- se donner une *partition ordonnée* $(J_1; \ldots; J_s)$ de $[\![1, \mathsf{N}]\!]$,
- faire évoluer en parallèle les cellules de J_1,
- puis faire évoluer en parallèle les cellules de J_2, en tenant compte de l'évolution déjà faite des cellules de J_1,
- *etc.*

Une fois toutes les cellules de $(J_1; \ldots; J_s)$ mises à jour à partir d'une configuration x^n, on obtient une nouvelle configuration x^{n+1}. Il s'agit en fait d'itérations séries par bloc.

Plus formellement :

DÉFINITION I.10 (ITÉRATIONS SÉRIES-PARALLÈLES) : Étant donnés $\mathsf{N} \in \mathbb{N}$, un ensemble d'états $\mathcal{X}$, une fonction $f : \mathcal{X} \to \mathcal{X}$ et une partition ordonnée $\tau = (J_1; \ldots; J_s)$ de l'ensemble $[\![1, \mathsf{N}]\!]$, les itérations *séries-parallèles* sont les itérations parallèles de la fonction $f_\tau : \mathcal{X} \to \mathcal{X}$ définie par :

$$f_\tau : f_{J_s} \circ \ldots \circ f_{J_2} \circ f_{J_1}$$

et telle que :

$$f_{J_i}(x) = \begin{pmatrix} y_1 \\ \vdots \\ y_{\mathsf{N}} \end{pmatrix}, \text{ où } y_i = \begin{cases} f_i(x) & \text{si } i \in J_i, \\ x_i & \text{sinon.} \end{cases}$$

♢

Les itérations séries-parallèles étant, une fois encore, des IP sur un ensemble fini, on peut à nouveau les représenter en utilisant les graphes d'itérations et de connexion.

Exemple I.6 : On reprend l'exemple I.1 avec $f : \mathbb{B}^3 \to \mathbb{B}^3$ définie par :

$$\begin{aligned} f_1(x_1, x_2, x_3) &= x_1\overline{x_2} + x_3 \\ f_2(x_1, x_2, x_3) &= x_1 + \overline{x_3} \\ f_3(x_1, x_2, x_3) &= x_2 x_3 \end{aligned}$$

avec le mode séries-parallèles défini par $\tau = ((2);(1,3))$. Ici $J_1 = (2)$ et $J_2 = (1;3)$.

On a successivement :

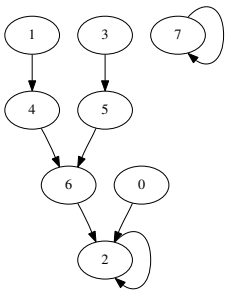
$$f_{J_1} = \begin{pmatrix} x_1 \\ x_1 + \overline{x_3} \\ x_3 \end{pmatrix} \quad f_{J_2} = \begin{pmatrix} x_1\overline{x_2} + x_3 \\ x_2 \\ x_2x_3 \end{pmatrix}$$

$$f_\tau = f_{J_2} \circ f_{J_1} = \begin{pmatrix} x_1.\overline{x_1 + \overline{x_3}} + x_3 \\ x_1 + \overline{x_3} \\ (x_1 + \overline{x_3})x_3 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_1 + \overline{x_3} \\ x_1x_3 \end{pmatrix}$$

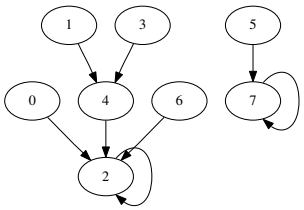
Les tables de f et f_τ sont les suivantes :

	x			$f(x)$			$f_\tau(x)$		
0	0	0	0	0	1	0	0	1	0
1	0	0	1	1	0	0	1	0	0
2	0	1	0	0	1	0	0	1	0
3	0	1	1	1	0	1	1	0	0
4	1	0	0	1	1	0	0	1	0
5	1	0	1	1	1	0	1	1	1
6	1	1	0	0	1	0	0	1	0
7	1	1	1	1	1	1	1	1	1

Les graphes d'itérations correspondants sont donnés à la figure 2.3.



Graphe d'itérations de f



Graphe d'itérations de f_τ

FIGURE 2.3 – Graphes d'itérations parallèles et séries-parallèles

CHAPITRE 3

Comportement asymptotique des itérations parallèles sur ensemble fini

Le bonheur est désir de répétition.

L'insoutenable légèreté de l'être
MILAN KUNDERA

Nous faisons le point, dans ce chapitre, sur le comportement asymptotique des itérations parallèles sur un ensemble fini. Nous regarderons quels peuvent être les différents cas limites, et sous quelles hypothèses on atteint tel ou tel cas. Nous en déduirons alors sous quelles conditions il est possible de faire diverger les itérations parallèles, et si de telles itérations sont intéressantes pour les applications que l'on vise.

Notre petite contribution consiste ici à avoir fait le point sur les situations de convergence (inconditionnelle), et en avoir déduit immédiatement des conditions suffisantes de divergence. Ces reformulations sont immédiates, mais l'approche n'en est pas moins originale, la distinction entre convergence et convergence inconditionnelle n'ayant jusqu'à présent pas été clairement établie dans ce contexte, et la non-convergence des itérations parallèles n'ayant jusqu'à présent pas fait l'objet de recherche en mathématiques appliquées. Ces résultats nous ont de plus guidés dans notre recherche d'exemples de systèmes itératifs au comportement désordonné intéressant : il s'agit, de fait, d'études préparatoires.

Le lecteur voulant approfondir le sujet pourra se reporter au livre de François Robert [Rob86].

I. ÉTUDE DES CAS LIMITES

1. Les cas limites possibles

Commençons par rappeler la définition suivante :

DÉFINITION I.11 (POINT FIXE D'UNE FONCTION) : Soit $f : \mathcal{X} \to \mathcal{X}$ une fonction définie sur un ensemble $\mathcal{X}$. Alors $x \in \mathcal{X}$ est un *point fixe* de f si $f(x) = x$. ◊

Considérons un système en itérations parallèles. Comme la fonction d'itération ne change pas d'une itérée à l'autre, et que l'on opère dans toute cette partie sur un ensemble fini, alors la suite des configu-

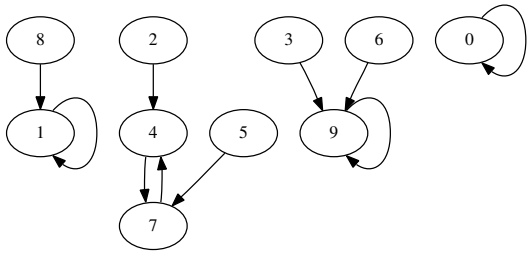


FIGURE 3.1 – Graphe d'itérations de l'exemple I.3.

rations $x^{n+1} = f(x^n)$ est périodique. En effet, les mêmes causes produisent les mêmes effets, ce qui nous permet de formuler la proposition suivante [Rob86] :

PROPOSITION I.2 : *Soit $\Sigma = (\mathcal{X}, f)$ un système en itérations parallèles, où $\mathcal{X}$ est fini et $x^0 \in \mathcal{X}$. Alors :*

- *Si le système $\Sigma(x^0)$ converge, alors ses configurations x^n sont stationnaires :*

$$\exists n_0 \in \mathbb{N}, \forall n \geqslant n_0, x^n = x^{n_0}.$$

Dans ce cas, la configuration terminale x^{n_0} est un point fixe de f.

- *Si le système $\Sigma(x^0)$ diverge, alors il entre dans un cycle : la suite de ses configurations est périodique au moins à partir d'un certain rang.*

Exemple I.7 : Dans l'exemple I.3, le système Σ peut évoluer soit vers une des trois configurations terminales 1, 9 ou 0, soit vers le cycle 4-7, selon la valeur de la configuration initiale choisie. On représente à nouveau ici le graphe d'itérations de cet exemple (figure 3.1), pour bien visualiser cela.

La proposition précédente nous incite à définir la notion d'attracteur :

DÉFINITION I.12 (ATTRACTEUR) : On appelle *attracteur* d'un système $\Sigma = (\mathcal{X}, f)$ en itérations parallèles, tout cycle ou point fixe de Σ. ◊

2. Bassins d'attraction de Σ

a. Configurations équivalentes

Il est possible d'étudier les itérées de f sur $\mathcal{X}$, en se ramenant à des configurations équivalentes plus simples à appréhender.

i. Rappels. La notion de relation d'équivalence sur un ensemble permet de mettre en relation des éléments de cet ensemble qui sont similaires pour une certaine propriété.

DÉFINITION I.13 (RELATION D'ÉQUIVALENCE) : Une *relation d'équivalence* $\mathfrak{R}$ dans un ensemble E est une relation binaire qui est :

Réflexive. Tout élément de E est associé à lui-même : $\forall\, x \in E, x\ \mathfrak{R}\ x$.

Symétrique. Tout élément de E est image de ses images : $\forall\, (x, y) \in E^2, (x\ \mathfrak{R}\ y) \Rightarrow (y\ \mathfrak{R}\ x)$.

Transitive. Toute image d'une image d'un élément de E est directement image de cet élément : $\forall\,(x,y,z)\in E^3, (x\,\mathfrak{R}\,y \wedge y\,\mathfrak{R}\,z) \Rightarrow (x\,\mathfrak{R}\,z)$. ◊

On pourra ainsi regrouper ces éléments par « paquets » qui se ressemblent, définissant ainsi la notion de classe d'équivalence :

DÉFINITION I.14 (CLASSE D'ÉQUIVALENCE) : La *classe d'équivalence* d'un élément x de E, notée $\mathfrak{R}\,(x)$, est l'ensemble des images de x par $\mathfrak{R}$: $\mathfrak{R}\,(x) = \{y \in E \,\big/\, x\,\mathfrak{R}\,y\}$. ◊

On peut alors construire de nouveaux ensembles en « assimilant » les éléments similaires à un seul et unique élément. On aboutit ainsi à la notion d'ensemble quotient :

DÉFINITION I.15 (ENSEMBLE QUOTIENT) : L'*ensemble quotient* de E par la relation d'équivalence $\mathfrak{R}$, noté $E/\mathfrak{R}$, est l'ensemble des classes d'équivalence de E suivant $\mathfrak{R}$:

$$E/\mathfrak{R} = \{\mathfrak{R}\,(x)\,|\,x \in E\}.$$

ii. Une relation d'équivalence. Pour simplifier les itérations de f, on peut penser à en donner une version quotientée selon une relation d'équivalence basée sur la notion de *descendance*.

DÉFINITION I.16 (DESCENDANCE) : On dit qu'une configuration $x' \in \mathcal{X}$ est un *descendant* d'une configuration $x \in \mathcal{X}$ s'il existe un entier $p \in \mathbb{N}$ tel que $x' = f^{(p)}(x)$. ◊

Notons qu'en particulier chaque configuration est son propre descendant ($p = 0$ dans ce cas). Par la suite,

DÉFINITION I.17 (CONFIGURATIONS ÉQUIVALENTES) : Deux configurations x, x' sont dites *équivalentes* si elles admettent un descendant commun. ◊

NOTATION I.4. On notera $x\,\mathfrak{R}\,x'$ quand x et x' seront deux configurations équivalentes.

Il est alors clair que :

PROPOSITION I.3 : *$\mathfrak{R}$ est une relation d'équivalence*

b. Les bassins d'attraction

Quand on réalise des itérations de f sur $\mathcal{X}$ fini, la partition en classes provenant de la relation d'équivalence de la définition I.17, ne peut donner qu'un nombre fini de classes, constituées d'un nombre fini d'éléments. Dans ce cas, dans le graphe d'itérations de f, les classes induites sont les composantes connexes du graphe, notion que l'on rappelle ci-dessous.

i. Composantes connexes d'un graphe. On peut définir une relation $\mathfrak{R}_0$ sur l'ensemble des sommets d'un graphe orienté, de la manière suivante :

DÉFINITION I.18 : Un sommet x est en relation $\mathfrak{R}_0$ avec un sommet x' si, et seulement s'il existe un chemin de x à x', et un chemin de x' à x. ◊

On a alors que :

PROPOSITION I.4 : *$\mathfrak{R}_0$ est une relation d'équivalence.*

PREUVE : *Sa définition même entraîne la symétrie et la réflexivité. Quant à la transitivité, elle est immédiate.*

On peut alors rappeler la définition d'une composante connexe d'un graphe orienté :

DÉFINITION I.19 (COMPOSANTE CONNEXE) : Les classes d'équivalence de $\mathfrak{R}_0$ sont appelées *composante connexe* du graphe. ◊

Un graphe connexe est un graphe ne possédant qu'une composante connexe pour $\mathfrak{R}_0$:

DÉFINITION I.20 (GRAPHE CONNEXE) : Un graphe est *connexe*, si toute paire ordonnée (a, b) de sommets distincts du graphe est reliée par au moins un chemin. ◊

En d'autres termes, tout sommet est atteignable depuis tous les autres sommets par au moins un chemin.

ii. Définition des bassins d'un graphe d'itérations. Cette notion de composantes connexes d'un graphe nous conduit à la définition suivante :

DÉFINITION I.21 (BASSIN) : Soit $\mathcal{X}$ un ensemble fini, et $f : \mathcal{X} \to \mathcal{X}$. Les composantes connexes du graphe d'itérations de f sont appelés *bassins* du système. ◊

Il n'existe que deux types de bassins dans le cas d'itérations parallèles sur un ensemble fini [Rob86] :

PROPOSITION I.5 : *Soit $\Sigma = (\mathcal{X}, f)$ un système en itérations parallèles, et C un de ses bassins d'attraction. Alors :*

- *Si C contient un point fixe ζ de f, alors il n'en contient qu'un seul, et pour tout x^0 dans C, le système $\Sigma(x^0)$ converge vers ζ.*
- *Sinon, C contient un unique cycle et, pour tout x^0 dans C, le système $\Sigma(x^0)$ finit par boucler indéfiniment sur ce cycle.*

En d'autres termes, tout bassin possède un et un seul attracteur.

Exemple I.8 : L'exemple I.3 rappelé à la figure 3.1 contient quatre bassins, chacun comportant un unique attracteur.

II. ÉTUDE DES SITUATIONS DE CONVERGENCE INCONDITIONNELLE

1. Première approche

Des résultats rappelés à la section précédente, on peut conclure un premier cas de convergence inconditionnelle des itérations parallèles sur un ensemble fini, notamment à partir de la proposition I.5 :

PROPOSITION I.6 : *On considère le système $\Sigma = (\mathcal{X}, f)$ des itérations parallèles de f sur $\mathcal{X}$ fini.*
Si le graphe d'itérations de f ne contient qu'un bassin, qui ne contient qu'un point fixe ζ, alors le système Σ converge inconditionnellement vers ζ.

Exemple I.9 : Les itérations parallèles de $f(x_1, x_2, x_3) = (x_2\overline{x_3}, 1, x_2)$ sur $\mathbb{R}^3$ sont dans cette situation.

En conséquence de quoi, on obtient une première condition nécessaire, originale bien qu'élémentaire, de *non convergence* inconditionnelle.

COROLLAIRE I.1 : *On considère le système $\Sigma = (\mathcal{X}, f)$ des itérations parallèles de f sur $\mathcal{X}$ fini.*
Pour que le système soit non convergent, il faut que son graphe d'itérations ne soit pas réduit à un bassin contenant un point fixe.

2. Puissances successives de f

Il peut s'avérer intéressant d'examiner non seulement les itérations parallèles de f sur $\mathcal{X}$ fini, mais aussi celles des composées f^k de f. En effet, on peut facilement établir les résultats suivants, rappelés dans [Rob86] :

PROPOSITION I.7 : *Tout point fixe de f est un point fixe de $f^{(k)}$, $(k = 1, 2, 3 \ldots)$.*

PROPOSITION I.8 : *Tout élément d'un cycle de longueur p est un point fixe de $f^{(p)}$, et réciproquement.*

REMARQUE. Si f admet un point fixe sur un ensemble fini $\mathcal{X}$, alors dans le passage de f à $f^{(k)}$, $k = 1, 2, 3 \ldots$, le bassin d'attraction de ce point fixe a tendance à se « contracter sur lui-même ».

On en déduit la proposition suivante :

COROLLAIRE I.2 : *Soit $\mathcal{X}$ un ensemble fini, $k \in \mathbb{N}^*$, et $f : \mathcal{X} \to \mathcal{X}$.*
Alors les itérations parallèles $(\mathcal{X}, f)$ convergent inconditionnellement si, et seulement si les itérations parallèles $(\mathcal{X}, f^{(k)})$ convergent inconditionnellement.

3. Cas où f est contractante

On se place dorénavant dans le cas $\mathcal{X} = \mathbb{B}^{\mathsf{N}}$, pour $\mathsf{N} \in \mathbb{N}^*$ fixé.

a. Rappels

i. Distance vectorielle. Nous allons présenter un résultat de convergence inconditionnelle dans le cas des applications contractantes. Il nous faudra donc au préalable définir ce qu'est une application contractante, et nous aurons besoin pour ce faire d'une distance.

DÉFINITION I.22 (DISTANCE VECTORIELLE) : On définit la notion de *distance vectorielle* entre les points $x = (x_1, \ldots, x_{\mathsf{N}})$ et $y = (y_1, \ldots, y_{\mathsf{N}})$ comme étant le vecteur [2] booléen :

$$d(x, y) = \begin{pmatrix} \delta(x_1, y_1) \\ \vdots \\ \delta(x_{\mathsf{N}}, y_{\mathsf{N}}) \end{pmatrix}$$

où δ représente la distance triviale (si $x = y$, alors $\delta(x, y) = 0$, sinon $\delta(x, y) = 1$). ◊

2. Cette distance vectorielle n'est donc pas une distance au sens topologique du terme, telle qu'elle sera rappelée dans la définition II.7.

PROPOSITION I.9 : *La distance vectorielle d vérifie les axiomes suivants :*

$$\begin{cases} d(x,y) &= 0 \Leftrightarrow x = y \\ d(x,y) &= d(y,x) \quad \forall x, y \in \mathcal{X} \\ d(x,z) &\leqslant d(x,y) + d(y,z) \quad \forall x, y, z \in \mathcal{X} \end{cases}$$

REMARQUE. Pour bien comprendre cette définition, remarquons que :

- le + de la dernière ligne est une somme vectorielle booléenne,
- l'inégalité $\leqslant$ est l'inégalité composante à composante.

Exemple I.10 : Soit x, y et z définis dans $\mathbb{B}^4$ par :

$$x = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, y = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \text{ et } z = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

On a

$$d(x,z) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \leqslant d(x,y) + d(y,z) = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

ii. Rappels d'analyse spectrale booléenne. Le résultat de convergence inconditionnelle que l'on va rappeler fait appel à de l'analyse spectrale, que l'on rappelle dans ce contexte.

DÉFINITION I.23 (VALEUR PROPRE, VECTEUR PROPRE) : Soit $B \in \mathcal{M}_n(\mathbb{B})$ une matrice booléenne, et $v \in \mathbb{B}^n$. On dit que v est un *vecteur propre* de B associé à la *valeur propre* $\lambda \in \mathbb{B}$, si $Bv = \lambda v$. ◇

La plus grande des valeurs propres joue un rôle particulier, ce qui justifie la notation suivante :

NOTATION I.5. On note $\rho(\mathbb{B})$ la plus grande des valeurs propres de B (c'est un booléen). $\rho(B)$ sera appelé *rayon spectral*.

Exemple I.11 : Soit $B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$. On peut vérifier que $v = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ est un vecteur propre de B associé à la valeur propre 1. Donc $\rho(B) = 1$.

iii. Applications contractantes. Il nous reste à redéfinir la notion d'application contractante, dans le cas de la distance vectorielle de la définition I.22.

DÉFINITION I.24 (APPLICATION CONTRACTANTE) : f est dite *contractante* relativement à la distance vectorielle d s'il existe une matrice booléenne $M \in \mathcal{M}_n(\mathbb{B})$ de rayon spectral booléen *nul*, telle que :

$$\forall (x,y) \in \mathbb{B}^n, d\left(f(x), f(y)\right) \leqslant M\, d(x,y)$$

Pour savoir si une application booléenne est contractante relativement à la distance vectorielle d, on peut utiliser la caractérisation suivante [Rob86] :

PROPOSITION I.10 : *Pour que f soit contractante relativement à d, il faut et il suffit que l'une des conditions suivantes soit réalisée :*

1. *le rayon spectral de $B(f)$ est nul,*
2. *il existe un entier $p \leqslant n$ tel que $[B(f)]^p = 0$,*
3. *il existe une matrice de permutation P de taille (n, n) telle que $P^t B(f) P$ soit triangulaire inférieure stricte,*
4. *le graphe de connexion de f est sans circuit.*

b. Convergence des itérations parallèles d'une application contractante

i. Un théorème du point fixe. Le théorème du point fixe suivant assure la convergence inconditionnelle des itérations parallèles dans le cas contractant [Rob86].

THÉORÈME I.1 (THÉORÈME DU POINT FIXE) : *Si f est contractante relativement à d, alors :*

- *il existe un entier $p \leqslant n$ tel que $f^{(p)}$ est constant,*
- *il existe $\zeta \in \mathcal{X}$ tel que $\forall x \in \mathcal{X}$, $f^{(p)}(x) = \zeta$.*

De plus,

- *ζ est l'unique point fixe de f dans $\mathcal{X}$,*
- *et pour tout x^0 dans $\mathcal{X}$, l'itération $x^{n+1} = f(x^n)$ stationne en ζ au bout d'au plus p itérations.*

On en déduit le corollaire suivant.

COROLLAIRE I.3 : *Supposons que $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ soit contractante relativement à la distance vectorielle d. Alors le système $(\mathbb{B}^{\mathsf{N}}, f)$ en itérations parallèles est inconditionnellement convergent, et atteint le point fixe de f en moins de N itérées.*

En d'autres termes, toute itération d'une fonction f contractante sur $\mathbb{B}^{\mathsf{N}}$ aboutit, en un temps fini inférieur à N, à une seule et unique configuration finale, qui ne dépend que de f (*i.e.*, pas de la configuration initiale).

ii. Condition nécessaire de non convergence des itérations parallèles. On déduit directement du théorème du point fixe précédent une condition nécessaire de *non convergence*, dans le cas des itérations parallèles.

PROPOSITION I.11 : *Soit $\Sigma = \left(\mathbb{B}^{\mathsf{N}}, f\right)$ un système en itérations parallèles. Si Σ est non convergent, alors f n'est pas une contraction.*

Cette condition nécessaire n'est pas suffisante, comme l'illustre l'exemple suivant.

Exemple I.12 : Soit $f : (x_1, x_2, x_3) \in \mathbb{B}^3 \mapsto (\overline{x_1}x_2 + x_1x_2\overline{x_3},\ 0,\ x_1 + x_2) \in \mathbb{B}^3$. Alors f n'est pas contractante, car sa matrice d'incidence :

$$B(f) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

possède un rayon spectral égal à 1 : $(1, 0, 1)^T$ est vecteur propre associé à la valeur propre 1 (la proposition I.1 permet de conclure). Cependant le système $(\mathbb{B}^3, f)$ converge inconditionnellement en itérations parallèles. En effet, on peut vérifier que, quelle que soit la configuration de départ, les itérées de f atteignent en temps fini la configuration finale $(0, 0, 0)$ que le système ne quitte plus.

4. Conclusion

Ces premiers résultats de non convergence ne sont valables que dans un cadre particulier, celui des systèmes en itérations parallèles sur un ensemble fini.

Cette petite étude prouve cependant que les itérations ne peuvent pas convenir pour les applications que l'on vise : nous avons vu dans la proposition I.2, que la divergence signifiait entrer dans un cycle, ce qui est une divergence beaucoup trop prévisible, inadéquate.

Nous avons donc recherché d'autres itérations, toujours élémentaires (rapides à calculer), mais qui ne possèdent pas ce défaut. Une généralisation de ces itérations parallèles, que l'on appelle *itérations chaotiques*, pourrait peut-être convenir.

Nous les étudierons dans les deux chapitres suivants, avec certains des outils présentés dans ce qui précède. Nous regarderons notamment ce qui peut être espéré en guise de divergence, et si la notion de contraction y joue encore un rôle important.

CHAPITRE 4

Les itérations chaotiques

Après avoir choisi on se contente du hasard de son existence et on peut l'aimer.

Vol de nuit
ANTOINE DE SAINT-EXUPERY

Les itérations précédentes (parallèles, séries, et séries-parallèles) finissaient toujours par répéter une certaine séquence dépendant de la fonction d'itérations et de la configuration initiale du système. La question à laquelle on s'intéresse ici est : que se passe-t-il si on fait évoluer les cellules non pas en série ou en parallèles, mais dans un ordre absolument arbitraire ?

Pour les modes précédents, le graphe d'itérations comportait toujours un point fixe ou un cycle. On sent bien qu'il n'en sera pas forcément de même pour ce nouveau mode opératoire, dit *chaotique* : nous verrons que, bien que le modèle soit fini (nombre fini de cellules, « temps » discret n), la suite des configurations du réseau engendrée par un mode opératoire chaotique n'est pas forcément convergente.

Ce chapitre rappelle la définition des itérations chaotiques introduites par D. Chazan et W. Miranker [CM69] et étudiées pour leur convergence sur $\mathbb{R}^N$ notamment par J.-C. Miellou [Mie75a, Mie75b] et F. Robert [Rob86]. Pour le cadre discret, on pourra se référer à [BT89]. Nous présentons nos premières réflexions sur ces objets. On considérera dans tout ce qui suit que $\mathcal{X} = \mathbb{B}^N$, où $N \in \mathbb{N}^*$.

I. LES STRATÉGIES CHAOTIQUES

Commençons par introduire les « stratégies chaotiques ». Elles vont désigner quelles cellules mettre à jour lors d'itérations chaotiques, et dans quel ordre le faire.

1. Approche théorique

a. Définition et notation

Soit $N \in \mathbb{N}^*$ fixé une fois pour toutes.

DÉFINITION I.25 (STRATÉGIE CHAOTIQUE) : On appelle *stratégie chaotique* toute suite de $[\![1; N]\!]$.$\diamond$

L'ensemble des stratégies chaotiques est donc l'ensemble des suites d'entiers naturels non nuls bornés par une valeur fixée N.

NOTATION I.6. On note $\mathcal{S}_{\mathsf{N}}$ l'ensemble des stratégies chaotiques des systèmes à N cellules.

Quand il n'y aura pas d'ambiguïté sur le nombre de cellules, ou qu'il sera clairement sous-entendu égal à une valeur fixée connue, on notera cet ensemble $\mathcal{S}$ tout simplement.

Par abus de langage et pour éviter trop de répétitions, on pourra utiliser le terme *suite* comme un synonyme de stratégie, quand le contexte le permettra.

REMARQUE. On pourrait aussi envisager, pour stratégies, des suites indexées par $\mathbb{Z}$, afin de pouvoir « remonter dans le temps » et faire en sorte que les itérations chaotiques deviennent une opération bijective. Cette étude, qui peut peut-être avoir son intérêt dans des applications où servent les fonctions bijectives (cryptographie par exemple), reste à faire.

Introduisons pour finir la généralisation suivante :

DÉFINITION I.26 (STRATÉGIE CHAOTIQUE GÉNÉRALISÉE) : On appelle *stratégie chaotique généralisée* toute suite de parties de $[\![1;\mathsf{N}]\!]$. ◇

NOTATION I.7. L'ensemble des stratégies chaotiques généralisées sera noté $\mathcal{S}^G$.

b. Nombre de stratégies chaotiques

Il est possible d'évaluer la taille de $\mathcal{S}$. Pour ce faire, on rappelle quelques notions de dénombrabilité.

i. Rappels.

DÉFINITION I.27 (DÉNOMBRABILITÉ) : Un ensemble E est dit *dénombrable* s'il existe une bijection de E sur une partie de l'ensemble des entiers naturels $\mathbb{N}$. Dans le cas contraire, cet ensemble est dit *indénombrable*. ◇

Pour les ensembles infinis, c'est-à-dire qui peuvent être mis en bijection avec une partie stricte d'eux-mêmes, on préfère parler de puissance plutôt que de cardinalité. Rappelons ci-dessous ce qu'est la *puissance du continu*.

DÉFINITION I.28 (PUISSANCE DU CONTINU) : Un ensemble a la *puissance du continu* c s'il peut être mis en bijection avec $\mathbb{R}$. ◇

Exemple I.13 : L'ensemble $\mathcal{P}(\mathbb{N})$ des parties de $\mathbb{N}$, comme l'ensemble $\mathbb{N}^{\mathbb{N}}$ des suites d'entiers, ont tous deux la puissance du continu.

ii. Puissance de $\mathcal{S}$. Prouvons maintenant le résultat suivant :

PROPOSITION I.12 : *L'ensemble $\mathcal{S}$ des stratégies chaotiques est infini indénombrable, il a la puissance du continu $c = card(\mathbb{R})$.*

PREUVE : *Nous reprenons la preuve de [Rob86], en la corrigeant un peu (l'application qu'il propose n'est pas bijective).*

Cette proposition provient du fait qu'à toute stratégie, on peut associer le réel de [0,1] dont la $n-$ième décimale en base N est le $n-$ième terme de la stratégie. Ce faisant, on définit une surjection (non une bijection, vu qu'une infinité de réels de [0,1] possèdent deux écritures en base N), et donc la puissance de $\mathcal{S}$ est supérieure ou égale à celle de [0,1], qui est celle de $\mathbb{R}$.

D'autres part, $\mathcal{S}$ est incluse dans l'ensemble $\mathbb{N}^{\mathbb{N}}$ des suites d'entiers. Comme ce dernier à la puissance du continu, on en déduit que la puissance de $\mathcal{S}$ est inférieure à c, et donc le résultat.

2. Approche pratique

En pratique, c'est-à-dire sur machine (ordinateur), on n'itère qu'un nombre fini de fois. Cela revient à considérer les stratégies ayant un nombre fini de termes. Il n'est pas certain qu'il faille distinguer l'approche théorique de l'approche pratique. Cela dit, cette distinction est aisée, et permet d'éviter certaines discussions sur le passage non trivial de la théorie aux nombres machines.

a. Définition et notation

DÉFINITION I.29 (STRATÉGIE FINIE) : On appelle stratégie finie toute suite finie de $[\![1;\mathsf{N}]\!]$. ◊

NOTATION I.8. On note $\tilde{\mathcal{S}}_{\mathsf{N}}$ l'ensemble des stratégies chaotiques finies des systèmes à N cellules. On préférera la notation $\tilde{\mathcal{S}}$ lorsqu'il n'y a pas ambiguïté sur le nombre de cellules.

b. Nombre de stratégies chaotiques finies

Commençons par un petit rappel.

DÉFINITION I.30 (ENSEMBLE DES DÉCIMAUX) : On appelle *nombre décimal* un nombre réel n'ayant qu'un nombre fini de décimales. ◊

NOTATION I.9. L'ensemble des décimaux est noté $\mathbb{D}$. Il est infini dénombrable.

Comme $\tilde{\mathcal{S}}$ est l'ensemble des suites entières dont les termes sont bornés par N, *et dont le nombre de termes est fini non borné*, c'est un ensemble infini dénombrable :

PROPOSITION I.13 : *L'ensemble $\tilde{\mathcal{S}}$ des stratégies chaotiques finies est infini dénombrable.*

PREUVE : *Exactement la même démonstration que pour la proposition I.12, mais en considérant l'ensemble $\mathbb{D}$ au lieu de $\mathbb{R}$.*

II. LES ITÉRATIONS CHAOTIQUES

Nous sommes maintenant en mesure de définir les itérations chaotiques.

1. Définition

a. La définition des itérations chaotiques

DÉFINITION I.31 (ITÉRATIONS CHAOTIQUES) : Soient $\mathsf{N} \in \mathbb{N}$, $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$ et $S \in \mathcal{S}$ une stratégie. Les *itérations chaotiques* $\left(f, (x^0, S)\right)$ sont définies par la suite récurrente :

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

Ainsi, au $n-$ième « battement », on n'actualise (avec f) que la cellule numérotée S^n.

NOTATION I.10. Par la suite, l'acronyme IC sera parfois utilisé pour désigner les itérations chaotiques.

REMARQUE. Précisons une fois encore que le terme « chaotique » a été utilisé dans les expressions « stratégies chaotiques » et « itérations chaotiques » sans rapport avec le sens de « chaos » de la théorie du même nom. Il s'agit ici d'un adjectif sensé qualifier l'apparent comportement erratique qu'un tel système peut avoir. Mais, jusqu'alors, rien n'a été fait pour approfondir le comportement apparemment désordonné que de telles itérations semblent induire sur un système en évolution (il s'agit là d'une de nos contributions).

REMARQUE. On peut remplacer, dans la définition I.31, $f(x^{n-1})_i$ par $f(x^k)_i$ où $k < n$, pour modéliser un retard à la transmission d'informations. On obtient alors des *itérations chaotiques asynchrones*.

REMARQUE. On peut aussi supposer que la suite S est une stratégie chaotique généralisée, *i.e.* une suite de parties de $[\![1, \mathsf{N}]\!]$. Nous n'irons pas plus avant dans cette généralisation, étant entendu qu'une itération chaotique généralisée y^n n'est qu'une sous-suite d'une itération chaotique x^n.

b. Les itérations chaotiques sont des systèmes itératifs

Ces itérations chaotiques peuvent se définir d'une autre manière, prouvant qu'il s'agit bien d'itérations au sens de la définition 3 :

PROPOSITION I.14 : *Soient $\mathsf{N} \in \mathbb{N}^*$, $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_{\mathsf{N}}$, $f : \mathcal{X} \to \mathcal{X}$, et $S \in \mathcal{S}_{\mathsf{N}}$ une stratégie. On pose $f = (f_1, \ldots, f_{\mathsf{N}})$, et l'on réutilise la notation du chapitre 2 :*

$$\begin{array}{rccc} F_i : & \mathcal{X} & \longrightarrow & \mathcal{X} \\ & (x_1, \ldots, x_{\mathsf{N}}) & \longmapsto & (x_1, \ldots, x_{i-1}, f_i(x_1, \ldots, x_{\mathsf{N}}), x_{i+1}, \ldots, x_{\mathsf{N}}). \end{array}$$

Alors les itérations chaotiques $\left(f, (x^0, S)\right)$ sont le système itératif $\left(\mathbb{B}^{\mathsf{N}}, (F_{S_n})_{n\in\mathbb{N}}, x^0\right)$. Et de même, les itérations chaotiques asynchrones correspondent bien à un système itératif asynchrone.

On pourra de plus signaler que :

REMARQUE. Pour un N quelconque, les itérations chaotiques définies par la stratégie $\underbrace{1;2;\ldots;\mathsf{N}}\underbrace{;1;2;\ldots;\mathsf{N}}\ldots$ redonnent les itérations séries de f. De même, les itérations parallèles peuvent être vues comme un cas particulier des itérations chaotiques, dans leur versions étendues aux stratégies chaotiques généralisées de la définition I.26.

Ainsi, il existe des cas où les itérations chaotiques se comportent « très raisonnablement » (convergence ou cycle). Nous verrons cependant plus loin que l'on ne peut pas faire de ces cas une généralité.

2. Exemple

Donnons un premier exemple pour lequel la situation idéale du théorème I.1 du point fixe ne se produit plus.

Exemple I.14 : On reprend l'exemple I.4 avec $f : \mathbb{B}^3 \to \mathbb{B}^3$ définie par :

$$\begin{aligned} f_1(x_1, x_2, x_3) &= x_1\overline{x_2} + x_3 \\ f_2(x_1, x_2, x_3) &= x_1 + \overline{x_3} \\ f_3(x_1, x_2, x_3) &= x_2 x_3. \end{aligned}$$

On a déjà vu que f admet les deux points fixes suivants :

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \text{ et } \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Soit la stratégie chaotique suivante $S = 1;3;1;3;\ldots$ Alors, partant de :

$$x^0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

l'itération chaotique considérée stationne en x^0, qui n'est pas un point fixe de f.

III. Le graphe de tous les possibles par itérations chaotiques (GTPIC)

Le graphe d'itérations des IP ne peut pas être utilisé tel quel pour décrire l'évolution des IC, il doit être adapté.

1. Présentation

Le graphe de tous les possibles par itérations chaotiques est une représentation graphique des itérations chaotiques (stratégies chaotiques non généralisées) sous la forme d'un graphe orienté, qui est due à Gotsman [GLS88]. On le présente au travers d'un exemple.

Exemple I.15 : On considère $\mathsf{N} = 3$, et $f : \mathbb{B}^3 \to \mathbb{B}^3$ définie par

$$\begin{aligned} f_1(x) &= \overline{x_2} \\ f_2(x) &= 1 \\ f_3(x) &= \overline{x_1} + x_2 \end{aligned}$$

Ce qui conduit aux tables de transition suivantes :

	x	$F(x)$	$F_1(x)$	$F_2(x)$	$F_3(x)$
0	0 0 0	1 1 1	1 0 0	0 1 0	0 0 1
1	0 0 1	1 1 1	1 0 1	0 1 1	0 0 1
2	0 1 0	0 1 1	0 1 0	0 1 0	0 1 1
3	0 1 1	0 1 1	0 1 1	0 1 1	0 1 1
4	1 0 0	1 1 0	1 0 0	1 1 0	1 0 0
5	1 0 1	1 1 0	1 0 1	1 1 1	1 0 0
6	1 1 0	0 1 1	0 1 0	1 1 0	1 1 1
7	1 1 1	0 1 1	0 1 1	1 1 1	1 1 1

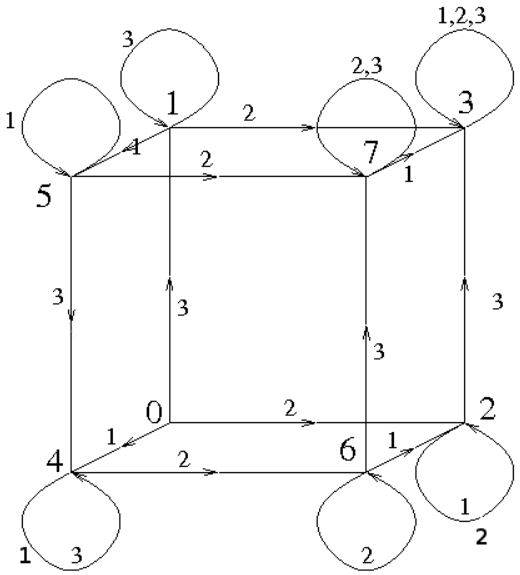


FIGURE 4.1 – Graphe de tous les possibles par itérations chaotiques du graphe I.15

que l'on peut représenter par le *graphe de tous les possibles par itérations chaotiques* de la figure 4.1.

Ainsi, si l'on est au sommet 5 – c'est-à-dire si la configuration de notre système est (1,0,1) – et si la stratégie chaotique vaut 1, alors on reste au sommet 5 (à la configuration (1,0,1)). Par contre, si la stratégie vaut 2, alors on passe du sommet 5 au sommet 7, et la nouvelle configuration du système sera donc (1,1,1), *etc.*

NOTATION I.11. Dans ce qui suit, on utilisera l'acronyme *GTPIC* pour *Graphe de Tous les Possibles par Itérations Chaotiques*. On notera $\mathcal{G}_f$ le GTPIC de $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$.

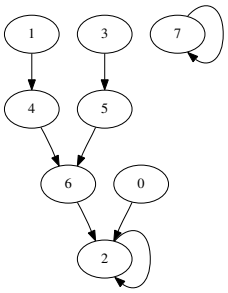
2. Exemple d'évolution d'une itération chaotique

Exemple I.16 : On reprend l'exemple I.4.

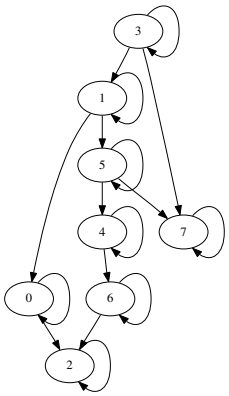
On a $F_1(x_1,x_2,x_3) = \begin{pmatrix} x_1\overline{x_2}+x_3 \\ x_2 \\ x_3 \end{pmatrix}$, $F_2(x_1,x_2,x_3) = \begin{pmatrix} x_1 \\ x_1+\overline{x_3} \\ x_3 \end{pmatrix}$ et $F_3(x_1,x_2,x_3) = \begin{pmatrix} x_1 \\ x_2 \\ x_2x_3 \end{pmatrix}$, ce qui donne les tables de transition suivantes :

	x	$F(x)$	$F_1(x)$	$F_2(x)$	$F_3(x)$
0	0 0 0	0 1 0	**0** 0 0	0 **1** 0	0 0 **0**
1	0 0 1	1 0 0	**1** 0 1	0 **0** 1	0 0 **0**
2	0 1 0	0 1 0	**0** 1 0	0 **1** 0	0 1 **0**
3	0 1 1	1 0 1	**1** 1 1	0 **0** 1	0 1 **1**
4	1 0 0	1 1 0	**1** 0 0	1 **1** 0	1 0 **0**
5	1 0 1	1 1 0	**1** 0 1	1 **1** 1	1 0 **0**
6	1 1 0	0 1 0	**0** 1 0	1 **1** 0	1 1 **0**
7	1 1 1	1 1 1	**1** 1 1	1 **1** 1	1 1 **1**

La figure 4.2 synthétise ces tables. On y retrouve les points fixes relatifs à 2 et 7 trouvés précédemment. De plus, on constate que chaque configuration est un point fixe d'une itération chaotique donnée.



Graphe d'itérations de f



GTPIC simplifié de f

FIGURE 4.2 – Graphes d'itérations parallèles et chaotiques de l'exemple I.14

Nous adaptons dans ce qui suit le GTPIC à notre étude, en proposant une caractérisation et une relation d'équivalence sur ces graphes.

3. Caractérisation

PROPOSITION I.15 (CARACTÉRISATION DES GTPIC) : *Soit $\mathcal{G}$ un graphe GTPIC. Alors il existe un N tel que :*

1. *$\mathcal{G}$ a 2^{N} sommets, que l'on numérote habituellement de 0 à $2^{\mathsf{N}} - 1$.*
2. *Tout sommet est à l'origine de N arcs (orientés) (numérotés de 1 à N), qui peuvent être :*
   - *soit une boucle, éventuellement multiple,*
   - *soit un arc simple (d'un sommet vers un autre sommet).*
3. *L'image d'un sommet s par un arc simple est un élément de $\widetilde{s}$: ensemble des sommets obtenus en niant au plus 1 bit dans l'écriture binaire de s.*

Réciproquement, tout graphe vérifiant les trois points précédents est le GTPIC d'une fonction f.

PREUVE : *Immédiat.*

Exemple I.17 : Dans l'exemple précédent, $\mathsf{N} = 3$. Il y a 2^3 sommets, et 3 arcs simples « sortant » de chaque sommet.

Cette caractérisation permet donc d'étudier les IC avec la théorie des graphes. Ce qui suit permet d'établir un lien avec l'algèbre linéaire.

4. Matrice d'adjacence d'un GTPIC

On peut représenter un graphe non orienté par une matrice d'adjacence.

DÉFINITION I.32 (MATRICE D'ADJACENCE) : Dans une *matrice d'adjacences*, les lignes et les colonnes représentent les sommets du graphe.

- Un 1 à la position (i,j) signifie que le sommet i est adjacent au sommet j, *i.e.* qu'il existe une arête entre i et j.
- Sinon, on place un 0.

En cas de boucle (pour le sommet i, par exemple), on place un 1 sur la diagonale (en position (i, i)). ◊

La matrice d'adjacence d'un GTPIC (plus exactement, de sa version non orienté) n'est pas quelconque, elle vérifie les propriétés suivantes :

- Sa taille est $2^{\mathsf{N}} \times 2^{\mathsf{N}}$.
- Elle est *très* creuse : entre 1 et N coefficients non nuls par ligne.
- Elle est constituée d'entiers naturels inférieurs ou égaux à N.
- Sa somme par ligne est toujours égale à N.
- Tout coefficient n'étant pas sur la diagonale est égal à 0 ou 1.
- Le coefficient de la diagonale est égal à N moins le nombre de coefficients non nuls (qui sont donc égaux à 1) de la ligne.

Tout cela peut se déduire de la caractérisation des GTPIC. On en conclut que pour mémoriser une ligne de la matrice, on peut se contenter de mémoriser les emplacements des 1 (la diagonale se déduit), soit au pire $\mathsf{N} - 1$ emplacements à sauvegarder pour chacune des 2^{N} lignes.

5. Une relation d'équivalence sur les fonctions booléennes

On découvre dans cette section une première utilité des GTPIC : deux fonctions peuvent donner les mêmes itérations chaotiques, mais deux GTPIC différents donnent des itérations différentes...

DÉFINITION I.33 : On définit une relation binaire $\mathfrak{R}$ sur les applications de $\mathbb{B}^{\mathsf{N}}$ par

$$\forall f, g : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}, \left[f \mathfrak{R} g \Longleftrightarrow \mathcal{G}_f = \mathcal{G}_g \right]$$ ◊

En d'autres termes, f et g sont en relation par $\mathfrak{R}$ si, et seulement si elles ont même GTPIC, ce qui se traduit par la proposition immédiate suivante :

PROPOSITION I.16 : *$\mathfrak{R}$ est une relation d'équivalence sur l'ensemble des applications de $\mathbb{B}^{\mathsf{N}}$ dans $\mathbb{B}^{\mathsf{N}}$.*

REMARQUE. On peut remarquer qu'il y a équivalence entre un graphe GTPIC $\mathcal{G}_f$ et une classe d'équivalence $\dot{f}$: «se donner une classe ou un graphe, c'est pareil ».

6. Utilité de cette relation d'équivalence

PROPOSITION I.17 : *Soient f_1 et f_2 deux fonctions appartenant à une même classe d'équivalence, c'est-à-dire ayant le même GTPIC.*

Alors f_1 et f_2 produisent exactement les mêmes itérations chaotiques : $\forall x^0 \in \mathbb{B}^{\mathsf{N}}$, $\forall S \in \mathcal{S}$, les itérations $\left(f_1, (x^0, S)\right)$ et $\left(f_2, (x^0, S)\right)$ sont les mêmes (elles présentent la même configuration au temps n, $\forall n \in \mathbb{N}$).

CHAPITRE 5

Comportement asymptotique des itérations chaotiques

Le désordre est le meilleur serviteur de l'ordre établi.

Le diable et le bon dieu
JEAN-PAUL SARTRE

Dans ce chapitre, nous rappelons les deux principaux résultats de convergence des itérations chaotiques, tels qu'on peut les trouver dans la littérature, le premier étant local, et le second global. Ces convergences dépendront de la contraction de la fonction d'itération, dans un sens à préciser, ainsi que d'une certaine notion de périodicité pour les stratégies chaotiques. Nous en déduirons alors des conditions nécessaires de non convergence des itérations chaotiques.

Ce chapitre contient donc des résultats classiques de convergence des itérations chaotiques, tels qu'on les trouvent dans le livre de François Robert [Rob86], et des réflexions sur les situations de non convergence de telles itérations, qui ont été notamment menées dans le cadre de l'encadrement, par Jean-François Couchot et moi-même, du stage de master recherche d'Olivier Grasset.

I. APPROCHE LOCALE

Pour réaliser une étude locale des situations de convergence des itérations chaotiques, il nous faut commencer par définir une notion de voisinage sur $\mathbb{B}^{\mathsf{N}}$.

1. Voisinage massif

DÉFINITION I.34 (VOISINAGE MASSIF) : Soit ζ un élément de $\mathbb{B}^{\mathsf{N}}$. $V \subset \mathbb{B}^{\mathsf{N}}$ est appelé *voisinage massif* de ζ si :

1. $\zeta \in V$,
2. $\forall u \in V, \forall v \in \mathbb{B}^{\mathsf{N}}$, si $d(\zeta, v) \leqslant d(\zeta, u)$, alors $v \in V$. ◊

NOTATION I.12. On associe à tout voisinage massif V de ζ un voisinage massif de 0, défini par :

$$V_0(\zeta) = \left\{ d(\zeta, u) \;\middle/\; u \in V \right\}.$$

2. Itérations contractantes et convergence des itérations chaotiques

a. Introduction

L'approche utilisée jusqu'à présent pour faire converger inconditionnellement les itérations chaotiques, a consisté à se ramener à une sorte de cas contractant, afin de se retrouver dans une situation similaire au chapitre 3. On rappelle que $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ est contractante suivant la définition I.24 si, et seulement si sa matrice d'incidence a un rayon spectral nul. On rappelle de plus qu'effectivement, un système contractant sur $\mathbb{B}^{\mathsf{N}}$ converge inconditionnellement en itérations parallèles.

La question qui se pose est alors comment définir la notion « d'application contractante » pour des itérations qui ne sont plus parallèles, *i.e.* dont la fonction d'itération est changeante (suite de fonction non constante pour le système itéré) ? Plus exactement, l'approche actuelle de recherche des situations de convergence pour les itérations chaotiques consiste à se demander : partant d'une application contractante f, quelles sont les manières de composer les F_i (notation introduite en I.1), pour que l'application résultante reste contractante ?

Un résultat d'algèbre prouve que quand on compose les F_i d'une fonction f contractante suivant certaines stratégies chaotiques dites « pseudo-périodiques », alors l'application résultante reste contractante. Voyons plus précisément de quoi il s'agit.

b. Les stratégies pseudo-périodiques

Les stratégies chaotiques pseudo-périodiques sont construites à partir de stratégies complètes :

Définition I.35 (Stratégie complète) : Une stratégie chaotique finie $S \in \tilde{\mathcal{S}}_{\mathsf{N}}$ est dite *complète* si toute valeur de $[\![1;\mathsf{N}]\!]$ apparaît au moins une fois dans S. ◊

On peut donc définir les stratégies pseudo-périodiques.

Définition I.36 (Stratégie pseudo-périodique) : La stratégie chaotique $S \in \mathcal{S}_{\mathsf{N}}$ est dite *pseudo-périodique* si elle est constituée par une succession indéfinie de stratégies (finies) complètes. ◊

c. Résultat de convergence locale

Commençons par introduire la notion de dérivée discrète :

Définition I.37 (Dérivée discrète) : Soit $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$. La *dérivée discrète* de f en un point $x \in \mathbb{B}^{\mathsf{N}}$, notée $f'(x)$, est la matrice booléenne de taille $\mathsf{N} \times \mathsf{N}$, dont l'élément (i, j) est égal à :

- 1 si $f_i(x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_{\mathsf{N}}) \neq f_i(x_1, \ldots, x_{j-1}, \overline{x_j}, x_{j+1}, \ldots, x_{\mathsf{N}})$,
- 0 sinon. ◊

On peut alors définir le résultat de convergence locale suivant :

Théorème I.2 : *Soient $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ une application possédant un point fixe ξ, et V un voisinage massif de ξ.*

Soit $\hat{V}$ le sous-ensemble de V obtenu en ôtant de V les éléments les plus éloignés de ξ au sens de d, et W le voisinage massif de 0 associé à V :

$$W = \{d(u, \xi), u \in V\}$$

On pose $M = \sup_{z\in\hat{V}}\{f'(z)\}$*, et* M_i *la matrice booléenne obtenue en remplaçant dans la matrice unité* (N,N) *la* i*-ième ligne par celle de* M*.*

Supposons que $M_i W \subset W$*, et que le rayon spectral de* W *est nul.*

Alors toute itération chaotique pseudo-périodique issue d'un sommet quelconque x^0 *de* V *reste dans* V*, et atteint* ξ *au bout d'au plus* n *pseudo-périodes.*
De plus, ξ *est l'unique point fixe de* f *dans* V*.*

Preuve : *Voir [Rob86].*

L'hypothèse de rayon spectral nul nous ramène à une application f contractante. La pseudo-périodicité est là pour faire en sorte que les F_i, composées dans l'ordre de la stratégie chaotique, donnent des itérations contractantes. Enfin, l'inclusion évoque la propriété de continuité (cette dernière étant l'inclusion d'un voisinage dans un autre) appliquée au cadre discret.

d. Obtention d'une non convergence

On déduit du théorème I.2 que pour éviter la convergence, on peut envisager d'attaquer le problème d'une des manières suivantes :

1. Prendre des applications non contractantes.
2. Faire en sorte qu'il y ait un voisinage massif répulsif.
3. Ne pas prendre de stratégie chaotique pseudo-périodique.

Le point 1 concerne la fonction d'itération, le point 2 la configuration initiale, et le dernier point touche à la stratégie chaotique.

II. Approche globale

1. Convergence globale inconditionnelle

Le théorème de convergence locale I.2 se généralise facilement : $\mathbb{B}^{\mathsf{N}}$ est un voisinage massif de tout point, et la borne supérieure de la dérivée sur $\mathbb{B}^{\mathsf{N}}$ est la matrice d'incidence. Ce qui donne :

Théorème I.3 : *Soit* $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ *une application contractante.*
Alors pour toute stratégie chaotique pseudo-périodique $(S^n)_{n\in\mathbb{N}}$*, les itérations chaotiques* $\left(\mathbb{B}^{\mathsf{N}}, (F_{S^n})_{n\in\mathbb{N}}\right)$ *convergent inconditionnellement, et atteignent l'unique point fixe de* f *en moins de* N *itérations.*

Ce résultat est une réécriture du théorème de convergence globale de [Rob86].

2. Condition nécessaire de non convergence

On peut déduire du théorème I.3 la condition nécessaire de non-convergence suivante.

COROLLAIRE I.4 : *Soit $f : \mathbb{B}^N \to \mathbb{B}^N$ et $S \in \mathcal{S}$. Si les itérations chaotiques $\left(\mathbb{B}^N, (F_{S^n})_{n\in\mathbb{N}}\right)$ sont non convergentes, alors*

- *soit f n'est pas contractante,*
- *soit S n'est pas pseudo-périodique.*

Lors du passage du local au global, on a perdu le moyen d'agir sur la convergence grâce aux voisinages. Il reste donc, comme moyen d'action, la pseudo-périodicité de la stratégie, et la contraction de la fonction.

À la fonction correspondra le programme informatique (algorithme), dans le cadre des applications que l'on vise. C'est ce programme, donc cette fonction, qui doit être imprévisible, la stratégie correspondant alors aux données qui lui seront appliquées en entrée. Nous allons donc agir sur la contraction de la fonction.

CONCLUSION DE L'APPROCHE MATHÉMATIQUES APPLIQUÉES

Les chapitres de la partie I ont porté sur la recherche de systèmes itératifs simples, pouvant présenter des comportements asymptotiques non élémentaires. La conclusion que l'on peut en tirer est qu'un bon candidat serait les itérations chaotiques, avec une fonction d'itération non contractante. Elles ont pour avantage d'être simples, et de présenter des situations asymptotiques autres que la convergence ou le cycle.

Nous voulons maintenant savoir si de telles itérations peuvent réellement se comporter de manière imprévisible. Il nous faut d'abord donner un sens à la notion d'imprévisibilité. Le candidat le plus sérieux est la notion de chaos, telle qu'elle est définie de plusieurs manières topologiques dans la théorie du même nom. Cette théorie mathématique du chaos sera rappelée dans la partie suivante.

Cependant, pour étudier le comportement chaotique des IC, il nous faut un exemple : la fonction d'itération doit être choisie, afin de pouvoir étudier le système itératif associé sous l'angle topologique. Nous rappelons qu'on ne peut pas prendre d'opérateur contractant, c'est-à-dire qu'il est nécessaire que le GTPIC de la fonction d'itération présente un circuit, si l'on ne veut pas converger. Introduisons pour cela la fonction *négation vectorielle*, qui se contente de nier chaque variable :

DÉFINITION I.38 (NÉGATION VECTORIELLE) : On définit la fonction *négation vectorielle* par :

$$\begin{array}{rccc} f_0 : & \mathbb{B}^{\mathsf{N}} & \longrightarrow & \mathbb{B}^{\mathsf{N}} \\ & (x_1, \ldots, x_{\mathsf{N}}) & \longmapsto & (\overline{x_1}, \ldots, \overline{x_{\mathsf{N}}}) \end{array}$$

Cette fonction négation a un graphe de connexion contenant N boucles. Elle n'est donc pas contractante. De plus, elle sera facile à programmer, et rapide, donc appropriée pour les applications que l'on vise. Enfin, vue l'allure du GTPIC, il sera possible d'atteindre toute configuration à partir de toute autre configuration. Voyons à présent ce que donne l'étude des IC sous l'angle théorie du chaos.

Deuxième partie

Introduction à la Théorie du Chaos

CHAPITRE 6

Les systèmes dynamiques discrets en topologie

> Ainsi s'écoule toute la vie : on cherche le repos en combattant quelques obstacles et si on les a surmontés, le repos devient insupportable par l'ennui qu'il engendre. Il en faut sortir et mendier le tumulte.
>
> *Les pensées*
> BLAISE PASCAL

Ce chapitre redonne un certain nombre de définitions topologiques, qui seront nécessaires à la définition des systèmes chaotiques. Nous en profiterons pour rappeler l'approche topologique des systèmes dynamiques discrets.

Le lecteur voulant approfondir la topologie pourra se référer au livre de Laurent Schwartz [Sch80].

I. ESPACES TOPOLOGIQUES, ESPACES MÉTRIQUES

Nous rappelons ici les types d'espaces qui nous intéressent, et les fonctions que l'on utilise sur ces espaces.

1. Espaces topologiques, ouverts et voisinages

a. Premières définitions

On introduit ici les termes de base de la topologie [Sch80].

DÉFINITION II.1 (ESPACE TOPOLOGIQUE) : On appelle *espace topologique* la donnée d'un couple (E, τ), où E est un ensemble et τ une famille de parties de E, qu'on appelle des *ouverts*, vérifiant :

- $\emptyset, E \in \tau$: l'ensemble vide et E sont des ouverts,
- une réunion quelconque d'ouverts est un ouvert,
- une intersection finie d'ouverts est un ouvert. ◊

DÉFINITION II.2 (FERMÉ) : On appelle *fermé* le complémentaire d'un ouvert. ◊

NOTATION II.1. Soit A une partie d'un espace topologique (E, τ). Le plus petit fermé contenant A existe toujours : c'est l'intersection de tous les fermés contenant A. On notera $\overline{A}$ cet ensemble, et l'on parlera de *fermeture topologique*, ou d'*adhérence* de A.

On peut munir un ensemble donné de plusieurs topologies différentes, ce qui conduit à la notion d'ordre suivante :

DÉFINITION II.3 (FINESSE DES TOPOLOGIES) : Une topologie $\tau \in \mathcal{P}(\mathcal{X})$ sur l'ensemble $\mathcal{X}$ est *plus fine* qu'une topologie $\tau' \in \mathcal{P}(\mathcal{X})$, si $\tau' \subset \tau$. On dira alors que τ' est *moins fine*, ou *plus grossière* que τ.◇

REMARQUE. Une topologie est donc plus fine qu'une autre si elle a plus d'ouverts. Cet ordre n'est pas total : il existe des topologies non comparables entre-elles.

Introduisons enfin la notion de voisinage :

DÉFINITION II.4 (VOISINAGE) : Soit (E, τ) un espace topologique. On appelle *voisinage de* $x \in E$ toute partie de E contenant un ouvert qui contient x. ◇

b. Exemples de topologies

Donnons dès à présent deux exemples de topologies sur un ensemble $\mathcal{X}$ donné [Sch80] :

DÉFINITION II.5 (TOPOLOGIE DISCRÈTE) : La *topologie discrète* sur un ensemble $\mathcal{X}$ est la topologie $\tau = \mathcal{P}(\mathcal{X})$ de toutes les parties de $\mathcal{X}$. ◇

La topologie discrète est la topologie de $\mathcal{X}$ possédant le plus d'ouverts. Elle est plus fine que toute autre topologie sur $\mathcal{X}$, et vérifie en particulier que tout sous-ensemble de $\mathcal{X}$ est à la fois ouvert et fermé. Cette topologie tire son nom du fait que tous les points sont isolés (chaque point est ouvert et fermé). A contrario :

DÉFINITION II.6 (TOPOLOGIE GROSSIÈRE) : La *topologie grossière* sur un ensemble $\mathcal{X}$ est la topologie $\tau = \{\varnothing, \mathcal{X}\}$. ◇

C'est la topologie la moins fine qui soit sur $\mathcal{X}$, elle ne contient que deux ouverts.

2. Distances, espaces métriques

Une manière commode de définir des topologies est d'utiliser des métriques.

DÉFINITION II.7 (DISTANCE) : Sur un ensemble E, une *distance*, aussi appelée *métrique*, est une application $d : E \times E \to \mathbb{R}^+$ possédant les propriétés suivantes :

Symétrie : $\forall x, y \in E, d(x, y) = d(y, x)$.

Séparation : $\forall x, y \in E, d(x, y) = 0 \Leftrightarrow x = y$.

Inégalité triangulaire : $\forall x, y, z \in E, d(x, z) \leqslant d(x, y) + d(y, z)$. ◇

DÉFINITION II.8 (ESPACE MÉTRIQUE) : On appelle *espace métrique* la donnée d'un couple (E, d), où E est un ensemble et d une distance sur E. ◇

Les espaces métriques, qui sont des espaces topologiques, possèdent des voisinages particuliers, appelés boules :

DÉFINITION II.9 (BOULE FERMÉE, BOULE OUVERTE) : Soit (E, d) un espace métrique. La *boule fermée* centrée en un point P, et de rayon réel r, est l'ensemble $\overline{\mathcal{B}}(P, r)$ des points dont la distance à P est inférieure ou égale à r : $\overline{\mathcal{B}}(P, r) = \left\{ M \in E \;\middle|\; d(M, P) \leqslant r \right\}$.

La boule ouverte est l'ensemble $\mathcal{B}(P, r) = \left\{ M \in E \;\middle|\; d(M, P) < r \right\}$. ◇

II. Compacité et complétude

1. Compacité

Certains espaces topologiques, dits compacts, joueront un rôle particulier dans la suite de ce document. On les définit dans ce qui suit [Sch80].

a. Définitions préliminaires

Il nous faut tout d'abord introduire les espaces séparés et la notion de recouvrement, afin de pouvoir définir la compacité.

Définition II.10 (Espaces séparés) : Un *espace séparé* est un espace topologique dans lequel deux points distincts quelconques admettent toujours des voisinages disjoints. ◊

Exemple II.1 : Les espaces métriques sont séparés.

Définition II.11 (Recouvrement) : Un *recouvrement* d'un ensemble $\mathcal{X}$ est un ensemble $\mathcal{P}$ de sous-ensembles non vides de $\mathcal{X}$ tel que l'union de ces sous-ensembles soit égale à $\mathcal{X}$. Un *recouvrement ouvert* est un recouvrement dont les sous-ensembles sont des ouverts. ◊

b. Compacité des espaces topologiques et métriques

Définition II.12 (Espaces compacts) : En topologie, on dit d'un espace séparé qu'il est *compact* si chaque fois qu'il est recouvert par des ouverts, il est recouvert par un nombre fini d'entre eux. ◊

Il existe une caractérisation séquentielle de la compacité pour les espaces métriques :

Proposition II.1 (Caractérisation séquentielle) : *Un espace métrique $(\mathcal{X}, d)$ est compact si de toute suite de $\mathcal{X}$ on peut extraire une sous-suite convergente dans $\mathcal{X}$.*

2. Complétude

On introduit maintenant la notion d'espace complet. Ce sont des espaces métriques tels que certaines suites particulières, dites suites de Cauchy, convergent.

Définition II.13 (Suite de Cauchy) : Une suite $(x^n)_{n\in\mathbb{N}}$ d'un espace métrique (E, d) est dite *de Cauchy* si pour tout réel $\varepsilon > 0$, il existe un entier naturel N tel que pour tous entiers $p, q \geqslant N$, la distance $d(x^p, x^q)$ soit inférieure à ε : $\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall p, q > N, d(x^p, x^q) < \varepsilon$. ◊

Une suite de Cauchy est donc une suite dont les termes se rapprochent à partir d'un certain rang.

Définition II.14 (Espace métrique complet) : Un espace métrique (E, d) est dit *complet* si toute suite de Cauchy de (E, d) a une limite dans (E, d). ◊

III. Continuité

1. Définition générale

La notion de continuité se définit dans les espaces topologiques de la manière suivante [Sch80].

Définition II.15 (Continuité) : Soit f une application entre deux espaces topologiques. Elle est dite *continue en x* si pour tout voisinage V de $f(x)$, il existe un voisinage de x dont l'image par f est incluse dans V. ◊

2. Le cas des espaces métriques

La continuité s'exprime plus simplement lorsque l'on considère des applications entre espaces métriques.

DÉFINITION II.16 (CONTINUITÉ) : Soient (E, d) et (E', d') deux espaces métriques, $a \in E$ et $f : E \to E'$. On dit que l'application f est *continue en a* si :

$$\forall \varepsilon > 0, \exists \eta > 0, \forall x \in E, d(x,a) \leq \eta \implies d'(f(x), f(a)) \leq \varepsilon.$$

Dans les espaces métriques, la continuité se montre facilement en utilisant la caractérisation séquentielle suivante :

PROPOSITION II.2 : *Soit $f : (E,d) \to (E',d')$ une application entre deux espaces métriques.*
Alors f est continue en $a \in E$ si et seulement si pour toute suite x^n convergent vers a, la suite $f(x^n)$ converge vers $f(a)$.

IV. LES SYSTÈMES DYNAMIQUES DISCRETS

Soit $f : \mathcal{X} \longrightarrow \mathcal{X}$ une application d'un espace topologique ou métrique $\mathcal{X}$ dans lui-même. On considère la suite des itérées définies par la relation de récurrence :

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

Le comportement de ces itérées dépend de la fonction f, et de l'espace sur lequel on itère. D'où la définition [For98] :

DÉFINITION II.17 (SYSTÈME DYNAMIQUE DISCRET) : Un *système dynamique discret* est un couple $(\mathcal{X}, f)$ formé par :

- un espace topologique non vide $(\mathcal{X}, \tau)$, appelé *espace des phases*,
- une fonction continue $f : \mathcal{X} \longrightarrow \mathcal{X}$, appelée *fonction successeur*. ◊

La fonction f peut, dans certains cas, être inversée, ce qui permet alors de « remonter dans le temps ». On parle alors de *réversibilité* [For98] :

DÉFINITION II.18 (SYSTÈME DYNAMIQUE DISCRET RÉVERSIBLE) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *réversible* si f est un homéomorphisme (topologique), i.e. si f est une bijection bicontinue. ◊

Enfin, on s'intéresse à la manière dont un point x considéré évolue au cours du temps. On parle d'orbite :

DÉFINITION II.19 (ORBITE) : Pour $x \in \mathcal{X}$ donné, la suite $(f^{(n)}(x))_{n\in\mathbb{N}}$ est appelée *mouvement positif*, ou *orbite* de x. Elle est notée γ_x. ◊

CHAPITRE 7

Le chaos selon Devaney

Ordo ab chao.

Prophétie
JEAN DE JÉRUSALEM

La théorie mathématique du chaos a connu de nombreux développements depuis la première apparition de ce terme en 1975, l'un des plus célèbres étant la définition de chaos donnée par Robert L. Devaney. Le but de ce chapitre est d'introduire cette définition. D'autres définitions feront l'objet du chapitre suivant. Le lecteur souhaitant approfondir le sujet pourra se référer au livre de Devaney [Dev03], ainsi qu'aux thèses d'Enrico Formenti [For98], [For03] et de Sylvie Ruette [Rue01].

I. PÉRIODICITÉ, ÉQUILIBRE, ET RÉGULARITÉ

La théorie du chaos cherche à savoir si le comportement d'un système dynamique discret peut être prévu ou pas, c'est-à-dire si l'on peut deviner quelle va être l'orbite γ_x d'un point x donné. En ce sens, les points dont le comportement est le plus facile à appréhender sont les points périodiques et les points d'équilibre.

1. Points périodiques

DÉFINITION II.20 (POINT PÉRIODIQUE) : Un point $p \in \mathcal{X}$ est dit *périodique* de *période* k si k est un entier naturel non nul tel que $f^{(k)}(p) = p$, et $\forall h \in [\![0; k-1]\!]$, $f^{(h)}(p) \neq p$. ◊

NOTATION II.2. On note $Per_k(f)$ l'ensemble des points k-périodiques de f, et $Per(f)$ l'ensemble des points périodiques de période quelconque.

La périodicité peut intervenir après une phase plus ou moins longue de transition, ce qui nous amène à introduire une variante à la précédente définition.

DÉFINITION II.21 (POINT ULTIMEMENT PÉRIODIQUE) : Un point est *ultimement périodique* s'il existe deux entiers n et p tels que $f^{(n+p)}(x) = f^{(p)}(x)$. Soit n_0 le plus petit n vérifiant cela. L'ensemble $\{x, f(x), \ldots, f^{(n_0)}(x)\}$ est alors appelé *transitoire* de x, et n_0 est la *longueur du transitoire*. ◊

2. Points d'équilibre

Les points d'équilibre sont les points ayant la plus simple des orbites.

Définition II.22 (Points d'équilibre) : Les points périodiques de période 1 sont appelés *points fixes* de f, ou encore *points d'équilibre*.
De même, les points ultimement périodiques de période 1 sont appelés *points ultimement fixes*. ◊

3. Systèmes réguliers

En topologie, le concept de densité d'un sous-ensemble A d'un espace topologique $\mathcal{X}$ permet de refléter l'idée que pour tout point x de $\mathcal{X}$ on peut trouver un point de A qui soit aussi proche de x que possible [Sch80].

Définition II.23 (Espace Dense) : Soit $\mathcal{X}$ un espace topologique et A un sous-ensemble de $\mathcal{X}$. On dit que A est *dense* dans $\mathcal{X}$ si pour tout élément x de $\mathcal{X}$, tout voisinage de x contient au moins un point de A. ◊

Nous sommes maintenant en mesure de définir un premier aspect du chaos, à partir des points périodiques [For98] :

Définition II.24 (Systèmes Réguliers) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *régulier* si l'ensemble des points périodiques de f est dense dans $\mathcal{X}$. ◊

Remarque. Dans un espace métrique $(\mathcal{X}, d)$, le système dynamique $(\mathcal{X}, f)$ est régulier si, et seulement si $\forall x \in \mathcal{X}, \forall \varepsilon > 0, \exists p \in Per(f), d(x,p) < \varepsilon$.

Bien que le terme « régulier » semble s'opposer à l'idée de « chaos », on pourrait quand même considérer que, sous un certain angle, la définition ci-dessus pourrait engendrer un certain type particulier d'imprévisibilité : si l'on fait une simulation numérique de l'évolution du système, de petites erreurs dans les conditions initiales *peuvent* mener sur une orbite radicalement différente de celle qui fait l'objet de la simulation. Par exemple, passer d'une petite période vers une grande période, ou une absence de période.

Cette régularité ne permet évidemment pas de définir, à elle seule, une notion de chaos. Il y a trop de cas pathologiques, et des systèmes élémentaires seraient chaotiques : l'identité, les permutations, *etc.* Cependant, il est possible de définir un certain chaos à partir de la variété des points périodiques, comme on le verra au chapitre 8.

II. Simplification des systèmes dynamiques discrets

Un point central de l'étude des systèmes dynamiques est de caractériser leurs comportements asymptotiques. On est ainsi amené à s'intéresser aux sous-ensembles de l'espace des phases qui restent stables sous l'action du système.

1. Invariance, sous-systèmes dynamiques

Définition II.25 (Invariance positive) : Soit $(\mathcal{X}, f)$ un système dynamique discret. Une partie A de $\mathcal{X}$ est dite *positivement invariante* si $f(A) \subset A$, et *strictement positivement invariante* en cas d'égalité. ◊

Exemple II.2 : Les ensembles $Per(f)$, $Per_k(f)$ sont positivement invariants.

Un ensemble positivement invariant est donc un « ensemble piège » : une fois tombé dedans, on y reste [For98]. Si $A \subset \mathcal{X}$ est positivement invariante, on a alors évidemment :

$$\forall n \in \mathbb{N}, f^n(A) \subset A$$

Le couple (A, f) peut donc être considéré comme un sous-système dynamique de $(\mathcal{X}, f)$. En d'autres termes, on peut étudier les parties positivement invariantes de $\mathcal{X}$ séparément ; elles sont plus petites, donc probablement plus faciles à appréhender.

2. (In)décomposabilité et transitivité

Certains systèmes sont simples à étudier, car on peut intégralement les décomposer en sous-systèmes positivement invariants. Ce n'est pas le cas des systèmes transitifs.

a. Décomposabilité

Les systèmes décomposables sont des systèmes admettant certains recouvrements (*c.f.* définition II.11) particuliers évoluant de manière prévisible sous l'action de f.

Définition II.26 (Décomposabilité) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *décomposable* s'il existe un recouvrement fini ouvert (contenant au moins deux éléments) de $\mathcal{X}$ tel que chaque ouvert du recouvrement est un ensemble positivement invariant de f. ◊

Des sous-systèmes dynamiques indépendants agissent donc sur chaque élément de ce recouvrement, et l'on peut étudier chaque ouvert séparément, et en déduire le comportement complet de $(\mathcal{X}, f)$. L'étude du système dans son ensemble s'en trouve ainsi simplifiée.

b. Transitivité

La *transitivité* est le « contraire » de la décomposabilité.

Définition II.27 (Transitivité) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *transitif* si pour chaque couple d'ouverts non vides $A, B \subset \mathcal{X}$, il existe $k \in \mathbb{N}$ tel que $f^{(k)}(A) \cap B \neq \varnothing$. ◊

La transitivité entraîne l'indécomposabilité, condition d'irréductibilité que l'on définit ainsi [For98] :

Définition II.28 (Indécomposabilité) : Un système dynamique discret $(\mathcal{X}, f)$ est *indécomposable* si et seulement s'il n'est pas la réunion de deux parties non vides, fermées et positivement invariantes. ◊

Donnons maintenant quelques conséquences du fait d'être transitif [For98], [For03], afin de mieux appréhender cette notion centrale dans la théorie du chaos.

c. Conséquences de la transitivité

Tout d'abord, quand un système dynamique discret est transitif, on peut être aussi proche que l'on veut de tout $x \in \mathcal{X}$ en « partant » de n'importe quel ouvert de $\mathcal{X}$. Plus précisément,

Proposition II.3 : *Un système dynamique discret $(\mathcal{X}, f)$ est transitif si et seulement si pour tout ouvert A non vide,*

$$\overline{\bigcup_{n\in\mathbb{N}} f^{(n)}(A)} = \mathcal{X}$$

En d'autres termes, un système est transitif si l'on peut rejoindre n'importe quel ouvert à partir de n'importe quel autre ouvert :

Proposition II.4 : *Un système dynamique discret $(\mathcal{X}, f)$ est transitif si et seulement si pour tout couple $(x, y) \in \mathcal{X}^2$, et toutes boules B_x, B_y respectivement centrées en x et y, on a :*

$$\exists z \in B_x, \exists n_0 \in \mathbb{N}, f^{(n_0)}(z) \in B_y$$

Enfin, les systèmes transitifs visitent tout l'espace, ils n'abandonnent aucun lieu [For98] :

Proposition II.5 : *Si $(\mathcal{X}, f)$ est transitif, alors $\overline{f(\mathcal{X})} = \mathcal{X}$*

d. Transitivité dans les espaces compacts

On rappelle certains résultats particuliers dans le cas où $(\mathcal{X}, f)$ est un système dynamique discret transitif avec $(\mathcal{X}, d)$ compact. Le lecteur trouvera les démonstrations de ces résultats dans [For98], [For03], et [Rue01].

Tout d'abord, la transitivité possède une formulation équivalente dans les espaces compacts :

Proposition II.6 : *Dans les espaces compacts, la transitivité est équivalente à avoir une orbite dense.*

En d'autres termes, vu que l'espace est relativement petit (compact), la transitivité se traduit par le fait qu'on peut trouver un point qui visitera « presque » tout l'espace. De plus, en cas de compacité, f est forcément surjective :

Proposition II.7 : *Soit $(\mathcal{X}, f)$ un système dynamique discret transitif, tel que $(\mathcal{X}, d)$ est compact. Alors $f(\mathcal{X}) = \mathcal{X}$.*

3. Formulations plus fortes de la transitivité

Il existe des versions plus fortes de la transitivité, qui conduisent à de plus fortes imprévisibilités.

a. Transitivité totale

Certains systèmes dynamiques peuvent être plus facilement appréhendés en n'étudiant pas directement la suite $x^n = f^{(n)}(x^0)$, mais certaines de ses sous-suites : les sous-suites des termes pairs et impairs, par exemple. La transitivité totale étend la propriété de transitivité à toutes les sous-suites de $x^n = f^{(n)}(x^0)$.

Définition II.29 (Transitivité totale) : f est dite *totalement transitive* quand $\forall n \geqslant 1$, l'application composée $f^{(n)}$ est transitive. ◇

b. Transitivité forte

Définition II.30 (Transitivité forte) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *fortement transitif* si

$$\forall x, y \in \mathcal{X}, \forall r > 0, \exists z \in B(x, r), \exists n \in \mathbb{N}, f^{(n)}(z) = y.$$

En d'autres termes, pour tout couple x, y, il existe un point aussi proche que l'on veut de x dont une itérée *est égale à* y : non seulement on peut rejoindre n'importe quel autre ouvert, mais on peut atteindre n'importe quel point précis.

Notons pour finir le résultat suivant [For98] :

Proposition II.8 : *Dans les espaces métriques compacts, transitivité et transitivité forte coïncident.*

c. Mélange topologique

Le mélange topologique est une autre version plus forte de transitivité :

Définition II.31 (Mélange topologique) : Un système dynamique discret est *topologiquement mélangeant* si et seulement si pour toute paire d'ouverts disjoints et non vides U et V, il existe $n_0 \in \mathbb{N}$ tel que $\forall n \geqslant n_0, f^{(n)}(U) \cap V \neq \varnothing$. ◊

4. Systèmes dynamiques discrets parfaits

Pour finir cette section, on introduit la définition d'espaces particuliers dits *parfaits*, dans lesquels la transitivité s'obtient dès que l'on a découvert un point d'orbite dense.

Définition II.32 (Point d'accumulation) : On dit qu'un point x d'un espace topologique (E, τ) est *point d'accumulation* d'une partie A de E si tout voisinage de x contient une infinité de points de A.◊

Définition II.33 (Systèmes parfaits) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *parfait* si chaque point de $\mathcal{X}$ est un point d'accumulation dans $\mathcal{X}$. ◊

On peut montrer que :

Proposition II.9 : *Si $(\mathcal{X}, f)$ est parfait et a une orbite dense, alors il est transitif.*

III. Stabilité, sensibilité et expansivité

Après avoir étudié les points périodiques dans la section 7.1 et les sous-ensembles stables dans la section 7.2, après avoir envisagé dans quelle mesure ces propriétés pouvaient produire de l'imprévisibilité, il nous reste à discuter de l'allure des orbites, et des conséquences que l'on peut en tirer. Plus précisément, un système sera stable quand deux points proches conduiront à deux orbites similaires. Dans le cas contraire on aura à nouveau de l'imprévisibilité : instabilité, sensibilité ou expansivité.

1. Stabilité et instabilité

Commençons par définir ce qu'est une orbite stable.

DÉFINITION II.34 (ORBITE STABLE) : Une orbite positive γ_x est dite *stable* si

$$\forall \varepsilon > 0, \exists \delta > 0, \forall y \in \mathcal{X}, d(x,y) < \delta \Longrightarrow \forall n \in \mathbb{N}, d(\gamma_x^n, \gamma_y^n) < \varepsilon$$

x est alors appelé *point stable* de f. ◊

En d'autres termes, si y est proche de x, alors l'orbite de y sera proche de celle de x. Au voisinage d'un point stable x, tous les points évoluent de la même manière : si l'on commet une petite erreur sur la condition initiale, on peut garantir que l'erreur entre le phénomène observé et l'évolution théorique reste minime.

L'instabilité est simplement la négation de la stabilité.

DÉFINITION II.35 (INSTABILITÉ) : Un mouvement positif γ_x est *instable* si

$$\exists \varepsilon > 0, \forall \delta > 0, \exists y \in \mathcal{X}, \exists n \in \mathbb{N}, d(x,y) < \delta \text{ et } d(\gamma_x^n, \gamma_y^n) \geqslant \varepsilon$$

Un système ayant tous ses points stables est fortement prévisible : on le qualifiera de «stable». Dans la situation diamétralement opposée, on parlera de système instable :

DÉFINITION II.36 (SYSTÈME (IN)STABLE) : Un système dynamique discret est *stable* si toutes ses orbites positives sont stables. Il est dit *instable* si toutes ses orbites positives sont instables. ◊

2. Sensibilité aux conditions initiales

La sensibilité aux conditions initiales est plus forte que l'instabilité. Ici, ε ne dépend plus du point x considéré :

DÉFINITION II.37 : Un système dynamique discret $(\mathcal{X}, f)$ est *sensible aux conditions initiales* s'il existe $\varepsilon > 0$ tel que

$$\forall x \in \mathcal{X}, \forall \delta > 0, \exists y \in \mathcal{X}, \exists n \in \mathbb{N}, d(x,y) < \delta \text{ et } d(\gamma_x^n, \gamma_y^n) \geqslant \varepsilon$$

ε est alors appelée *constante de sensibilité*. ◊

Un système est donc sensible si pour chaque x, il existe des points arbitrairement proches de x dont les orbites respectives sont séparées d'au moins ε pendant l'évolution du système [For98]. Cet ε est fixé une fois pour toutes, il est le même pour chaque point x.

REMARQUE. Tous les points voisins de x ne sont pas forcément séparés de ε pendant l'évolution du système : il suffit qu'il en existe au moins un dans chaque boule ouverte de centre x.

3. Expansivité

a. Définition

Dans un système expansif, *toute* erreur sur la position initiale est amplifiée, jusqu'à atteindre (au moins) la constante d'expansivité ε :

Définition II.38 (Expansivité) : Un système dynamique discret est *expansif* si

$$\exists \varepsilon > 0, \forall x \neq y, \exists n \in \mathbb{N}, d(\gamma_x^n, \gamma_y^n) \geqslant \varepsilon$$

ε est alors appelée la *constante d'expansivité*. ◊

Remarque. L'expansivité ne suffit pas pour définir le chaos. En effet, certains systèmes linéaires sont expansifs. Or, la non-linéarité est reconnue, par un certain nombre d'auteurs, comme une caractéristique importante des phénomènes chaotiques. De plus, un système expansif est quelque part un peu prévisible, dans la mesure où toutes les erreurs sont amplifiées d'au moins ε : il n'y a pas d'incertitude à ce niveau, et l'on connaît ainsi *quelque chose* du comportement du système.

b. Exemple

Introduisons l'application suivante, dite *doublement de l'angle* :

Définition II.39 (Doublement de l'angle) : Soit $\mathbb{S}^1 = \mathbb{R}/\mathbb{Z}$ le cercle unité, muni de la distance naturelle. L'application *doublement de l'angle* est

$$\begin{cases} f : & \mathbb{S}^1 & \longrightarrow & \mathbb{S}^1 \\ & x & \longmapsto & 2x \end{cases} \tag{7.1}$$

◊

Ce doublement de l'angle est expansif. En effet, toute « erreur » double à chaque itération.

c. Cas des systèmes parfaits

On peut très bien avoir des systèmes expansifs qui ne soit pas sensibles aux conditions initiales, comme le montre l'exemple suivant extrait de la thèse de E. Formenti [For98] :

Exemple II.3 : Soit $\mathcal{X}$ un espace fini muni de la distance triviale δ ($\delta(x, y) = 0$ si $x = y$, et $\delta(x, y) = 1$ sinon). Soit $f : \mathcal{X} \to \mathcal{X}$ bijective.

- Alors $(\mathcal{X}, f)$ est expansif, avec une constante d'expansivité égale à 1.
- Par contre, $(\mathcal{X}, f)$ n'est pas sensible aux conditions initiales, car $\forall x \in \mathcal{X}, \forall \zeta < 1$, il n'y a aucun y tel que $\delta(x, y) < \zeta$ et $\delta\left(f^n(x), f^n(y)\right) \geqslant \varepsilon, \forall n, \forall \varepsilon$.

Pour éviter d'avoir des systèmes expansifs mais pas sensibles aux conditions initiales, il faut par exemple supposer $\mathcal{X}$ parfait [For98].

IV. Le chaos selon Devaney (1989)

Nous sommes dorénavant en mesure de donner la plus célèbre définition de chaos mathématique, celle de Devaney [Dev03].

Définition II.40 (système dynamique discret chaotique) : Un système dynamique discret est *chaotique selon Devaney* s'il est régulier et transitif. ◊

La définition originelle de Devaney comprenait en plus la sensibilité aux conditions initiales. Cependant, Banks *et al.* ont prouvé dans [BBCS92] que la définition était redondante sur les espaces métriques :

PROPOSITION II.10 (BANKS *et al.*) : *Si un système dynamique discret sur un espace métrique est chaotique au sens de la définition II.40, alors il est sensible aux conditions initiales.*

REMARQUE. La régularité et la transitivité sont des propriétés topologiques, la sensibilité aux conditions initiales est une propriété métrique. Le chaos, tel qu'on le définit ici, est donc purement topologique, et a d'importantes conséquences métriques.

Citons quelques mots de Devaney pour conclure [Dev03] : « A chaotic dynamical system is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or simplified into two subsystems which do not interact because of topological transitivity. And in the midst of this random behavior, we nevertheless have an element of regularity ». Ainsi, des comportements fondamentalement différents sont possibles (périodicité, divergence, *etc.*), et ils se produisent de manière imprévisible.

V. EXEMPLES DE SYSTÈMES CHAOTIQUES

1. Le doublement de l'angle

Le doublement de l'angle sera notre premier exemple de système dynamique chaotique selon Devaney. En effet, on peut prouver que [Dev03] :

PROPOSITION II.11 : *Le doublement de l'angle est chaotique au sens de Devaney.*

REMARQUE. Le doublement de l'angle est l'un des exemples les plus classiques de sensibilité aux conditions initiales (une erreur est doublée à chaque itération).

On peut considérer que le doublement de l'angle consiste en des itérations sur un intervalle réel, *i.e.* des itérations de la fonction suivante :

$$\begin{array}{rccc} f : & [0;1[& \longrightarrow & [0;1[\\ & x & \longmapsto & 2x \pmod 1 \end{array}$$

dont le graphe est donné en figure 7.1(a). Les figures 7.1(b) à 7.1(d) illustrent la sensibilité en prenant différentes valeurs de x^0 assez proches, menant à des itérations complètement différentes.

2. La fonction tente

DÉFINITION II.41 (FONCTION TENTE) : La *fonction tente* (« tent map ») est définie sur $[0;1]$ par :

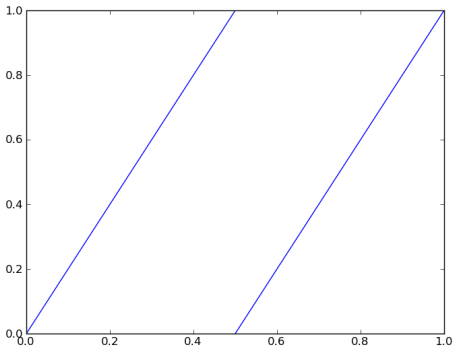
$$T(x) = \begin{cases} 2x & 0 \leqslant x \leqslant \frac{1}{2} \\ 2(1-x) & \frac{1}{2} \leqslant x \leqslant 1 \end{cases} \tag{7.2}$$

♢

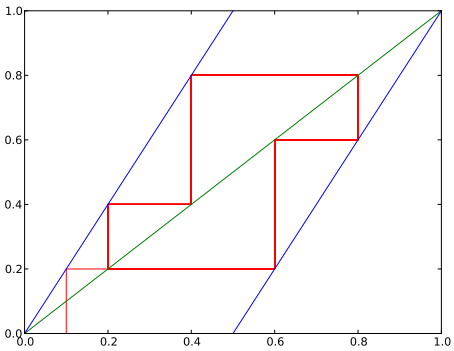
Le nom de cette fonction vient simplement de sa forme (*c.f.* figure 7.2). On peut alors facilement démontrer que [Dev03] :

PROPOSITION II.12 : *Le système dynamique $([0;1], T)$ de la fonction tente est un système chaotique au sens de Devaney.*

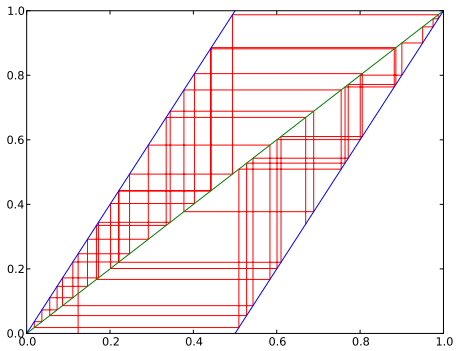
La courbe de la fonction tente, et des exemples d'itérations du système dynamique associé, sont fournis à la figure 7.2.



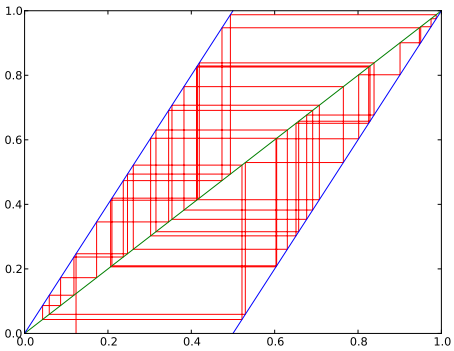
(a) La fonction doublement de l'angle.



(b) Itérations pour $x^0 = 0,1$.



(c) Itérations pour $x^0 = 0,12344$.



(d) Itérations pour $x^0 = 0,12345$.

FIGURE 7.1 – Le doublement de l'angle.

3. Le chat d'Arnold (1968)

Introduisons pour finir le *chat*[3] *d'Arnold*.

DÉFINITION II.42 (LE CHAT D'ARNOLD) : Le *chat d'Arnold* est la suite à valeurs dans $[0;1]^2$, définie par :

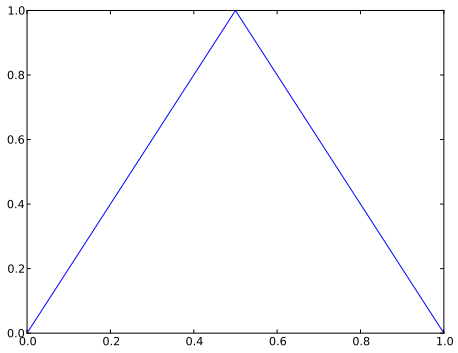
$$\begin{cases} x^{n+1} &= x^n + y^n \pmod 1 \\ y^{n+1} &= x^n + 2\,y^n \pmod 1 \end{cases}$$

En d'autres termes, le système dynamique consiste ici à itérer la fonction $f((x,y)) = (x+y, x+2y)$ sur le carré unité (voir figure 7.3). Il est possible de montrer que :

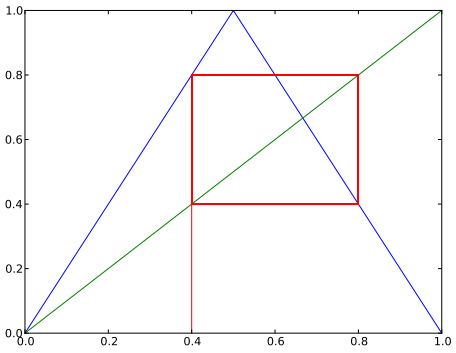
PROPOSITION II.13 : *Le chat d'Arnold est chaotique au sens de Devaney.*

Nous verrons dans les états de l'art des différentes applications à venir (parties IV à VI) que ce chat d'Arnold est l'une des suites chaotiques les plus utilisées à l'heure actuelle (cela est principalement dû au fait qu'il s'agit là d'une suite à deux composantes).

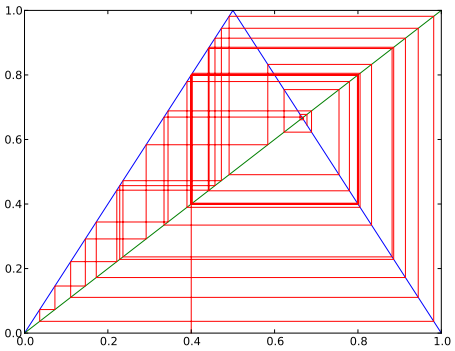
3. Arnold utilisait le mot *cat* comme abréviation de « Continuous Automorphisms of the Torus ».



(a) La fonction tente.

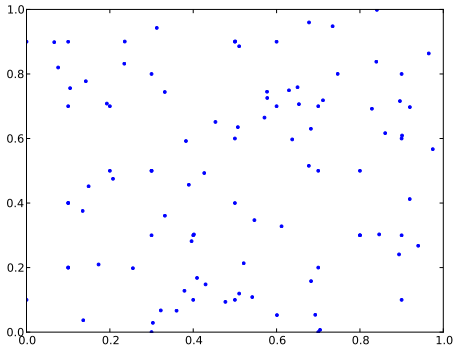


(b) Itérations pour $x^0 = 0,4$.

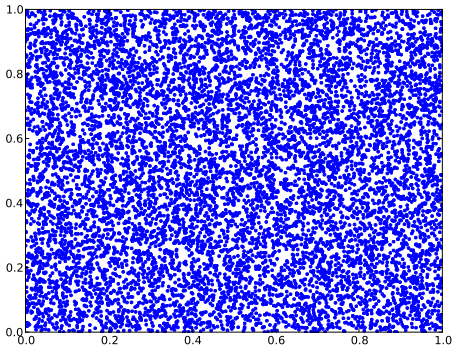


(c) Itérations pour $x^0 = 0,40001$.

FIGURE 7.2 – La fonction tente.



(a) 100 itérations pour $(x^0, y^0) = (0,1; 0,2)$.



(b) 10000 itérations pour $(x^0, y^0) = (0,1; 0,2)$.

FIGURE 7.3 – Itérations du chat d'Arnold.

CHAPITRE 8

Autres définitions topologiques du chaos

L'occulte a toujours fasciné l'inculte.

Les Pensées
JACQUES STERNBERG

Dans ce chapitre, on montre que la définition de Devaney n'est pas complètement satisfaisante, et l'on propose d'autres définitions topologiques de chaos. Il n'y a pas vraiment de définition parfaite, chacune éclaire un comportement chaotique d'une certaine lumière.

I. CRITIQUE DE LA DÉFINITION DE DEVANEY

La définition de Devaney est bien établie, mais n'est pas universellement reconnue. Certains auteurs se contentent de la sensibilité aux conditions initiales, d'autres en physique exigent une non-linéarité, voire un attracteur étrange.

Une des raisons qui font que la définition de Devaney ne s'est pas complètement imposée provient du fait suivant (Knudsen [Knu94b], rapporté dans [For98]).

PROPOSITION II.14 : *Soient* $(\mathcal{X}, f)$ *et* $(\mathcal{Y}, f)$ *deux systèmes dynamiques discrets tels que* $\mathcal{Y}$ *est dense dans* $\mathcal{X}$*. Alors :*

- *Si* $(\mathcal{X}, f)$ *est transitif, alors* $(\mathcal{Y}, f)$ *l'est aussi.*
- *Si* $(\mathcal{X}, f)$ *est sensible aux conditions initiales, alors* $(\mathcal{Y}, f)$ *l'est aussi.*

Donc si $(\mathcal{X}, f)$ est chaotique au sens de Devaney, alors le sous-système $(Per(f), f)$ l'est aussi. Et pourtant, toute orbite de $(Per(f), f)$ est périodique, ce qui n'est pas ce que l'on attend habituellement d'un comportement chaotique [Knu94a].

Il se peut évidemment qu'à proximité de tout point périodique, il existe un autre point périodique de période aussi grande que l'on veut, ce qui est bien une sorte de chaos. Cependant, le chaos de ce système semble moins prononcé que celui d'un système où de vraies orbites non-périodiques côtoient des orbites périodiques.

Finalement, comme le « hasard », le « chaos » est un terme très difficile à définir mathématiquement, et l'on ne saurait parvenir à enfermer des comportements si complexes et si variés sein d'une définition bien choisie : si tel était le cas, ces chaos ne seraient pas si difficiles à appréhender.

Nous devrons donc faire avec une pluralité de définitions, chacune illustrant un des aspects de l'imprévisibilité au sens large. Nous obtiendrons alors des chaos plus ou moins prononcés, suivant le nombre de propriétés d'imprévisibilité que de tels systèmes possèdent.

II. D'AUTRES DÉFINITIONS DE CHAOS DU TYPE DEVANEY

On donne ici des définitions proches de celle de Devaney. Ces dernières se trouvent dans la littérature, et ont chacune leur propre intérêt.

1. Chaos de la sensibilité aux conditions initiales

La définition suivante de chaos est sans doute celle qui est la plus utilisée, notamment par les physiciens [MDS98] :

DÉFINITION II.43 (CHAOS DE LA SENSIBILITÉ AUX CONDITIONS INITIALES) : On dit qu'un système dynamique $(\mathcal{X}, f)$ présente le *chaos de la sensibilité aux conditions initiales* dans un ensemble positivement invariant $\mathcal{X}$ si la fonction f est sensible aux conditions initiales dans $\mathcal{X}$. ◊

Autrement dit, on peut avoir des orbites très différentes pour des points très proches au départ. Il en découle que l'évolution du système est imprévisible, puisque de petites erreurs de mesures aux conditions initiales sont inévitables. De plus, cette imprévisibilité se vérifie numériquement.

En dépit de ses avantages, cette définition de chaos n'est pas tout à fait satisfaisante, comme on peut le voir dans l'exemple qui va suivre.

Exemple II.4 : Soit $D = \{x \in \mathbb{R}^2 : \|x\| \leqslant 2\}$. Utilisons les coordonnées polaires pour définir $f : D \to D$ par $f(x) = f(\rho, \theta) = (\rho, \theta + \rho)$.

On remarque que pour tout $\rho \in]0; 2]$, l'ensemble défini par $C_\rho = \left\{x \in \mathbb{R}^2 \,\middle/\, \|x\| = \rho\right\}$ est invariant et que le système défini par f est une rotation dans C_ρ. Dans ce cas on ne peut pas dire que f soit réellement imprévisible sur C_ρ. Pourtant, on peut prouver que le système a une sensibilité aux conditions initiales dans C_ρ avec une constante $r_0 = \rho$ [Led02].

2. Chaos au sens de Wiggins

Une autre idée consiste à retirer la régularité, qui ne semble pas si naturelle que cela. Ainsi, à la différence de Devaney, Wiggins ne pense pas que la densité des orbites périodiques est un élément clé dans l'apparition des phénomènes chaotiques. Il conserve le fait que le système ne puisse pas être simplifié (transitivité), et y adjoint la sensibilité aux conditions initiales [MDS98] :

DÉFINITION II.44 (CHAOS SELON WIGGINS) : Un système dynamique discret est dit *chaotique selon Wiggins* s'il est transitif et sensible aux conditions initiales. ◊

Cette définition s'accorde assez bien avec l'idée que l'on peut se faire d'un système chaotique : sensible aux conditions initiales, et non simplifiable. Malheureusement, la définition de Wiggins inclut des chaos plus dégénérés, plus pauvres que ceux reconnus par Devaney. Ainsi, on peut reconnaître des comportements chaotiques (selon Wiggins) sur un ensemble fini de points, voir même sur un singleton, comme le montre l'exemple suivant.

Exemple II.5 : Le système $(\mathcal{X}, f)$, où $\mathcal{X} = \left\{\frac{1}{3}\right\}$ et f est définie sur $[-1;1]$ par $f(x) = -2|x| + 1$, est chaotique selon Wiggins.

3. Chaos au sens de Knudsen

Une autre approche consiste à essayer de solutionner le problème soulevé par la proposition II.14. C'est ce que Knudsen a fait dans [Knu94a] et [Knu94b], en proposant la définition suivante de chaos :

DÉFINITION II.45 (CHAOS SELON KNUDSEN) : Un système dynamique discret est dit *chaotique selon Knudsen* s'il a une orbite dense et s'il est sensible aux conditions initiales. ◊

Ce chaos est moins exigeant que celui de Devaney quand on est sur espace compact [For98] :

PROPOSITION II.15 : *Si $\mathcal{X}$ est compact, alors être chaotique au sens de Devaney implique être chaotique au sens de Knudsen.*

4. Chaos expansif

Enfin, dans la définition originale de Devaney [Dev03], il y avait la sensibilité aux conditions initiales. Cette dernière étant une conséquence de la régularité et de la transitivité, elle ne fait plus partie des formulations actuelles du chaos selon Devaney. Certains auteurs regrettent qu'une propriété de type « effet papillon » ne soit plus clairement présente dans la définition même de chaos, et la réintroduisent au-travers de l'expansivité [For98] :

DÉFINITION II.46 (CHAOS EXPANSIF) : Un système dynamique discret est en *chaos expansif* s'il est transitif, régulier et expansif. ◊

REMARQUE. Le rajout de l'expansivité à la définition de Devaney aboutit, dans une certaine mesure, à un chaos « plus fort » que celui de Devaney. D'un autre côté, nous l'avons déjà signalé précédemment : un système en chaos expansif a quelque chose de prévisible...

III. CHAOS DE LA MULTIPLICITÉ DES PÉRIODES

1. Présentation

C'est en 1973, dans leur article « Period three implies chaos », que Li et Yorke furent les premiers à utiliser le terme chaos pour décrire le comportement désordonné d'un système dynamique [LY75]. Leur résultat est relié au théorème de Sarkovskii.

Le théorème de Sarkovskii est un théorème de mathématiques portant sur l'itération des fonctions continues. Il donne des contraintes sur la présence de points périodiques lorsque l'on itère la fonction f. Pour énoncer ce théorème, il nous faut tout d'abord introduire l'ordre de Sarkovskii.

2. Ordre de Sarkovskii

L'*ordre de Sarkovskii* est une relation d'ordre définie sur les entiers strictement positifs de la façon suivante :

$$3 \triangleleft 5 \triangleleft 7 \triangleleft \ldots \triangleleft 2\times 3 \triangleleft 2\times 5 \triangleleft 2\times 7 \triangleleft \ldots \triangleleft 2^n\times 3 \triangleleft 2^n\times 5 \triangleleft 2^n\times 7 \triangleleft \ldots$$
$$\ldots \triangleleft 2^{n+1}\times 3 \triangleleft 2^{n+1}\times 5 \triangleleft \ldots \triangleleft 2^n \triangleleft 2^{n-1} \triangleleft \ldots \triangleleft 2^2 \triangleleft 2 \triangleleft 1$$

Autrement dit, on place d'abord les impairs à partir de 3 par ordre croissant, puis les impairs multipliés par 2, puis par 4, *etc.* et on termine par les puissances de 2 par ordre décroissant.

3. Le théorème de Sarkovskii

Le théorème de Sarkovskii s'énonce ainsi :

PROPOSITION II.16 (THÉORÈME DE SARKOVSKII) : *Soit f une fonction continue sur un intervalle réel I, à valeurs dans I.*

Si f admet un point périodique de période n, alors pour tout m succédant à n dans l'ordre de Sarkovskii, f admet un point périodique de période m.

Ainsi, si f admet un point périodique de période 3, alors f admet des points périodiques de n'importe quelle période. De cette pluralité de périodes se dégage un certain chaos...

4. Le chaos de la multiplicité des périodes

DÉFINITION II.47 (CHAOS DE LA MULTIPLICITÉ DES PÉRIODES) : Un système dynamique discret $(\mathcal{X}, f)$ est dit *chaotique au sens de la multiplicité des périodes* s'il possède des points périodiques de période $k, \forall k \in \mathbb{N}$. ◇

Pour illustrer ce chaos, commençons par rappeler la définition de la suite logistique :

DÉFINITION II.48 (SUITE LOGISTIQUE) : Soit $\mu \in [0,4]$, et $I = [0,1]$. On pose :

$$\begin{array}{rccc} g_\mu : & I & \longrightarrow & I \\ & x & \longmapsto & \mu\, x\, (1-x) \end{array}$$

On appelle *suite logistique* le système dynamique (I, g_μ), *i.e.* la suite définie par $x^{n+1} = \mu\, x^n\, (1 - x^n)$. Des exemples d'évolution de ce système sont données aux figures 8.1 et 8.2. ◇

Exemple II.6 : Dans l'application logistique, pour $\mu > 3,82845$, il y a apparition d'un 3-cycle. On en déduit, en utilisant le théorème de Sarkovskii, que ce système dynamique discret est chaotique sur [0,1], au sens de la multiplicité des périodes. Voir la figure 8.2 pour une illustration de cela.

Li et Yorke n'ont pas défini de notion de chaos dans leur article de 1975, ils ont juste introduit le terme dans le titre de leur papier. Il existe cependant un chaos selon Li et Yorke, que l'on détaille dans la section ci-dessous [Rue01].

IV. CHAOS SELON LI ET YORKE

1. Rappels sur les notions de limite inférieure et supérieure

La notion de limite d'une suite, telle qu'on la connaît, a été introduite en 1821 par Cauchy. Toutes les suites ne convergent cependant pas. Pour étudier la variation d'une suite non-convergente, plus précisément ses oscillations, Cauchy a introduit la même année la notion de limite supérieure, que l'on définit comme suit.

DÉFINITION II.49 (LIMITE SUPÉRIEURE) : Considérons une suite bornée de réels (x^n), et notons y^n la borne supérieure de l'ensemble constitué de tous les termes de la suite d'indice supérieur ou égal à n :

$$y^n = \sup_{p \geqslant n} x^p$$

Quand n augmente, le cardinal de l'ensemble $A_n = \left\{ x^p \;\middle/\; p \geqslant n \right\}$ diminue, et donc sa borne supérieure dimunue également : la suite (y^n) décroît. Comme la suite (x^n) est minorée, il en est de même pour (y^n), ce qui montre la convergence de cette dernière.

Sa limite L, notée $\limsup_{n\to+\infty} x^n$, s'appelle la *limite supérieure* de cette dernière. ◊

Remarque. On définit de la même façon la limite inférieure.

Nous utiliserons par la suite la proposition suivante :

Proposition II.17 : $\limsup_{n\to+\infty} x^n$ *et* $\liminf_{n\to+\infty} x^n$ *sont respectivement la plus grande et la plus petite des valeurs d'adhérence de la suite* $(x^n)_{n\geqslant 0}$. *En particulier, si la suite* $(x^n)_{n\geqslant 0}$ *possède une sous-suite qui converge vers l, alors :*

$$\limsup_{n\to+\infty} x^n \geqslant l \geqslant \liminf_{n\to+\infty} x^n$$

Nous avons maintenant les outils pour introduire la notion de chaos au sens de Li-Yorke.

2. Chaos de Li-Yorke

La définition de Li et Yorke fait intervenir la notion de couple de Li-Yorke [Rue01] :

Définition II.50 (Couple de Li-Yorke) : Soit $(\mathcal{X}, f)$ un système dynamique topologique. Si $x, y \in \mathcal{X}$, on dit que (x, y) est un *couple de Li-Yorke* si

$$\limsup_{n\to+\infty} d(f^{(n)}(x), f^{(n)}(y)) > 0 \text{ et } \liminf_{n\to+\infty} d(f^{(n)}(x), f^{(n)}(y)) = 0$$

Les ensembles brouillés sont des ensembles constitués exclusivement de couples de Li-Yorke :

Définition II.51 (Ensemble brouillé) : On appelle *ensemble brouillé* un ensemble $B \subset \mathcal{X}$ tel que tout couple de points distincts de B est un couple de Li-Yorke. ◊

Nous sommes maintenant en mesure de définir la notion de chaos au sens de Li-Yorke [Rue01] :

Définition II.52 (Chaos au sens de Li-Yorke) : Soit $\mathcal{X}$ un espace métrique compact, et $f : \mathcal{X} \to \mathcal{X}$ une application continue.

Le système $(\mathcal{X}, f)$ est dit *chaotique au sens de Li-Yorke* s'il contient un ensemble brouillé non dénombrable. ◊

V. Exposant de Lyapunov

On l'a vu, certains systèmes dynamiques sont très sensibles aux petites variations de leur condition initiale. Les constantes de sensibilité aux conditions initiales et d'expansivité illustrent cela. Cependant, dans certains cas, ces variations peuvent rapidement prendre d'énormes proportions, croître de manière exponentielle, et aucune des propriétés énoncées ci-dessus ne permet d'illustrer cela.

Le mathématicien russe Alexander Lyapunov s'est penché sur ce phénomène et a proposé l'exposant du même nom [Spi74]. Cet exposant permet de mesurer la vitesse à laquelle ces petites variations peuvent s'amplifier :

DÉFINITION II.53 (EXPOSANT DE LYAPUNOV) : L'*exposant de Lyapunov* du système :

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

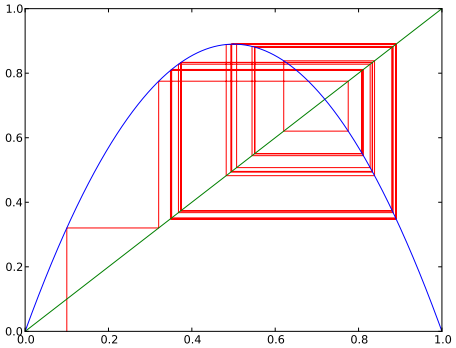
est défini par :

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

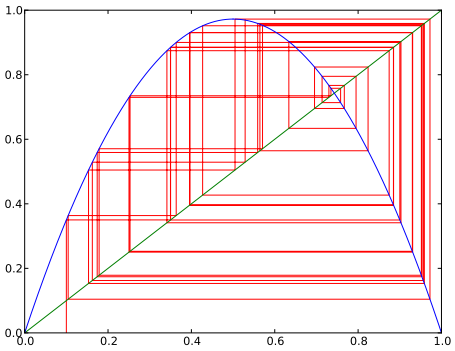
Considérons un système dynamique quelconque, dont la condition initiale x^0 est entachée d'une erreur infinitésimale. Lorsque l'exposant de Lyapunov est positif, l'erreur du début ira en augmentant. Dans le cas contraire, l'erreur ira en diminuant, et le système n'aura pas un comportement chaotique. Bref, plus l'exposant est grand, plus les répercussions d'une petite modification de la condition initiale se feront sentir rapidement.

Exemple II.7 : L'exposant de Lyapunov de la suite logistique devient positif pour $\mu > 3,54$, mais on peut montrer qu'il reste toujours plus petit que 1, quelle que soit la condition initiale du système. Quant à la fonction tente et au doublement de l'angle, ils ont un exposant de Lyapunov égal à $\ln(2)$ indépendamment du choix de la condition initiale (le calcul est immédiat).

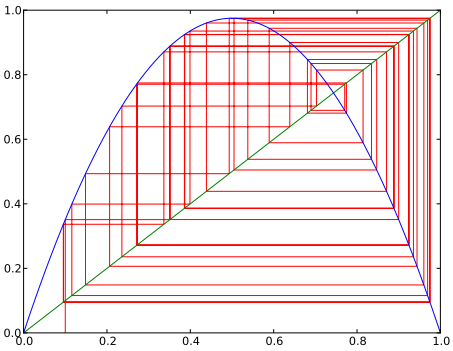
Nous avons vu dans les deux précédents chapitres un certain nombre de notions de chaos au sens topologique du terme. Établir ce genre de propriétés en revenant systématiquement à leurs définitions n'est pas toujours des plus aisés, et il s'avère souvent plus habile de se ramener à des systèmes connus. Ainsi, le chapitre suivant introduit un outil permettant d'établir le chaos d'un système étudié, simplement en le comparant à d'autres systèmes déjà prouvés chaotiques.



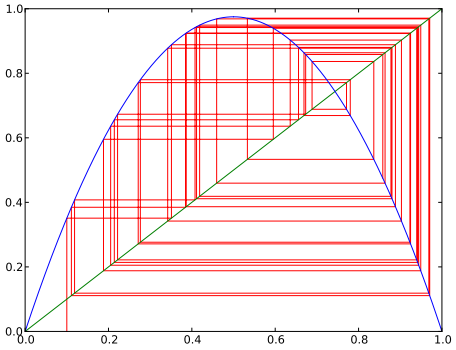
(a) Itérations pour $\mu = 3,56$ et $x^0 = 0,1$.



(b) Itérations pour $\mu = 3,89$ et $x^0 = 0,1$.

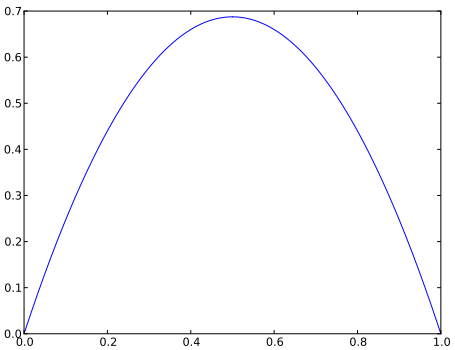


(c) Itérations pour $\mu = 3,90$ et $x^0 = 0,1$.

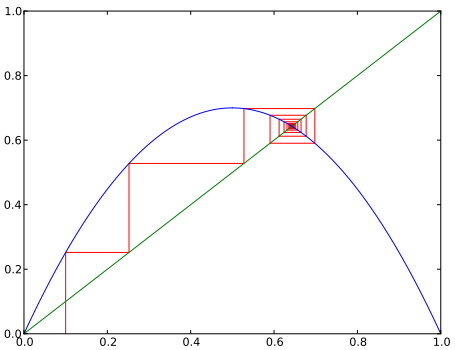


(d) Itérations pour $\mu = 3,90$ et $x^0 = 0,10001$.

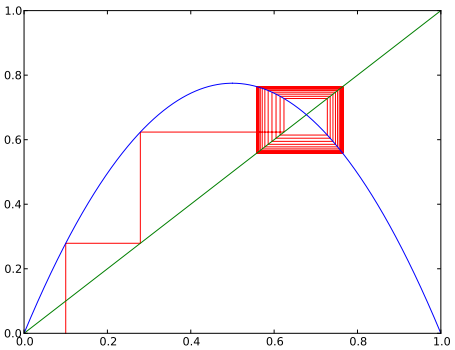
FIGURE 8.1 – Itérations de la fonction logistique.



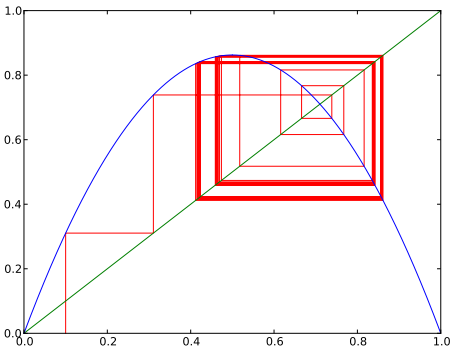
(a) La fonction logistique ($\mu = 2,75$).



(b) Itérations pour $\mu = 2,8$ et $x^0 = 0,1$.



(c) Itérations pour $\mu = 3,1$ et $x^0 = 0,1$.



(d) Itérations pour $\mu = 3,45$ et $x^0 = 0,1$.

FIGURE 8.2 – La fonction logistique et ses doublements de périodes.

CHAPITRE 9

Les conjugaisons topologiques et métriques

Les idées que tu défends sont-elles vraiment tes idées ?

La morale
WOLINSKI

Parfois, au lieu d'essayer de prouver directement des propriétés sur le système lui-même, on préfère réduire le problème initial à un autre dont les caractéristiques sont connues ou semblent accessibles. Un tel outil de réduction se nomme, dans la théorie mathématique du chaos, la (semi-)conjugaison.

Nous commencerons par introduire les notions de conjugaison topologique et métrique, avant de préciser quelles propriétés sont préservées par de telles conjugaisons.

I. LA SEMI-CONJUGAISON TOPOLOGIQUE

1. Définition

DÉFINITION II.54 (SEMI-CONJUGAISON TOPOLOGIQUE) : Le système dynamique discret $(\mathcal{X}, f)$ est dit *topologiquement semi-conjugué* au système $(\mathcal{Y}, g)$ s'il existe une fonction continue surjective $\varphi : \mathcal{X} \longrightarrow \mathcal{Y}$ telle que :

$$\varphi \circ f = g \circ \varphi$$

c'est-à-dire qui rende le diagramme suivant commutatif [For98] :

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{f} & \mathcal{X} \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\varphi} \\ \mathcal{Y} & \xrightarrow[g]{} & \mathcal{Y} \end{array}$$

Dans ce cas, le système $(\mathcal{Y}, g)$ est appelé *facteur* du système $(\mathcal{X}, f)$. ◊

2. Utilité de la semi-conjugaison

Plusieurs comportements dynamiques sont hérités par les systèmes facteurs [For98]. Ils sont résumés ci-dessous :

PROPOSITION II.18 : *Soit $(\mathcal{Y}, g)$ un facteur du système $(\mathcal{X}, f)$. Alors :*

1. $p \in Per_k(f) \Longrightarrow \varphi(p) \in Per_j(g)$, *où* $j \leqslant k$,
2. $(\mathcal{X}, f)$ *régulier* $\Longrightarrow (\mathcal{Y}, g)$ *régulier,*
3. $(\mathcal{X}, f)$ *transitif* $\Longrightarrow (\mathcal{Y}, g)$ *transitif,*
4. *Donc* $(\mathcal{X}, f)$ *chaotique selon Devaney* $\Longrightarrow (\mathcal{Y}, g)$ *chaotique selon Devaney,*

3. Exemple d'utilisation

La semi-conjugaison topologique permet par exemple de montrer que la suite logistique est chaotique selon Devaney, en se ramenant à un autre système que l'on connaît mieux.

On peut alors retrouver les résultats de la section 7.5 en ne partant que du chaos du doublement de l'angle :

1. $g_4(x) = 4x(1-x)$ est reliée au doublement de l'angle f, de la manière suivante. Posons $h(\theta) = (1 - cos\theta)/2$. On a alors :

$$(h \circ f)(q) = \frac{1 - cos(2q)}{2} = 1 - cos^2 q = 4\left(\frac{1}{2} - \frac{cos(q)}{2}\right)\left(\frac{1}{2} + \frac{cos(q)}{2}\right) = (g_4 \circ h)(q)$$

   On sait que f est chaotique, et l'on en déduit que cette propriété se transpose à g_4, car h est une semi-conjugaison.

2. g_4 est topologiquement semi-conjuguée à la fonction tente, donc la fonction tente est chaotique selon Devaney.

II. CONJUGAISON TOPOLOGIQUE

La conjugaison (topologique et métrique) est une version plus forte que la semi-conjugaison. On la définit, avant de voir ce que cela implique.

1. Définitions

DÉFINITION II.55 (CONJUGAISON TOPOLOGIQUE) : Deux systèmes dynamiques discrets sont *conjugués topologiquement* si la fonction φ de la semi-conjugaison est un homéomorphisme (bijection bicontinue). ◊

Il existe une autre forme de conjugaison, dite métrique :

DÉFINITION II.56 (CONJUGAISON MÉTRIQUE) : Deux systèmes dynamiques discrets sont *conjugués métriquement* si la fonction φ de la semi-conjugaison est une isométrie surjective. ◊

2. Propriétés conservées par conjugaison

a. Rappel : stabilité des points fixes

Quand f est dérivable, on peut ranger ses points fixes suivant plusieurs catégories, selon leur stabilité :

Définition II.57 (Stabilité et instabilité) : Un point fixe p de f est dit *stable*, *quasi-stable* ou *instable* selon que $|f'(p)|$ est inférieur, égal ou supérieur à 1. ◊

Selon leur stabilité, les points fixes ont tendance soit à attirer les itérées, soit à les repousser, ce qui conduit à la définition-proposition suivante :

Définition II.58 (Attractivité et répulsivité) : Si p est stable, alors il est *attractif* : il existe un voisinage V de p tel que $\forall x \in V, f^{(n)}(x) \longrightarrow p$ lorsque $n \longrightarrow +\infty$.

Si p est instable, alors il est *répulsif* : il existe un voisinage V de p dont les points distincts de p « sortent » de ce voisinage lorsqu'on les itère par f. ◊

b. Les propriétés conservées par conjugaison topologique

Proposition II.19 : *Soit la conjugaison topologique suivante :*

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{f} & \mathcal{X} \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\varphi} \\ \mathcal{Y} & \xrightarrow[g]{} & \mathcal{Y} \end{array}$$

On a alors :

1. *Pour tout n, $\varphi \circ g^{(n)} = f^{(n)} \circ \varphi$,*
2. *p est un point périodique de g de période n si et seulement si $\varphi(p)$ est un point périodique de f de période n.*
3. *Si f, g et φ sont C^1 (*i.e. *continûment dérivables), φ' ne s'annulant pas, alors pour p point périodique de g de période n, on a $(g^{(n)})'(p) = (f^{(n)})'(\varphi(p))$.*

   Dit autrement, p et $\varphi(p)$ sont donc de même nature.

La conjugaison traduit donc le fait que f et g ont la même dynamique. On a affaire à une relation d'équivalence :

Proposition II.20 : *La conjugaison, qu'elle soit topologique ou métrique, est une relation d'équivalence sur l'ensemble des systèmes dynamiques discrets.*

Exemple II.8 : Il est équivalent d'étudier l'une ou l'autre des formes suivantes, qui appartiennent toutes à la famille quadratique : $x \mapsto \mu x(1-x)$, $x \mapsto 1-\lambda x^2$, $x \mapsto x^2 + c$, ou encore $x \mapsto 2\mu x + 2x^2$... En effet, on peut toujours trouver une fonction affine reliant deux de ces formes.

Signalons pour finir que :

Proposition II.21 : *La stabilité, la sensibilité et l'expansivité sont des propriétés métriques : elles sont préservées par conjugaison métrique, mais pas par conjugaison topologique.*

CHAPITRE 10

L'entropie topologique

> Tu ne peux posséder jeunesse et savoir en même temps.
> Car la jeunesse est trop passionnée de la vie pour savoir,
> et le savoir est trop avide de lui-même pour vivre.
>
> *Le sable et l'écume*
> KALIL GIBRAN

Dans ce chapitre, on rappelle les définitions de l'entropie topologique et quelques-unes de ses propriétés. L'idée de la définition de l'entropie d'un système dynamique est la suivante : on considère que la position initiale du système n'est pas connue avec une précision infinie, mais que le comportement que l'on va observer en itérant le système va nous renseigner de mieux en mieux sur le point dont on est parti. Cela n'est pas possible si l'entropie topologique est élevée.

Nous présenterons les résultats sans démonstration, l'étude de l'entropie topologique n'étant pas le but premier de ce travail de recherche. Le lecteur intéressé par cette notion pourra consulter la thèse de Sylvie Ruette [Rue01], le rapport de DEA de Boris Saulnier [Sau02], ou le rapport de TIPE d'Alain Frisch [Fri98a]. Signalons pour finir que l'entropie topologique n'est pas la même chose que l'entropie de la théorie de l'information (cette dernière sera définie au chapitre 19).

I. DÉFINITION ORIGINALE DE L'ENTROPIE TOPOLOGIQUE

1. Introduction

Adler, Konheim et McAndrew ont défini [AKM65] en 1965 l'entropie topologique d'un système dynamique sur un espace métrique compact ; elle mesure la quantité moyenne d'information que l'on gagne à chaque itération. Intuitivement, si l'on récupère « trop » d'informations à chaque itération, beaucoup trop pour nos capacités d'étude, c'est-à-dire si l'entropie du système est trop élevée, alors on ne pourra pas prévoir le comportement futur dudit système.

Dans ce chapitre, on considérera un espace métrique compact $(\mathcal{X}, d)$, et une fonction continue $f : \mathcal{X} \to \mathcal{X}$. Nous commençons par introduire la définition originelle de l'entropie topologique, avant d'en donner par la suite quelques variantes permettant de mieux la comprendre.

2. L'entropie d'un recouvrement ouvert

a. Définition et notations

On rappelle que la définition d'un recouvrement ouvert $\mathcal{U}$ d'un ensemble $\mathcal{X}$ a été donnée en II.11 : c'est une famille d'ouverts de $\mathcal{X}$ dont la réunion est égale à $\mathcal{X}$. Nous noterons $N(\mathcal{U})$ le nombre minimal d'ouverts du recouvrement $\mathcal{U}$ nécessaire pour recouvrir tout $\mathcal{X}$:

NOTATION II.3. Si $\mathcal{U} = \{U_1; \ldots; U_p\}$ est un recouvrement ouvert, on note

$$N(\mathcal{U}) = \min\left\{n \in \mathbb{N} \,\middle/\, \exists i_1, \ldots; i_n \in [\![1; p]\!], \mathcal{X} = U_{i_1} \cup \ldots \cup U_{i_n}\right\}$$

REMARQUE. Cet entier existe bien, vu que l'espace est compact.

Il nous reste à définir ce qu'est un recouvrement ouvert joint, avant de pouvoir introduire l'entropie topologique d'un recouvrement :

DÉFINITION II.59 (RECOUVREMENT OUVERT JOINT) : Soient $\mathcal{U} = \{U_1; \ldots; U_p\}$ et $\mathcal{V} = \{V_1; \ldots; V_q\}$ deux recouvrements ouverts de $\mathcal{X}$. On définit un nouveau recouvrement ouvert, appelé *recouvrement ouvert joint*, en intersectant ces ouverts deux à deux :

$$\mathcal{U} \vee \mathcal{V} = \left\{U_i \cap V_j \,\middle/\, 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q\right\}$$

NOTATION II.4. On note $N_n(\mathcal{U}) = N\left(\mathcal{U} \vee f^{-1}(\mathcal{U}) \vee \ldots \vee f^{-(n-1)}(\mathcal{U})\right)$

b. L'entropie d'un recouvrement

On peut dorénavant définir l'entropie d'un recouvrement :

DÉFINITION II.60 (ENTROPIE D'UN RECOUVREMENT) : L'*entropie du recouvrement ouvert* $\mathcal{U}$ est donnée par la formule :

$$h_{top}(\mathcal{U}, f) = \lim_{n \to +\infty} \frac{\log N_n(\mathcal{U})}{n} = \inf_{n \geqslant 1} \frac{\log N_n(\mathcal{U})}{n}$$

3. L'entropie topologique

DÉFINITION II.61 (ENTROPIE TOPOLOGIQUE) : L'*entropie topologique* du système $(\mathcal{X}, f)$ est définie comme étant la borne supérieure des entropies des recouvrements ouverts de $\mathcal{X}$:

$$h_{top}(\mathcal{X}, f) = \sup\left\{h_{top}(\mathcal{U}, f) \,\middle/\, \mathcal{U} \text{ recouvrement ouvert fini de } \mathcal{X}\right\}$$

Donnons une explication imagée de cette définition de l'entropie topologique. Supposons que les positions que peut prendre un point x de notre système ne nous sont connues qu'imparfaitement, par exemple que l'on observe notre système au-travers d'un instrument de mesure ne pouvant retourner que $\mathcal{U}$ valeurs (expression imagée de notre recouvrement $\mathcal{U}$).

Le nombre $N_n(\mathcal{U})$ représente le nombre minimal de mots de longueur n nécessaires pour encoder les points de $\mathcal{X}$ d'après le comportement de leurs $n-1$ premières itérées. Dit autrement, ce nombre $N_n(\mathcal{U})$ mesure le nombre de « scénarios » différents que l'on est capable d'observer pour ces $n-1$ itérées.

Vu sous cet angle, l'entropie topologique est la quantité moyenne d'information nécessaire par itération pour être capable de décrire l'évolution à long terme du système. Elle peut être infinie.

Une entropie positive impose une démultiplication exponentielle du nombre d'orbites visibles à chaque itération. Ce qui conduit à une nouvelle définition de chaos :

DÉFINITION II.62 (CHAOS AU SENS DE L'ENTROPIE TOPOLOGIQUE) : Un système dynamique est dit *chaotique au sens de l'entropie topologique* si son entropie est strictement positive. Ce système sera d'autant plus chaotique que l'entropie topologique sera grande. ◊

4. Quelques exemples

Nous donnons ici quelques exemples de mesures d'entropies topologiques classiques.

Exemple II.9 : Les applications 1-Lipschitziennes ont une entropie nulle.

Exemple II.10 : L'application doublement de l'angle, donnée dans la définition II.39, possède une entropie topologique égale à $\ln 2$.

Exemple II.11 : L'application logistique $f(x) = \mu x(1-x)$ de la définition II.48, dans le cas $\mu > 2+\sqrt{5}$, a elle aussi une entropie $\ln 2$ sur l'ensemble $\left\{x \in [0;1] \,\middle/\, \forall n \in \mathbb{N}, f^{(n)}(x) \in [0,1]\right\}$

Il n'existe pas de méthode générale pour calculer l'entropie d'une application. On essaye souvent de se ramener à des fonctions dont l'entropie est plus facile à mesurer, mais dont on peut prouver qu'elles ont même entropie que la fonction qui nous intéresse. Il existe aussi certains résultats permettant d'obtenir une borne pour l'entropie, tel le suivant :

PROPOSITION II.22 : *Dans le cas des fonctions expansives, l'entropie est supérieure où égale à la limite supérieure de* $\frac{p_n}{n}$*, où* p_n *est le nombre de points de période* n*.*

L'entropie topologique n'étant pas si aisée à appréhender, nous allons en donner une autre définition dans ce qui suit. Ce sera cette définition, non historique, que l'on utilisera par la suite.

II. DÉFINITION À PARTIR D'ENSEMBLES SÉPARÉS

On introduit dans cette section une définition de l'entropie topologique basée sur la séparation de points, qu'il nous faut introduire avant toute chose.

1. Points séparés

On dira de deux points qu'ils sont $\varepsilon-$séparés s'ils sont à distance (au moins) ε l'un de l'autre, et qu'ils sont $\varepsilon-$séparés en temps n s'il existe une itération inférieure à n qui les sépare :

DÉFINITION II.63 (POINTS $\varepsilon-$SÉPARÉS) : Soit $\varepsilon > 0$. On dit que deux points $x, y \in \mathcal{X}$ sont $\varepsilon-$*séparés* si $d(x,y) > \varepsilon$. ◊

DÉFINITION II.64 (POINTS $\varepsilon-$SÉPARÉS EN TEMPS n) : Soient $(\varepsilon, n) \in \mathbb{R}^+ \times \mathbb{N}$. Deux points $x, y \in \mathcal{X}$ sont $\varepsilon-$*séparés en temps* n s'il existe un $k \leqslant n$ tel que $d\left(f^{(k)}(x), f^{(k)}(y)\right) > \varepsilon$. ◊

Cela nous amène à introduire une nouvelle distance :

NOTATION II.5. On note $d_n(x,y) = \max\limits_{0 \leqslant k \leqslant n} d\left(f^{(k)}(x), f^{(k)}(y)\right)$ la mesure du plus grand éloignement entre les orbites de x et y au cours des n premières itérations de f.

2. Ensembles séparés

Les ensembles $(n, \varepsilon)-$séparé sont des ensembles de points qui seront tous $\varepsilon-$séparés en temps n :

DÉFINITION II.65 (ENSEMBLE $(n, \varepsilon)-$SÉPARÉ) : Une partie E de $\mathcal{X}$ est un *ensemble* $(n, \varepsilon)-$*séparé* si $\forall x, y \in E, x \neq y, \exists k \in [\![0; n-1]\!], d\left(f^{(k)}(x), f^{(k)}(y)\right) > \varepsilon$. ◊

REMARQUE. Le moment k pour lequel deux points x et y d'un ensemble $(n, \varepsilon)-$séparé seront effectivement $\varepsilon-$séparés dépend de x et y : $k = k(x, y)$. De plus, il se peut très bien qu'au temps $k(x, y) + 1$ ces deux points ne soient plus $\varepsilon-$séparés. Cependant on est sûr que pour tout couple de points il y aura eu une séparation avant le temps n.

NOTATION II.6. On note $s_n(\varepsilon, Y)$ le cardinal maximal d'un ensemble $(n, \varepsilon)-$séparé inclus dans Y. $s_n(\varepsilon, Y)$ est donc le nombre de points du plus grand ensemble de points $\varepsilon-$séparés en temps n de Y. Ou encore, c'est l'ensemble des points qui se sont éloignés d'ε à un moment donné au cours des n premières itérations de f. $s_n(\varepsilon, Y)$ est donc le nombre d'orbites de longueur n que l'on peut distinguer, quand on suppose ne pas être capable de distinguer deux points dès qu'ils ne sont pas $\varepsilon-$séparés.

On peut redéfinir l'entropie topologique à partir de cette notion de séparation.

3. Entropie topologique

Dans [Bow71a] et [Bow71b], Bowen montre en 1971 que l'entropie topologique peut être calculée à l'aide d'ensembles $(n, \varepsilon)-$séparés :

PROPOSITION II.23 (FORMULE DE BOWEN) : *Soit $f : \mathcal{X} \longrightarrow \mathcal{X}$ une fonction continue sur un espace métrique compact $(\mathcal{X}, d)$. Alors :*

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

REMARQUE. La fonction $\varepsilon \mapsto \limsup_{n \to +\infty} \left(\frac{1}{n} \log s_n(\varepsilon, \mathcal{X}) \right)$ croît quand ε décroît : cette limite est donc bien définie.

L'entropie topologique mesure donc la croissance exponentielle moyenne du nombre d'orbites distinguables : si l'on compte le nombre maximum de points dont les orbites ne restent pas collées pendant un temps n, et que l'on étudie la vitesse de croissance de cette quantité par rapport à n, on dispose d'une mesure du caractère dispersif du système.

Le chaos au sens de l'entropie topologique consiste donc, sous un certain angle, à considérer qu'un système est complexe lorqu'il disperse des états proches.

Troisième partie

Apport théorique

CHAPITRE 11

Modélisation des itérations chaotiques

Les bêtises des gens intelligents sont fascinantes.

Peplum
AMÉLIE NOTHOMB

Dans ce chapitre, les itérations chaotiques (définition I.31) sont réécrites sous la forme d'un système dynamique discret sur un espace topologique $\mathcal{X}$:

$$\begin{cases} X^0 \in \mathcal{X} \\ X^{n+1} = F(X^n) \end{cases}$$

où F est une fonction continue sur $\mathcal{X}$. Ce faisant, on sera en mesure de les étudier au sein de la théorie du chaos. La définition de $\mathcal{X}$, sa topologie, et F sont donnés, la taille de $\mathcal{X}$ est évaluée et la continuité de F est établie.

Ces travaux ont été en partie publiés dans [GB10].

I. MODÉLISATION DES ITÉRATIONS CHAOTIQUES

Cette section contient la modélisation des IC sous la forme d'une suite récurrente $X^0 \in \mathcal{X}$, $X^{n+1} = F(X^n)$ [GB10]. La définition d'une topologie τ sur $\mathcal{X}$, et les propriétés de F sur l'espace topologique $(\mathcal{X}, \tau)$, feront l'objet des sections suivantes.

Pour parvenir à modéliser les IC sans trop alourdir les notations, nous introduisons deux nouvelles fonctions sur l'ensemble $\mathcal{S}$ des stratégies chaotiques (voir notation I.6), à savoir les fonctions *décalage* et *initiale*.

1. Définitions et notations

a. Fonctions décalage *et* initiale

DÉFINITION III.1 (FONCTION DÉCALAGE) : On appelle *décalage* la fonction :

$$\begin{array}{rccc} \sigma : & \mathcal{S} & \longrightarrow & \mathcal{S} \\ & (S^n)_{n\in\mathbb{N}} & \longmapsto & (S^{n+1})_{n\in\mathbb{N}} \end{array}$$

Cette fonction se contente donc de décaler la stratégie vers la gauche. Dans le cas d'une stratégie indexée par $\mathbb{N}$, cela revient donc à perdre le premier terme.

Exemple III.1 : Soit Π la suite des chiffres du nombre π : $\Pi^0 = 3, \Pi^1 = 1, \Pi^2 = 4, \Pi^3 = 1, \Pi^4 = 5$,... Alors $u = \sigma(\Pi)$ correspond à la suite décalée : $u^0 = 1, u^1 = 4, u^2 = 1, u^3 = 5$,...

Cette fonction décalage σ, encore appelée *fonction shift*, sera fondamentale lors de l'étude du comportement des itérations chaotiques sous l'angle topologique. En effet, cette fonction est un exemple bien connu de chaos au sens de Devaney [Dev03].

Définissons à présent la fonction *initiale*, qui permet d'obtenir le premier terme d'une suite.

DÉFINITION III.2 (FONCTION INITIALE) : On appelle *initiale* la fonction qui à une stratégie associe son premier terme :

$$\begin{array}{rccc} i : & \mathcal{S} & \longrightarrow & [\![1;\mathsf{N}]\!] \\ & (S^n)_{n\in\mathbb{N}} & \longmapsto & S^0 \end{array}$$

b. Notation : la fonction F_f

Il nous reste à introduire une dernière notation, avant d'en venir à la modélisation à proprement parler.

NOTATION III.1. On commence par rappeler que δ désigne la *distance triviale* :

$$\delta(x,y) = \begin{cases} 0 & \text{si } x = y, \\ 1 & \text{sinon.} \end{cases}$$

C'est une distance, au sens topologique du terme (voir définition II.7).

NOTATION III.2. On note F_f la fonction suivante :

$$\begin{array}{rccc} F_f : & [\![1;\mathsf{N}]\!] \times \mathbb{B}^{\mathsf{N}} & \longrightarrow & \mathbb{B}^{\mathsf{N}} \\ & (k,E) & \longmapsto & \left(E_j.\delta(k,j) + f(E)_k.\overline{\delta(k,j)}\right)_{j\in[\![1;\mathsf{N}]\!]} \end{array}$$

où les opérateurs . et + représentent respectivement le *et* et le *ou* booléen.

2. La modélisation d'une itération chaotique par un système dynamique discret

On est maintenant en mesure de modéliser les itérations chaotiques.

THÉORÈME III.1 (MODÉLISATION DES IC) : *Soit*

$$\mathcal{X} = [\![1;\mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}},$$

appelé espace des phases, et G_f la fonction successeur :

$$G_f(S,E) = \left(\sigma(S), F_f(i(S),E)\right).$$

On modélise les itérations chaotiques $\left(f,(S,x^0)\right)$ par le système dynamique discret :

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

La différence avec la définition I.31 des itérations chaotiques est que, dans cette écriture, on applique toujours la même fonction G_f à chaque itération.

Remarque. On définira de la même manière un espace des phases généralisé $\mathcal{X}^G$ à partir de l'ensemble des stratégies chaotiques généralisées. C'est-à-dire : $\mathcal{X}^G = \mathcal{S}^G \times \mathbb{B}^{\mathsf{N}}$.

Cette manière de voir les IC sous la forme d'un système dynamique discret peut se comprendre de la manière suivante : à chaque itération,

1. on prend le premier terme S^0 de la stratégie S, grâce à la fonction initiale i,
2. on met à jour la cellule S^0 de l'état E de notre système,
3. enfin, on décale la stratégie d'une case, à l'aide de σ.

Nous avons réussi à écrire les itérations chaotiques sous la forme d'une itération de fonction sur un ensemble $\mathcal{X}$. Afin d'être pleinement dans le cadre de la théorie du chaos, ce qui permettrait d'étudier l'imprévisibilité des IC, il nous faut nous assurer que la fonction d'itération G_f ainsi définie, est continue sur $\mathcal{X}$. Mais pour parler de continuité, il nous faut une topologie ou une métrique adéquate sur $\mathcal{X}$. Ce sera l'enjeu de la section 11.2.

3. *A parte* concernant le lien parallèle-chaotique

Nous avions rappelé au chapitre 4 de la partie I que les itérations parallèles (et donc séries et séries-parallèles) pouvaient être vues comme un cas particulier des itérations chaotiques, pourvu que nous laissions ces dernières arriver dans l'ensemble des parties de $[\![1;\mathsf{N}]\!]$. Nous établissons avec ce résultat le fait que les itérations chaotiques sont des itérations parallèles, et que ces deux objets désignent en fait la même chose. Nous ne savons pas si un tel résultat élémentaire existe déjà dans la littérature : nous l'avons recherché sans succès.

Nous avions montré au chapitre 3 de la partie I que les itérations parallèles étaient inadaptées à nos objectifs visés, car ces dernières étaient par trop prévisibles : leur comportement asymptotique se résume à soit de la convergence, soit des cycles. Cette modélisation et cette constatation ne viennent pas contredire les conclusions de ce chapitre, qui avait pour cadre les ensembles d'états finis : en modélisant les itérations chaotiques par des itérations parallèles, nous obtenons un ensemble d'états $\mathcal{X}$ infini indénombrable.

II. Définition d'une métrique sur $\mathcal{X}$

Définition III.3 : On définit une distance d entre deux points $(S;E)$ et $(\check{S};\check{E})$ de notre espace des phases $\mathcal{X}$ par la formule

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

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

Remarque. d est bien une distance, comme somme de d_e et d_s, qui le sont de manière évidente.

La distance d a été construite comme somme de d_e et de d_s, notamment pour la raison suivante :

PROPOSITION III.1 : *Pour tous points* (S, E) *et* $(\check{S}; \check{E})$ *de* $\mathcal{X}$ *:*

- $d_e(E, \check{E})$ *est égale à la partie entière de* $d((S, E); (\check{S}; \check{E}))$*,*
- $d_s(S, \check{S})$ *en est sa partie fractionnaire.*

PREUVE : *d_e prend clairement uniquement des valeurs entières.*
D'autres part, $0 \leqslant d_s(S, \check{S})$*, et*

$$d_s(S, \check{S}) = \frac{9}{N} \sum_{k=1}^{\infty} \frac{|S^k - \check{S}^k|}{10^k} \leqslant \frac{9}{N} \sum_{k=1}^{\infty} \frac{N}{10^k} = 9 \sum_{k=1}^{\infty} \frac{1}{10^k} = 9 \times \frac{1}{10} \times \frac{1}{1 - \frac{1}{10}} = 1.$$

Ainsi, la partie entière de d mesure le nombre de cellules différentes entre deux points X et Y de l'espace $\mathcal{X}$, et la partie décimale mesure la divergence entre les stratégies. Plus exactement, si $\lfloor d(X, Y) \rfloor = n$, alors il y a exactement n cellules qui ne présentent pas le même aspect entre X et Y. Quant à la partie décimale de d, elle sera d'autant plus faible que les stratégies associées à X et Y diffèrent « tardivement » :

- Si la $k-$ième décimale est non nulle, alors les deux stratégies considérées ont un $k-$ième terme différent.
- Cette partie décimale est inférieure à 10^{-k} si et seulement si les k premiers termes des stratégies sont identiques.

REMARQUE. d_e peut être construite de telle sorte que la valeur de la partie entière de d donne quelles cellules diffèrent. Par exemple, $d_e(E, \check{E}) = \sum_{k=1}^{N} 10^{\delta(E_k, \check{E}_k) \times (k-1)}$. On y gagne de la précision (savoir quelles sont les cellules différentes), mais on y perd le fait que deux états ayant n cellules différentes sont à une distance comprise entre n et $n + 1$. Or, cela semble important pour les applications que l'on vise (par exemple, pour mesurer le nombre de bits différents entre une image d'origine et une image tatouée).

REMARQUE. On pourrait construire une autre distance, qui varie d'autant plus fortement que les deux stratégies divergent rapidement : on inverserait les rôles des parties entières et décimales de d, et des cellules diversement activées ne feraient varier que faiblement ladite distance. Cette distance nous semble cependant moins pertinente que d, dans la mesure où les valeurs des cellules sont ce qui caractérise vraiment un système donné.

Nous avons muni notre espace des phases $\mathcal{X}$ d'une distance d pertinente, car telle que deux points sont proches (pour d) si et seulement si leurs itérations chaotiques évoluent de la même manière.

En dernier lieu, pour être exactement dans le cadre de la théorie du chaos et étudier la prévisibilité des itérations chaotiques, il faut que G_f soit continue sur l'espace métrique $(\mathcal{X}, d)$.

III. CONTINUITÉ DE G_f SUR $(\mathcal{X}, d)$

Pour démontrer ce résultat, on utilise la caractérisation séquentielle de la continuité, qui est valable sur les espaces métriques. On renvoie le lecteur aux définitions II.8 et II.16, et à la proposition II.2.

THÉORÈME III.2 : *La fonction $G_f : (\mathcal{X}, d) \to (\mathcal{X}, d)$ est continue.*

PREUVE : *On prouve la continuité de G_f avec la caractérisation séquentielle.*

*Soit donc $(S^n, E^n)_{n\in\mathbb{N}}$ une suite de points de l'espace des phases $\mathcal{X}$ qui converge vers (S, E). Montrons que $\left(G_f(S^n, E^n)\right)_{n\in\mathbb{N}}$ converge vers $\left(G_f(S, E)\right)$. Remarquons pour commencer que pour tout n, S^n est une stratégie : on a affaire à une suite de stratégies (*i.e. *une suite de suites).*

La distance $d((S^n, E^n); (S, E))$ tend vers 0, donc chacune des distances $d_e(E^n, E)$ et $d_s(S^n, S)$ tend vers 0. Or, $d_e(E^n, E)$ ne prend que des valeurs entières, donc cette distance est nulle à partir d'un certain rang n_0.

Ainsi, à partir de cet instant n_0, plus aucune cellule ne change d'état : $\exists n_0 \in \mathbb{N}, n \geqslant n_0 \Longrightarrow E^n = E$. De plus, $d_s(S^n, S) \longrightarrow 0$ donc $d_s(S^n, S) < 10^{-1}$ à partir d'un certain rang n_1. Cela signifie qu'à partir du rang n_1, les S^n ont toutes le même premier terme, qui est celui de S : $\forall n \geqslant n_1, (S^n)^0 = S^0$.

Bref, à partir du rang $max(n_0, n_1)$, les états E^n et E sont les mêmes, et les stratégies S^n et S ont le même premier élément. En conséquence, à partir de ce rang, les états de $G_f(S^n, E^n)$ et de $G_f(S, E)$ sont les mêmes, donc la distance d entre ces points est inférieure à 1.

Montrons maintenant que la distance entre $\left(G_f(S^n, E^n)\right)$ et $\left(G_f(S, E)\right)$ tend bien vers 0 quand n tend vers $+\infty$. Soit $\varepsilon > 0$.

- *Si $\varepsilon \geqslant 1$, alors on vient de voir que la distance entre $\left(G_f(S^n, E^n)\right)$ et $\left(G_f(S, E)\right)$ est strictement plus petite que 1 à partir du rang $max(n_0, n_1)$ (puisque ces points ont même état).*

- *Si $\varepsilon < 1$, alors $\exists k \in \mathbb{N}, 10^{-k} \geqslant \varepsilon \geqslant 10^{-(k+1)}$. Comme $d_s(S^n, S)$ tend vers 0, il existe un rang n_2 à partir duquel $d_s(S^n, S) < 10^{-(k+2)}, \forall n \geqslant n_2$: à partir de ce rang, les $k + 2$ premiers termes de S^n sont ceux de S.*

  En conséquence de quoi, les $k + 1$ premiers termes des stratégies de $G_f(S^n, E^n)$ et de $G_f(S, E)$ sont les mêmes (puisque G_f opère un décalage sur les stratégies), et vue la définition de d_s, la partie décimale de la distance entre les points (S^n, E^n) et (S, E) est inférieure à $10^{-(k+1)} \leqslant \varepsilon$.

Pour conclure, $\forall \varepsilon > 0, \exists N_0 = max(n_0, n_1, n_2) \in \mathbb{N}, \forall n \geqslant N_0, d\left(G_f(S^n, E^n); G_f(S, E)\right) \leqslant \varepsilon$. La distance entre $\left(G_f(S^n, E^n)\right)$ et $\left(G_f(S, E)\right)$ tend bien vers 0, ce qui permet d'en déduire la continuité de G_f par la caractérisation séquentielle.

Nous sommes dorénavant en mesure d'étudier les itérations chaotiques sous l'angle de la théorie du chaos. Les résultats de cette étude sont donnés dans les chapitres suivants. Avant de clore ce chapitre, on étudie l'espace métrique $(\mathcal{X}, d)$, et l'on précise quelques pistes pour une généralisation du cadre d'étude des itérations chaotiques.

IV. Étude de l'espace métrique $(\mathcal{X}, d)$

Dans cette section, on cherche à connaître un peu mieux notre espace des phases. On étudie quelques propriétés topologiques de $\mathcal{X}$, telles que la compacité et la complétude. Ces propriétés seront requises pour conduire une étude approfondie du chaos des IC. On en profite pour évaluer la taille de $\mathcal{X}$. Ces travaux ont été en partie publiés dans [GB10].

1. Puissance de l'espace des phases $\mathcal{X}$

Avant d'aller plus loin, on cherche à savoir si $\mathcal{X}$ est un espace de grande taille, c'est-à-dire s'il a beaucoup d'éléments. Ce résultat aura son importance pour les applications que l'on vise. Ainsi, $\mathcal{X}$ sera notre espace des clés, lors du chiffrement du filigrane (l'information que l'on souhaite cacher dans un média) : cet espace ne saurait être petit.

Proposition III.2 : *L'espace des phases $\mathcal{X}$ est de puissance infinie indénombrable, si l'on considère le cadre théorique $\mathcal{X} = \mathcal{S}_{\mathsf{N}} \times \mathbb{B}^{\mathsf{N}}$, et infinie dénombrable, dans le cadre pratique $\mathcal{X} = \tilde{\mathcal{S}}_{\mathsf{N}} \times \mathbb{B}^{\mathsf{N}}$.*

Preuve : *La puissance d'un produit cartésien est égale au produit des puissances. Or, $\mathbb{B}^{\mathsf{N}}$ est de cardinal 2^{N}, qui est fini. La puissance de $\mathcal{X}$ est donc celle de $\mathcal{S}_{\mathsf{N}}$ ou $\tilde{\mathcal{S}}_{\mathsf{N}}$, suivant le cas considéré, car ces derniers sont infinis. Les propositions I.12 et I.13 permettent alors d'en conclure le résultat énoncé.*

2. Compacité

Établissons maintenant la compacité de $(\mathcal{X}, d)$: l'entropie topologique, que l'on calculera plus loin, nécessite une telle propriété pour être pleinement définie, et nous avons vu à la partie II qu'un certain nombre de résultats de la théorie du chaos s'appuient sur cette compacité.

Pour vérifier que $(\mathcal{X}, d)$ est un espace compact, on utilisera la caractérisation séquentielle de la compacité pour les espaces métriques, que l'on a rappelée précédemment (proposition II.1).

Proposition III.3 : *$(\mathcal{X}, d)$ est un espace métrique compact.*

Preuve : *Soit $X = (S^n, E^n)_{n\in\mathbb{N}}$ une suite de $\mathcal{X}$.*

1. *Il existe un état apparaissant une infinité de fois dans cette suite. Soit $\tilde{n}$ le premier indice où cet état apparaît, et $E^{\tilde{n}}$ la valeur de cet état. Notons I l'ensemble des termes (S^n, E^n) de la suite X tels que $E^n = E^{\tilde{n}}$. Les premiers termes $(S^n)^0$ des stratégies S^n des couples de I appartiennent à $[\![1, \mathsf{N}]\!]$, et I est infini. Il existe donc un $\tilde{k} \in [\![1, \mathsf{N}]\!]$ pour lequel il y a une infinité de stratégies de I dont le premier terme est égal à $\tilde{k}$. Soit n_0 le plus petit entier n vérifiant $E^n = E^{\tilde{n}}$ et $(S^n)^0 = \tilde{k}$.*
2. *L'ensemble $I' = \left\{(S^n, E^n) \,\middle|\, E^n = E^{n_0} \text{ et } (S^n)^0 = (S^{n_0})^0\right\}$ est infini, donc un des chiffres de $[\![1, \mathsf{N}]\!]$ apparaît une infinité de fois dans les $(S^n)^1$ de I' : appelons-le $\tilde{l}$. Soit n_1 le plus petit entier n vérifiant $(S^n, E^n) \in I'$ et $(S^n)^1 = \tilde{l}$.*
3. *L'ensemble $I'' = \left\{(S^n, E^n) \,\middle|\, E^n = E^{n_0}, (S^n)^0 = (S^{n_0})^0, (S^n)^1 = (S^{n_1})^1\right\}$ est infini,* etc.

Soit $l = \left(\left((S^{n_k})^k ; E^{n_0}\right)_{k\in\mathbb{N}}\right)$, alors la suite extraite (S^{n_k}, E^{n_k}) converge vers l.

3. Complétude

PROPOSITION III.4 : *$(\mathcal{X}, d)$ est un espace métrique complet.*

PREUVE : *Soit X^n une suite de Cauchy :*

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n, p \in [\![N; +\infty[\![, d(X^n, X^p) < \varepsilon.$$

Alors, pour $\varepsilon = 1$, il existe un rang N_1 tel que

$$\forall n, p \in [\![N_1, +\infty[\![, d(X^n, X^p) < 1$$

En particulier, l'état du système ne change plus après le rang N_1, et vaut $\tilde{E}$.
De même, pour $\varepsilon_k = 10^{-k}$, il existe un rang N_k tel que

$$\forall n, p \in [\![N_k, +\infty[\![, d(X^n, X^p) < 10^{-k}$$

Ainsi, les stratégies de X^n et X^p ont les k mêmes premiers termes, $\forall n, p \in [\![N_k, +\infty[\![$.

Soit donc $\tilde{X} = (\tilde{S}, \tilde{E})$, où $\tilde{E}$ est l'état ci-dessus, et $\tilde{S}^k = \left(S^{N_k}\right)^k$ une stratégie construite par un procédé diagonal. Alors X^n converge vers $(\tilde{S}, \tilde{E})$. En effet, soit $\varepsilon > 0$, alors il existe $N = -\log_{10}(\varepsilon), n \geqslant N \Rightarrow d(X, (\tilde{S}, \tilde{E})) < \varepsilon$

4. $\mathcal{X}$ est parfait

On montre pour finir que :

PROPOSITION III.5 : *$(\mathcal{X}, G_f)$ est un système dynamique discret parfait.*

PREUVE : *Soit $X = (S; E) \in \mathcal{X}$, et $r > 0$ Alors l'ensemble*

$$\left\{(\tilde{S}; E) \in \mathcal{X} \,\middle|\, \forall k \leqslant -\log_{10}(r) + 1, \tilde{S}^k = S^k\right\}$$

est infini et inclus dans $B(x, r)$. Donc X est un point d'accumulation.

V. QUELQUES PISTES POUR UNE GÉNÉRALISATION

1. Le retard

On imagine ici que l'état futur du système (à l'instant $n+1$) dépend de son état actuel (à l'instant n) et de l'état précédent (instant $n-1$). Pour prendre en compte ce retard, on considérera que chaque terme de la stratégie est un *couple* d'entiers pris dans $[\![1; \mathsf{N}]\!]$: le premier élément correspond à la transmission sans retard, et le deuxième à celle avec retard. On prendra alors pour F_f la fonction :

$$\begin{array}{rccc} F_f : & [\![1;\mathsf{N}]\!]^2 \times \left[\mathbb{B}^{\mathsf{N}}\right]^2 & \longrightarrow & \left[\mathbb{B}^{\mathsf{N}}\right]^2 \\ & \left(\begin{pmatrix} k \\ k' \end{pmatrix}; \begin{pmatrix} E \\ E' \end{pmatrix}\right) & \longmapsto & \begin{pmatrix} \left(E_j.\delta(k,j).\delta(k',j) + f(E)_k.\overline{\delta(k,j)} + f(E')_{k'}.\overline{\delta(k',j)}\right)_{j \in [\![1;\mathsf{N}]\!]} \\ E \end{pmatrix} \end{array}$$

où les opérateurs . et + représentent, une fois encore, le *et* et le *ou* booléen. L'espace des phases sera alors $\mathcal{X} = \left([\![1;\mathsf{N}]\!]^2\right)^{\mathbb{N}} \times \left[\mathbb{B}^{\mathsf{N}}\right]^2$, et la fonction successeur à nouveau : $G_f(S,\mathcal{E}) = \left(\sigma(S), F_f(i(S),\mathcal{E})\right)$, dans laquelle, on le rappelle, la stratégie S est une suite de *couples*, et $\mathcal{E}$ le couple (E,E') où E est l'état actuel du système et E' son état à l'instant précédent. On modélise alors les itérations chaotiques avec retard par le système dynamique discret : $X^0 \in \mathcal{X}, X^{k+1} = G_f(X^k)$.

REMARQUE. Si, dans l'expression de F_f, on remplace : $E_j.\delta(k,j).\delta(k',j) + f(E)_k.\overline{\delta(k,j)} + f(E')_{k'}.\overline{\delta(k',j)}$ par $E_j.\delta(k',j) + f(E')_{k'}.\overline{\delta(k',j)}$, alors on ne considère plus que la partie retard (absence de transmission sans retard).

REMARQUE. On peut généraliser encore plus cette notion de retard. Par exemple, si l'on suppose que l'état futur du système dépend de l'état actuel, et des deux précédents états, alors la stratégie deviendra une suite d'éléments de $\left([\![1;\mathsf{N}]\!]^3\right)^{\mathbb{N}}$, et la fonction F_f s'écrira :

$$\begin{array}{rccl} F_f : & [\![1;\mathsf{N}]\!]^3 \times \left[\mathbb{B}^{\mathsf{N}}\right]^3 & \mapsto & \left[\mathbb{B}^{\mathsf{N}}\right]^3 \\ & \left(\begin{pmatrix} k \\ k' \\ k'' \end{pmatrix};\begin{pmatrix} E \\ E' \\ E'' \end{pmatrix}\right) & \to & \begin{pmatrix} \left(E_j.\delta(k,j).\delta(k',j).\delta(k'',j) + f(E)_k.\overline{\delta(k,j)} + f(E')_{k'}.\overline{\delta(k',j)} + f(E'')_{k''}.\overline{\delta(k'',j)}\right)_{j\in[\![1;\mathsf{N}]\!]} \\ E \\ E' \end{pmatrix} \end{array}$$

On peut poursuivre ainsi ce procédé pour prendre en compte des retards d'ordre quelconque.

Dans cette généralisation, la distance d peut être conservée, moyennant le remplacement de la valeur absolue par une norme adéquate.

2. Cas où plusieurs cellules sont modifiées à chaque itération

Nous l'avons déjà dit, la stratégie peut être une suite de *parties* de $[\![1;\mathsf{N}]\!]$, pour prendre en considération le fait que plusieurs cellules peuvent changer à chaque coup d'horloge. La théorie est la même, et seules changent les distances. Ainsi, la distance triviale devient :

$$\delta(x,S^k) = \begin{cases} 0 & \text{si } x \in S^k, \\ 1 & \text{sinon,} \end{cases}$$

et pour la distance d, il suffit de remplacer, comme ci-dessus, les valeurs absolues par des normes.

3. Le cas général

Dans le cas général où plusieurs cellules sont modifiées et pour lequel il existe un retard, il suffit de « composer » les précédentes constructions.

CHAPITRE 12

Un exemple fondamental : la fonction G_{f_0}
Étude du chaos selon la multiplicité des périodes

L'existence précède l'essence.

L'existentialisme est un humanisme
JEAN-PAUL SARTRE

On s'intéresse à nouveau à la fonction négation vectorielle f_0 définie en I.38, qui n'est pas dans une situation de convergence globale des itérées (*c.f.* partie I, chapitre 5.2.2). On souhaite se familiariser avec les itérations chaotiques G_{f_0} et leur système dynamique discret $(\mathcal{X}, G_{f_0})$, où $(\mathcal{X}, d)$ est l'espace métrique introduit dans le chapitre 11 de la présente partie. On en profite pour établir que les itérations chaotiques généralisées sont du chaos au sens de la multiplicité des périodes.

Cet exemple sera fondamental dans l'étude topologique des itérations chaotiques. Il sera aussi utilisé dans les applications de la fin de ce document.

I. ÉTUDE DE LA BIJECTIVITÉ DE G_{f_0}

On rappelle pour commencer que la négation vectorielle est la fonction suivante (définition I.38) :

$$\begin{array}{rccc} f_0 : & \mathbb{B}^{\mathsf{N}} & \longrightarrow & \mathbb{B}^{\mathsf{N}} \\ & (x_1, \ldots, x_{\mathsf{N}}) & \longmapsto & (\overline{x_1}, \ldots, \overline{x_{\mathsf{N}}}) \end{array}$$

Le résultat suivant montre que G_{f_0} n'est pas bijective :

PROPOSITION III.6 : *Supposons que* $\mathsf{N} \geqslant 2$. *Alors* $G_{f_0} : \mathcal{X} \to \mathcal{X}$ *est surjective, mais pas injective.*

PREUVE : *Montrons pour commencer la surjectivité de* G_{f_0}. *Soit* (S, E) *un point de* $\mathcal{X}$. *Alors le point* $(\check{S}, \check{E})$, *défini ainsi :*

- $\check{S}$ *commence par un 1, et se termine par* S,
- *les cellules de* $\check{E}$ *sont toutes dans le même état que celles de* E, *sauf* $\check{E}_0 = \overline{E_0}$,

est tel que $G_{f_0}\left(\check{S},\check{E}\right) = (S,E)$. Tout point de $\mathcal{X}$ possède donc bien au moins un antécédent par G_{f_0}, ce qui prouve que cette dernière est surjective.

Montrons maintenant que G_{f_0} n'est pas injective. Soit pour premier couple le point défini ainsi :

- *la stratégie constante égale à 1,*
- *l'état dont toutes les cellules sont égales à 1.*

et pour second point le couple :

- *la stratégie commençant par 2, puis toujours égale à 1,*
- *l'état dont toutes les cellules sont égales à 1, sauf les deux premières cellules (qui sont égales à 0).*

Alors ces deux points sont différents, mais leur image par G_{f_0} est la même.

REMARQUE. Dans le cas dégénéré $\mathsf{N} = 1$, G_{f_0} est bijective.

REMARQUE. $G_{f_0}(\mathcal{X}) = \mathcal{X}$, du fait de la surjectivité.

Comme G_{f_0} n'est pas injective, on en déduit que :

PROPOSITION III.7 : *$(\mathcal{X}, G_{f_0})$ n'est pas réversible.*

PREUVE : *Immédiat, d'après la définition II.18.*

On pourrait rendre facilement ce système réversible : il suffirait de considérer pour ensemble $\mathcal{S}$ de stratégies des suites indexées par $\mathbb{Z}$. Cette adaptation ne devrait pas changer nos résultats en profondeur, et pourrait s'avérer intéressante pour certaines applications, par exemple en cryptographie : un cryptosystème peut se construire à partir d'une application bijective.

On étudie maintenant les points périodiques de G_{f_0}, tels qu'ils ont été définis dans le chapitre 6 de la partie II, ce qui nous mène sur le chemin de notre première étude de chaos.

II. CHAOS AU SENS DE LA MULTIPLICITÉ DES PÉRIODES

1. Premiers résultats d'existence de points périodiques

Pour qu'un point soit de période n dans le cadre de nos itérations chaotiques, il faut et il suffit qu'après un décalage de n « cases » la stratégie soit la même, et qu'alors toutes les cellules se retrouvent dans le même état. En particulier, chaque cellule (booléenne) doit être modifiée un nombre pair de fois, ce qui nous permet d'affirmer immédiatement les divers résultats suivants.

PROPOSITION III.8 : *G_{f_0} possède ni point fixe, ni points périodiques de période impaire :*

$$\forall k \in \mathbb{N}, Per_{2k+1}(G_{f_0}) = \varnothing$$

PROPOSITION III.9 : *Le nombre de points périodiques pour G_{f_0} de période $2k$ est égal au nombre de suites de $[\![1;\mathsf{N}]\!]^{\mathbb{N}}$, $2k$-périodiques, telles que dans les $2k$ premiers termes, chaque terme apparaît un nombre pair de fois.*

On peut en déduire que :

PROPOSITION III.10 : *G_{f_0} possède au moins un point périodique de période $2k$ exactement, $\forall k \in \mathbb{N}$.*

PREUVE : *En effet, il suffit de prendre pour stratégie la répétition de $2(k-1)$ uns, suivi de deux 2 (ce qui n'est possible que si $|\mathsf{N}| > 1$).*

2. Chaos selon la multiplicité des périodes

a. Premier résultat négatif

L'absence de point périodique de période impaire nous conduit au résultat suivant :

THÉORÈME III.3 : *Les itérations chaotiques G_{f_0} ne sont pas du chaos au sens de la multiplicité des périodes sur $\mathcal{X} = [\![1;\mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$.*

PREUVE : *En effet, G_{f_0} devrait posséder des points périodiques de période k, $\forall k \in \mathbb{N}$, ce qui n'est pas le cas.*

Ce premier résultat est négatif, et c'est regrettable. Cependant, cette forme de chaos est la plus ancienne qui soit, et est avant tout adaptée à la droite réelle. L'existence de points périodiques de toutes périodes a été remplacé, dans les formulations modernes, par la densité des points périodiques (régularité). En effet, on se prémunit ainsi des situations où les points périodiques se concentrent dans un coin de l'espace, laissant se comporter les autres points de l'espace de la manière la plus régulière qui soit.

De plus, nous avons jusqu'à présent privilégié le cas le plus simple d'itérations chaotiques. En effet, nous avons considéré l'ensemble $\mathcal{X} = [\![1;\mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$ tel qu'à chaque itérée, une et une seule cellule change. Nous avons rappelé à la section 11.5 de la partie I, que cette formulation admettait certaines généralisations : on pouvait prendre en compte le retard à la transmission, ou le fait que plusieurs cellules changent à chaque itération. Nous allons voir dans ce qui suit que, dans ce cadre, les itérations chaotiques sont du chaos selon la multiplicité des périodes.

b. Cas de chaos selon la multiplicité des périodes

Nous pouvons énoncer le résultat suivant :

THÉORÈME III.4 : *Sur l'espace des phases généralisé $\mathcal{X}^G = \mathcal{S}^G \times \mathbb{B}^{\mathsf{N}}$, les itérations chaotiques sont du chaos selon la multiplicité des périodes.*

PREUVE : *On rappelle (définition I.26) que $\mathcal{S}^G$ désigne l'ensemble des stratégies chaotiques généralisées, c'est-à-dire des suites de parties de $[\![1;\mathsf{N}]\!]$.*

Tout point de la forme $X = (S, E) \in \mathcal{X}^G$, avec $S = (\emptyset, \emptyset, \emptyset, \ldots)$, est un point fixe pour G_{f_0}. De même, tout point dont la stratégie se termine par une infinité de $\emptyset$ est un point ultimement fixe.

Montrons maintenant que pour tout entier k, il existe un point périodique de période impaire $2 \times k + 1$. Soit $X = (S, E) \in \mathcal{X}$ un point périodique de période $2 \times k$. On définit un point de $\mathcal{X}^G$ de période $2 \times k + 1$ en rajoutant à S un terme $\emptyset$ à chaque fin de période chaque $2k$. On rajoute de cette manière à chaque période une itérée dans laquelle on ne fait rien, on ne change aucune cellule, ce qui nous permet ainsi de transformer le point X en un point $2 \times k + 1$ périodique. De cette manière, G_{f_0} possède des points périodiques de n'importe quelle période sur $\mathcal{X}^G = \mathcal{S}^G \times \mathbb{B}^{\mathsf{N}}$.

III. ÉTUDE DES POINTS PÉRIODIQUES DE G_{f_0} SUR $\mathcal{X}$

1. Points de période 2 et 4

PROPOSITION III.11 : *Il y a $p_2 = \mathsf{N} \times 2^{\mathsf{N}}$ points périodiques de période 2.*

PREUVE : *L'état des cellules n'importe pas, seule compte la stratégie : $(S^n)_{n\in\mathbb{N}}$ doit être une suite périodique de période 2. De plus, si $S^0 \neq S^1$, alors au bout de deux itérations, les états E_{S^0} et E_{S^1} auront tous les deux changé : on n'a pas de période 2. On peut donc en conclure que les points de période 2 de G_{f_0} sont les points où la stratégie chaotique est constante. Il y a N stratégies possibles, et 2^{N} états différents, ce qui donne le résultat.*

PROPOSITION III.12 : *G_{f_0} possède $p_4 = 3\mathsf{N} \times (\mathsf{N} - 1) \times 2^{\mathsf{N}}$ points périodiques de période 4.*

PREUVE : *La stratégie chaotique doit être une suite de période 4 exactement : ses 4 premiers termes la définissent complètement. De plus, toute cellule devant être modifiée un nombre pair de fois, on n'a que trois possibilités (on met ici de côté les points de période 2) :*

- *$S^0 = S^1 \neq S^2 = S^3$: N choix pour S^0 et $\mathsf{N} - 1$ choix pour S^2, soit $\mathsf{N} \times (\mathsf{N} - 1)$ stratégies possibles.*
- *On a ce même nombre de stratégies possibles losque $S^0 = S^2 \neq S^1 = S^3$ et $S^0 = S^3 \neq S^1 = S^2$.*

On trouve donc un total de $3\mathsf{N} \times (\mathsf{N} - 1)$ stratégies possibles, et 2^{N} états différents, ce qui donne $p_4 = 3\mathsf{N} \times (\mathsf{N} - 1) \times 2^{\mathsf{N}}$ points périodiques de période 4.

REMARQUE. Il existe sûrement une formule générale pour p_n. Cependant, nous n'avons pas encore eu le temps de la trouver.

2. Une minoration du nombre de points de période $2n$

On termine ce chapitre par un résultat légèrement plus général, concernant le nombre de points de période $2n$ de G_{f_0}. Outre son intérêt propre et son aide à mieux cerner la périodicité des itérations chaotiques, ce résultat nous sera utile pour mesurer l'entropie de ces dernières.

PROPOSITION III.13 : *Le nombre de points périodiques de période $2n$ de G_{f_0} est supérieur à n^2, pour n grand.*

PREUVE : *On suppose que l'on a au moins deux cellules ($\mathsf{N} \geqslant 2$), et on considère les stratégies $2n$ périodiques, telles que le premier terme et le $2n$-ième terme sont égaux à 2, entre ces termes il y a un autre couple de 2, et tous les autres termes sont égaux à 1.*

Pour le couple de 2, on a le choix entre $\dfrac{(2n-2)(2n-3)}{2}$ positions possibles. Le nombre total de points $2n-$périodiques est donc supérieur à $2^{\mathsf{N}-1}(2n-2)(2n-3)$, ce qui est bien supérieur à n^2 pour n grand.

CHAPITRE 13

Itérations chaotiques et chaos selon Devaney

> Moi qui reçois cette sentence, j'ai peut-être moins peur que vous qui la prononcez.
>
> GIORDANO BRUNO

On présente, dans ce chapitre, l'un des résultats de base de nos travaux : les itérations chaotiques sont du chaos au sens de Devaney, quand on considère la négation vectorielle [GB10]. On en profite pour mesurer la constante de sensibilité de telles itérations [GFB10].

I. RÉGULARITÉ DES ITÉRATIONS CHAOTIQUES G_{f_0}

On rappelle qu'un système dynamique discret $(\mathcal{X}, f)$ est dit régulier si l'ensemble des points périodiques de f est dense dans $\mathcal{X}$ (définition II.24). Nous allons démontrer le résultat suivant :

PROPOSITION III.14 : *$(\mathcal{X}, G_{f_0})$ est régulier.*

PREUVE : *Soit $(\check{S}, \check{E}) \in \mathcal{X}$ et $\varepsilon > 0$. On cherche un point périodique $(\widetilde{S}, \widetilde{E})$ tel que $d((\check{S}, \check{E}); (\widetilde{S}, \widetilde{E})) < \varepsilon$. Comme ε peut être strictement plus petit que 1, on doit choisir $\widetilde{E} = \check{E}$. Soit $k_0 = \lfloor -\log_{10}(\varepsilon) \rfloor + 1$ la plus grande puissance de 10 inférieure à ε, et*

$$\mathcal{S}_{\check{S},k_0} = \left\{ S \in \mathcal{S} \,\middle/\, S^k = \check{S}^k, \forall k \leqslant k_0 \right\}$$

l'ensemble des stratégies chaotiques dont les k_0 premiers termes coïncident avec ceux de $\check{S}$. Ainsi définis, tous les points $(S, \check{E})$ sont proches de notre point de départ à ε près : $\forall S \in \mathcal{S}_{\check{S},k_0}$, $d((S, \check{E}); (\check{S}, \check{E})) < \varepsilon$. Il reste à choisir $\widetilde{S} \in \mathcal{S}_{\check{S},k_0}$ tel que $(\widetilde{S}, \widetilde{E}) = (\widetilde{S}, \check{E})$ soit un point périodique pour G_{f_0}.

Soit $\mathcal{J} = \left\{ i \in [\![1, \mathsf{N}]\!] \,\middle/\, E_i \neq \check{E}_i, \text{ où } (S, E) = G_{(f_0)}^{k_0}(\check{S}, \check{E}) \right\}$, $i_0 = card(\mathcal{J})$ et $j_1 < j_2 < ... < j_{i_0}$ les éléments de $\mathcal{J}$. Alors, $\widetilde{S} \in \mathcal{S}_{\check{S},k_0}$ défini par :

- *$\widetilde{S}^k = \check{S}^k$, si $k \leqslant k_0$,*

- $\widetilde{S^k} = j_{k-k_0}$, *si* $k \in [\![k_0+1, k_0+i_0]\!]$,
- *et* $\widetilde{S^k} = \widetilde{S^j}$, *où* $j \leqslant k_0 + i_0$ *satisfait* $j \equiv k \ (mod\, k_0 + i_0)$, *si* $k > k_0 + i_0$,

est tel que $(\widetilde{S}, \widetilde{E})$ *est un point périodique de période* $k_0 + i_0$, *qui est à une distance inférieure à* ε *de* $(\check{S}, \check{E})$. *On a ainsi trouvé un point périodique à distance* ε *de tout point* (S, E), *et ce pour tout* $\varepsilon > 0$: $(\mathcal{X}, G_{f_0})$ *est régulier.*

Il existe de ce fait au moins une fonction dont les itérations chaotiques sont régulières. On notera $\mathcal{R}$ l'ensemble des fonctions satisfaisant cette propriété :

NOTATION III.3. $\mathcal{R}$ est l'ensemble non vide $\left\{ f : \mathbb{B}^{\mathbb{N}} \longrightarrow \mathbb{B}^{\mathbb{N}} \,\middle/\, (\mathcal{X}, G_f) \text{ est régulier} \right\}$.

II. ITÉRATIONS CHAOTIQUES ET TRANSITIVITÉ

Soient $\mathcal{B}_A = \mathcal{B}(X_A, r_A)$ et $\mathcal{B}_B = \mathcal{B}(X_B, r_B)$ deux boules ouvertes de $\mathcal{X}$. Existe-t-il $X_0 \in \mathcal{B}_A$ tel que $G_f^{(k)}(X_0) \in B$? Intuitivement, on comprend bien que cela dépend de G_f, c'est-à-dire de f. Ainsi, si l'on prend pour f la fonction identité, alors il n'y a aucun mélange (pas de transitivité). On montre ci-dessous qu'il existe des fonctions f telles que G_f soit transitive.

1. Un exemple d'itérations transitives

Il existe au moins une fonction d'itération rendant les IC transitives : c'est la négation vectorielle.

PROPOSITION III.15 : $(\mathcal{X}, G_{f_0})$ *est transitif.*

PREUVE : *Pour plus de lisibilité, définissons* $\mathcal{E} : \mathcal{X} \to \mathbb{B}^{\mathbb{N}}$ *par* $\mathcal{E}(S, E) = E$. *Soient* $\mathcal{B}_A = \mathcal{B}(X_A, r_A)$ *et* $\mathcal{B}_B = \mathcal{B}(X_B, r_B)$ *deux boules ouvertes de* $\mathcal{X}$, *avec* $X_A = (S_A, E_A)$ *et* $X_B = (S_B, E_B)$. *On cherche* $\widetilde{X} = (\widetilde{S}, \widetilde{E}) \in \mathcal{B}_A$ *tel que* $\exists n_0 \in \mathbb{N}, G_{f_0}^{(n_0)}(\widetilde{X}) \in \mathcal{B}_B$.

$\widetilde{X}$ *doit appartenir à* $\mathcal{B}_A$ *et* r_A *peut être strictement plus petit que 1, donc* $\widetilde{E} = E_A$. *Soit* $k_0 = \lfloor -\log_{10}(r_A) + 1 \rfloor$. *Alors* $\forall S \in \mathcal{S}$, *si* $S^k = S_A^k, \forall k \leqslant k_0$, *alors* $(S, \widetilde{E}) \in \mathcal{B}_A$. *Posons* $(\check{S}, \check{E}) = G_{f_0}^{(k_0)}(S_A, E_A)$ *et* $c_1, ..., c_{k_1}$ *les éléments de l'ensemble* $\left\{ i \in [\![1, \mathsf{N}]\!] \,\middle/\, \check{E}_i \neq \mathcal{E}(X_B)_i \right\}$. *Alors tout point* X *de l'ensemble* $\{(S, E_A) \in \mathcal{X} / \forall k \leqslant k_0, S^k = S_A^k \text{ et } \forall k \in [\![1, k_1]\!], S^{k_0+k} = c_k\}$ *vérifie* $X \in \mathcal{B}_A$ *et* $\mathcal{E}\left(G_{f_0}^{(k_0+k_1)}(X)\right) = E_B$.

Enfin, posons $k_2 = \lfloor -\log_{10}(r_B) \rfloor + 1$. *Alors* $\widetilde{X} = (\widetilde{S}, \widetilde{E}) \in \mathcal{X}$ *défini par :*

1. $\widetilde{E} = E_A$,
2. $\forall k \leqslant k_0, \widetilde{S^k} = S_A^k$,
3. $\forall k \in [\![1, k_1]\!], \widetilde{S^{k_0+k}} = c_k$,
4. $\forall k \in \mathbb{N}^*, \widetilde{S^{k_0+k_1+k}} = S_B^k$,

est tel que $\widetilde{X} \in \mathcal{B}_A$ *et* $G_{f_0}^{(k_0+k_1)}(\widetilde{X}) \in \mathcal{B}_B$.

REMARQUE. $(\mathcal{X}, G_{f_0})$ est donc indécomposable (définition II.28) : on ne peut pas, en particulier, réduire son étude à de plus petits systèmes $\left(\mathcal{Y}, G_{f_0}\right)$, où $\mathcal{Y} \subset \mathcal{X}$.

Parler de l'ensemble des applications dont le système associé est transitif a dorénavant un sens, puisque nous savons maintenant qu'il existe au moins une fonction dont les IC sont transitives. Nous noterons cet ensemble $\mathcal{T}$:

NOTATION III.4. $\mathcal{T}$ désigne l'ensemble non vide $\left\{ f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}} \,\middle/\, (\mathcal{X}, G_f) \text{ est transitif} \right\}$.

2. Transitivité forte

Reconsidérons la preuve de la proposition III.15. On a construit un point $\widetilde{X}$ aussi proche que l'on veut de X_A et dont une itérée $G_{f_0}^{(k_0+k_1)}(\widetilde{X})$ est en fait *égale à* X_B. C'est la transitivité forte :

PROPOSITION III.16 : *$(\mathcal{X}, G_{f_0})$ est fortement transitif.*

REMARQUE. On a vu aussi que $(\mathcal{X}, d)$ est un espace métrique compact (proposition III.3). D'après la propriété II.8, on peut directement en déduire la proposition précédente. D'autre part, $\mathcal{T}$ est donc aussi égal à $\left\{ f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}} \,\middle/\, (\mathcal{X}, G_f) \text{ est fortement transitif} \right\}$.

III. Chaos au sens de Devaney

Des résultats précédents, on peut en déduire que :

THÉORÈME III.5 : *$(\mathcal{X}, G_{f_0})$ est chaotique au sens de Devaney.*

NOTATION III.5. On pose $\mathcal{C} = \mathcal{R} \cap \mathcal{T}$ l'ensemble des fonctions f telles que $(\mathcal{X}, G_f)$ est chaotique au sens de Devaney. D'après le théorème précédent, $\mathcal{C}$ est non vide. Il nous faudra caractériser les ensembles $\mathcal{R}$ et $\mathcal{T}$ au chapitre suivant, pour mieux connaître l'ensemble qui nous intéresse le plus, à savoir $\mathcal{C}$.

IV. La sensibilité aux conditions initiales

On rappelle qu'un système est sensible aux conditions initiales si pour chaque x, il existe des points arbitrairement proches de x dont les orbites respectives sont séparées au moins de ε pendant l'évolution du système (définition II.37). Tous les points voisins de x ne sont pas forcément séparés de ε pendant l'évolution du système : il suffit qu'il en existe au moins un dans chaque boule ouverte de centre x. Nous savons que cette propriété est une conséquence de la régularité et de la transitivité (théorème de Banks II.10). Cependant, cela ne nous dit pas quelle est la constante de sensibilité. C'est pourquoi l'on va redémontrer la sensibilité « à la main » :

PROPOSITION III.17 : *G_{f_0} est sensible aux conditions initiales sur $(\mathcal{X}, d)$ et sa constante de sensibilité est supérieure ou égale à $\mathsf{N}-1$.*

PREUVE : *Soit $\check{X} = (\check{S}, \check{E}) \in \mathcal{X}$. On recherche $\widetilde{X} = (\widetilde{S}, \widetilde{E}) \in \mathcal{X}$ tel que $d\left((\check{X}, \widetilde{X})\right) \leqslant \delta$ et $\exists n_0 \in \mathbb{N}$, $d\left(G_{f_0}^{(n_0)}(\check{X}); G_{(f_0)}^{n_0}(\widetilde{X})\right) \geqslant \mathsf{N}-1$. Posons $k_0 = \lfloor -\log_{10}(\delta) \rfloor + 1$, tel que si une stratégie S coïncide avec*

$\check{S}$ *sur les* k_0 *premiers termes* $\left(S \in \left\{S \in \mathcal{S} \,\middle/\, \forall k \leqslant k_0, S^k = \check{S}^k\right\}\right)$, *alors* $(S, \check{E})$ *et* $(\check{S}, \check{E})$ *sont proches à* δ *près* ($d\left((S, \check{E}), (\check{S}, \check{E})\right) \leqslant \delta$).

Soit $\mathcal{J} = \left\{i \in [\![1, \mathsf{N}]\!] \,\middle/\, \mathcal{E}\left(G_{f_0}^{(k_0)}(\check{S}, \check{E})\right)_i \neq \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{S}, \check{E})\right)_i\right\}$ *l'ensemble des indices des cellules différentes entre les états de* $G_{f_0}^{(k_0)}(\check{S}, \check{E})$ *et de* $G_{f_0}^{(k_0+\mathsf{N})}(\check{S}, \check{E})$, *et soit* p *le nombre de différences :* $p = card\,(\mathcal{J})$. *Si* $p = \mathsf{N}$, *alors le point* $(\widetilde{S}, \widetilde{E}) \in \mathcal{X}$ *défini par :*

1. $\widetilde{E} = \check{E}$, *pour avoir les mêmes cellules que* $\check{X}$, *et donc être à une distance inférieure à 1 de* $\check{X}$,
2. $\forall k \leqslant k_0, \widetilde{S}^k = \check{S}^k$, *pour être proche de* $\check{X}$ *à* δ *près,*
3. $\forall k \in [\![1, \mathsf{N}]\!], \widetilde{S}^{k_0+k} = k$, *pour avoir ensuite un état dont les cellules sont toutes différentes de* $\check{X}$,
4. $\forall k > k_0 + \mathsf{N}, \widetilde{S}^k = 1$, *pour finir de définir pleinement* $(\widetilde{S}, \widetilde{E})$,

satisfait $d\left((\widetilde{S}, \widetilde{E}); (\check{S}, \check{E})\right) < \delta$ *(c'est-à-dire que* $\widetilde{X}$ *est proche de* $\check{X}$*), et* $\forall i \in [\![1, \mathsf{N}]\!]$, $\mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\widetilde{S}; \widetilde{E})\right)_i \neq \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{S}; \check{E})\right)_i$, *(c'est-à-dire que l'itérée* $k_0 + \mathsf{N}$ *de* $\widetilde{X}$ *est éloignée d'une distance supérieure à* N *de l'itérée* $k_0 + \mathsf{N}$ *de* $\check{X}$*), donc le résultat est obtenu.*

Sinon, soit $j_1 < j_2 < ... < j_p$ *les éléments de* $\mathcal{J}$ *et* $j_0 \notin \mathcal{J}$. *Alors* $\widetilde{X} = (\widetilde{E}, \widetilde{S}) \in \mathcal{X}$ *défini, pour les mêmes raisons que précédemment, par :*

1. $\widetilde{E} = \check{E}$,
2. $\forall k \leqslant k_0, \widetilde{S}^k = \check{S}^k$,
3. $\forall k \in [\![1, p]\!], \widetilde{S}^{k_0+k} = j_k$,
4. $\forall k \in \mathbb{N}^*, \widetilde{S}^{k_0+p+k} = j_0$.

est tel que $d(\check{X}, \widetilde{X}) < \delta$. *De plus,* $\forall i \in [\![1, p]\!], \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{X})\right)_{j_i} \neq \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\widetilde{X})\right)_{j_i}$, *car :*

- $\forall i \in [\![1, \mathsf{N}]\!], \mathcal{E}\left(G_{f_0}^{(k_0)}(\check{X})\right)_i = \mathcal{E}\left(G_{f_0}^{(k_0)}(\widetilde{X})\right)_i$, *du fait de la définition de* k_0.
- $\forall i \in [\![1, p]\!], j_i \in \mathcal{J} \Rightarrow \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{X})\right)_{j_i} = \mathcal{E}\left(G_{f_0}^{(k_0)}(\check{X})\right)_{j_i}$, *d'après la définition de* $\mathcal{J}$.
- $\forall i \in [\![1, p]\!]$, j_i *apparaît exactement une fois dans* $\widetilde{S}^{k_0}, \widetilde{S}^{k_0+1}, ..., \widetilde{S}^{k_0+\mathsf{N}}$, *donc*

$$\mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\widetilde{X})\right)_{j_i} \neq \mathcal{E}\left(G_{f_0}^{(k_0)}(\widetilde{X})\right)_{j_i}.$$

Enfin, $\forall i \in [\![1, \mathsf{N}]\!] \setminus \{j_0, j_1, ..., j_p\}, \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\widetilde{X})\right)_i \neq \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{X})\right)_i$, *car :*

- $\forall i \in [\![1, \mathsf{N}]\!], \mathcal{E}\left(G_{f_0}^{(k_0)}(\check{X})\right)_i = \mathcal{E}\left(G_{f_0}^{(k_0)}(\widetilde{X})\right)_i$,
- $i \notin \mathcal{J} \Rightarrow \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{X})\right)_i \neq \mathcal{E}\left(G_{f_0}^{(k_0)}(\check{X})\right)_i$,
- $i \notin \{\widetilde{S}^{k_0}, \widetilde{S}^{k_0+1}, ..., \widetilde{S}^{k_0+\mathsf{N}}\} \Rightarrow \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\widetilde{X})\right)_i = \mathcal{E}\left(G_{f_0}^{(k_0)}(\widetilde{X})\right)_i$.

Donc, dans ce cas, $\forall i \in [\![1, \mathsf{N}]\!] \setminus \{j_0\}, \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\widetilde{S}; \widetilde{E})\right)_i \neq \mathcal{E}\left(G_{f_0}^{(k_0+\mathsf{N})}(\check{S}; \check{E})\right)_i$ *et le résultat est obtenu.*

CHAPITRE 14

Caractérisation des IC chaotiques selon Devaney
Étude de C

Soit heureux un instant, cet instant c'est ta vie.

Rubaiyat
OMAR KHAYYAM

Nous savons donc qu'il existe des fonctions pour lesquelles les itérations chaotiques sont régulières et transitives, donc engendrent du chaos au sens de Devaney. Ce sont les fonctions de C. Notre prochain travail consiste à caractériser C.

I. CARACTÉRISATION DES FONCTIONS DE $\mathcal{T}$

Notons pour commencer le résultat immédiat suivant (on rappelle que le GTPIC a été défini en 4.3) :

PROPOSITION III.18 : *Dire qu'il existe un chemin de E_1 à E_2 dans le GTPIC $\mathcal{G}_f$ revient à dire qu'il existe une stratégie S et $n \in \mathbb{N}$ tels que $G_f^{(n)}(S, E_1)$ a pour état E_2. C'est-à-dire qu'il existe une stratégie « envoyant E_1 sur E_2 ».*

Nous sommes en mesure de caractériser $\mathcal{T}$ à partir de cette proposition :

PROPOSITION III.19 : *$(\mathcal{X}, G_f)$ est (fortement) transitive si, et seulement si le GTPIC $\mathcal{G}_f$ est fortement connexe.*

REMARQUE. On rappelle que, $\mathcal{X}$ étant compact, transitivité et transitivité forte coïncident.

PREUVE : *$\Longleftarrow$: Supposons que $\mathcal{G}_f$ soit fortement connexe, et montrons que $(\mathcal{X}, G_f)$ est fortement transitive. Soit donc (S, E) et (S', E') deux points de $\mathcal{X}$, et $\varepsilon > 0$. On va construire un point $(\hat{S}, \hat{E})$ proche de (S, E) dont une itérée est (S', E') (c'est la définition de la transitivité forte).*

Pour ce faire, on pose $\hat{E} = E$, et on recopie suffisamment de termes de S dans $\hat{S}$, pour que $d\Big((S, E); (\hat{S}, \hat{E})\Big) < \varepsilon$ ($n_1 = \lfloor -\log_{10}(\varepsilon) \rfloor + 1$ convient).

Soit $(\check{S},\check{E}) = G_f^{(n_1)}\left(\hat{S},\hat{E}\right)$ *l'image de* $\left(\hat{S},\hat{E}\right)$ *après* n_1 *itérations de* G_f.

Comme $\mathcal{G}_f$ *est totalement connexe, il existe un chemin de* $\check{E}$ *vers E'. En d'autres termes, il existe une stratégie finie* $(s_1,\dots,s_{n_2})$ *qui permet de transformer* $\check{E}$ *en E'. Concaténons cette stratégie à* $\hat{S}$*, puis rajoutons S' à la fin de* $\hat{S}$.

Alors $(S',E') = G_f^{(n_1+n_2)}\left(\hat{S},\hat{E}\right)$*, et l'on obtient bien le résultat escompté :* $(\mathcal{X},G_f)$ *est fortement transitive.*

$\underline{\Longrightarrow}$*: Réciproquement, si* $\mathcal{G}_f$ *n'est pas fortement connexe, alors il existe deux états* E_1, E_2 *pour lesquels il n'existe pas de chemin de* E_1 *à* E_2.

Il n'existe conséquemment pas de stratégie permettant de transformer E_1 *en* E_2*, et toute itération d'une stratégie quelconque sur* E_1 *sera toujours à une distance plus grande que 1 de* E_2.

Remarque. Cela revient à exiger une seule composante connexe pour $\mathcal{G}_f$. Remarquons qu'il existe des algorithmes efficaces qui déterminent si un graphe est fortement connexe ou non, par exemple en utilisant l'arborescence de Trémaux.

On peut donc en déduire que :

Corollaire III.1 : $\mathcal{T} = \left\{ f : \mathbb{B}^{\mathbb{N}} \to \mathbb{B}^{\mathbb{N}} \;\middle/\; \mathcal{G}_f \text{ est fortement connexe} \right\}$.

II. Lien entre $\mathcal{R}$ et $\mathcal{T}$. Caractérisation de $\mathcal{C}$

Nous savons dorénavant quelles fonctions f rendent les itérations chaotiques G_f transitives, c'est-à-dire que nous connaissons l'ensemble $\mathcal{T}$. Cependant, ça n'est pas $\mathcal{T}$ qui nous intéresse, mais l'ensemble $\mathcal{C} = \mathcal{T} \cap \mathcal{R}$ des fonctions rendant les IC chaotiques au sens de Devaney. Il nous reste donc à connaître plus en détail $\mathcal{R}$. En fait, nous allons démontrer que :

Proposition III.20 : $\mathcal{T} \subset \mathcal{R}$.

Preuve : *Supposons que* $f : \mathbb{B}^{\mathbb{N}} \to \mathbb{B}^{\mathbb{N}}$ *soit élément de* $\mathcal{T}$, i.e. *que* G_f *soit transitive sur* $(\mathcal{X},d)$. *Soit* $X = (S,E) \in \mathcal{X}$*, et* $\varepsilon > 0$. *On souhaite prouver qu'il existe un point périodique* $\check{X} = (\check{S},\check{E})$ *dans la boule ouverte* $\mathcal{B}(X,\varepsilon)$ *de centre X, rayon* ε. *Ceci étant vrai pour tout X et* ε*, on pourra en déduire la densité des points périodiques dans* $\mathcal{X}$. *On veut que* $\check{X}$ *soit dans* $\mathcal{B}(X,\varepsilon)$ *et* ε *peut être plus petit que* 1. *Il faut donc que* $\check{E} = E$. *Soit maintenant* $k_0 = \lfloor -\log_{10}(\varepsilon) \rfloor + 1$. *On pose alors* $\forall k \leqslant k_0(\varepsilon), \check{S}^k = S^k$*, toujours pour s'assurer que* $\check{X} \in \mathcal{B}(X,\varepsilon)$.

Notons $\widetilde{X} = (\widetilde{S},\widetilde{E})$ *le point* $G_f^{(k_0)}(X)$. f *étant dans* $\mathcal{T}$*, on en conclut que* $\mathcal{G}_f$ *est un graphe fortement connexe. En particulier, il existe un chemin reliant le sommet de* $\widetilde{E}$ *au sommet de E. Soit* $(s_1,s_2,...,s_i)$ *la succession ordonnée des arêtes de ce chemin. On pose :*

- $\forall k \in [\![1,i]\!], \check{S}^{k_0+k} = s_k$,
- $\forall k \geqslant k_0 + i + 1, \check{S}^k = \check{S}^{k \bmod k_0+i}$.

Ainsi construit, le point $\check{X}$ *est périodique, et est dans* $\mathcal{B}(X,\varepsilon)$.

Remarque. L'inclusion de la proposition III.20 est stricte, du fait de l'identité (elle est régulière, mais pas transitive).

On peut donc en déduire que $\mathcal{C} = \mathcal{R} \cap \mathcal{T} = \mathcal{T}$, ce qui fournit la caractérisation :

Théorème III.6 (Caractérisation des IC chaotiques) : *Les fonctions $f : \mathbb{B}^N \to \mathbb{B}^N$ telles que l'itération chaotique G_f est chaotique au sens de Devaney, sont les fonctions dont le GTPIC $\mathcal{G}_f$ est fortement connexe.*

Nous avons caractérisé $\mathcal{C}$, mais nous n'en connaissons pour l'instant pas grand chose d'autre. Nous allons maintenant voir ce que l'on peut dire de plus le concernant. Plus précisément, nous allons déterminer sa taille et évaluer le nombre de points périodiques de G_f pour $f \in \mathcal{C}$.

III. Détermination de la taille de $\mathcal{C}$

On commence par regarder la taille de $\mathcal{C}$, en calculant le nombre d'éléments de l'ensemble quotient $\mathcal{T}/\mathfrak{R}$, *i.e.* le nombre de graphes GTPIC. Cela revient à compter non pas tous les éléments de $\mathcal{C}$, mais uniquement les fonctions qui génèrent des itérations différentes pour une condition initiale donnée.

On rappelle que f et g sont en relation par $\mathfrak{R}$ si, et seulement si elles ont même GTPIC (définition I.33).

1. Dénombrement

Déterminons le nombre de classes d'équivalences de $\mathfrak{R}$ en comptant le nombre de graphes GTPIC. Cela peut se faire grâce à la caractérisation des GTPIC, que nous avions énoncé au chapitre 4 de la partie I (proposition I.15).

Pour chaque sommet $E \in [\![0; 2^{N-1}]\!]$, et pour chaque arc $\varepsilon_k, k \in [\![1; N]\!]$, correspondant à l'activation de la cellule k, il y a deux possibilités :

- La k-ième cellule de l'état du système E ne change pas quand la stratégie vaut k (ε_k est donc une boucle sur E).
- La k-ième cellule change d'état, ε_k est donc un arc de $E = (E_1, \dots, E_{k-1}, E_k, E_{k+1}, \dots, E_n)$ vers $(E_1, \dots, E_{k-1}, \overline{E_k}, E_{k+1}, \dots, E_n)$.

On a donc deux possibilités pour chacune des N arêtes des 2^N sommets. D'où...

Théorème III.7 : *Soit N le nombre de cellules du système. Alors $\mathfrak{R}$ possède $\left(2^N\right)^{2^N}$ classes d'équivalence.*

2. Conséquences

Le théorème III.7 signifie que si notre système admet N cellules, alors il n'y a que $\left(2^N\right)^{2^N}$ « types d'évolution future », ou itérations chaotiques fondamentalement différentes. Ainsi, si l'on a le choix entre

une infinité de fonctions f, cette infinité ne se résume au final qu'à $\left(2^{\mathsf{N}}\right)^{2^{\mathsf{N}}}$ classes d'itérations différentes dans le cas d'un système à N cellules.

Bien sûr, une fois un de ces $\left(2^{\mathsf{N}}\right)^{2^{\mathsf{N}}}$ graphes choisi, il nous reste une infinité de choix de points de départ (S, E) (car $\mathcal{X}$ est infini), qui donneront tous des itérations différentes. Reste qu'il peut être possible de cataloguer ces $\left(2^{\mathsf{N}}\right)^{2^{\mathsf{N}}}$ GTPIC fondamentalement différents, en fonction de leurs propriétés de graphes, par exemple : on pourrait peut-être en déduire quelque chose concernant le comportement des itérations chaotiques, obtenir un classement, préciser quelles fonctions choisir suivant quelles applications visées... Mais ce travail, sûrement conséquent, reste à faire.

Nous allons maintenant chercher à savoir si les fonctions de C engendrent des systèmes dynamiques discrets avec beaucoup de points périodiques. Si tel est le cas, l'utilisation des itérations chaotiques pour la sécurité informatique pourrait être discutable, du fait d'une trop grosse densité des points périodiques.

Nous verrons qu'il n'en est rien : les graphes GTPIC vont nous permettre de démontrer la dénombrabilité (infinie) des points périodiques de $(\mathcal{X}, G_f), \forall f \in C$. Or, on le rappelle, $\mathcal{X}$ est infini indénombrable (proposition III.2). Bref, sous un certain angle, il n'y a donc presque aucun point « raisonnable » chez les itérations chaotiques, même si l'on peut tomber dessus de manière imprévisible et qu'il y en a partout.

IV. Dénombrabilité des points périodiques de G_f, pour $f \in C$

Commençons par signaler que $\forall f \in C$, il n'y a jamais plus d'un arc reliant deux sommets distincts donnés (*c.f.* la caractérisation I.15 des GTPIC). Nous allons maintenant montrer que :

Théorème III.8 : *$\forall f \in C$, l'ensemble des points (ultimement) périodiques de G_f est infini dénombrable.*

Preuve : *Soit (E, S) un point (ultimement) périodique de G_f.*

À partir d'un certain rang n_0 éventuellement nul, on parcourt indéfiniment un cycle de sommets $(E_0, E_1, \ldots, E_k)$ dans le graphe des itérations. Soient :

- *ε_0 l'arc (orienté, unique) menant de E_0 à E_1,*
- *ε_1 l'arc (orienté, unique) menant de E_1 à E_2,* etc.

Alors, nécessairement, à partir du rang n_0, la stratégie boucle indéfiniment sur les valeurs $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{k-1}$. En associant à toute stratégie d'un point périodique le nombre réel de $\left[0, 2^{\mathsf{N}}\right]$:

- *dont la partie entière est le nombre entier dont l'écriture en base 2 correspond à la suite binaire ordonnée des états de E,*
- *et dont la partie décimale est ainsi construite : sa $n^{\text{ième}}$ décimale est le n^{ieme} terme de la stratégie S,*

on crée ainsi une bijection entre les points périodiques de G_f et les rationnels de l'intervalle $[0, 2^{\mathsf{N}}]$, puisque les rationnels sont les nombres dont le développement décimal est fini ou périodique à partir d'un certain rang.

CHAPITRE 15

Étude du désordre des itérations chaotiques

La sagesse consiste souvent à suivre sa folie plutôt que sa raison.

Le visiteur
ERIC EMMANUEL SCHMITT

Les itérations chaotiques G_f sont donc du chaos au sens de Devaney dès que $f \in C$. On a vu au chapitre 8 de la partie II, que cette définition ne suffit pas à pleinement appréhender le comportement de systèmes dynamiques désordonnés. Notre travail, dans ce qui suit, consiste à qualifier et quantifier plus précisément le chaos de nos itérations.

I. INSTABILITÉ, SENSIBILITÉ ET EXPANSIVITÉ

1. Instabilité

L'instabilité de la définition II.35 est plus faible que la sensibilité aux conditions initiales : dans l'instabilité, le ε dépend d'un point x. De cette relation d'implication, on en déduit que :

PROPOSITION III.21 : $\forall f \in C, (\mathcal{X}, G_f)$ *est instable.*

2. Chaos de la sensibilité aux conditions initiales

Nous avons vu que tout système dynamique discret transitif et régulier sur un espace métrique était sensible aux conditions initiales (c'est le théorème de Banks II.10). On peut donc en déduire que (*c.f.* définition II.43) :

PROPOSITION III.22 : $\forall f \in C, (\mathcal{X}, G_f)$ *est chaotique au sens de la sensibilité aux conditions initiales.*

REMARQUE. Cette proposition ne fournit pas la constante de sensibilité de G_f.

3. Chaos selon Wiggins

Le chaos selon Wiggins (définition II.44) est immédiat, dans la mesure où il requiert des systèmes transitifs et sensibles aux conditions initiales :

PROPOSITION III.23 : *$\forall f \in \mathcal{C}, (\mathcal{X}, G_f)$ est chaotique au sens de Wiggins.*

4. Chaos expansif

On rappelle que dans l'expansivité (définition II.38), *toute* erreur sur la position initiale est amplifiée, jusqu'à atteindre (au moins) la constante d'expansivité ε.

THÉORÈME III.9 : *Le système dynamique discret $(\mathcal{X}, G_{f_0})$ est expansif, et sa constante d'expansivité vaut 1.*

PREUVE : *On doit montrer que $\exists \varepsilon > 0, \forall X \neq Y, \exists n \in \mathbb{N}, d(G_{f_0}^{(n)}(X), G_{f_0}^{(n)}(Y)) \geqslant \varepsilon$.*

Soient $X = (S; E)$ et $Y = (\check{S}; \check{E})$ deux points distincts de $\mathcal{X}$. Alors :

- *Soit $E \neq \check{E}$, alors au moins une cellule ne présente pas le même état dans E et $\check{E}$. Donc la distance entre $(S; E)$ et $(\check{S}; \check{E})$ est supérieure ou égale à 1 : la propriété est montrée avec $n = 0$.*
- *Sinon, forcément, $E = \check{E}$. Et comme $X \neq Y$, on en déduit que les stratégies S et $\check{S}$ ne sont pas égales. Soit n_0 le premier indice de divergence entre les termes de S et ceux de $\check{S}$ (*i.e. *l'indice où pour la première fois le terme de S est différent de celui de $\check{S}$:*

$$\begin{cases} \forall k < n_0, G_{f_0}^{(k)}(S, E) = G_{f_0}^{(k)}(\check{S}, \check{E}), \\ \text{les états de } G_{f_0}^{(n_0+1)}(S, E) \text{ et de } G_{f_0}^{(n_0+1)}(\check{S}, \check{E}) \text{ sont différents.} \end{cases}$$

Comme $E = \check{E}$, la cellule qui a changé dans E à l'itérée n_0 n'est pas la même que la cellule qui a changé dans $\check{E}$, donc la distance entre $G_{f_0}^{(n_0+1)}(S, E)$ et $G_{f_0}^{(n_0+1)}(\check{S}, \check{E})$ est supérieure ou égale à 2.

On en conclut donc que $(\mathcal{X}, G_{f_0})$ est expansif, et que sa constante d'expansivité est supérieure ou égale à 1. Montrons que $(\mathcal{X}, G_{f_0})$ n'est pas A-expansif, avec $A > 1$. Si l'on considère deux points définis ainsi : même stratégie, et un seul état différent, alors il est clair que toutes les itérées de ces deux points restent à distance 1 exactement : même stratégie, et un seul état différent à chaque itération.

On en déduit naturellement le corollaire suivant.

COROLLAIRE III.2 : *Les itérations chaotiques sont du chaos expansif, quand la fonction d'itération est la négation vectorielle.*

REMARQUE. Il faudrait à un moment donné chercher à étendre l'étude de l'expansivité à toutes les fonctions de $\mathcal{C}$, et essayer de comprendre ce qui fait qu'une itération sera plus expansive qu'une autre. En effet, certaines applications nécessitent des systèmes expansifs (soit avec une grande constante d'expansivité, soit avec une petite). D'autres au contraire demandent des systèmes non-expansifs. On renvoie le lecteur aux parties finales de cette thèse, dans lesquelles ce point est détaillé.

II. Mélange topologique

On rappelle que le mélange topologique est une des versions plus fortes de transitivité (définition II.31). Ce mélange topologique sera une conséquence du résultat suivant :

Proposition III.24 : *Pour toute boule ouverte B de $\mathcal{X}$, il existe un indice n tel que $G_{f_0}^{(n)}(B) = \mathcal{X}$.*

Preuve : *En effet, soit $B = \mathcal{B}((S,E),\varepsilon)$ une telle boule ouverte, dont on peut supposer le rayon inférieur à 1 (qui peut le plus, peut le moins). Les éléments de B ont donc tous le même état E, et sont tels qu'il existe un indice $k\left(=-\log_{10}(\varepsilon)\right)$ vérifiant :*

- *Toutes les stratégies de B ont les k mêmes premiers termes.*
- *Au-delà du rang k, tous les termes sont possibles.*

Donc, au bout de k itérées, l'état du système est maintenant $G_{f_0}^{(k)}(S,E)_2$, et toutes les stratégies sont possibles (tout point de la forme $(G_{f_0}^{(k)}(S,E)_2, \hat{S})$, avec $\hat{S} \in \mathcal{S}$ quelconque, est atteignable à partir de B.

Soit maintenant un point quelconque (S',E') de $\mathcal{X}$. On va prouver qu'on peut l'atteindre à partir d'un élément de B. En effet, soit s la liste des cellules différentes entre $G_{f_0}^{(k)}(S,E)_2$ et E', et $|s|$ sa taille. Le point de B dont l'état est E, et la stratégie :

- *coïncide avec S sur les k premiers termes,*
- *se poursuit avec les éléments de s,*
- *se termine avec S'*

est tel que l'image par $G_{f_0}^{(k+|s|)}$ est exactement (S',E').

Ce résultat nous semble assez fort : aussi petite que soit la boule B de départ, il existe une itérée $G_{f_0}^{(k)}(B)$ qui recouvrira tout $\mathcal{X}$. On en déduit en particulier que :

Théorème III.10 : *$(\mathcal{X}, G_{f_0})$ est un système dynamique topologiquement mélangeant.*

Preuve : *Il s'agit de montrer que pour toute paire d'ouverts disjoints et non vides U et V, il existe $n_0 \in \mathbb{N}$ tel que $\forall n \geqslant n_0, G_{f_0}^{(n)}(U) \cap V \neq \varnothing$.*

Ceci est une application immédiate du précédent résultat.

III. Chaos au sens de Knudsen

Théorème III.11 : *$\forall f \in C$, $(\mathcal{X}, G_f)$ est chaotique au sens de Knudsen.*

Preuve : *$\forall f \in C$, $\mathcal{X}$ est compact, et $(\mathcal{X}, G_f)$ est chaotique au sens de Devaney, donc $(\mathcal{X}, G_f)$ est chaotique au sens de Knudsen (c'est la proposition II.15).*

La preuve ci-dessus n'est pas constructive : elle énonce l'existence d'une orbite dense, sans l'exhiber. Pour cette raison on donne une démonstration particulière de ce résultat dans le cas où la fonction d'itération f est la négation vectorielle f_0. On commence par rappeler la définition suivante :

DÉFINITION III.4 (NOMBRE DE CHAMPERNOWNE) : Le *Nombre de Champernowne* en base 10 se définit comme le nombre dont la partie entière est nulle, et dont la partie fractionnaire est la concaténation de tous les entiers naturels non nuls rangés dans l'ordre : $C_{10} = 0,1\,2\,3\,4\,5\,6\,7\,8\,9\,10\,11\,12\,13\,14\,15\,16\ldots$ ◊

Nous allons maintenant prouver que :

PROPOSITION III.25 : *$(\mathcal{X}, G_{f_0})$ possède une orbite dense.*

PREUVE : *On démontre ce résultat dans le cas où* $\mathsf{N} = 2$. *Il se généralise sans soucis au cas* N *quelconque, mais la démonstration en est alourdie. Soit $X = (S, E) \in \mathcal{X}$ définit comme suit :*

- *l'état E a toutes ses cellules mises à 0,*
- *S est la stratégie dont le n-ième terme est la n-ième décimale du nombre de Champernowne C_2, dans lequel on a enlevé les deux premiers termes (01), et que l'on modifie ainsi :*

$$
\begin{array}{cccc}
S = & & 00\ 01\ 10\ 11 & \\
& 0 & 00\ 01\ 10\ 11 & 0 \\
& 1 & 00\ 01\ 10\ 11 & 1 \\
& 01 & 00\ 01\ 10\ 11 & 01 \\
& & 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111 & \\
& 0 & 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111 & 0 \\
& 1 & 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111 & 1 \\
& 01 & 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111 & 01 \\
& \textit{etc.} & &
\end{array}
$$

Bref on encadre chaque « bloc » de décimales : 00011011, puis 000001010011100101110111, etc., *du nombre de Champernowne C_2, par chaque modification possible du système :*

- rien *: n'activer aucune cellule,*
- *0 : activer seulement la première cellule,*
- *1 : activer seulement la deuxième cellule,*
- *01 : activer les deux cellules.*

Chaque bloc de décimales contient un nombre pair de chiffres (0 et 1), donc l'état du système est le même au début et à la fin de chaque bloc. L'ajout des « rien, 0, 1, et 01 » en tête de chaque bloc de décimales permet de faire en sorte que, pour chaque bloc de décimales B, chaque état possible du système sera atteint. Enfin, l'ajout des « rien, 0, 1, et 01 » en fin de ligne remet l'état du système dans sa configuration d'origine (vecteur nul).

(Insistons sur le fait que la stratégie S n'est pas généralisée. Nous avons artificiellement regroupé par bloc les termes de S dans l'explication ci-dessus, pour mieux en comprendre sa définition, mais S n'est autre que la suite :

$$(0,0,0,1,1,0,1,1,0,0,0,0,1,1,0,1,1,0,1,0,0,0,1,1,0,1,1,1...).)$$

Soit donc un point $\check{X} = (\check{S}, \check{E}) \in \mathcal{X}$, et $\varepsilon > 0$. Alors il existera un n tel que $G_{f_0}^{(n)}(X)$ sera proche de $\check{X}$ à moins de ε. En effet, soit $k_0 = \lfloor -\log_{10}(\varepsilon) \rfloor + 1$. On cherche, dans l'écriture ci-dessus de la stratégie S, les lignes correspondant aux blocs de décimales de C_2 de taille k_0. Une de ces lignes aura une entête correspondant à l'état $\check{E}$:

- *l'entête « rien » pour* $\check{E} = (0,0)$,
- *l'entête « 0 » pour* $\check{E} = (1,0)$,
- *l'entête « 1 » pour* $\check{E} = (0,1)$,
- *l'entête « 01 » pour* $\check{E} = (1,1)$.

Soit n_0 *le nombre de chiffres séparant le début de* S *de la fin de cette entête. Alors* $G_{f_0}^{(n_0)}(X)$ *est proche de* $\check{X}$ *à moins de* ε*, ce qui conclut la preuve de la densité.*

IV. Chaos au sens de Li-Yorke

Nous allons maintenant montrer que les itérations chaotiques G_{f_0} sont du chaos au sens de Li et Yorke. À cette fin, il nous faut trouver des ensembles brouillés (*c.f.* définition II.51) indénombrables.

1. Définition des points d'un ensemble brouillé

Pour $x \in [0,1]$, on pose $(u(x)^n)_{n\in\mathbb{N}} = \left(\frac{\lfloor 10^n x \rfloor}{10^n}\right)_{n\in\mathbb{N}^*}$. En d'autres termes, le $n-$ième terme $u(x)^n$ de la suite $u(x)$ est égal à x, tronqué à sa n^{ieme} décimale, et $\forall x \in [0,1]$, la suite $(u(x)^n)_{n\in\mathbb{N}}$ converge (par défaut) vers x. L'ensemble des suites $\left\{u(x) \,\middle/\, x \in [0,1]\right\}$ est infini indénombrable, puisqu'il est en bijection avec $[0,1]$.

Déterminons un ensemble de points (S,E) de $\mathcal{X}$ candidat à être brouillé. E pourra être quelconque, mais fixé une fois pour toute. On définit alors, $\forall x \in [0;1]$, la stratégie $S(x)$ de la manière suivante :

- on prend deux fois les décimales de $u(x)^1$,
- on insère 2×1 zéros.
- On prend ensuite deux fois les décimales de $u(x)^2$,
- et on insère 2×2 zéros, *etc.*

Exemple III.2 : Pour $x = \dfrac{1}{\pi} = 0,3183098...$, on a

$$\left\{\begin{array}{lcl} u(x)^1 &=& 0,3 \\ u(x)^2 &=& 0,31 \\ u(x)^3 &=& 0,318, etc. \end{array}\right.$$

donc $S(x) = 3\ 3\ 00\ 31\ 31\ 0000\ 318\ 318\ 000000\ 3183\ 3183\ 00000000\ \ldots$

Remarque. L'application $x \in [0,1] \longmapsto S(x)$ est bijective.

2. Couples de points de limite inférieure nulle

Proposition III.26 : *Pour la négation vectorielle* f_0*, la fonction* S *ci-dessus, et pour un état quelconque* E *du système, on a :* $\forall x, x' \in [0,1]$, $\liminf_{n\to+\infty} d\left(G_{f_0}^{(n)}(S(x),E)\,;G_{f_0}^{(n)}(S(x'),E)\right) = 0$.

PREUVE : *En effet, au bout de deux itérations, chaque cellule a été activée un nombre pair de fois (0 ou 2 fois, puisque la stratégie commence par deux fois le même chiffre - c.f. l'exemple ci-dessus). Donc* $G_{f_0}^{(2)}(S(x),E)$ *et* $G_{f_0}^{(2)}(S(x'),E)$ *ont le même état, qui est* E.

De plus, les stratégies $S(x)$ *et* $S(x')$ *ont été décalées deux fois, donc commencent toutes les deux par 00. Ainsi, la distance entre* $G_{f_0}^{(2)}(S(x),E)$ *et* $G_{f_0}^{(2)}(S(x'),E)$ *est inférieure à* 10^{-2}.

De même, en itérant 2+2+4 = 8 fois à partir de $(S(x),E)$ *et de* $(S(x'),E)$, *on trouve dans les deux cas le même état, qui est à nouveau* E *(puisque chaque cellule a été activée un nombre pair de fois), et les deux stratégies décalées* $\sigma^{(8)}(S(x))$ *et* $\sigma^{(8)}(S(x'))$ *commencent par 0000. Ainsi, la distance entre* $G_{f_0}^{(8)}(S(x),E)$ *et* $G_{f_0}^{(8)}(S(x'),E)$ *est inférieure à* 10^{-4}.

On retrouve le même résultat à la troisième série de zéros, soit après 2+2+4+4+6 = 18 itérations, avec une distance intérieure à 10^{-6}.

On peut ainsi extraire une sous-suite $\left(G_{f_0}^{(\tau(n))}(S(x),E)\,;G_{f_0}^{(\tau(x))}(S(x'),E)\right)_{n\in\mathbb{N}}$, *où :*

$$\tau(n) = 2(1+2+3+...+n)+2(1+2+3+...+(n-1)) = n(n+1)+n(n-1) = 2n^2,$$

qui est telle que $d\left(G_{f_0}^{(\tau(n))}(S(x),E)\,;G_{f_0}^{(\tau(n))}(S(x'),E)\right)\leqslant 10^{-2n}$.

Il existe donc une sous-suite de $\left(d\left(G_{f_0}^{(n)}(S(x),E)\,;G_{f_0}^{(n)}(S(x'),E)\right)\right)_{n\in\mathbb{N}}$ *qui converge vers 0. De ce fait, la limite inférieure de cette suite est inférieure à 0 (*c.f. proposition II.17*).*

Comme cette suite est constituée d'éléments positifs (une distance est positive), sa limite inférieure est positive. Finalement, on trouve que cette limite est nulle.

3. Limite supérieure de ces points

PROPOSITION III.27 : *Pour la fonction « négation de chaque variable »* f_0, *et pour un état quelconque* E *du système, on a :* $\forall x,x'\in[0,1], \limsup_{n\to+\infty} d\left(G_{f_0}^{(n)}(S(x),E)\,;G_{f_0}^{(n)}(S(x'),E)\right)\geqslant 1$.

PREUVE : *Soit* $x\neq x'$, *et posons* $k_0 = \inf\left\{n\in\mathbb{N}\,/\,\lfloor 10^n x\rfloor \neq \lfloor 10^n x'\rfloor\right\}$ *:* k_0 *est l'indice de la première décimale différente entre* x *et* x'. *Alors les stratégies* $S(x)$ *et* $S(x')$ *seront égales jusqu'au rang* $4\times 1+4\times 2+\ldots+4\times(k_0-1)+k_0-1 = 2(k_0-1)k_0+k_0-1 = 2k_0^2-k_0-1$, *et vont différer au rang suivant :* $d\left(G_{f_0}^{(2k_0^2-k_0)}(S(x),E)\,;G_{f_0}^{(2k_0^2-k_0)}(S(x'),E)\right)\geqslant 1$.

À l'issue de la prochaine séquence de 0, les états sont les mêmes, et les stratégies (décalées) vont coïncider sur les k_0-1 *premiers termes, et diverger sur le terme* k_0.

Plus exactement, soit $\tau(n)=2n^2$ *la suite extraite de la preuve précédente. Alors, pour* $n\geqslant k_0$ *:*

- $\tau(n)$ *correspond au nombre d'itérations nécessaires pour atteindre les* n *successions de 0. A ce moment, les états de* $G_{f_0}^{(\tau(n))}(S(x),E)$ *et de* $G_{f_0}^{(\tau(x))}(S(x'),E)$ *sont les mêmes.*
- $\tau(n)+n$ *est le nombre nécessaire d'itérations pour atteindre la fin de la succession de ces* n *zéros. Les états sont toujours les mêmes.*
- $\tau(n)+n+k_0$ *est le nombre nécessaire d'itérations pour atteindre le premier chiffre différent, après cette succession de* n *zéros, dans les stratégies. A cette itérations, les états ne sont plus les mêmes.*

Au final, on a construit une sous-suite $d\left(G_{f_0}^{(\tau(n)+n+k_0)}(S(x),E)\,;G_{f_0}^{(\tau(x)+n+k_0)}(S(x'),E)\right)_{n\geqslant k_0}$ *de* $\left(d\left(G_{f_0}^{(n)}(S(x),E)\,;G_{f_0}^{(n)}(S(x'),E)\right)\right)_{n\in\mathbb{N}}$*, qui est infiniment souvent supérieure à 1. Donc la limite supérieure de cette dernière est plus grande que 1.*

4. Résultat de chaos

PROPOSITION III.28 : $\forall E\in\mathbb{B}^{\mathsf{N}}$*, les ensembles* $\mathsf{B}_E=\left\{(E,S(x))\;\middle/\;x\in[0;1]\right\}$ *sont brouillés et indénombrables. Il y en a donc (au moins)* 2^{N}.

Nous pouvons donc en déduire que :

THÉORÈME III.12 : $(\mathcal{X},G_{f_0})$ *est chaotique au sens de Li-Yorke.*

V. ENTROPIE DES ITÉRATIONS CHAOTIQUES

On fournit dans cette section deux preuves du fait que l'entropie topologique des itérations chaotiques est infinie.

1. Premier calcul de l'entropie

On rappelle que dans le cas des fonctions expansives, l'entropie est supérieure ou égale à la limite supérieure de $\dfrac{p_n}{n}$, où p_n est le nombre de points de période n au sens large (proposition II.22). À partir de ce résultat, on peut montrer que :

THÉORÈME III.13 : *L'entropie de* $(\mathcal{X},G_{f_0})$ *est infinie.*

PREUVE : *On a obtenu précédemment une majoration du nombre de points de période* $2n$ *par* n^2 *(c'est la proposition III.13). Alors :*

$$\limsup_n \frac{p_n}{n}=\lim_{n\to+\infty}\sup_{k\geqslant n}\frac{p_k}{k}\geqslant\lim_{2n\to+\infty}\sup_{k\geqslant 2n}\frac{p_k}{k}\geqslant\lim_{2n\to+\infty}\frac{p_{2n}}{2n}\geqslant\lim_{2n\to+\infty}\frac{n^2}{2n}=+\infty.$$

2. Deuxième calcul de l'entropie

Nous n'avons pas trouvé la preuve du fait que, dans le cas des fonctions expansives, l'entropie est supérieure égale à la limite sup de $\dfrac{p_n}{n}$, où p_n est le nombre de points de période n (au sens large). Il y a bien une bibliographie dans [Fri98b] (où le résultat est cité), mais nous n'avons pas encore pu trouver et parcourir chaque ouvrage référencé. N'ayant vu ce résultat nulle part ailleurs, nous préférons redémontrer l'infinité de l'entropie par une autre voie :

THÉORÈME III.14 : *L'entropie de* $(\mathcal{X}, G_f)$ *est infinie.*

PREUVE : *Soient deux états E et Ě différant d'une cellule. Alors pour toute stratégie S et Š,*

$$d((E,S);(\check{E},\check{S})) \geqslant 1.$$

Le nombre de stratégie S et Š est infini, donc plus grand que e^{n^2}, $\forall n$.
Donc $\forall n$, *le nombre maximal* $H(n,1)$ *de points* $(n,1)-$*séparé est supérieur à* e^{n^2}, *d'où*

$$h_{top}(G_f,1) = \limsup_{n\to+\infty} \frac{H(n,1)}{n} > \limsup_{n\to+\infty} \frac{\ln\left(e^{n^2}\right)}{n} = \limsup_{n\to+\infty}(n) = +\infty,$$

et comme $h_{top}(G_f,\varepsilon)$ *est croissante quand* ε *décroît, on en déduit que*

$$h_{top}\left(G_f\right) = \lim_{h\to 0} h_{top}(G_f,\varepsilon) > h_{top}(G_f,1) = +\infty.$$

CHAPITRE 16

De la relativité du chaos

> Ce qui est crime ici est souvent vertu quelque cent lieues plus bas, et les vertus d'un autre hémisphère pourraient bien réversiblement être des crimes pour nous.
>
> *La philosophie dans le boudoir*
> SADE

Dans ce chapitre, on se pose la question de savoir si le choix de la topologie a une importance pour les travaux que l'on mène. On constate que le fait d'être chaotique dépend de la finesse de la topologie que l'on considère. Que les systèmes sont toujours chaotiques, quand on considère la topologie grossière, et qu'ils ne le sont en général jamais avec la plus fine des topologies, à savoir la topologie discrète. Ce constat étant établi, on se demande à partir de quand considérer un système chaotique, et comment comparer le caractère chaotique de deux systèmes donnés.

Cette discussion ne porte que sur le chaos selon Devaney, mais devrait pouvoir s'étendre à d'autres formes de chaos.

I. PRÉSENTATION DU PROBLÈME

1. Approches relatives et absolues

Comme nous l'avions évoqué au chapitre 1 de l'introduction générale, nous souhaiterions par la suite appliquer le cadre de la théorie du chaos à la sécurité informatique (entre autres choses). On peut en effet voir un programme œuvrant sur des données comme une fonction itérant sur un ensemble – c'est du moins ce que nous apprend Turing (une formalisation plus précise de cela est donnée au chapitre 20 de la partie IV).

Nous souhaiterions considérer ce programme sûr quand, et seulement quand sa fonction associée est imprévisible, c'est-à-dire chaotique, pour au moins l'une des acceptions exposées dans la partie II. Plus précisément, étant donné un programme f œuvrant sur une collection d'objets $\mathcal{X}$, nous considérerons que ce programme est sûr si, et seulement si $(\mathcal{X}, f)$ est un système chaotique, suivant la définition de Devaney. Les raisons justifiant ce point de vue seront détaillées dans la partie IV.

Que l'ensemble $\mathcal{X}$ intervienne dans cette notion de sécurité c'est, sommes toutes, assez naturel : que cet ensemble ait une dizaine ou une infinité d'éléments ne signifie pas la même chose, que ce soit en termes de possibilités d'un comportement chaotique riche, ou bien en termes de sécurité – ne serait-ce que du point de vue d'une attaque de type brute-force, par exemple.

Supposons que l'on veuille comparer deux algorithmes f et g pour une tâche donnée qui nécessite un certain niveau de « sécurité » au sens large. Nous supposerons que f conviendra plus que g si son comportement à terme se déduit moins facilement que celui de g. C'est-à-dire que nous voudrions supposer f plus sûre que g si, et seulement si elle est plus chaotique.

Outre ce point de vue relatif, nous souhaiterions une approche plus absolue permettant de déterminer si un algorithme f donné est sûr ou pas – c'est-à-dire s'il est, ou non, chaotique.

2. Les problèmes soulevés par cette approche

Ces approches relatives et absolues supposent donc que l'on puisse déterminer :

1. Si un système $(\mathcal{X}, f)$ est chaotique, et pour quelle topologie.
2. Si un système $(\mathcal{X}, f)$ est plus chaotique qu'un système $(\mathcal{Y}, g)$.

Il se pose précisément trois problèmes :

- Les ensembles $\mathcal{X}$ et $\mathcal{Y}$ ne sont pas forcément les mêmes.
- Quand bien même ces ensembles seraient égaux, les topologies ne sont pas forcément les mêmes.
- Enfin, dans l'absolu, établir que $(\mathcal{X}, f)$ est chaotique doit dépendre du choix de la topologie, et ce choix n'est pas sans conséquence.

En d'autres termes, on ne regarde pas forcément la même chose. Et quand bien même, on ne la regarde pas forcément aux mêmes endroits. Et si tel est le cas, encore faut-il la regarder avec les mêmes yeux.

Le premier point ne pose pas vraiment problème, car les fonctions que l'on souhaite comparer sont censées s'appliquer aux mêmes types de données. De plus, si tel n'est pas le cas, ce problème peut être solutionné à l'aide d'une semi-conjugaison topologique vers un espace de référence (nous illustrerons cela au chapitre suivant). Les autres points soulèvent la question de savoir si le choix de la topologie sur $\mathcal{X}$ affecte la propriété de chaos, par exemple telle qu'elle est définie par Devaney, et le cas échéant quelles en sont les conséquences. La réponse à cette question est l'objet de la section 16.2. Enfin, ces interrogations nous poussent à nous intéresser à la question de la comparaison de deux algorithmes donnés, sachant une fois encore que l'on ne regarde en général ni la même chose, ni de la même manière. La fin de ce chapitre étudie cela.

II. Le désordre est relatif

1. Impact de la finesse de la topologie

Nous énonçons ici quelques résultats élémentaires, qui existent sûrement déjà dans la littérature, bien que nous n'en ayons pas trouvé trace. Introduisons deux notations pour commencer.

NOTATION III.6. On notera $\mathcal{X}_\tau$ l'espace topologique $(\mathcal{X}, \tau)$, et ce pour alléger les notations (diminuer le nombre de parenthèses).

NOTATION III.7. Nous noterons $\mathcal{V}_\tau(x)$ l'ensemble des voisinages de x pour la topologie τ. S'il n'y a pas ambiguïté, on utilisera simplement la notation $\mathcal{V}(x)$.

THÉORÈME III.15 : *Soient $\mathcal{X}$ un ensemble, et τ, τ' deux topologies sur $\mathcal{X}$ telles que τ' est plus fine que τ. Soit $f : \mathcal{X} \to \mathcal{X}$, continue à la fois pour τ et pour τ'.*

Si $(\mathcal{X}_{\tau'}, f)$ est chaotique selon Devaney, alors $(\mathcal{X}_\tau, f)$ l'est aussi.

PREUVE : *Commençons par établir la transitivité de $(\mathcal{X}_\tau, f)$.*

Soient ω_1, ω_2 deux ouverts de τ. Alors $\omega_1, \omega_2 \in \tau'$, car τ' est plus fine que τ. Comme f est $\tau'-$transitive, on en déduit que $\exists n \in \mathbb{N}, \omega_1 \cap f^{(n)}(\omega_2) = \emptyset$. En conséquence de quoi, f est $\tau-$transitive.

Établissons maintenant la régularité de $(\mathcal{X}_\tau, f)$, i.e. *pour tout $x \in \mathcal{X}$, et tout $\tau-$voisinage V de x, il existe un point périodique pour f dans V.*

Soit donc $x \in \mathcal{X}$, et $V \in \mathcal{V}_\tau(x)$ un $\tau-$voisinage de x. Par définition de la notion de voisinage, $\exists \omega \in \tau, x \in \omega \subset V$.

Or $\tau \subset \tau'$, donc $\omega \in \tau'$, et de ce fait $V \in \mathcal{V}_{\tau'}(x)$. Mais $(\mathcal{X}_{\tau'}, f)$ est régulier, donc il existe un point périodique pour f dans V, et la régularité de $(\mathcal{X}_\tau, f)$ est prouvée.

2. Comme quoi un système peut toujours être chaotique

Soit f une fonction itérant sur $\mathcal{X}$, ayant au moins un point fixe. Alors elle est chaotique (sous un certain angle) :

THÉORÈME III.16 : *Soit $\mathcal{X}$ un ensemble non vide, et $f : \mathcal{X} \to \mathcal{X}$ une application possédant au moins un point fixe. Alors f est τ_0-chaotique, où τ_0 est la topologie grossière sur $\mathcal{X}$.*

PREUVE : *f est transitive si $\forall \omega, \omega' \in \tau_0 \setminus \{\emptyset\}, \exists n \in \mathbb{N}, f^{(n)}(\omega) \cap \omega' \neq \emptyset$. Comme $\tau_0 = \{\emptyset, \mathcal{X}\}$, cela revient à chercher un entier n tel que $f^{(n)}(\mathcal{X}) \cap \mathcal{X} \neq \emptyset$. Par exemple, $n = 0$ convient.*

D'autres part, soit $x \in \mathcal{X}$, et $V \in \mathcal{V}_{\tau_0}(x)$. Alors $V = \mathcal{X}$, donc V possède un point fixe (à savoir, l'unique point fixe de f). De ce fait, f est régulière, et l'on en conclut le résultat.

3. Comme quoi un système peut toujours ne jamais être chaotique

THÉORÈME III.17 : *Soit $\mathcal{X}$ un ensemble, et $f : \mathcal{X} \to \mathcal{X}$ une application. Si $\mathcal{X}$ est infini, alors $(\mathcal{X}_{\tau_\infty}, f)$ n'est pas chaotique selon Devaney, où τ_∞ désigne la topologie discrète.*

PREUVE : *Montrons-le par l'absurde, en supposant $(\mathcal{X}_{\tau_\infty}, f)$ transitif et régulier.*

Soit $x \in \mathcal{X}$, et $\{x\}$ un de ses voisinages. Ce voisinage doit contenir un point périodique pour f, si l'on veut que $(\mathcal{X}_{\tau_\infty}, f)$ soit régulier. Donc x doit être un point périodique de f.

Soit $I_x = \left\{f^{(n)}(x), n \in \mathbb{N}\right\}$. Cet ensemble est fini car x est périodique, et $\mathcal{X}$ est infini, donc $\exists y \in \mathcal{X}, y \notin I_x$.

$(\mathcal{X}_{\tau_\infty}, f)$ devant être transitif, pour tous ouverts non vide A et B, il doit exister un entier n tel que $f^{(n)}(A) \cap B \neq \varnothing$. Or $\{x\}$ et $\{y\}$ sont des ouverts, et $y \notin I_x \Rightarrow \forall n, f^{(n)}(\{x\}) \cap \{y\} = \varnothing$.

III. RÉFLEXIONS AUTOUR D'UN DÉSORDRE ABSOLU

1. Quels sont les problèmes auxquels on fait face

Nous concluons des résultats de la section précédente qu'un système $(\mathcal{X}, f)$ peut être ou non chaotique, suivant la « richesse » de la topologie utilisée. Cela pose problème pour définir de manière absolue le désordre d'une application sur un ensemble.

Un deuxième problème se pose quand on veut comparer le désordre de deux fonctions définies sur un même ensemble. Pour que cette comparaison soit toujours possible, il faudrait que la relation « être plus chaotique que », basée sur la finesse de la topologie rendant chaque fonction chaotique, soit une relation d'ordre totale, dont la définition est rappelée ci-dessous :

DÉFINITION III.5 (RELATION D'ORDRE) : Soit E un ensemble, et une relation binaire sur cet ensemble notée $\leq$. Cette relation est une *relation d'ordre* si pour tous x, y et z éléments de E :

(réflexivité) $x \leq x$

(antisymétrie) $x \leq y$ et $y \leq x \Rightarrow x = y$

(transitivité) $(x \leq y$ et $y \leq z) \Rightarrow x \leq z$

Cette relation d'ordre est dite *totale* si tout élément de E est comparable à tout autre élément de E : $\forall x, y \in E, x \leq y$ ou $y \leq x$. Dans le cas contraire, elle est dite *partielle*. ◊

Cependant, la notion de finesse est une relation d'ordre partielle, et non totale, sur l'ensemble des topologies de $\mathcal{X}$: on a en fait affaire à un espace dit « réticulé » (un treillis), dont on rappelle la définition ci-dessous.

DÉFINITION III.6 (ESPACE RÉTICULÉ) : Un *espace réticulé* est un ensemble partiellement ordonné dans lequel chaque couple d'éléments admet une borne supérieure et une borne inférieure. ◊

Deux topologies données ne sont pas forcément comparables, donc le désordre de deux systèmes dynamiques ne l'est pas forcément. Nous précisons ci-dessous nos réflexions pour aboutir à une solution théorique à ces problèmes. Le chapitre suivant en apportera une solution plus pratique.

2. Quelles peuvent être les solutions à ces problèmes

Nous n'avons pas, à l'heure actuelle, trouvé de solution complète et pleinement satisfaisante aux problèmes évoqués ci-dessus. Nous avons pensé à deux pistes, que nous évoquons ci-dessous, mais qui n'ont pour l'instant pas pleinement abouti.

a. La plus fine topologie rendant une fonction chaotique

Sur un ensemble $\mathcal{X}$ donné, on aimerait vouloir dire qu'une fonction f est plus chaotique qu'une autre g, si la plus fine topologie rendant chaotique f, est plus fine que la plus fine topologie rendant g chaotique. En d'autres termes, s'il nous faut une topologie plus fine pour lever le chaos apparent de f, que pour lever celui de g. Dit de manière plus imagée, s'il nous faut un microscope plus puissant pour comprendre comment f évolue, par rapport à celui nécessaire pour observer g.

Encore faudrait-il que cela ait un sens, c'est-à-dire que pour un ensemble et une fonction donnée sur cet ensemble, il existe une plus fine topologie $\mathcal{T}$ la rendant chaotique.

Nous avons essayé de montrer qu'une telle topologie existe toujours, en notant $(\tau_i)_{i\in I}$ la collection de topologies pour lesquelles f est chaotique (I désignant un ensemble d'indices quelconque), et en considérant la topologie constituée des ensembles $\cup_{i\in I}\omega_i$, où $\forall i \in I, \omega_i \in \tau_i$. C'est-à-dire les réunions quelconques d'ouverts des τ_i.

Il nous a semblé que l'on définissait bien ainsi une topologie, mais cependant nous n'avons pas encore réussi à montrer que, pour cette topologie, f était bien chaotique. Nous avons essayé de regarder les réunions finies de tels ouverts, mais cela n'a pas non plus abouti. Peut-être faudrait-il être sur un compact, pour obtenir la transitivité ?

De plus, dans la définition de Devaney, f est sensée être continue. Ce qui signifierait dans ce contexte que $\forall i \in I, f : (\mathcal{X}, \tau_i) \to (\mathcal{X}, \tau_i)$ est continue, et qu'alors il faudrait que $f : (\mathcal{X}, \mathcal{T}) \to (\mathcal{X}, \mathcal{T})$ soit à son tour continue. Nous ne savons pas si tel est le cas.

b. Être chaotique, c'est l'être pour la topologie de l'ordre

La deuxième solution que l'on a envisagé est de considérer que toutes les topologies ne sont pas d'égale importance, mais qu'il y a des topologies plus naturelles que d'autres. Et que donc, être chaotique signifierait l'être pour la topologie la plus naturelle qui soit sur un ensemble $\mathcal{X}$ donné.

i. Pertinence de l'approche. Cette approche peut sembler arbitraire, mais c'est en fait l'approche qui a habituellement cours en sécurité informatique, nous semble-t-il : la sécurité s'exprime habituellement en termes de probabilités. Cependant, qui dit probabilité, dit espace probabilisé, donc tribu. Or, à notre connaissance, le choix de la tribu n'est jamais discuté. À vrai dire, ces études se font habituellement sur $\mathbb{R}$, et l'on suppose que la tribu adéquate est la tribu des boréliens de la topologie de l'ordre sur $\mathbb{R}$. Nous trouvons cela un peu réducteur, pour les raisons suivantes :

- Les programmes proposés ne sont pas tous définis sur $\mathbb{R}$, ou $\mathbb{R}^n$. On étudie donc pas ces programmes, mais une version plus ou moins modifiée pour les besoins de notre étude. Et certains programmes sont trop particuliers pour pouvoir être étudiés dans ce cadre.
- Une fois implantés sur machine, ces programmes n'œuvrent pas dans l'ensemble sur lequel l'étude de sécurité a été menée.
- Ces deux couches d'approximations soulèvent la question de la pertinence de la comparaison de deux programmes donnés.
- L'usage ne suffit pas à faire de la tribu des boréliens de la topologie de l'ordre la meilleure tribu qui soit. Une tribu plus riche permettrait peut-être d'obtenir une garantie de sécurité plus élevée.
- Enfin, un programme sûr pour la tribu des boréliens ne l'est peut-être plus pour une tribu différente.

Nous ne remettons pas en cause ce qui a été fait jusqu'à présent, nous nous posons juste des questions, et cherchons à partir sur de bonnes bases pour notre approche.

ii. L'approche en tant que telle. Notre deuxième approche, celle que nous retiendrons par la suite, consiste donc à considérer la topologie de l'ordre comme la plus raisonnable, celle servant de base à définir ce qu'est une fonction chaotique sur $\mathcal{X}$: elle l'est si et seulement si $(\mathcal{X}_{\tau_o}, f)$ est chaotique, où τ_o est la topologie de l'ordre.

Ainsi, l'étude de la sécurité d'une fonction f sur $\mathcal{X}$ serait la suivante :

- Supposer que l'ensemble considéré est un produit d'ensembles ordonnés.
- Considérer la topologie de l'ordre sur chaque ensemble ordonné, et la topologie produit sur l'ensemble $\mathcal{X}$.
- Dire qu'un système est chaotique s'il l'est pour cette topologie produit.

REMARQUE. À proprement parler, nous n'avons fait que déplacer le problème, en considérant que la topologie naturelle est celle de l'ordre. Pour commencer, cela suppose que $\mathcal{X}$ est ordonné. De plus, plusieurs relations binaires sur $\mathcal{X}$ peuvent faire de $\mathcal{X}$ un ensemble ordonné, et si l'on part du principe que la bonne topologie est celle de l'ordre, il faut alors préciser de quel ordre il s'agit, *i.e.*, quel est le bon ordre... Cependant, en faisant le choix de cette seconde approche, on se rapproche des conventions d'usage.

CHAPITRE 17

Une semi-conjugaison topologique ou comment passer de $\mathcal{X}$ à un intervalle réel

L'homme est un petit homme triste.

Les yeux plus grands que le ventre
CAVANNA

On présente dans ce chapitre une semi-conjugaison topologique entre nos itérations chaotiques œuvrant sur $\mathcal{X}$ et son équivalent sur $\mathbb{R}$. Les raisons d'être de cette semi-conjugaison sont multiples :

- En avoir une vision plus simple.
- Permettre de mieux comprendre leur dynamique, notamment en les étudiant avec les outils de l'analyse classique : continuité, dérivabilité, tracés de courbes, *etc.*
- Rendre possible le calcul de l'exposant de Lyapunov, réalisé à la section 17.4, car il nous faut une fonction dérivable pour cela.
- Faciliter la comparaison théorique de nos itérations chaotiques avec d'autres systèmes dynamiques, usuellement définis sur $\mathbb{R}$. Cette comparaison se fera principalement sous l'angle du désordre topologique, en tirant les conclusions du chapitre 16.
- Faciliter la comparaison pratique de la sécurité de nos algorithmes avec ceux déjà existants.

On suppose, dans la suite de ce chapitre, que $\mathsf{N} = 10$, pour se simplifier la vie, et parce que cela ne réduit en rien notre propos : on obtiendrait sans problème une formulation équivalente de ce qui suit en remplaçant l'écriture en base 10 par celle en base N.

I. NOTRE ESPACE DES PHASES EST UN INTERVALLE DE $\mathbb{R}$

1. Vers une semi-conjugaison topologique

Nous cherchons à montrer dans ce qui suit que les itérations chaotiques sur $\mathcal{X}$ peuvent être vues comme des itérations sur un intervalle réel, grâce à la semi-conjugaison topologique (*c.f.* chapitre 9). Pour cela, on doit introduire quelques définitions :

DÉFINITION III.7 : On définit la fonction φ de $\mathcal{S}_{10} \times \mathbb{B}^{10}$ dans $\left[0, 2^{10}\right[$ par :

$$\begin{array}{rccc} \varphi : & \mathcal{X}_{10} = \mathcal{S}_{10} \times \mathbb{B}^{10} & \longrightarrow & \left[0, 2^{10}\right[\\ & (S, E) = \left((S^0, S^1, \ldots); (E_0, \ldots, E_9)\right) & \longmapsto & \varphi\left((S, E)\right) \end{array}$$

où $\varphi\left((S, E)\right)$ est le réel :

- dont la partie entière e est $\displaystyle\sum_{k=0}^{9} 2^{9-k} E_k$, c'est-à-dire que le développement en base 2 de e est $E_0\ E_1\ \ldots\ E_9$.
- dont la partie décimale s est : $s = 0, S^0\ S^1\ S^2\ \ldots = \sum_{k=1}^{+\infty} 10^{-k} S^{k-1}$. ◊

La fonction φ permet donc d'associer un point de $\mathcal{X}_{10}$ à un réel de $\left[0, 2^{10}\right[$ (on rappelle que la notation $\mathcal{S}_{\mathsf{N}}$ désigne l'ensemble des suites à valeurs dans $[\![1; \mathsf{N}]\!]$, et que $\mathcal{X}_{\mathsf{N}} = \mathcal{S}_{\mathsf{N}} \times \mathbb{B}^{\mathsf{N}}$). Une fois que l'on se retrouve sur cet intervalle réel, il faut encore traduire les itérations chaotiques G_{f_0}. Nous devons pour cela commencer par introduire deux fonctions sur $\left[0, 2^{10}\right[$:

DÉFINITION III.8 : Soit $x \in \left[0, 2^{10}\right[$, et :

- $e_0, \ldots, e_9$ les chiffres de l'écriture en base 2 de la partie entière de x : $\lfloor x \rfloor = \displaystyle\sum_{k=0}^{9} 2^{9-k} e_k$.
- $(s^k)_{k \in \mathbb{N}}$ la suite des décimales de x, en choisissant l'écriture décimale de x qui ne se termine pas par une infinité de 9 : $x = \lfloor x \rfloor + \displaystyle\sum_{k=0}^{+\infty} s^k 10^{-k-1}$. ◊

On définit alors les fonctions e et s de la manière suivante :

$$\begin{array}{rccc} e : & \left[0, 2^{10}\right[& \longrightarrow & \mathbb{B}^{10} \\ & x & \longmapsto & (e_0, \ldots, e_9) \end{array}$$

et

$$\begin{array}{rccc} s : & \left[0, 2^{10}\right[& \longrightarrow & [\![0, 9]\!]^{\mathbb{N}} \\ & x & \longmapsto & (s^k)_{k \in \mathbb{N}} \end{array}$$

Nous pouvons maintenant définir la fonction g, dont le rôle est de traduire les itérations chaotiques G_{f_0} sur un intervalle de $\mathbb{R}$.

DÉFINITION III.9 : On définit la fonction g de $\left[0, 2^{10}\right[$ dans $\left[0, 2^{10}\right[$ par :

$$\begin{array}{rccc} g : & \left[0, 2^{10}\right[& \longrightarrow & \left[0, 2^{10}\right[\\ & x & \longmapsto & g(x) \end{array}$$

où g(x) est le réel de $\left[0, 2^{10}\right[$ défini de la manière suivante :

- sa partie entière a pour écriture binaire : $e'_0, \ldots, e'_9$, avec :

$$e'_i = \begin{cases} e(x)_i & \text{si } i \neq s^0 \\ e(x)_i + 1 \pmod 2 & \text{si } i = s^0 \end{cases}$$

- dont la suite des décimales est $s(x)^1, s(x)^2, \ldots$ ◊

En d'autres termes, si $x = \sum_{k=0}^{9} 2^{9-k} e_k + \sum_{k=0}^{+\infty} s^k \, 10^{-k-1}$, alors :

$$g(x) = \sum_{k=0}^{9} 2^{9-k}(e_k + \delta(k, s^0) \pmod 2) + \sum_{k=0}^{+\infty} s^{k+1} 10^{-k-1}.$$

2. Métriques sur $\left[0, 2^{10}\right[$

On peut munir l'ensemble $\left[0, 2^{10}\right[$ de plusieurs métriques, la plus usuelle étant la distance euclidienne que l'on rappelle ci-dessous :

NOTATION III.8. Nous noterons Δ la distance euclidienne sur $\left[0, 2^{10}\right[: \Delta(x, y) = |y - x|^2$.

Cette distance euclidienne ne reproduira pas fidèlement la notion de proximité et d'éloignement de notre distance d sur $\mathcal{X}$: elle n'est pas assez fine. C'est pour cela que l'on introduit une nouvelle distance ci-dessous :

DÉFINITION III.10 : Soient $x, y \in \left[0, 2^{10}\right[$. Nous noterons D la fonction de $\left[0, 2^{10}\right[^2$ dans $\mathbb{R}^+$ définie par : $D(x, y) = D_e\left(e(x), e(y)\right) + D_s\left(s(x), s(y)\right)$, où :

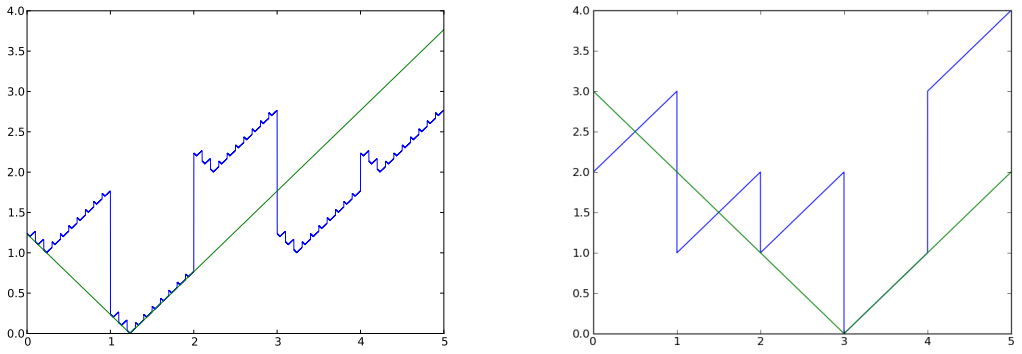
$$D_e(E, \check{E}) = \sum_{k=0}^{9} \delta(E_k, \check{E}_k), \quad \text{et} \quad D_s(S, \check{S}) = \sum_{k=1}^{\infty} \frac{|S^k - \check{S}^k|}{10^k}. \qquad \text{◊}$$

PROPOSITION III.29 : *D est une distance sur* $\left[0, 2^{10}\right[$.

PREUVE : *On vérifie les trois axiomes l'un après l'autre.*

- *$D \geqslant 0$, car tout est positif dans sa définition.*
- *Si $D(x, y) = 0$, alors $D_e(x, y) = 0$, donc les parties entières de x et y sont égales (elles ont même écriture en base 2). De plus, $D_s(x, y) = 0$, donc $\forall k \in \mathbb{N}^*, s(x)^k = s(y)^k$. En d'autres termes, x et y ont la même $k-$ième décimale en base 10, et ce pour tout k dans $\mathbb{N}^*$. Donc $x = y$.*
- *$D(x, y) = D(y, x)$.*
- *Enfin, l'inégalité triangulaire est satisfaite, car δ et la distance $\Delta(x, y) = |x - y|$ la vérifient.* □

La convergence pour D n'est pas la même que la convergence usuelle des suites pour la distance euclidienne. Par exemple, si $x^n \to x$ pour D, alors forcément la partie entière des x^n est égale à celle de x (à partir d'un certain rang), et la partie décimale de x^n coïncide avec celle de x « autant que l'on veut ». Pour illustrer cela, une comparaison de la distance D avec la distance euclidienne est donnée à la figure 17.1. Ces graphes illustrent le fait que D est plus riche et raffinée que la distance euclidienne, et permet des mesures plus précises.



(a) Application $x \to dist(x; 1,234)$ sur l'intervalle $(0;5)$. (b) Application $x \to dist(x;3)$ sur l'intervalle $(0;5)$.

FIGURE 17.1 – Comparaison des distances D (en bleu) et euclidienne (en vert).

3. La semi-conjugaison

À partir des définitions ci-dessus, il est possible d'exhiber une semi-conjugaison topologique entre $\mathcal{X}$ et un intervalle de $\mathbb{R}$:

THÉORÈME III.18 : *Les itérations chaotiques sur l'espace des phases $\mathcal{X}$ sont en fait de simples itérations sur $\mathbb{R}$. C'est ce que traduit la semi-conjugaison du diagramme suivant :*

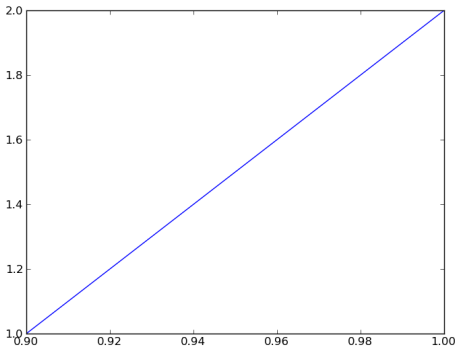
$$\begin{array}{ccc} \left(\mathcal{S}_{10} \times \mathbb{B}^{10}, d\right) & \xrightarrow{G_{f_0}} & \left(\mathcal{S}_{10} \times \mathbb{B}^{10}, d\right) \\ \varphi \downarrow & & \downarrow \varphi \\ \left(\left[0, 2^{10}\right[, D\right) & \xrightarrow[g]{} & \left(\left[0, 2^{10}\right[, D\right)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

PREUVE : *φ est clairement continue et surjective, elle a été construite à cette fin.*

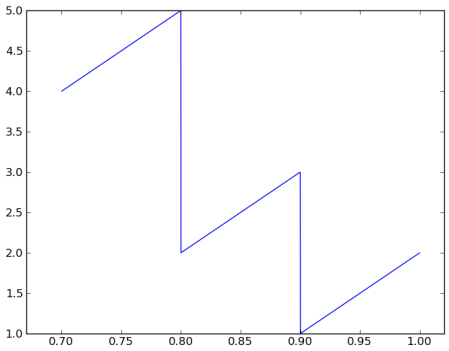
En d'autres termes, $\mathcal{X}$ est égal à $\left[0, 2^{\mathsf{N}}\right[$, à peu de choses près.

II. ÉTUDE DES ITÉRATIONS CHAOTIQUES VUES COMME UNE FONCTION DE $\mathbb{R}$

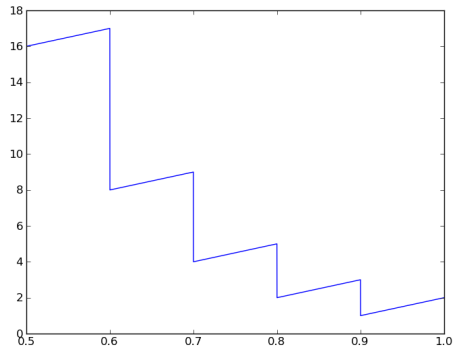
Nous avons écrit un programme en python nous permettant de représenter les itérations chaotiques (avec la négation vectorielle f_0) sur $\mathbb{R}$. Diverses représentations de ces IC sont données aux figures 17.2, 17.3 et 17.4. On y constate que la fonction g est affine par morceaux : elle est linéaire sur chaque intervalle de la forme $\left[\frac{n}{10}, \frac{n+1}{10}\right[$, $n \in [\![0; 2^{10} \times 10]\!]$, et sa droite représentative est de pente 10. Justifions cela :



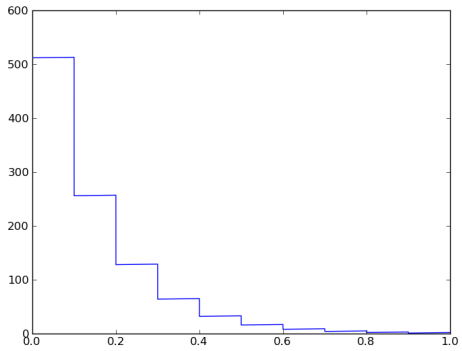
(a) Les IC sur l'intervalle $(0,9;1)$.



(b) Les IC sur l'intervalle $(0,7;1)$.



(c) Les IC sur l'intervalle $(0,5;1)$.



(d) Les IC sur $(0;1)$

FIGURE 17.2 – Représentation des itérations chaotiques.

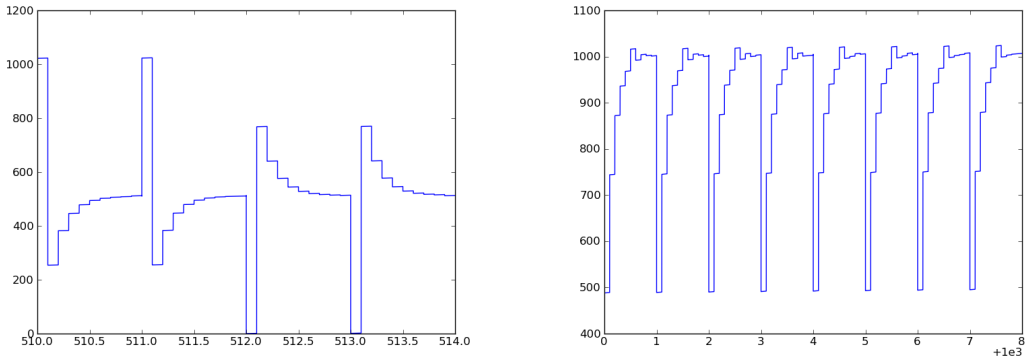
PROPOSITION III.30 : *Les itérations chaotiques g définies sur $\mathbb{R}$ sont infiniment dérivables sur $\left[0,2^{10}\right[$, sauf aux 10241 points de l'ensemble I défini par $\left\{\frac{n}{10} \,/\, n \in [\![0;2^{10}\times 10]\!]\right\}$.*

De plus, sur chaque intervalle de la forme $\left[\frac{n}{10},\frac{n+1}{10}\right[$, avec $n \in [\![0;2^{10}\times 10]\!]$, la fonction g est affine. C'est une droite de pente 10 : $\forall x \notin I, g'(x) = 10$.

PREUVE : *Soit $I_n = \left[\frac{n}{10},\frac{n+1}{10}\right[$, avec $n \in [\![0;2^{10}\times 10]\!]$. Tous les points de I_n ont la même partie entière e et la même première décimale s^0 : sur I_n les fonctions $e(x)$ et $x \mapsto s(x)^0$ de la définition III.8 ne dépendent que de n. Donc toutes les images $g(x)$ de ces points x :*

- *Ont la même partie entière, qui est e au bit s^0 près. Dit autrement, cet entier a la même écriture en base 2 que e, à l'exception du chiffre s^0 (ce nombre est donc $e + 2^{10-s^0}$, ou $e - 2^{10-s^0}$, suivant la parité de s^0, c'est-à-dire qu'il est égal à $e + (-1)^{s^0} \times 2^{10-s^0}$).*
- *La partie fractionnaire y à été décalée d'une case vers la gauche, le premier chiffre s^0 commun à tous étant perdu. En d'autres termes, y a été transformée en $10 \times y - s^0$.*

Ainsi, l'action de g sur les points de I est la suivante : multiplier par 10, et rajouter la même constante à tous, qui est $\frac{1}{10}\left(e + (-1)^{s^0} \times 2^{10-s^0}\right) - s^0$.



(a) Les IC sur l'intervalle $(510; 514)$.

(b) Les IC sur $(1000; 1008)$

FIGURE 17.3 – IC sur de petits intervalles.

REMARQUE. Ainsi, les itérations chaotiques appartiennent finalement à la grande famille des fonctions chaotiques linéaires par morceaux, comprenant la fonction tente (définition II.41), le doublement de l'angle (définition II.39), *etc.*

III. COMPARAISON DES MÉTRIQUES SUR $\left[0, 2^{\mathsf{N}}\right[$

Les deux propositions suivantes permettent de comparer nos deux distances sur $\left[0, 2^{\mathsf{N}}\right[$:

PROPOSITION III.31 : *Id* : $\left(\left[0, 2^{\mathsf{N}}\right[, \Delta\right) \to \left(\left[0, 2^{\mathsf{N}}\right[, D\right)$ *n'est pas continue.*

PREUVE : *La suite* $x^n = 1,999\ldots999$ *constituée de* n *9 comme décimales, est telle que :*

- $\Delta(x^n, 2) \to 0$.
- *Mais* $D(x^n, 2) \geqslant 1$, *donc* $D(x^n, 2)$ *ne tend pas vers 0.*

La caractérisation séquentielle de la continuité nous permet de conclure.

A contrario :

PROPOSITION III.32 : *Id* : $\left(\left[0, 2^{\mathsf{N}}\right[, D\right) \to \left(\left[0, 2^{\mathsf{N}}\right[, \Delta\right)$ *est continue.*

PREUVE : *Si* $D(x^n, x) \to 0$, *alors* $D_e(x^n, x) = 0$ *à partir d'un certain rang, car* D_e *ne prend que des valeurs entières. Donc, à partir d'un certain rang, les parties entières des* x^n *sont toutes égales à celle de* x.

D'autres part, $D_s(x^n, x) \to 0$, *donc* $\forall k \in \mathbb{N}^*, \exists N_k \in \mathbb{N}, n \geqslant N_k \Rightarrow D_s(x^n, x) \leqslant 10^{-k}$. *Ce qui signifie que pour tout* k, *il existe un rang* N_k *à partir duquel les* x^n *ont toutes les mêmes* k *premières décimales, qui sont celles de* x. *On déduit de tout cela la convergence* $\Delta(x^n, x) \to 0$, *et le résultat annoncé.*

On peut conclure des propositions suivantes que notre métrique est plus précise que la distance euclidienne. En d'autres termes :

COROLLAIRE III.3 : *La distance D est plus fine que la distance euclidienne Δ.*

On peut reformuler ce corollaire de l'une des manières suivantes :

- La topologie engendrée par Δ est incluse dans celle engendrée par D.
- D a plus d'ouverts que Δ.
- De manière imagée, D permet de mieux observer, de voir plus de détails que Δ.
- Enfin, il est plus difficile de converger pour la topologie τ_D engendré par D, que dans celle engendrée par Δ, et notée τ_Δ.

IV. CHAOS DES ITÉRATIONS CHAOTIQUES SUR $\mathbb{R}$

1. Chaos au sens de Devaney

Nous avons vu au chapitre 13 que les itérations chaotiques $\left(G_{f_0}, \mathcal{X}_d\right)$ étaient du chaos selon Devaney. Nous en déduisons qu'il en est de même pour les itérations chaotiques sur $\mathbb{R}$ pour la topologie de l'ordre, vu que :

- $\left(G_{f_0}, \mathcal{X}_d\right)$ et $\left(g, \left[0, 2^{10}\right[_D\right)$ sont semi-conjugués par φ (*c.f.*, chapitre 17),
- Donc $\left(g, \left[0, 2^{10}\right[_D\right)$ est un système chaotique selon Devaney, car la semi-conjugaison préserve ce caractère (chapitre 9).
- Or la topologie engendrée par D est plus fine que celle engendrée par la distance euclidienne Δ – qui est la topologie de l'ordre (chapitre 17).
- D'après le théorème III.15, on en déduit que les itérations chaotiques g sont du chaos, au sens de Devaney, pour la topologie de l'ordre sur $\mathbb{R}$.

On formule ce résultat de la manière suivante :

THÉORÈME III.19 : *Les itérations chaotiques g sur $\mathbb{R}$ sont du chaos au sens de Devaney, quand on munit $\mathbb{R}$ de sa topologie usuelle (la topologie de l'ordre).*

À vrai dire, ce résultat est moins fort que le théorème III.5, qui établit le chaos des itérées sur une topologie plus fine. On peut expliquer en quoi le théorème III.5 est meilleur que celui ci-dessus, de la manière imagée suivante. En utilisant les outils qui ont cours habituellement dans le domaine, on ne constate que désordre dans les IC de f_0 (théorème III.19) ; et quand bien même on passerait à une plus forte résolution, à des outils plus puissants que ceux qui sont usuellement utilisés, on ne parviendrait toujours pas à retrouver de l'ordre dans ce chaos (théorème III.5).

Le résultat du théorème III.19 nous est pourtant précieux. En effet, nous sommes partis d'un ensemble autre que celui habituellement considéré ($\mathcal{X}$ au lieu de $\mathbb{R}$), afin d'être au plus proche de la machine, et de ne pas perdre les propriétés de désordre en passant du papier au programme. On pouvait craindre que cette introduction du discret ne se paye uniquement par l'obtention de désordres de moins bonne qualité. En d'autres termes, passer d'une situation de bon désordre perdu lors du passage à la machine, à un désordre préservé mais de mauvaise qualité. Le théorème III.19 affirme exactement le contraire.

2. Calcul de l'exposant de Lyapunov des IC

Nous terminons ce chapitre et l'étude théorique des itérations chaotiques par le calcul de leur exposant de Lyapunov λ, que l'on a rappelé à la définition II.53, et qui vaut :

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

où $x^0 \in \left[0, 2^{10}\right[$ et $x^{i+1} = g(x^i)$.

Soit $\mathcal{L} = \left\{x^0 \in \left[0, 2^{10}\right[\;\middle/\; \forall n \in \mathbb{N}, x^n \notin I\right\}$, où I est l'ensemble des points de l'intervalle où g est non dérivable (c.f. proposition III.30). Alors :

THÉORÈME III.20 : *$\forall x^0 \in \mathcal{L}$, l'exposant de Lyapunov des itérations chaotiques ayant x^0 pour condition initiale vaut $\lambda(x^0) = \ln(10)$.*

PREUVE : *On rappelle que g est linéaire par morceaux, de pente 10 ($g'(x) = 10$ là où la fonction g est dérivable, c'est la proposition III.30). Alors : $\forall x \in \mathcal{L}$,*

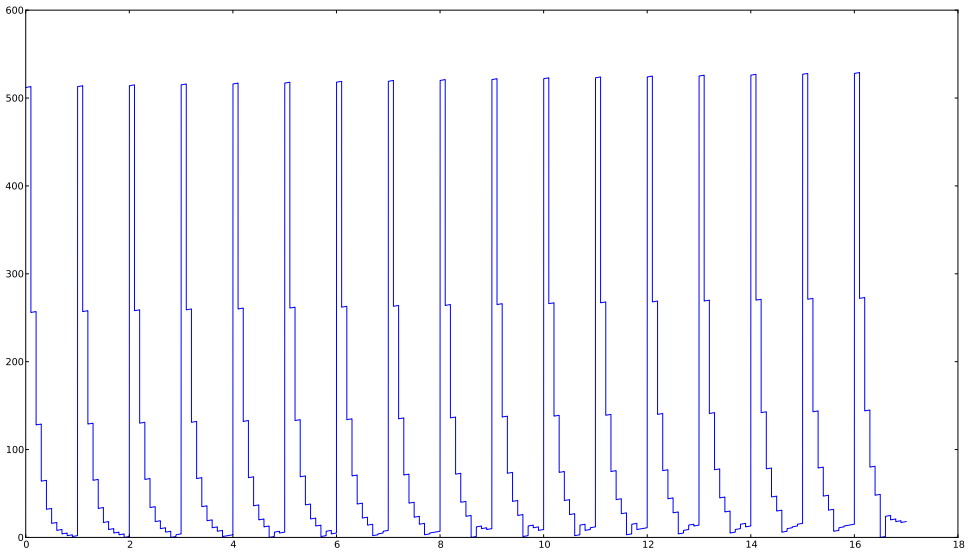
$$\lambda(x) = \lim_{n\to+\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| g'\left(x^{i-1}\right)\right| = \lim_{n\to+\infty} \frac{1}{n} \sum_{i=1}^{n} \ln|10| = \lim_{n\to+\infty} \frac{1}{n} n \ln|10| = \ln 10.$$

REMARQUE. L'ensemble des conditions initiales pour lesquelles cet exposant n'est pas calculable est dénombrable. Il s'agit en effet des conditions initiales telles qu'une itérée aboutira à un nombre de la forme $\frac{n}{10}$, avec n entier. On ne peut aboutir à ce genre de réels qu'en commençant sur un nombre décimal, car pour cela, il faut avoir une partie fractionnaire finie.

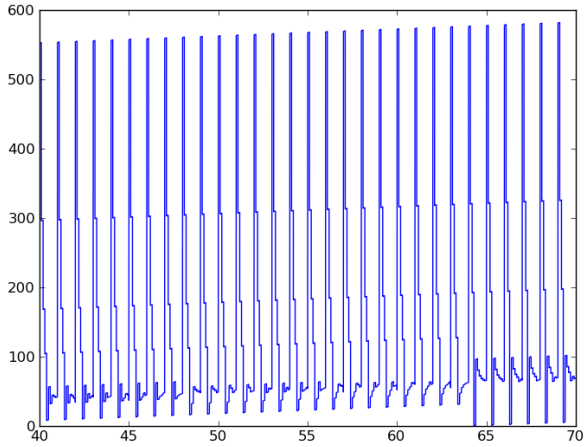
REMARQUE. Pour un système ayant N cellules, on trouvera un ensemble infini indénombrable de conditions initiales $x^0 \in \left[0; 2^{\mathsf{N}}\right[$ telles que $\lambda(x^0) = \ln(\mathsf{N})$.

On peut donc rendre l'exposant de Lyapunov des IC aussi grand que l'on veut, suivant le nombre de cellules de notre système. On rappelle à titre de comparaison que :

- l'exposant de Lyapunov de la suite logistique devient positif pour $\mu > 3,54$, mais qu'il reste toujours plus petit que 1.
- La fonction tente et le doublement de l'angle ont tous deux un exposant de Lyapunov égal à $\ln(2)$, indépendamment du choix de la condition initiale.



(a) Les IC sur l'intervalle $(0;16)$.



(b) Les IC sur l'intervalle $(40;70)$.

FIGURE 17.4 – Allure générale des itérations chaotiques.

CHAPITRE 18

Les aspects pratiques

Accepte l'épreuve afin d'y échapper.

CRATÈS

Dans ce chapitre, nous exposons quels sont les problèmes limitant à l'heure actuelle l'usage du chaos en informatique, qu'ils soient d'ordre théorique ou pratique, et quelles ont été les solutions que nous avons proposé pour résoudre ces problèmes soulevés. Nous montrerons, ce faisant, qu'il est possible de concevoir un algorithme réellement chaotique, tel que le programme associé ait un comportement sur machine qui soit aussi réellement chaotique.

I. PROBLÈMES DES APPROCHES EXISTANTES

1. Présentation des problèmes

Jusqu'à présent, la plupart des méthodes présentées comme chaotiques perdent, au moins en partie, leurs propriétés de désordre quand elles sont exécutées sur ordinateur (*i.e.* dans l'ensemble fini des nombres machine). De plus, le chaos n'a pas toujours bonne presse en informatique, et son utilisation est fréquemment critiquée. Cela est imputable, nous semble-t-il, aux raisons suivantes, que l'on a déjà partiellement évoquées dans les précédents chapitres :

Au **niveau théorique**, c'est-à-dire lors des réflexions autour de la conception de l'algorithme, nous pouvons relever les erreurs suivantes :

- La théorie du chaos, ses définitions et leur conséquences, ses raffinements, sont en général mal connus. Aussi, le sens même de chaos n'est pas précisé, les raisons pour lesquelles ce chaos est nécessaire ne sont pas claires, de même que l'intérêt de son utilisation.
- Les auteurs imaginent souvent qu'il suffit d'utiliser, à un moment donné, une fonction chaotique, pour obtenir à la fin un algorithme chaotique au complet.

Au **niveau pratique**, c'est-à-dire lors de l'exécution du programme en machine, les sources d'erreurs, rarement discutées, sont les suivantes :

- Toute machine ayant un nombre fini d'états finit par entrer dans un cycle.

- Les nombres machine ne sont qu'une approximation des réels, ce qui pose problème quand le programme est sensé être chaotique sur $\mathbb{R}$.

Nous détaillons dans ce qui suit chacun de ces problèmes, puis nous évoquons quelles sont nos solutions pour y faire face.

2. Le problème de l'absence de définition mathématique du chaos

Nous avons vu dans la partie II que du fait même de sa nature, le chaos est très difficile à définir, à enfermer dans une formulation comprenant tous ses aspects complexes. Cette complexité en fait sa force et son intérêt. Les multiples définitions rigoureuses de chaos, établies de longue date et étudiées en profondeur par la communauté mathématicienne s'y intéressant, sont toutes nécessaires et ne sont pas interchangeables. Le chaos de Devaney ne comporte ni expansivité, ni entropie. Le chaos au sens de l'exposant de Lyapunov n'entraîne pas la régularité ou la transitivité totale, *etc.* Suivant l'application considérée, toutes ces propriétés ne sont pas forcément nécessaires et, on le verra, certaines peuvent même être contraires aux objectifs visés. Aussi il conviendrait, lors d'une utilisation rigoureuse de la théorie du chaos pour l'informatique, de clairement définir ses besoins, et de les relier sans ambiguïté à divers aspects clairement définis de la théorie du chaos.

Cependant, nous n'avons pas connaissance de travaux mentionnant cela. La plupart du temps, le chaos n'est pas défini [ZZX+04, CJQZ06, LX07]. Dans le meilleur des cas, le moins fréquent, le chaos est défini en utilisant la simple sensibilité aux conditions initiales [DGW04, LYGL07], couplée éventuellement par une sorte de transitivité (« the outspreading of orbits over the entire space ») ni définie ni nommée [WG07], ou enfin avec l'exposant de Lyapunov. Ces termes ne sont pas précisés, et ces choix sont exclusifs : les papiers faisant mention de la sensibilité aux conditions initiales ignorent Devaney, Lyapunov, et toute autre propriété issue de la théorie du chaos. Nous trouvons cette approche beaucoup trop réductrice, et ce d'autant plus que le fait d'avoir choisi telle définition plutôt que telle autre n'est pas justifié. En d'autres termes, les auteurs ayant choisi de considérer la définition de Devaney n'expliquent pas pourquoi ils ont fait ce choix, en quoi cette définition répond à leurs besoins, et pourquoi les autres définitions ne les intéressent pas. À vrai dire, la question semble se résumer, dans les applications du chaos à l'informatique, à : quelle est la bonne définition de chaos. Or, nous avons vu que dans la communauté mathématicienne, cette question ne se pose pas : il n'y a pas de définition « la plus légitime qui soit » (signalons cependant que la notion de Devaney est considérée comme un bon point de départ pour initier une étude de chaos).

Cette méconnaissance de la théorie du chaos lors de son application en informatique ne se résume pas à la seule définition de chaos. Elle s'étend aussi au choix des outils utilisés. Plus précisément, dans tous les papiers que nous avons parcourus, les seules suites utilisées sont la suite logistique et le chat d'Arnold, que l'on a rappelées aux définitions II.48 et II.42. Or, d'une part, ces choix devraient être justifiés, mais ne le sont pas. Et d'autre part, ces choix sont critiquables, au moins en ce qui concerne la suite logistique : diverses études ont été menées au sein de la théorie du chaos, établissant que cette dernière ne devraient pas être utilisée, du fait de ses nombreux travers topologiques et statistiques, et des nombreuses failles de sécurité qu'elle présente (voir à ce sujet, par exemple, [AAF08]). Signalons de plus que :

- Cette fonction est quadratique, et il existe bon nombre de fonctions chaotiques linéaires par morceaux. La fonction logistique est donc beaucoup plus lente que la plupart de ses coreligionnaires.
- Cette suite logistique n'est pas magique, elle ne saurait être adaptée à tous les problèmes. Ses diverses caractéristiques topologiques ont été étudiées en profondeur, et de nombreuses autres suites chaotiques possèdent de meilleures propriétés qualitatives et quantitatives qu'elle.

3. Le problème de la partie et du tout

Le second problème qui nous est apparu dans les utilisations informatiques de la théorie du chaos, au moins dans les domaines du tatouage numérique et des fonctions de hachage, concerne le fait que la suite logistique ou le chat d'Arnold ne sont qu'une partie d'un tout, ce tout étant l'algorithme déclaré chaotique du fait même de la présence de l'une de ces deux suites. Cette assertion ne nous semble pas évidente, et mériterait d'être prouvée au cas par cas.

En d'autres termes, il s'agit de montrer que l'algorithme au complet est chaotique : il ne suffit pas d'utiliser un ingrédient chaotique, et de laisser supposer que sa présence suffit à rendre tel le tout.

Par exemple, les auteurs de [WG07] proposent l'algorithme de tatouage suivant :

1. un chat d'Arnold est utilisé pour sélectionner les pixels du support hôte qui embarqueront le filigrane,
2. une suite logistique est utilisée pour fixer quel bit du pixel hôte sélectionné sera remplacé par le bit considéré du filigrane.

On peut leur reprocher les points suivants :

- Dans le second point, la suite logistique est transformée en une suite d'entiers pris dans $[\![1;8]\!]$. Qu'est-ce qui permet de penser que cette suite d'entiers est elle-aussi chaotique, et pour quelle topologie ? Le simple contre exemple associant à tout terme u^n d'une suite logistique l'entier 0 devrait inciter à se montrer prudent, et à faire une preuve.
- La « composée » de cette suite d'entiers avec le chat d'Arnold est-elle chaotique ?
- Et, surtout, l'algorithme se résume avant tout au remplacement de bits de l'hôte par ceux du filigrane. C'est cette fonction de remplacement qui devrait être chaotique, qui devrait être prouvé tel.

Cette approche est commune à la majorité des méthodes de tatouage dites chaotiques, telles qu'elles ont été principalement proposées dans le courant des années 90 (un état de l'art de ces méthodes est donné en 22.1.1). Ce même genre de problème se retrouve dans d'autres domaines de la sécurité informatique pour lesquels l'utilisation du chaos a été proposée : les fonctions de hachage, les algorithmes de chiffrement, les générateurs de nombres pseudo-aléatoires...

4. Le problème des machines à ensemble fini d'états

Les programmes présentés comme chaotiques agissent généralement comme suit. Après avoir reçu ses données initiales, la machine travaille seule, sans interaction avec le monde extérieur. À la fin de ses itérations, elle renvoie le dernier résultat calculé. L'exemple de [WG07] agit de cette façon.

Le principal problème qui empêche de parler de chaos dans cette situation particulière est que, quand une machine à états finis atteint un même état interne à deux reprises, elle évolue alors deux fois de suite de la même manière. En d'autres termes, une telle machine finit toujours par entrer dans un cycle. Ce comportement trop prévisible, relié à l'ensemble fini sur lequel on itère, ne peut pas être défini comme chaotique, au moins selon l'exposant de Lyapunov. Et il semble à l'opposé du caractère désordonné attendu, même si à vrai dire un tel comportement, peu riche, n'est pas complètement incompatible avec la théorie du chaos.

5. Le problème de l'utilisation des réels

Le dernier problème induit par la programmation d'algorithmes se voulant chaotiques sur machine intervient, selon nous, du fait de l'utilisation de réels dans l'algorithme : ces derniers n'existent pas sur machine. Ainsi, quand on itère une fonction « logistique » ou une fonction « tente » dans un programme donné :

- On restreint l'ensemble des conditions initiales aux réels ayant un nombre fini borné de chiffres, alors que ces fonctions sont chaotiques sur $[0,1]$.
- Quand on itère ces fonctions, le nombre de décimales des termes obtenus n'est pas borné, sauf en de rares occasions. La version programme ne contient donc pas ces itérations, mais leur version tronquée à la précision limite, et rien ne garantit que cette version tronquée reste chaotique.
- Enfin, au niveau sécurité opérationnelle, les programmes supposés sûrs car se basant sur des fonctions réelles chaotiques, présentent des failles de sécurité liées à l'utilisation même des nombres réels. En effet, du fait que la définition et le traitement de ces derniers sont dépendants des processeurs, des malwares « sur mesure », plus spécifiques et plus virulents, sont ainsi envisageables, en tirant profit de la présence de ces réels [DEF10].

Quelques tentatives de définition d'un chaos discret ont été proposées [KSAT06], mais elles ne sont pas complètement satisfaisantes et sont moins reconnues que les notions exposées dans le présent document. D'autre part, nous avons eu connaissance de l'existence d'un résultat, appelé parfois le *lemme de filature*, affirmant que la version tronquée des itérées d'une fonction chaotique est aussi proche que l'on veut des itérées de cette suite pour une autre condition initiale. Nous n'avons cependant pas réussi à trouver de formulation exacte de ce résultat, et à notre connaissance il ne s'applique qu'à la notion de Devaney.

Nous avons donc préféré attaquer ce problème sous un autre angle, et tenter de réussir à faire fonctionner la machine de manière imprévisible. La prochaine section détaille notre approche.

II. Notre solution

Les problèmes évoqués à la précédente section peuvent être résolus des manières suivantes.

1. Réponse au problème des machines à ensemble fini d'états

Pour contourner le problème nous semblant le plus délicat, celui lié à la finitude de la machine, nous nous sommes dès le début intéressés aux itérations chaotiques : ces dernières, qui peuvent être fortement imprévisibles, opèrent sur un ensemble fini d'états, de la forme $\mathbb{B}^{\mathsf{N}}$. Cet ensemble fini représente la mémoire de la machine, et la fonction d'itération f est le programme considéré. Il reste à régler le problème de la stratégie, mais nous avons à l'esprit qu'au temps n, il n'est nécessaire d'avoir que son $n-$ième terme. Nous pouvons faire en sorte qu'à chaque nouvelle itération, une nouvelle donnée soit prise du monde extérieur. Plusieurs choix sont possibles, nous avons retenu le suivant : le $n-$ième terme de la stratégie sera, à peu de choses près, le $n-$ième coefficient d'une représentation numérique choisie du média sur lequel on itère (dans cette courte discussion, nous nous sommes restreints aux programmes dont les entrées sont des médias, mais le propos se généralise sans peine à tout type d'entrées).

L'ensemble des stratégies sera donc en correspondance biunivoque avec l'ensemble des médias numériques. Ces derniers sont des suites, finies mais non bornées, d'entiers :

- Une image TIFF en 256 niveaux de gris de Lena, de taille 256×256 sera, les en-têtes mises à part, une suite de 256^2 entiers pris dans $[\![0, 255]\!]$, chaque terme correspondant, de manière précise, au niveau de gris d'un pixel donné.
- Pour une image de même type, mais deux fois plus grande, on obtiendrait ainsi une suite de 512^2 entiers.
- Pour une image JPEG de Lena, les termes de la stratégie seraient obtenus à partir des coefficients DCT de chaque bloc de 8×8 pixels.
- Pour une image JPEG2000, les coefficients DWT seraient utilisés,
- *etc.*

Pour qu'il y ait effectivement correspondance biunivoque, il faudrait que la stratégie embarque aussi le type de média (image, son, *etc.*) et son mode de représentation choisi. On pourrait aussi imaginer que les stratégies soient construites à partir des bits représentant l'objet considéré en mémoire.

Chaque média a une taille finie, mais il n'y a pas de borne aux tailles des médias. Ainsi, chaque opéra a une durée limitée finie, mais les compositeurs peuvent écrire des opéras aussi long qu'ils veulent (*Licht*, de Stockhausen, dure 29 heures). Ce qui se traduit par le fait que l'ensemble des médias numériques est l'ensemble des suites finies d'entiers, telles que :

- Chaque terme est borné : ce terme est un chiffre, dans une base N, donc cet entier est borné par N.
- Le nombre de termes de la suite est fini, *mais non borné*.

L'ensemble des stratégies de nos programmes, correspondant à l'ensemble des médias numériques, est donc l'ensemble $\mathcal{S}_\mathsf{N}$ des suites de $[\![1, \mathsf{N}]\!]$ ayant un nombre fini non borné de termes. C'est un ensemble infini dénombrable : c'est $\mathbb{D}$.

Revenons plus précisément sur un point de détail de notre approche, mais qui a son importance. On pourrait croire que, la machine ayant un nombre fini d'états, dans la pratique on se retrouve limités par les suites de $\mathcal{S}_\mathsf{N}$ ayant un nombre de termes inférieur à ce nombre d'états. Et qu'une fois encore l'on se retrouve dans la situation d'une machine à états finis, qui finit forcément par boucler.

Il n'en est rien car, on le rappelle, le média sur lequel on travaille n'a pas besoin d'être stocké au complet dans la mémoire. Seul son $n-$ième coefficient nous intéresse à la $n-$ième itérée. On pourrait croire qu'il ne s'agit là que d'une vision de l'esprit, mais tel n'est pas le cas. Il arrive effectivement en pratique que l'on n'utilise qu'une partie du média sur lequel on itère, et que cette partie varie au cours du temps, comme l'illustre chacune des situations particulières suivantes :

- Un programme travaille sur un texte, par exemple le chiffre ou le hache. Il n'y a aucune obligation pour que *tout* le texte soit connu avant d'entamer le chiffrement et le hachage : on peut très bien, à chaque nouvelle lettre saisie par l'utilisateur, calculer le nouveau terme de la stratégie, et faire ensuite aussitôt la nouvelle itération du système sans attendre la suite. Ainsi fonctionnait la machine Enigma.

  Notons que, dans cet exemple, on est capable de hacher des textes dont la taille dépasse la mémoire disponible. Le texte n'est pas stocké, seule sa $n-$ième lettre et le condensé ont besoin de l'être.
- Cet exemple s'étend à tout programme agissant sur un flux de données : son enregistré en continu à partir du micro d'un ordinateur, vidéo d'une webcam, flux multimédia émis par un site web, *etc.*

Il nous semble que les exemples précédents illustrent bien le fait que le nombre fini d'états de la machine ne limite en rien la taille des données en entrée, à partir du moment où l'on considère qu'il n'est pas nécessaire, ni forcément toujours souhaitable, de séparer l'étape de saisie des entrées, de celle des opérations de la machine.

Cette généralisation s'étend aussi aux sorties du programme. On pourrait penser que ce qui a rendu possible l'exemple ci-dessus (hachage d'un texte aussi grand que l'on veut) réside dans le fait que dans le programme considéré, la sortie est finie bornée : par définition, la valeur hachée fait un nombre fini fixé de bits (512 bits, pour la plupart des fonctions de hachage modernes). Tel n'est pas le cas : pour les itérations chaotiques, la machine n'a besoin que du dernier état obtenu et de la dernière entrée reçue pour travailler. La suite des états précédemment obtenus ne lui est ici d'aucune utilité, et n'a pas lieu d'être stockée. Par contre, chaque nouvel état peut être publié en sortie, tel quel ou modifié [4]. C'est le cas lorsqu'une machine réalise des traitements en continu :

- Un serveur reçoit un flux vidéo en entrée, et publie un flux traité en sortie : vidéo compressée, taguée, chiffrée ou tatouée, *etc.*
- Cloud computing : les serveurs peuvent traiter les données en continu.

Dans chacune de ces situations, la machine est à nombre fini d'états, mais l'ensemble des entrées et l'ensemble des sorties sont infinis. Et si l'on prend garde à faire entrer à chaque nouvelle itérée un nouvel élément dans la machine, on obtient donc finalement que :

- la machine a certes une mémoire finie,
- mais cette dernière œuvre finalement sur un espace ayant un nombre infini d'états : sa mémoire, couplée avec le monde extérieur (on retrouve $\mathcal{X} = \mathbb{B}^{\mathsf{N}} \times [\![1;\mathsf{N}]\!]^{\mathbb{N}}$).

On comprend bien qu'ainsi la mémoire interne du système peut très bien se retrouver deux fois de suite dans le même état, cela ne voudra pas dire pour autant que la machine se mettra à évoluer deux fois de suite de la même manière, vu que :

- Cette évolution dépend certes de l'état interne de la machine, mais aussi de la valeur actuelle de l'entrée du système.
- Cette entrée ne sera pas forcément la même à ces deux instants-là (et quand bien même, cela indiquerait juste que la prochaine itération sera identique, sans rien laisser présager pour la suite).

En fait, l'argument selon lequel le chaos est irréalisable sur machine, du fait de la finitude de cette dernière, est erroné, car il ne considère que le cas particulier des programmes recevant leurs entrées à la première itérée, itérant ensuite en vase clos, et publiant enfin leur dernier état calculé une fois le programme terminé. Tous les exemples de ce chapitre sont des programmes n'entrant pas dans cette catégorie. À vrai dire, les programmes œuvrant en vase clos ne sont que des cas particuliers de l'ensemble de tous les programmes que conçoit l'homme. On peut très bien concevoir des programmes qui soient difficilement prévisibles, si l'on prend garde à ne pas se restreindre à cette première catégorie.

Pour mieux appréhender cela, considérons-nous comme objet d'étude. Nous observons le monde au-travers de nos pigments et de nos photo-récepteurs. Ou, d'une manière plus générale, de nos cinq sens, qui retournent des valeurs discrètes si l'on considère être constitué d'atomes. Notre cerveau est constitué d'un nombre fini de neurones, et nous sommes un nombre fini d'atomes. Bref, nous pouvons imaginer de

4. Nous avons considéré cela dans nos réflexions sur les générateurs de nombres pseudo-aléatoires. En effet, le générateur proposé dans [BGW09], et étudié plus en détail dans [BGW10a], [WBGF10] et [BGW10b], publie, à chaque intervalle non régulier, son état interne vers la sortie du système, récupérant ainsi petit à petit, au fil des itérations du système, une suite de bits.

prime abord être une machine à états finis. Faut-il en conclure que chaque individu se mette à boucler en un temps fini ? Non, car à chaque instant, l'on interagit avec le monde extérieur. Et le monde n'est pas fini, bien qu'on l'observe et qu'on interagisse avec lui avec des ressources finies. Et c'est notamment de ces interactions entre notre finitude et le monde extérieur que naît nos comportements parfois si complexes, voire chaotiques chez certains.

2. Réponse au problème de l'utilisation des réels

Nous avons tout fait pour ne manipuler que des entiers bornés par une valeur N. Ce problème ne nous concerne donc plus.

3. Réponse au problème de la partie et du tout

Nous avons étudié divers aspects de chaos pour les itérations chaotiques, et en avons déduit un comportement riche et intéressant. Le but consistera dans ce qui suit à voir ces itérations chaotiques comme des programmes : considérer que les IC sont des fonctions de hachage, ou des générateurs de nombres pseudo-aléatoires, ou encore des algorithmes de tatouage, *etc.* C'est-à-dire résumer, le plus possible, l'algorithme à des itérations chaotiques. Nous donnons ci-dessous deux exemples, afin de bien comprendre de quoi il retourne.

a. Les IC vues comme des générateurs de nombres pseudo-aléatoires

Un exemple simple de générateur de nombres pseudo-aléatoires (PRNG) peut être le suivant :

- L'état initial est un vecteur de taille 256, construit à partir d'une graine.
- La stratégie est obtenue à l'aide d'un autre PRNG aux bonnes propriétés statistiques, ou à l'aide de mesures matérielles.
- La fonction d'itérations est la négation vectorielle.

On publie alors vers la sortie la suite des états obtenus, ou une de ses sous-suites. Le pari est alors de préserver les bonnes propriétés statistiques du PRNG utilisé pour constituer la stratégie, et d'obtenir de surcroît un programme au comportement chaotique. On a prouvé dans [BGW09], [BGW10a], [WBGF10], et [BGW10b] qu'un tel pari peut se gagner en faisant intervenir la graine, de taille non fixée, à chaque itération dans les IC.

b. Les IC vues comme des fonctions de hachage

Un autre exemple simple d'utilisation directe des itérations chaotiques est la constitution de fonctions de hachage. On peut par exemple imaginer que :

- L'état initial est le vecteur nul de taille 256.
- La stratégie est obtenue en regroupant 8 par 8 les bits de l'hôte.
- La fonction d'itérations est la négation vectorielle.

On constate dans l'exemple ci-dessus que l'algorithme de hachage est exactement réduit à la réalisation d'itérations chaotiques. Il est donc chaotique. De plus, la solution évoquée à la section 18.2.1 permettant d'éviter de tomber dans le cadre d'une machine à états finis a pu être mise en place : l'hôte est découvert par petits bouts, à chaque itérée. Il reste cependant à discuter du bon choix de l'obtention de la condition initiale (état et stratégie) à partir du média considéré. Une telle discussion est menée à la partie V.

4. Bilan : la démarche que nous proposons d'adopter

Nous proposons de concevoir des programmes résumés le plus possible aux itérations chaotiques, et tels qu'à chaque nouvelle itération, la nouvelle valeur de la stratégie soit calculée à partir du monde extérieur. Le nombre d'itérations ne doit pas être fixe, il doit dépendre du média considéré. Enfin, la fonction d'itération ne doit manipuler que des entiers, tout en appartenant à C.

Notons pour finir que les itérations chaotiques ne sont sûrement pas les seuls outils permettant d'atteindre nos objectifs. Il existe sûrement d'autres manières de concevoir des programmes au comportement imprévisible. Il faut juste bien veiller à répondre à chacun des problèmes énoncés à la section 18.1.

Quatrième partie

Application aux techniques de la dissimulation d'information

CHAPITRE 19

La science de l'information dissimulée

> La chose la plus banale devient délicieuse dès l'instant qu'on la dissimule.
>
> *Le portrait de Dorian Gray*
> OSCAR WILDE

On présente dans ce chapitre introductif les tenants et les aboutissants des techniques de dissimulation de l'information, que ce soit dans les domaines du tatouage numérique ou de la stéganographie. Des exemples d'applications sont donnés, et les notions de robustesse et de sécurité sont introduites. Les définitions utiles à la compréhension de la suite de cette partie sont elles aussi rappelées.

On dresse ensuite un état de l'art de la sécurité pour la dissimulation de l'information. On rappelle, ce faisant, quelques d'outils et contextes classiques dans ce domaine, et l'on explique enfin quelle est notre contribution.

I. INTRODUCTION

Nous appellerons *information dissimulée* (« information hiding ») toute étude sur les techniques visant à insérer, de manière discrète, une quelconque information au sein d'un contenant donné. Cette information dissimulée regroupe des domaines variés (stéganographie et stéganalyse, tatouage numérique, fingerprinting, *etc.*), dont certains seront définis ci-dessous, car étudiés par la suite.

1. Hôte et filigrane

L'information dissimulée consistant à cacher un message dans un autre, il faut avant toutes choses commencer par donner des noms à ces deux types de messages. Nous ne formulerons pas ici de définitions précises, nous souhaitons juste introduire qualitativement les objets de notre étude. Des définitions rigoureuses seront données au chapitre 22.

DÉFINITION IV.1 : Supposons que l'on camoufle un message dans un support donné, par une quelconque technique d'information dissimulée.

- Le support recevant le message s'appelle l'*hôte*, ou encore la *couverture*.

- Le message caché est aussi appelé la *marque*, le *filigrane*, ou le *tatouage*. ◊

2. Stéganographie

Nous donnons ci-dessous la définition de la stéganographie proposé par C. Cachin [Cac98], qui est en quelque sorte le père de la stéganalyse :

DÉFINITION IV.2 (STÉGANOGRAPHIE) : La stéganographie est l'art et la science de communiquer de telle sorte que la présence d'un message ne peut pas être détectée. ◊

En d'autres termes, la *stéganographie* est la technique consistant à cacher des messages de sorte que, en dehors de l'expéditeur et du destinataire, nul ne puisse en soupçonner l'existence. L'objet de la stéganographie est donc de faire passer inaperçu un message dans un autre, et non de rendre un message uniquement intelligible à qui-de-droit – ce qui est le rôle de la cryptographie. Cette technologie, quoique très ancienne, connaît un regain d'intérêt à l'air du numérique [Fil08].

Pour qu'une technique de stéganographie soit viable, il faut bien sûr que les adversaires ne puissent découvrir le contenu caché. Mais cela ne suffit pas : pour bien faire, il faudrait que nos adversaires ne puissent même pas savoir qu'un contenu est caché, et qu'ils ne puissent pas empêcher sa transmission (volontairement ou non). Enfin, ils ne devraient pas être capable d'envoyer une fausse information, en se faisant passer pour nous et en utilisant notre canal caché. La fin de ce chapitre et le suivant donneront plus de détails concernant ces pré-requis.

3. Tatouage numérique

a. Définitions

Le tatouage numérique est une discipline assez récente, qui serait née au début des années 1990 avec les articles de Tanaka *et al.* [TNM90] et de Caronni, Tirkel *et al.* [KP00]. Le terme *digital watermark* (tatouage numérique) fut pour la première fois employé en 1992 par Andrew Tirkel et Charles Osborne [Van93]. Enfin, Ingemar Cox [CMK+97] popularisa une de ses techniques les plus célèbres : l'étalement de spectre, que l'on étudiera plus loin. Précisons maintenant ce terme de tatouage, en reprenant la définition de Teddy Furon [Fur05] :

DÉFINITION IV.3 (TATOUAGE NUMÉRIQUE) : Le *tatouage numérique* est l'art de cacher des métadonnées dans du contenu numérique de manière robuste. ◊

REMARQUE. Le terme *marquage* est synonyme de tatouage. L'anglicisme *watermarking* est, lui aussi, fréquemment utilisé.

Cette définition en entraîne une autre, celle de robustesse. La définition de Kalker [Kal01] fait autorité [Fur05, PFCTPPG06] :

DÉFINITION IV.4 (ROBUSTESSE (TATOUAGE)) : Le *tatouage robuste* est le mécanisme consistant à créer un canal de communication multiplexé dans le contenu original, et dont la capacité se dégrade comme une fonction continue de la dégradation du contenu tatoué. ◊

T. Furon s'empresse d'ajouter que le terme « cacher » a plusieurs sens, suivant les auteurs. Il signifie tantôt que l'embarquement des métadonnées ne cause aucune distorsion perceptible, et tantôt sous-entend « d'une manière sûre » [Fur05]. Cette « sécurité » dans le tatouage a été formulée d'une manière concise par Kalker [Kal01] :

DÉFINITION IV.5 (SÉCURITÉ (TATOUAGE)) : La sécurité se réfère à l'incapacité qu'on les utilisateurs non autorisés d'avoir accès au canal de tatouage. Un tel accès fait référence à la tentative de supprimer, détecter et estimer, écrire et modifier les bits de tatouage. ♢

Cette définition, qui fait autorité [Fur05, PFCTPPG06], est cependant trop générale et pas assez technique pour être utilisée en théorie ou en pratique. Ainsi, comme nous le verrons dans la prochaine section, des définitions plus mathématiques ont été proposées ces dix dernières années pour donner plus de consistance à cette notion. Avant cela, nous tenterons d'expliquer plus concrètement la différence entre la robustesse et la sécurité.

b. Robustesse et sécurité

Le tatouage contenu dans un média peut être soumis à deux grandes familles d'attaques, ce qui conduit à la distinction entre les notions de robustesse et de sécurité.

i. Les menaces. Les premières menaces, dites *aveugles*, consistent en la tentative de suppression de l'information dissimulée en utilisant la force plutôt que l'intelligence. Cela revient à enlever le tatouage de manière agressive et/ou involontaire, majoritairement en déformant l'hôte. Par exemple, on peut « espérer » enlever le copyright d'une photographie en lui faisant subir des rotations, des découpages, des compressions, *etc.* Ces attaques sont aveugles dans la mesure où elles s'appliquent sans discernement à tout tatouage, et ne cherchent pas à tirer profit d'une quelconque connaissance supplémentaire (par exemple, connaissance de l'algorithme utilisé, ou possession de plusieurs images tatouées). Les schémas de tatouage résistant à ce genre d'attaques sont dits *robustes*.

Il existe cependant d'autres manières d'attaquer un support contenant de l'information dissimulée. Ces attaques, forcément volontaires, sont plus fines, utilisent toute source d'information supplémentaire mises à leur disposition, et sont bien plus difficiles à prévenir. Elles sont détaillées dans la section suivante.

ii. Robustesse versus sécurité. Initialement, lorsqu'il s'agissait de savoir si une technique d'information dissimulée donnée était apte à faire face à diverses attaques d'un adversaire, on se contentait d'évaluer la robustesse de ladite technique [AKS06, BCGG99]. En d'autres termes, on altérait le support hôte de différentes manières, et l'on regardait si la marque restait malgré ces modifications. Ces études ne prenaient pas en compte le caractère malicieux, l'intelligence, les moyens et la détermination de l'attaquant. Dernièrement, la notion de *sécurité* est apparue en stéganographie [Cac98, Mit99], et s'est ensuite développée à toute l'information dissimulée [Fur05, PFCTPPG06, PFPGFC06], comme nous le verrons dans la section 19.3. Elle prend en considération toutes les catégories possibles d'attaques.

La sécurité et la robustesse sont des concepts relativement voisins, leurs définitions faisant encore débat [PFCTPPG06]. De prime abord, on peut cependant considérer que la sécurité comprend la robustesse et les attaques intentionnelles [CPFPG05a, Kal01]. Les tentatives de définir les différences entre la robustesse et la sécurité, de clarifier les classes d'attaques, et de donner quelque consistance à la notion de sécurité, illustrent le besoin pressant d'un cadre théorique rigoureux et complet pour l'étude des attaques auxquelles peuvent faire face les techniques d'information dissimulée. Nous allons maintenant rappeler ces différentes notions de sécurité.

II. Quelques rappels préliminaires

Il nous faut commencer par introduire quelques outils issus de la théorie de l'information et des probabilités, afin d'être en mesure de dresser l'état de l'art de la sécurité pour la dissimulation d'information.

1. Le principe de Kerckhoffs

Le principe de Kerckhoffs, qui a été formulé par Auguste Kerckhoffs à la fin XIXe siècle [Ker83], énonce le fait suivant :

Définition IV.6 (Le principe de Kerckhoffs) : La sécurité d'un cryptosystème ne doit reposer que sur le secret de la clef. ◊

Autrement dit, tous les autres paramètres doivent être supposés publiquement connus. Il a été reformulé, peut-être indépendamment, par Claude Shannon : « l'adversaire connaît le système ». Cette autre formulation est connue sous le nom de *maxime de Shannon*. C'est un principe raisonnable, celui le plus souvent adopté par les cryptologues, par opposition à la sécurité par l'obscurité, qui n'est sérieusement envisageable que dans le cadre de l'armée.

2. La divergence de Kullback-Leibler

La divergence de Kullback-Leibler, encore appelée *entropie relative*, est une bonne mesure de dissimilarité entre deux distributions de probabilités P et Q. Elle est définie de la manière suivante :

Définition IV.7 (Divergence de Kullback-Leibler) : Pour deux distributions de probabilités discrètes P et Q, la *divergence de Kullback–Leibler* de Q par rapport à P est définie par :

$$D_{\mathrm{KL}}(P|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}.$$

On l'utilise habituellement de la manière suivante :

- P représente les vraies valeurs : les données, les observations, ou une distribution de probabilité calculée avec précision.
- Q est une théorie, un modèle, une description ou une approximation de P.

Cette divergence est une grandeur positive, qui n'est nulle que quand $P = Q$.

3. L'information de Fisher

L'information de Fisher est une notion de statistique introduite par R.A. Fisher, qui quantifie l'information relative à un paramètre contenu dans une distribution.

Soit $f(x, \theta)$ la distribution de vraisemblance d'une grandeur x, paramétrée par θ. La « technique d'estimation de θ par le maximum de vraisemblance », introduite par Fisher, consiste à choisir la valeur maximisant la vraisemblance des observations X :

$$\mathbf{E}\left[\frac{\partial \log f(X;\theta)}{\partial \theta}|\theta\right] = 0.$$

où $\mathbf{E}$ désigne l'espérance mathématique. L'information de Fisher est quant à elle définie à partir de la variance associée à ce maximum :

DÉFINITION IV.8 (INFORMATION DE FISHER) : L'information de Fisher est la grandeur :

$$I(\theta) = \mathbf{E}\left[\left(\frac{\partial \log f(X;\theta)}{\partial \theta}\right)^2 |\theta\right].$$

4. Entropie de Shannon

L'entropie de Shannon correspond à la quantité d'information contenue ou délivrée par une source d'information à un récepteur : pour ce dernier, plus la source émet d'informations différentes non reçues, plus l'entropie (ou incertitude sur ce qui est émis) est grande. Inversement, plus le récepteur reçoit d'informations sur le message transmis, plus l'entropie diminue :

DÉFINITION IV.9 (ENTROPIE DE SHANNON) : Pour une variable aléatoire discrète X comportant n symboles (*i.e.* n réalisations possibles pour la variable aléatoire X), un symbole i ayant une probabilité $P(i)$ d'apparaître, l'*entropie de Shannon* h (en bits) de la source X est définie par :

$$h(X) = -\mathbf{E}[\log_2 P(X = x_i)] = -\sum_{i=1}^{n} P(i) \log_2 P(i).$$

$h(X)$ est positive, nulle pour une distribution résumée en un point, et maximale pour une distribution uniforme, *i.e.* quand tous les symboles sont équiprobables. Enfin, elle augmente avec le nombre d'états possibles, et diminue avec la redondance de la source.

5. L'information mutuelle

L'information mutuelle de deux variables aléatoires est une quantité mesurant (en bits) la dépendance statistique de ces variables, leur degré de dépendance au sens probabiliste.

DÉFINITION IV.10 (INFORMATION MUTUELLE) : Soit (X, Y) un couple de variables aléatoires de distributions marginales $P(X = x)$, $P(Y = y)$, et de densité de probabilité jointe $P(X = x, Y = y)$.

L'*information mutuelle* est égale à :

$$I(X, Y) = \sum_{x,y} P(X = x, Y = y) \log \frac{P(X = x, Y = y)}{P(X = x)\, P(Y = y)}.$$

L'information mutuelle est nulle si et seulement si les variables sont indépendantes (*i.e.* si la réalisation de l'une n'apporte aucune information sur la réalisation de l'autre), et croît lorsque la dépendance augmente. Elle généralise la notion de *corrélation*, plus connue, qui se restreint à certaines dépendances (strictement monotones).

Notons que l'information mutuelle peut aussi être exprimée par la divergence de Kullback-Leibler : $I(X, Y)$ mesure une sorte de « distance » entre les distributions $P(X, Y)$ et $P(X) \times P(Y)$, alors que la divergence de Kullback-Leibler donne le nombre de bits d'information apportés par la connaissance de $P(X, Y)$ lorsque l'on connaît déjà $P(X)$ et $P(Y)$.

III. Sécurité dans la dissimulation : état de l'art

1. Cadre : le problème du prisonnier de Simmons

Dans le problème dit *du prisonnier de Simmons* [Sim84], Alice et Bob sont en prison, et veulent concevoir un plan d'évasion en échangeant des messages cachés (filigranes) dans des documents hôtes d'allure innocente. Les documents sont transmis par une gardienne, Eve, qui les étudie, et peut choisir d'interrompre la communication quand elle le veut.

2. Sécurité en stéganographie

a. Approche de Cachin

La stéganographie est historiquement la première discipline à avoir reçue une théorie de sécurité, basée sur l'idée suivante (C.Cachin, fin des années 90 [Cac98]). Alice envoie à Bob un contenu C. Le but d'Eve est de déterminer si ce contenu est le support d'une communication cachée entre Alice et Bob. Nous avons affaire ici à un test d'hypothèse [Fur05] [5] :

- soit C contient une information cachée,
- soit C est une image naturelle.

On peut évaluer la performance de ce test en mesurant :

- la probabilité P_{fa} d'une fausse alarme, *i.e.* la probabilité d'accuser à tord Alice,
- la puissance du test P_p, *i.e.* la probabilité d'accuser Alice avec raison.

Le test d'hypothèse effectué par l'attaquant est efficace quand $P_{fa} \approx 0$ et $P_p \approx 1$. Cependant, la performance de ce test est limitée par la capacité de l'attaquant de pouvoir distinguer les lois de X et de Y, ce qui se formule de la manière suivante :

$$D_{\mathrm{KL}}(p_X|p_Y) \geqslant D_{\mathrm{KL}}(P_{fa}|P_p) \geqslant 0.$$

Ainsi, si Alice arrive à produire un contenu Y ayant une information cachée proche de X, *i.e.* si les densités p_X, p_Y de probabilité de X et Y sont égales presque-partout, alors $D_{\mathrm{KL}}(p_X|p_Y) = 0$, et donc $P_{fa} = P_p$, ce qui rend les tests de l'attaquant inutiles. Il y a donc *sécurité inconditionnelle*, au sens de Cachin et pour les raisons évoquées ci-dessus, quand $D_{\mathrm{KL}}(p_X|p_Y) = 0$.

b. Approche de Mittelholzer

Mittelholzer [Mit99] est à l'origine de la seconde tentative de définir la sécurité d'une méthode de stéganographie [Fur05]. Inspiré par les travaux de Cachin, il considère que la stéganographie est sûre lorsque, en l'absence de la clé secrète, le contenu marqué Y n'est source d'aucune fuite d'information concernant le message caché M. Cela se formalise de la manière suivante : la *stéganographie parfaite* est obtenue pour $I(M;Y) = 0$, où I est l'information mutuelle de la définition IV.10.

3. Sécurité et tatouage numérique

Ces efforts pour apporter un cadre théorique pour la sécurité dans la stéganographie se sont traduits ensuite dans le cadre du tatouage numérique, de la manière suivante [Fur05].

5. Démarche consistant à rejeter une hypothèse statistique, appelée hypothèse nulle, en fonction d'un jeu de données (échantillon).

On suppose que l'adversaire connaît la technique de tatouage, et qu'il ne lui manque que la clé secrète (principe de Kerckhoff). L'idée à la base de la notion de sécurité dans le tatouage numérique consiste à considérer que des pirates, observant un contenu tatoué, peuvent éventuellement déduire certaines informations concernant la clé secrète. Cette fuite d'information est certainement très faible, mais si ces pirates peuvent observer un grand nombre de contenus tatoués avec la même clé, alors la quantité d'information qu'ils posséderont concernant la clé secrète risque d'augmenter. Ainsi, à partir d'un certain nombre de contenus tatoués avec la même clé, la quantité d'information obtenue par les pirates peut être suffisante pour leur permettre de réaliser des attaques contre la clé secrète. Ce nombre peut se trouver à l'aide d'outils issus des travaux de Shannon ; il fixe une frontière au-delà de laquelle le schéma n'est plus sûr. Cette approche pour la sécurité des algorithmes de tatouage numérique est simplement l'adaptation de la théorie de Shannon, utilisée depuis longtemps en cryptanalyse, au domaine du watermarking.

En pratique, la fuite d'information à la base de cette notion de sécurité se calcule à l'aide des outils suivants, rappelés dans la section 19.2 : information mutuelle, information de Fisher, divergence de Kullback Leibler, *etc.* D'un point de vue théorique, chaque outil a ses avantages et ses inconvénients, qui les rendent d'égal intérêt. L'interprétation pratique n'est cependant pas la même, suivant l'outil utilisé.

- Considérons par exemple le cas de l'information mutuelle. Soit $h(k)$ l'entropie de Shannon mesurant l'ignorance que le pirate a de la clé secrète avant son observation, $h(K|Y^{N_0})$ celle qu'il a après ses observations de N_0 contenus tatoués : $Y^{N_0} = \{Y_1, \ldots, Y_{N_0}\}$. On peut établir que
$$h(K|Y^{N_0}) = h(k) - I(K; Y^{N_0}).$$
Ainsi, s'il y a fuite d'information ($I(K; Y^{N_0}) \geqslant 0$), alors le pirate a acquis de la connaissance sur la clé secrète, il a réduit son ignorance. Dans le cas contraire ($I(K; Y^{N_0}) = 0$), la technique de tatouage est dite *complètement sûre*.
- Dans le cas de l'information de Fisher, l'inégalité de Cramer-Rao établit que, quel que soit l'estimateur non biaisé $\check{K}$ de la clé secrète K, la meilleure précision d'estimation atteignable est majorée par l'information de Fisher $I(K, Y^{N_0})$. Ainsi, plus la « fuite d'information » ($I(K, Y^{N_0})$) est petite, plus les estimation de la clé secrète seront mauvaises.

4. L'impact du contexte

La détermination du niveau de sécurité d'une technique de tatouage dépend du contexte dans lequel cette dernière est sensée œuvrer, des catégories d'attaques auxquelles elle est sensée résister.

Ainsi, les efforts évoqués ci-dessus pour apporter un cadre théorique à l'étude de la sécurité d'un schéma de tatouage ont été suivis par Kalker [Kal01] qui tenta de clarifier les concepts (robustesse *versus* sécurité, *c.f.* définitions IV.4 et IV.5), et de classifier les attaques sur le tatouage. Ce travail a été approfondi par Furon *et al.* [Fur02], qui ont transposés le principe de Kerckhoffs de la cryptographie vers la dissimulation d'information. Ils ont utilisé la méthodologie de Diffie et Hellman, et la théorie de Shannon [Sha49], pour classer les attaques sur le tatouage en diverses catégories, selon le type d'information auquel l'adversaire a accès [CFF05], [PFPgC06] :

Attaque de l'objet tatoué seul. L'adversaire n'a accès qu'à des contenus tatoués pour réaliser ses attaques. Noté, dans la littérature, WOA : *Watermarked Only Attack*.

Attaque du message connu. L'adversaire a accès à des couples de contenus tatoués avec leurs messages cachés correspondants. Noté KMA : *Known Message Attack*.

Attaque de l'original connu. L'adversaire a accès à des couples de contenus tatoués avec leurs messages originaux. Noté KOA, pour *Known Original Attack*.

Attaque du message constant. L'adversaire observe plusieurs contenus tatoués, et sait seulement que le message caché est le même dans tous les contenus. Noté CMA : *Constant-Message Attack*.

Cette prise en considération du contexte revient à remplacer, dans ce qui précède, Y^{N_0} par O^{N_0}, où $O = Y$ dans le WOA, $O = \{Y, X\}$ dans le KOA et $O = \{Y, M\}$ dans le KMA. Cette manière de procéder permet de définir d'autres contextes, tels que l'attaque de l'original estimé (Estimated Original Attack, EOA), pour lequel $O = \{Y, \check{X}\}$.

5. Sécurité de l'information dissimulée

François Cayre et Patrick Bas ont ensuite proposé dans [CB08] une approche de la sécurité basée sur la théorie des probabilités, dans le contexte WOA, valable à la fois pour la stéganographie et le tatouage numérique. Le cadre est une fois encore le problème du prisonnier de Simmons.

a. Notations préliminaires

Commençons par introduire quelques notations dans le cadre du problème du prisonnier [CB08] :

- N_c est la taille disponible (en bits) dans les vecteurs hôtes, pour y cacher des messages, N_v est la taille du vecteur hôte, et N_0 est le nombre de contenus observés.
- $\mathbf{X}$ est un ensemble de vecteurs représentant une collection de N_0 contenus originaux (hôtes), chaque élément x de $\mathbf{X}$ étant un vecteur tiré aléatoirement dans un espace E de dimension N_v.
- $\mathbf{Y}$ est un ensemble de vecteurs représentant une collection de N_0 contenus tatoués, chaque élément y de $\mathbf{Y}$ étant un vecteur aléatoire à N_v coordonnées.
- $\mathbb{K} \subset \mathbb{R}$ est un ensemble fini de taille N_K, dit « ensemble des clés ».
- Enfin, $e : \mathbb{K} \times E \to E$ représente une fonction de tatouage.

Le respect du principe de Kerckhoffs [Ker83] appliqué à la fonction de tatouage utilisée par Alice et Bob, permet de supposer qu'Alice et Eve peuvent toutes deux construire une estimation parfaite des différentes lois de probabilités entrant ici en jeu : celle des contenus originaux, celle des contenus tatoués, *etc.* Listons plus précisément ces lois de probabilités [CB08] :

- $p(X)$ est la loi de probabilité de N_0 contenus originaux (les hôtes). Bien que nous nous plaçons dans la configuration WOA, nous supposons quand même Eve capable de modéliser la loi de X : $p(X) = p(x_0, \ldots, x_{N_0-1})$. Cette hypothèse est un cas le pire, du point de vue d'Alice et de Bob, pour lequel *Eve a pu estimer quelle pouvait être la loi des contenus originaux, rien qu'en observant attentivement les contenus tatoués qu'elle transporte*.
- $p(Y)$ est la loi de probabilité de N_0 contenus tatoués, chaque contenu l'ayant été avec une clé différente. Cette loi peut être obtenue par Eve en utilisant suffisamment la fonction de tatouage qu'elle a à sa disposition (principe de Kerckhoffs).
- $p(Y_K)$ est la loi de probabilité de N_0 contenus tatoués, chaque contenu l'ayant été avec une seule et unique clé secrète $K \in \mathbb{K}$. Ce modèle peut être construit par Eve, simplement en observant les contenus qu'elle transporte, vu que l'*on suppose qu'Alice et Bob ne changent pas leur clé secrète K au cours de leurs échanges*.

- $p(Y|K_i)$ est la loi de probabilité de N_0 contenus retatoués, chaque contenu tatoué (par Alice et Bob) ayant été retatoué par Eve en utilisant la même clé connue $K_i \in \mathbb{K}$. Ce modèle peut être construit par Eve, car l'on suppose le principe de Kerckhoffs respecté : l'adversaire a accès à la fonction de tatouage e, et peut retatouer, avec sa propre clé K_i, des messages tirés aléatoirement dans une collection Y de contenus déjà tatoués.

Comme les supports hôtes sont supposés indépendants les uns des autres, les lois précédentes sont en fait des lois marginales : $p(X) = p(x_0) \times \ldots \times p(x_{N_0-1})$. La même chose vaut pour $p(Y)$ et $p(Y|K)$.

b. Quatre notions de sécurité

Le but d'Eve est de trouver la constante K_e qui maximise la probabilité $p(Y_K|K_e)$, c'est-à-dire la clé K_e qui a été la plus sûrement utilisée pour tatouer les Y_K.

Une des originalités de l'approche de Cayre et Bas consiste à considérer que la geôlière peut être passive ou *active* (Eve peut détecter la présence d'un message caché, estimer ce dernier, puis agir éventuellement sur la communication). Cette considération conduit les auteurs de [CB08] à considérer quatre niveaux de sécurité dans le WOA, le plus faible niveau étant l'insécurité :

DÉFINITION IV.11 (INSÉCURITÉ) : La fonction de marquage e est *non-sûre* (insecure) si et seulement si $\exists K_1 \in \mathbb{K}, p(Y|K_1) = p(Y_K)$, et $\forall K_2 \in \mathbb{K}, p(Y|K_2) \neq p(Y_K)$. ◊

Dans ce cas de figure, il existe une unique clé K_1 dont le modèle associé $p(Y|K_1)$ des contenus tatoués avec cette clé, correspond exactement au modèle $p(Y_K)$ des observations. Ainsi, la méthode consistant à rechercher, par exemple de manière exhaustive, la clé aboutissant à la meilleure correspondance, a des chances sérieuses d'aboutir : l'estimation de K_e est possible [CB08].

La deuxième définition introduite par Cayre et Bas est la *key-security* :

DÉFINITION IV.12 (KEY-SECURITY) : La fonction de marquage e est *key-secure* si et seulement si $\exists \mathcal{S}_K \subset \mathbb{K}, card(\mathcal{S}_K) > 1, \forall K_1 \in \mathcal{S}_K, p(Y|K_1) = p(Y_K)$. ◊

Dans la définition précédente, $K \in \mathcal{S}_K$, et $\mathcal{S}_K$ correspond à l'ensemble des clés qui ne modifient pas le modèle probabiliste des observations. Si le schéma est non-sûr, alors un tel S_K n'existe pas. On peut constater aussi que, dans ce cas, quand bien même il est impossible d'estimer la clé secrète, il est cependant possible de trouver $\mathcal{S}_K$, ce qui est un risque (bien qu'un attaquant ne puisse pas retrouver K à partir de $\mathcal{S}_K$ [CB08]).

Dans ce cas de figure, la sécurité d'un schéma utilisant la clé privé K repose sur la taille de $\mathcal{S}_K$. Cette propriété de sécurité est aussi reliée, dans une certaine mesure, à la robustesse, dans le sens où une telle sécurité autorise une faible distorsion du support. Cette classe est le niveau le plus élémentaire de sécurité selon Cayre et Bas [CB08], au moins lorsque l'on ne désire pas permettre d'accès non autorisé en lecture/écriture dans le canal secret d'échange (Eve est passive).

Dans le cas particulier où $\mathcal{S}_K = \mathbb{K}$, on parle alors de « subspace-security ».

DÉFINITION IV.13 (SUBSPACE-SECURITY) : La fonction de marquage e est *subspace-secure* si et seulement si $\forall K_1 \in \mathbb{K}, p(Y|K_1) = p(Y_K)$. ◊

En cas de « subspace-security », Eve ne sera pas capable de distinguer la bonne clé de toute autre clé, même en cas d'étude exhaustive. Il lui est donc impossible d'estimer le sous-espace $\mathcal{S}_K$ associé à la clé secrète K, car Y et K sont indépendants. La subspace-security entraîne la key-security, et conduit au fait qu'il n'y a aucune fuite d'information entre les contenus tatoués et la clé secrète :

$$\text{subspace-security} \iff I(Y_K, K) = 0.$$

En effet, dans ce cas de figure, la clé secrète est équivalente à toute autre clé, et donc Eve ne peut rien déduire de ses observations.

D'un autre côté, si le schéma considéré est key-secure, mais pas subspace-secure, alors Eve pourra estimer, à partir d'un nombre suffisant d'observations, le sous-espace $\mathcal{S}_K$, mais pas K. Elle pourra alors concentrer tous ses efforts sur $\mathcal{S}_K$.

Enfin, la *stégo-sécurité* est ce qui peut se produire de pire pour Eve :

Définition IV.14 (Stégo-sécurité) : La fonction de tatouage e est *stégo-sûre* (stego-secure) si et seulement si $\forall K_1 \in \mathbb{K}, p(Y|K_1) = p(X)$. ◇

La stégo-sécurité signifie notamment que la connaissance de K n'aidera pas à faire la différence entre $p(X)$ et $p(Y)$. Cette définition entraîne la propriété suivante :

$$p(Y|K_1) = \cdots = p(Y|K_{N_k}) = p(Y) = p(X).$$

On peut montrer que cette propriété est équivalente à une divergence de Kullback-Leibler nulle, ce qui est la définition du *secret parfait* proposée par Cachin [Cac98].

Proposition IV.1 : *Stégo-sécurité* $\Longrightarrow$ *Secret parfait* $(D_{KL}(p(Y)|p(X)) = 0)$.

Dans ce cas de figure, il est impossible pour Eve de savoir si le contenu qu'elle étudie est passé par la fonction de tatouage, ou pas : elle ne peut tirer aucune information à partir des contenus transmis.

Signalons enfin que si le schéma considéré est subspace-secure, mais pas stégo-sûr, alors Eve, qui ne pourra estimer ni K, ni $\mathcal{S}_K$, pourra quand même distinguer les contenus innocents des contenus marqués. En d'autres termes, d'un point de vue stéganalytique, le schéma ne serait pas sûr, car la fonction d'embarquement ne respecterait pas le secret parfait de Cachin. Par contre, le schéma sera sûr du point de vue du tatouage numérique.

IV. Contribution à la sécurité de l'information dissimulée

Les études de sécurité présentés dans la section précédente ne peuvent avoir lieu que dans la configuration WOA, et ont forcément pour cadre le problème des prisonniers de Simmons [Sim84]. De plus, il s'agit dans ces études d'un cadre particulier de ce problème : les messages véhiculés par Eve ont été tatoués avec une seule et unique clé secrète pour tous les messages. Ce cadre est relativement restrictif, comme l'ont indiqué Cayre et Bas eux-mêmes [CB08] : « Comme dans les autres travaux de ce genre, nous considérons qu'Alice et Bob n'utilisent qu'une seule clé. Bien sûr, dans les applications réelles, en particulier dans la stéganographie, il est hautement souhaitable de modifier la clé à chaque communication entre Alice et Bob. »

En outre, l'existence d'une loi de probabilité pour la couverture est nécessaire et, comme l'a déclaré Cachin [Cac98], « Supposer l'existence d'une loi de probabilité pour la couverture semble rendre notre modèle peu réaliste en pratique » : il n'y a pas de modèle canonique pour la couverture, qui puisse être utilisé dans tous les scénarios. Cette existence même peut faire défaut, les couvertures n'ayant aucune nécessité à se plier à pareille exigence. Enfin, Alice et Bob peuvent chercher soit à induire en erreur Eve, l'incitant par un quelconque moyen à supposer une mauvaise loi de probabilité, soit s'intéresser à des ensembles de couvertures tels qu'Eve ne puisse jamais déterminer $p(\mathbf{X})$.

Ces remarques n'invalident pas les méthodes probabilistes pour autant, et prouver qu'une telle chose est insensée et n'est pas notre propos, et que la plupart de ces problèmes peuvent être contournés en s'y prenant bien. Mais il est indéniable qu'un certain nombre de questions se posent, d'ordre théorique et pratique, qui ne sont pas toujours faciles à résoudre.

Pour y répondre, nous avons proposé un nouveau cadre théorique pour la sécurité dans la dissimulation d'informations, afin d'être en mesure d'étudier les configurations KMA, KOA, et CMA. Un algorithme de dissimulation d'informations est ici considéré comme une machine dont le mécanisme est public. Cette machine reçoit le message à cacher, la clé secrète et le support hôte à utiliser, et renvoie le contenu tatoué. Pour nous, la sécurité du système dépend du comportement imprévisible de la machine : il y a une faille de sécurité qui nous semble incontestable si un adversaire est capable de prédire les lieux où le filigrane peut se trouver, c'est-à-dire prédire l'image de la machine pour une entrée quelconque.

Pour donner plus de consistance à la notion d'imprévisibilité, cette machine est modélisée sous la forme d'un système dynamique discret :

$$x^0 \in \mathcal{X}, x^{n+1} = f(x^n).$$

Cette reformulation est toujours possible, comme on le prouvera dans la section 20.1. On peut donc alors relier l'imprévisibilité du schéma de tatouage à certains aspects topologiques de sa fonction associée f, aspects issus de la théorie mathématique du chaos rappelée dans la partie II.

Ce nouveau cadre théorique pour une certaine approche de la sécurité pour la dissimulation d'information respecte le principe de Kerckhoffs. Il est basé sur une description topologique, alors que la plupart des études dans ce domaine ont généralement utilisé la théorie des probabilités [PFCTPPG06, Fur05]. Le but de cette recherche est avant tout de combler l'absence de notion de sécurité dans les configurations CMA, KOA et KMA. Accessoirement, on obtiendra de ce fait un outil supplémentaire permettant d'évaluer la sécurité dans le cadre WOA, ce qui ne nous semble pas sans intérêt : ne peut-on penser qu'en matière de sécurité, plus grand est le nombre, la variété et la différence de points de vue, mieux c'est ?

Ainsi, contrairement aux modèles proposés dans la section précédente la chaos-sécurité, qui sera définie au chapitre suivant, peut être utilisée dans les configurations KOA, KMA et CMA. De plus, dans le cas particulier des algorithmes de tatouage basés sur le chaos, on sera en mesure de vérifier si l'affirmation d'un comportement chaotique tient ou non : une telle vérification semble naturelle, de même qu'évaluer la force de ce comportement chaotique semble intéressant quand on commence une étude de sécurité d'un algorithme supposé tel.

CHAPITRE 20

La chaos-sécurité pour la dissimulation

> Rappelons à ce propos l'histoire de ce soldat allemand qui, au milieu d'une bataille, reste immobile, debout, une ficelle de trente centimètres environ tenue entre le pouce et l'index des deux mains, et s'écrit : « Es reicht nicht, es reicht nicht ! » (Ça ne suffit pas, ça ne suffit pas !)
> Intrigués par ce manège insolite, ses camarades font appeler un infirmier, lequel en réfère au médecin capitaine qui adresse le soldat à la consultation psychiatrique. Le militaire est réformé est renvoyé à la vie civile. Quelques mois plus tard, ses camarades reçoivent une carte postale, expédiée de Suisse ; seuls quelques mots y figurent : « Und es hat doch gereicht » (Et pourtant, c'était suffisant).
>
> *La tentation nihiliste*
> ROLAND JACCARD

Nous allons considérer qu'un algorithme de dissimulation de données est chaos-sûr lorsque son comportement est imprévisible. La raison pour laquelle cette caractéristique est appropriée provient notamment du fait que, quand un système est prévisible, il peut être possible de découvrir au moins une partie du filigrane dans les configurations KOA, KMA, ou CMA, comme nous le montrerons ici.

Le domaine mathématique qui étudie l'imprévisibilité est la théorie du chaos (*c.f.* partie II), qui décrit le comportement d'un système dynamique en termes topologiques. On l'a vu, une des descriptions les plus réputées d'un tel comportement chaotique est celle de Devaney [Dev03]. Son intérêt pour le problème étudié, ainsi que la notion de chaos-sécurité qui en découle, seront détaillés dans ce qui suit.

I. LA CHAOS-SÉCURITÉ

1. Pertinence de la définition de Devaney

Dans notre point de vue, tout algorithme de dissimulation devrait au moins être chaotique selon Devaney : il serait alors aussi difficile pour un adversaire de retrouver le message caché et les coefficients modifiés après n itérations de l'algorithme de tatouage, que de prédire le comportement d'un système

chaotique sur une longue période. Et cela, on ne sait pas le faire : cette prévision devient impossible dans la pratique lorsque n augmente. Suivant Devaney (définition II.40), un tel algorithme de dissimulation satisfera donc les propriétés suivantes : sensibilité aux conditions initiales, régularité et transitivité.

a. Utilité de la sensibilité aux conditions initiales

La sensibilité aux conditions initiales est utile, entre autre, pour résister aux attaques dites *de sensibilité* [CPFPG05b]. Les attaques de sensibilité sont des attaques très puissantes, qui peuvent rendre illisible un message caché en ne faisant subir au support tatoué qu'une très faible distorsion. Différentes stratégies ont été présentées dans la littérature pour faire face à ce genre d'attaques, mais elles ne sont pas complètement satisfaisantes. La meilleure approche à ce jour est, à notre connaissance, celle proposée par Furon *et al.* [FB08]. Cependant, le score de 50,24 dB, obtenu par cette méthode lors d'un concours public, est jugée par certains trop faible. Nous pensons que la sensibilité aux conditions initiales pourrait permettre d'améliorer ce score : la constante de sensibilité pourrait être choisie de telle sorte que la distorsion finale, dans l'attaque de sensibilité, ne pourra pas être faible, devra forcément être telle que l'hôte en soit fortement détérioré.

La sensibilité aux conditions initiales nous sera utile aussi pour obtenir une bonne authentification des données lors d'un tatouage, lorsque cette dernière est nécessaire (voir notre définition IV.31). En effet, dans ce cas, l'insertion et l'extraction du filigrane seront fortement tributaires de l'information contenue dans le support hôte.

Enfin, signalons que pour obtenir un tatouage fragile, il suffit d'utiliser une méthode avec une grande constante de sensibilité.

b. Utilité de la transitivité

Quand l'algorithme utilisé pour le tatouage est transitif, Eve ne peut espérer supprimer le filigrane en découpant le média tatoué. En effet, le système visitera tout l'espace, de sorte que le filigrane sera réparti sur l'ensemble du média. On constate donc que la transitivité est liée à la robustesse, même si cette relation est difficile à quantifier, notamment en raison de l'absence de définition rigoureuse pour cette dernière. Ce lien est partiellement établi dans ce qui suit et au chapitre 23, en utilisant des exemples-jouets : annulation de pixels (zeroing attack), attaques par rotation, bruit blanc gaussien additif, et compression JPEG. Il a fait l'objet des publications [BG10a] et [BG10b].

De plus, grâce à la transitivité, on devrait pouvoir vérifier l'authenticité d'un bout de document tatoué que l'on ne posséderait pas en entier, vue que la marque se retrouve partout. Enfin, la transitivité participe encore à renforcer la sécurité au sens large, de la manière suivante : Eve ne peut espérer abaisser la complexité de l'opération de recherche du filigrane, en réduisant la taille de l'objet tatoué, c'est-à-dire en n'étudiant qu'une partie bien choisie de ce dernier.

c. Utilité de la régularité

La régularité, lorsqu'elle rencontre la transitivité, conduit à l'imprévisibilité. Cette dernière peut aider Alice et Bob à résister aux attaques de type KOA et KMA, vu qu'il serait alors impossible de déterminer quels coefficients ont été modifiés lors de l'insertion du filigrane, et de quelle manière ils l'ont été. Ce point est détaillé à la section 20.3.

Toutes ces propriétés participent à la sécurité de la machine de dissimulation d'information, ce qui rend à notre sens la notion de Devaney pertinente pour initier l'étude de sécurité telle que nous l'entendons. Cette approche originale de la sécurité est définie dans la section suivante.

2. La définition de la chaos-sécurité

Voyons maintenant plus en détail cette nouvelle notion de chaos-sécurité.

a. Formalisation préliminaire d'un processus itératif

Pour vérifier si un schéma de dissimulation d'information existant S est chaotique ou non, nous proposons tout d'abord de l'écrire comme un processus itératif :

$$x^{n+1} = f(x^n).$$

Il est possible de prouver que cette formulation peut toujours se faire, de la manière suivante [6].

Considérons un quelconque algorithme de dissimulation d'information. Vu qu'il doit être un jour programmé, il est donc nécessairement possible de le traduire sous la forme d'une machine de Turing. Or toute machine de Turing peut être mise sous la forme $x^{n+1} = f(x^n)$, de la manière suivante.

Soit (w, i, q) la configuration actuelle de la machine de Turing (Fig. 20.1), où :

- $w = \sharp^{-\omega} w(0) \dots w(k) \sharp^{\omega}$ 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.

On définit f par :

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

Ainsi la machine de Turing peut être écrite sous la forme d'un processus itératif $x^{n+1} = f(x^n)$ sur un ensemble bien défini $\mathcal{X}$, avec x^0 comme configuration initiale de la machine.

NOTATION IV.1. On note $\mathcal{T}(S)$ le processus itératif du schéma S.

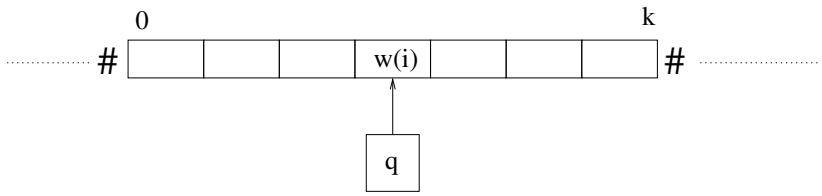


FIGURE 20.1 – Machine de Turing

6. Cette formulation a été rendue possible grâce à l'aide apportée par Pierre-Cyril Heam.

b. Chaos-sécurité : la définition

Soit τ une topologie sur l'ensemble $\mathcal{X}$ défini ci-dessus. Alors le comportement du système dynamique $(\mathcal{X}_\tau, f)$ peut être étudié, pour savoir si oui ou non le schéma associé est $\tau-$imprévisible. Ce qui nous mène à la définition suivante :

DÉFINITION IV.15 (CHAOS-SÉCURITÉ) : Un schéma de dissimulation d'information S est dit *chaos-sûr* sur $\mathcal{X}_\tau$, si son processus itératif $\mathcal{T}(S)$ a un comportement chaotique, tel que le définit Devaney. ◊

REMARQUE. Nous avons fait le choix de la notion de Devaney, pour les raisons évoquées à la section 20.1. Il se peut qu'une autre notion de chaos soit plus pertinente encore pour entamer une étude de sécurité basée sur l'imprévisibilité : les mérites respectifs de chaque définition candidate devraient être comparés entre eux à un moment donné.

c. Universalité de la notion

Théoriquement parlant, la chaos-sécurité peut toujours être étudiée, car elle exige seulement que les deux points suivants soient respectés :

1. Tout d'abord, le schéma doit être mis sous la forme d'une itération sur un ensemble $\mathcal{X}$. Nous avons vu que cela est toujours possible.
2. Ensuite, une métrique ou une topologie doit être définie sur $\mathcal{X}$. Cela est toujours possible, ne serait-ce qu'en prenant soit la topologie discrète, soit la topologie grossière, même si ces dernières ne sont pas des plus pertinentes.

Ainsi la chaos-sécurité peut toujours s'étudier, mais la question du meilleur cadre d'étude se pose. Cette question, qui est celle du choix d'une bonne topologie, a été initiée au chapitre 16, et sera approfondie ci-dessous et plus tard dans le cadre qui nous intéresse ici.

II. DISCUSSIONS SUR LA DÉFINITION

1. Du bon choix de la topologie

La chaos-sécurité est évidemment affectée par le choix de la distance ou de la topologie sur $\mathcal{X}$, et cette dépendance doit être scrutée avec attention : ce choix doit être considéré avec soin, par exemple en créant un lien fort entre la proximité de deux points et les objectifs visés.

Toutefois, comme nous l'avions signalé au chapitre 16, certaines topologies sont plus naturelles et raisonnables que d'autres. Ainsi, l'on a pu considérer la topologie de l'ordre comme la plus naturelle sur un espace ordonné. De plus, le théorème III.15 permet souvent, partant d'une preuve de chaos-sécurité établie sur $(\mathcal{X}, \tau)$, de se ramener à cette dernière. Enfin, l'équivalence des normes en dimension finie réduit l'impact de ce choix.

D'autres part, il a été remarqué précédemment que les autres approches de sécurité supposent le même genre d'hypothèse : manipuler des probabilités implique la définition d'une tribu (sigma-algèbre). L'algèbre de Borel est toujours choisie, même si ce choix n'est ni spécifié, ni justifié. Comme un espace topologique est nécessaire pour définir des ensembles de Borel (la topologie de l'ordre est toujours choisie, sans discussion préalable), on peut dire qu'au moins, quand une étude de sécurité au sens classique du terme est réalisée, alors une étude du chaos peut être réalisé elle-aussi, avec la même topologie, et la même pertinence.

2. Des divers niveaux de sécurité

La chaos-sécurité est le premier niveau de notre échelle de sécurité fondée sur l'imprévisibilité. Cette propriété est requise, mais n'est pas suffisante : c'est seulement la première étape de l'évaluation du comportement imprévisible du schéma de dissimulation.

Cette étude doit être suivie par l'établissement de la liste précise des propriétés chaotiques que le système présente. En effet, être imprévisible est une chose difficile à définir, comme on l'a vu dans la partie II, et le nombre de candidats qui donnent consistance à cette notion est important : les entropies topologiques et métriques, l'ergodicité, le mélange topologique, l'exposant de Lyapunov, l'expansivité, la transitivité et la transitivité forte, la théorie de la bifurcation, ou le chaos tel qu'il est défini par Li-Yorke, par Devaney ou par Knudsen, *etc*.

Comme chaque définition illustre un aspect particulier d'un comportement imprévisible et a son propre intérêt, chaque notion de chaos offre un nouvel éclairage sur la sécurité d'un système de dissimulation des données. Ainsi, nous considérons qu'un schéma de tatouage sera plus sûr qu'un autre s'il présente un plus grand nombre de qualités chaotiques et si ses valeurs quantitatives sont meilleures. On ne définit pas ainsi un ensemble totalement ordonné, mais plutôt un ensemble réticulé.

Cette considération doit cependant être tempérée, et ce point nous semble important : la sécurité est une notion toute relative, dépendant du contexte, des objectifs visés et des risques encourus. Les propriétés chaotiques requises devraient donc dépendre des objectifs que l'on souhaite réaliser (tatouage fragile, robustesse, résistance contre les attaques dans la configuration CMA, *etc.*), ces propriétés à vérifier étant plus ou moins pertinentes suivant ce que l'on cherche à faire. Nous essayerons dans ce qui suit d'expliquer à chaque fois pourquoi nous cherchons à établir telle ou telle propriété et, par la suite, nous chercherons à dresser des listes de propriétés à satisfaire suivant le contexte considéré.

III. Imprévisibilité et classes d'attaques

1. Pourquoi la stégo-sécurité ne suffit pas

Nous reconsidérons maintenant les quatre classes d'attaques rappelées au chapitre 19. La stégo-sécurité est clairement pertinente, et nécessaire, dans la configuration WOA, dans le problème du prisonnier, et quand une seule clé est utilisée au cours des échanges : Eve a seulement accès à des contenus tatoués et, en raison de la stégo-sécurité, il est impossible pour elle de décider si un contenu a été tatoué ou non. Ainsi, dans la configuration WOA, un algorithme stégo-sûr pourrait contrer sans trop de problème les attaques d'Eve.

Toutefois, la stégo-sécurité ne s'applique pas aux configurations KOA, KMA et CMA. En effet, dans ces configurations, Eve tente de profiter de ses observations sur les contenus tatoués, quand elle change certaines conditions initiales dans l'algorithme de tatouage.

2. Exemples de situations où la chaos-sécurité est nécessaire

Nous donnons dans cette section deux exemples dans lesquels la chaos-sécurité nous semble nécessaire, respectivement lors d'une attaque *homme du milieu* à travers un canal caché, ou lors d'une tentative de suppression de DRM. Il s'agira à chaque fois de situations dans lesquelles Eve désire avoir une connaissance suffisante du système pour être en mesure de prédire son comportement.

a. Attaque de l'homme du milieu

Expliquons par exemple comment Eve peut essayer de réussir une attaque de type *homme du milieu*, en tirant partie des avantages du comportement prévisible d'un schéma de dissimulation dans la configuration KOA. Nous supposons qu'Alice et Bob communiquent entre eux par un canal caché, au-travers de tableaux de maîtres (Van Gogh, Modigliani, Blake, *etc.*) d'allure innocente. Eve peut avoir ainsi accès aux contenus tatoués *et* originaux :

- en utilisant une base de connaissance, contenant les peintures originales,
- et en observant le canal de communication.

Supposons maintenant qu'Alice envoie un tableau tatoué P_o à Bob. Si Eve est capable de prédire le comportement de l'algorithme de tatouage utilisé par Alice et Bob, alors elle peut :

1. Intercepter P_o.
2. Utiliser le même tableau P qu'Alice.
3. Et tenter de prédire comment son propre message devrait changer P.
4. Le résultat de cette prédiction est envoyé à Bob.

Il est vrai que les chances de succès de cette attaque sont faibles, mais le risque augmente avec la prévisibilité de l'algorithme de tatouage, et lorsque la taille disponible pour l'information secrète est petite (quelques bits), révélant ainsi une faille certaine de sécurité.

b. Attaque de DRM

Supposons maintenant qu'Eve veuille attaquer des DRM. Elle a accès à plusieurs paires de contenus : les filigranes – les copyrights, qui sont normalement publiques – et les médias tatoués associés : nous sommes dans la configuration KMA. En outre, elle a accès à la machine de tatouage, mais il lui manque la clé secrète : on suppose en effet que l'algorithme de DRM respecte le principe de Kerckhoffs. Cette clé secrète détermine la manière dont le copyright sera inséré dans le média.

Eve veut insérer son propre copyright dans ce média protégé, pour rendre impossible de déterminer si elle en est le propriétaire ou pas, ce qui invalide finalement le DRM d'un point de vue légal. En effet, il y aurait alors deux auteurs potentiels (dont l'un usurpe le rôle de l'autre), et il serait impossible de les départager sans doute. Eve ne sait pas exactement comment ces copyrights sont appliqués sur le support d'origine car, comme nous l'avons indiqué précédemment, la machine DRM fonctionne avec une clé secrète. Mais Eve peut atteindre son objectif si elle est capable de prédire le comportement de la machine de DRM. Elle sera alors en mesure de déterminer, approximativement, ce que devrait être le média qu'elle considère, s'il était tatoué avec son propre copyright. Ainsi, pour bien faire, la prédiction du comportement de la machine de copyright ne doit pas être une chose aisée.

Suivant Kerckhoffs, la clé doit être ce qui détermine ce comportement, et deux clés données devraient conduire à deux comportements fondamentalement différents : c'est exactement ce qui se passe avec un système chaotique. En outre, même si normalement contourner la sécurité de la clé ne doit pas être facile, il y a une faille de sécurité indéniable si Eve peut prédire approximativement le comportement du système avec une estimation partielle de la clé : de nombreux pirates ont réussi leur challenge grâce à une connaissance partielle d'une donnée secrète (par exemple, en étudiant les résidus mémoire). Enfin, un droit d'auteur n'est pas un message totalement aléatoire, il contient certaines informations prévisibles. La chaos-sécurité peut, nous semble-t-il, résoudre ces problèmes.

En conclusion, au moins dans les situations similaires à celles évoquées ci-dessus, des algorithmes imprévisibles sont, selon nous, nécessaires pour assurer un niveau de sécurité suffisant.

CHAPITRE 21

Chaos-sécurité de l'étalement de spectre

> Dieu est le seul être qui, pour régner, n'ait pas besoin d'exister.
>
> *Fusées I*
> CHARLES BAUDELAIRE

Dans ce qui suit, nous donnons un premier exemple d'évaluation de chaos-sécurité : nous établissons que l'étalement de spectre (spread-spectrum), un célèbre algorithme de tatouage [CB08], est chaos-sûr.

Ce faisant, nous prouvons que notre approche est prête à être appliquée à des exemples concrets, et nous établissons de plus un premier lien entre les notions de chaos-sécurité et de stégo-sécurité. Quelques conséquences, concernant l'utilisation des techniques d'étalement de spectre dans les configurations KOA, KMA et CMA, sont données à la fin de ce chapitre.

I. UNE PREMIÈRE PREUVE DE CHAOS-SÉCURITÉ

1. L'étalement de spectre pour la dissimulation d'information

Introduisons quelques notations pour commencer. Soit $x \in \mathbb{R}^{N_v}$ un vecteur hôte, dans lequel on souhaite cacher un message $m \in \{0;1\}^{N_c}$. Un générateur de nombres pseudo-aléatoires est utilisé afin d'obtenir des vecteurs dits « porteurs de secret » : $\left\{u^i \in \mathbb{R}^{N_v}, i \in [\![0; N_c - 1]\!]\right\}$.

DÉFINITION IV.16 (ÉTALEMENT DE SPECTRE) : L'*étalement de spectre* est une famille de techniques dans laquelle le message m est dissimulé dans l'hôte x, pour obtenir le contenu tatoué y, défini par :

$$y = x + w, \tag{21.1}$$

où $+$ est la somme de vecteurs de $\mathbb{R}^{N_v}$, et où le filigrane w a été construit à partir de m, de l'une des manières suivantes, selon la technique retenue.

Étalement de spectre dit « classique » : $w = \sum_{i=0}^{N_c-1} \gamma(-1)^{m^i} u^i$, où $\gamma \in \mathbb{R}$ est un niveau de distorsion fixé. On parle dans ce cas de « modulation BPSK » [CB08].

Étalement de spectre amélioré : $w = \sum_{i=0}^{N_c-1} \left((-1)^{m^i} \alpha - \lambda \frac{< x, u^i >}{\|u^i\|^2} \right) u^i$, où $\alpha \in \mathbb{R}$ et $\lambda \in \mathbb{R}$ sont calculés pour réaliser une distorsion moyenne acceptable, et minimiser la probabilité d'erreur lors de l'extraction du filigrane [CB08]. Cette technique se note aussi ISS (*Improved Spread Spectrum*, voir [MF03]).

Étalement de spectre dit « naturel » : $w = \sum_{i=0}^{N_c-1} -\left(1 + \eta \, (-1)^{m^i} \frac{< x, u^i >}{| < x, u^i > |} \right) \frac{< x, u^i >}{\|u^i\|^2} \, u^i$, où $\eta \in \mathbb{R}$ sert à nouveau à fixer un certain niveau de distorsion donné. ♢

Nous allons dans ce qui suit reformuler ces techniques de dissimulation, en nous inspirant de ce qui a été fait au chapitre 11, et ce afin de pouvoir étudier leur chaos-sécurité.

2. Modélisation des techniques d'étalement de spectre

Supposons que la taille des filigranes soit bornée par une valeur finie N_b donnée :

$$\max\left(\left\{ \|w\|_\infty \,\middle/\, w \in \mathbb{R}^{N_v} \right\} \right) \leqslant \mathsf{N}_b.$$

Cette borne peut être aussi grande que l'on veut ; cependant une très grande valeur de N_b est en contradiction avec les objectifs de dissimulation des données.

Soient [7] $\overline{\mathcal{X}} = \left([\![-\mathsf{N}_b, \mathsf{N}_b]\!]^{N_v} \right)^{\mathbb{N}} \times \mathbb{R}^{N_v}$ et $\overline{G}((S,E)) = (\overline{\sigma}(S); \overline{\iota}(S) + E)$, où :

- $\overline{\sigma}$ est la fonction *décalage* (shift) définie, de manière semblable au chapitre 11, par :

$$\begin{array}{rccc} \overline{\sigma} : & \left([\![-\mathsf{N}_b; \mathsf{N}_b]\!]^{N_v} \right)^{\mathbb{N}} & \longrightarrow & \left([\![-\mathsf{N}_b; \mathsf{N}_b]\!]^{N_v} \right)^{\mathbb{N}} \\ & (S^n)_{n\in\mathbb{N}} & \longmapsto & (S^{n+1})_{n\in\mathbb{N}}, \end{array}$$

- et la fonction *initiale* $\overline{\iota}$ est l'application qui transforme une suite en son premier terme :

$$\begin{array}{rccc} \overline{\iota} : & \left([\![-\mathsf{N}_b; \mathsf{N}_b]\!]^{N_v} \right)^{\mathbb{N}} & \longrightarrow & [\![-\mathsf{N}_b; \mathsf{N}_b]\!]^{N_v} \\ & (S^n)_{n\in\mathbb{N}} & \longmapsto & S^0. \end{array}$$

Nous sommes dorénavant en mesure de modéliser l'étalement de spectre sous la forme d'un système itératif, permettant d'en faire son étude de chaos-sécurité :

Proposition IV.2 : *Les techniques d'étalement de spectre sont les résultats de N_c itérations du système dynamique :*

$$\begin{cases} X^0 \in \overline{\mathcal{X}}, \\ X^{n+1} = \overline{G}(X^n), \end{cases}$$

où X^0 dépend des données initiales et de la technique choisie (cette dépendance est précisée à la section suivante), et le média tatoué est la deuxième coordonnée de X^{N_c}.

Remarque. La deuxième coordonnée de X^k correspond à l'image hôte après k modifications, tandis que sa première coordonnée explique comment modifier l'hôte à la prochaine itération.

7. Nous utiliserons le surlignement pour ne pas confondre ce nouvel espace des phases, et cette nouvelle fonction d'itérations, avec celles définissant les itérations chaotiques.

3. Conditions initiales et variantes des techniques d'étalement de spectre

On constate immédiatement que choisir la technique d'étalement de spectre revient à choisir la condition initiale du système de la proposition IV.2, ce que l'on résume dans les résultats suivants :

PROPOSITION IV.3 : *L'étalement de spectre classique correspond aux itérations de la proposition IV.2, où la condition initiale $X^0 = (S^0, E^0)$ est définie ainsi : E^0 est le vecteur hôte x, et S^0 est la suite $\left((-1)^{m^0}\gamma\, u^0, (-1)^{m^1}\gamma\, u^1, \ldots, (-1)^{m^{N_c-1}}\gamma\, u^{N_c-1}\right)$, complétée indéfiniment par des vecteurs nuls de $\mathbb{R}^{N_v}$.*

PROPOSITION IV.4 : *La technique d'étalement de spectre amélioré correspond au système de la proposition IV.2, où la condition initiale $X^0 = (S^0, E^0)$ est définie par : E^0 est le vecteur hôte x, et S^0 est la suite $\left(\left((-1)^{m^i}\alpha - \lambda\dfrac{< x, u^i >}{\|u^i\|^2}\right) u^i\right)_{i=0,\ldots,N_c-1}$, complétée avec des vecteurs nuls.*

PROPOSITION IV.5 : *La technique d'étalement de spectre dit naturel correspond au système de la proposition IV.2, où la condition initiale $X^0 = (S^0, E^0)$ est définie par : E^0 est le vecteur hôte x, et S^0 est la suite $\left(-\left(1 + \eta(-1)^{m^i}\dfrac{< x, u^i >}{| < x, u^i > |}\right)\dfrac{< x, u^i >}{\|u^i\|^2}\, u^i\right)_{i=0,\ldots,N_c-1}$, complétée, comme ci-dessus, avec des vecteurs nuls.*

4. Stégo-sécurité de l'étalement de spectre

Nous rappelons le résultat suivant [CB08] :

THÉORÈME IV.1 : *L'étalement de spectre naturel, avec $\eta = 1$, est une technique de dissimulation stégo-sûre.*

Nous allons maintenant prouver dans ce qui suit que les techniques d'étalement de spectre sont chaos-sûres, établissant ainsi un premier lien entre les deux notions de sécurité (intersection non vide).

II. PREMIÈRE PREUVE DE CHAOS-SÉCURITÉ

1. Une métrique sur $\overline{\mathcal{X}}$

On définit une nouvelle distance entre deux points $X = (S, E), Y = (\check{S}, \check{E}) \in \overline{\mathcal{X}}$ par :

$$\overline{d}(X, Y) = d_\infty(E, \check{E}) + \overline{d_s}(S, \check{S}),$$

où :

$$\begin{cases} d_\infty(A, B) = max\left\{ |A_i - B_i| \,\middle/\, i \in [\![1; N_v]\!] \right\}, \\ \overline{d_s}(S, \check{S}) = \dfrac{9}{\mathsf{N}_b}\displaystyle\sum_{k=0}^{\infty}\frac{d_\infty(S^k, \check{S}^k)}{10^k}. \end{cases}$$

Le choix de d_∞ sur $\mathbb{R}^{N_v}$ n'est pas important, du fait de l'équivalence des normes en dimension finie : les topologies sont les mêmes, donc les propriétés de chaos ne changent pas en utilisant une autre distance issue d'une norme sur $\mathbb{R}^{N_v}$. $\overline{d_s}$ a été choisie de sorte que $\overline{d}(X,Y)$ soit petit quand la distance entre les images tatouées résultantes sont proches (pour n'importe quelle distance issue d'une norme sur $\mathbb{R}^{N_v}$, vu qu'elles sont toutes équivalentes). Enfin, $\dfrac{9}{\mathsf{N}_b}$ n'est qu'un coefficient de normalisation.

2. Continuité de l'étalement de spectre

Nous allons maintenant prouver que,

THÉORÈME IV.2 : *$\overline{G}$ est continue sur $(\overline{\mathcal{X}}, \overline{d})$.*

PREUVE : *On utilise la continuité séquentielle.*

*Soit $(S_n, E_n)_{n\in\mathbb{N}}$ une suite sur l'espace des phases $\overline{\mathcal{X}}$, qui converge vers (S,E). Nous allons prouver que $\left(\overline{G}(S_n,E_n)\right)_{n\in\mathbb{N}}$ converge vers $\overline{G}(S,E)$. Notons que pour tout n, S_n est une stratégie, ainsi nous considérons une suite de stratégies (*i.e. *une suite de suites).*

Comme $\overline{d}((S_n,E_n);(S,E))$ converge vers 0, chaque suite $d_\infty(E_n,E)$ et $\overline{d_s}(S_n,S)$ converge aussi vers 0.

1. *Si* $\dfrac{9}{\mathsf{N}_b}\displaystyle\sum_{k=0}^{\infty}\frac{d_\infty(S_n^k,S^k)}{10^k}\to 0$ *quand* $n\to+\infty$, *alors* $\dfrac{9}{\mathsf{N}_b}\displaystyle\sum_{k=1}^{\infty}\frac{d_\infty(S_n^k,S^k)}{10^k}\to 0$.
   Ainsi $\dfrac{9}{\mathsf{N}_b}\displaystyle\sum_{k=0}^{\infty}\frac{d_\infty(S_n^{k+1},S^{k+1})}{10^{k+1}} = \frac{1}{10}\overline{d_s}(\overline{\sigma}(S_n);\overline{\sigma}(S))\to 0$. *De ce fait,* $\overline{d_s}(\overline{\sigma}(S_n),\overline{\sigma}(S))$ *converge vers 0.*
2. *Prouvons maintenant que* $d_\infty\left(\overline{\iota}(S_n)+E_n;\overline{\iota}(S)+E\right)\to 0$.

$$
\begin{aligned}
&d_\infty\left(\overline{\iota}(S_n)+E_n;\overline{\iota}(S)+E\right)\\
&\quad= max\left\{\left|(\overline{\iota}(S_n)_k+(E_n)_k)-(\overline{\iota}(S)_k+E_k)\right| \;\middle/\; k\in[\![1;N_v]\!]\right\}\\
&\quad= max\left\{\left|(\overline{\iota}(S_n)_k-\overline{\iota}(S)_k)+((E_n)_k-E_k)\right| \;\middle/\; k\in[\![1;N_v]\!]\right\}\\
&\quad\leqslant max\left\{\left|\overline{\iota}(S_n)_k-\overline{\iota}(S)_k\right|+\left|(E_n)_k-E_k\right| \;\middle/\; k\in[\![1;N_v]\!]\right\}\\
&\quad\leqslant max\left\{\left|\overline{\iota}(S_n)_k-\overline{\iota}(S)_k\right| \;\middle/\; k\in[\![1;N_v]\!]\right\}+d_\infty(E_n,E)\\
&\quad= d_\infty(S_n^0,S^0)+d_\infty(E_n,E)\\
&\quad\leqslant \overline{d_s}(S_n,S)+d_\infty(E_n,E)\\
&\quad= \overline{d}\left((S_n,E_n);(S,E)\right)\to 0.
\end{aligned}
$$

Nous sommes donc pleinement dans le cadre d'étude des systèmes dynamiques d'après Devaney (itérations d'une fonction continue sur un espace topologique ou métrique). Nous allons nous intéresser maintenant à la régularité et la transitivité de ce système.

3. Régularité

PROPOSITION IV.6 : *Les points périodiques de $\overline{G}$ sont denses dans $(\overline{\mathcal{X}},\overline{d})$, donc $\overline{G}$ est régulier.*

PREUVE : *Soit* $(S,E) \in \overline{\mathcal{X}}$ *et* $\varepsilon > 0$. *On recherche un point périodique* $(\check{S},\check{E}) \in \overline{\mathcal{X}}$ *tel que* $\overline{d}\left((S,E),(\check{S},\check{E})\right) < \varepsilon$. *Soit* $\check{E} = E$ *et* S^n *représentant la suite (de suites) définie par :*

$$\begin{cases} [S^n]^k = S^k & \forall k \leqslant n, \\ [S^n]^k = (\mathsf{N}_b,\dots,\mathsf{N}_b) & \text{si } k > n \text{ et } k \equiv 0 \ (mod\ 2); \\ [S^n]^k = (-\mathsf{N}_b,\dots,-\mathsf{N}_b) & \text{sinon.} \end{cases}$$

Alors $\overline{d_s}(S^n,S) = \dfrac{9}{\mathsf{N}_b}\displaystyle\sum_{k=n+1}^{\infty}\dfrac{d_\infty([S^n]^k,S^k)}{10^k} \leqslant \dfrac{9}{\mathsf{N}_b}\sum_{k=n+1}^{\infty}\dfrac{\mathsf{N}_b}{10^k} = \dfrac{1}{10^n} \to 0$ *quand* $n \to \infty$. *Donc* $\exists\, n_0 \in \mathbb{N}$ *tel que* $\overline{d_s}(S^{n_0},S) < \varepsilon$. *Le point* $(\check{S},\check{E}) = (S^{n_0},E)$ *est donc un point périodique de* $\overline{\mathcal{X}}$ *pour* $\overline{G}$ *(évident), qui est* ε*-proche du point donné* (S,E).

4. Transitivité

Nous allons maintenant prouver que,

PROPOSITION IV.7 : $\overline{G}$ *est transitive sur* $(\overline{\mathcal{X}},\overline{d})$.

PREUVE : *Soient* $B_A = \mathcal{B}(X_A,r_A)$ *et* $B_B = \mathcal{B}(X_B,r_B)$ *deux boules ouvertes de* $\overline{\mathcal{X}}$, *où* $X_A = (S_A,E_A)$ *et* $X_B = (S_B,E_B)$. *On recherche* $\tilde{X} = (\tilde{S},\tilde{E}) \in B_A$ *tel que* $\exists n_0 \in \mathbb{N}, \overline{G}^{(n_0)}(\tilde{X}) \in B_B$.

Soit $k_0 \in \mathbb{Z}$ *tel que* $10^{-k_0} \leqslant r_A < 10^{-k_0+1}$ *et* $\left(\check{S},\check{E}\right) = \overline{G}^{(k_0)}(X_A)$. *On définit* $\tilde{X} = (\tilde{S},\tilde{E})$ *comme suit :*

- $\tilde{E} = E_A$,
- $\forall k \leqslant k_0, \tilde{S}^k = S_A^k$,
- $\forall k \in [\![1,N_v]\!], \tilde{S}^{k_0+k} = (-\check{E}^k + E_B^k)\times(0,\dots,0,1,0,\dots,0)$, i.e. *le vecteur* $\tilde{S}^{k_0+k}$ *a toutes ses composantes nulles, sauf sa* $k-$*ième, égale à* $(-\check{E}^k + E_B^k)$,
- $\forall k \in \mathbb{N}, \tilde{S}^{k_0+N_v+k+1} = S_B^k$.

Avec une telle définition, $\tilde{X}$ *est dans* B_A *et vérifie* $\overline{G}^{(k_0+N_v)}\left(\tilde{X}\right) \in B_B$.

5. Conclusion

Comme $\overline{G}$ et régulière et transitive sur $(\overline{\mathcal{X}},\overline{d})$, on peut en conclure que $\overline{G}$ est chaotique selon Devaney, ce qui prouve ainsi le théorème :

THÉORÈME IV.3 : *Les techniques d'étalement de spectre sont chaos-sûres.*

REMARQUE. Toutes les techniques d'étalement de spectre sont chaos-sûres, alors que seul l'étalement de spectre naturel avec $\eta = 1$ est stégo-sûr. Cela provient du fait que le choix de la technique n'impacte que la condition initiale, et que la théorie du chaos ne traite pas des conditions initiales, mais s'intéresse « seulement » aux fonctions d'itérations. Il ne faut pas pour autant penser que la chaos-sécurité est plus

faible que la stégo-sécurité. En d'autres termes, que tout algorithme stégo-sûr est aussi chaos-sûr, relation d'inclusion réduisant drastiquement l'intérêt de cette dernière.

Déjà, cette inclusion est peu probable, vue la grande différence des outils utilisés (topologie *vs* probabilités) ; la finesse de chacune des deux notions et le formalisme qu'ils supposent, rendent difficilement imaginable une relation de cause à effet. Ensuite, une fois encore, la stégo-sécurité se restreint à la configuration WOA, et même à un cadre particulier de cette dernière, ce qui n'est pas le cas de la chaos-sécurité. Ces propriétés de sécurité ne sont donc que rarement comparables, et sont complémentaires quand elles le sont. De plus, il peut très bien exister un algorithme stégo-sûr qui ne soit pas chaos-sûr. Seulement, le très faible nombre d'algorithmes prouvés sûr suivant l'une ou l'autre des approches, ne nous a pas encore permis d'en trouver.

III. Étude approfondie de la chaos-sécurité

Comme nous l'avons précisé au chapitre précédent, la preuve que l'algorithme d'étalement de spectre est chaos-sûr n'est que le début de l'étude de sa sécurité telle que nous l'entendons. Il ne permet pas de déterminer les configurations (KOA, KMA, CMA, ou WOA) dans lesquelles l'algorithme de tatouage peut être utilisé sans problème.

La prochaine étape consiste à évaluer la qualité de ce comportement chaotique, en utilisant les nombreux outils qualitatifs et quantitatifs offerts par la théorie du chaos. Ces outils permettent de comparer deux algorithmes chaos-sûrs donnés et d'en déduire, ce faisant, quel algorithme préférer pour une configuration d'attaques choisie. Pour illustrer ce propos, quelques propriétés seront étudiées dans cette section, à savoir la transitivité forte et les constantes d'expansivité et de sensibilité. Nous les utiliserons afin d'avoir une meilleure compréhension de l'imprévisibilité des techniques à étalement de spectre, et d'en déduire s'il est raisonnable de les utiliser dans les configurations KOA, KMA et CMA.

1. Propriétés qualitatives

a. La transitivité forte

Nous rappelons (définition II.30) qu'un système itératif f est dit *fortement transitif* sur l'espace topologique $(\mathcal{X}, \tau)$ si et seulement si, pour tous points $A, B \in \mathcal{X}$ et tout voisinage V de B, $n_0 \in \mathbb{N}$ et $X \in V$ peuvent être trouvés tels que $f^{n_0}(X) = A$.

Nous pouvons énoncer que,

Théorème IV.4 : *L'étalement de spectre $\overline{G}$ est fortement transitif sur $(\overline{\mathcal{X}}, \overline{d})$.*

Preuve : *Reprenons la preuve de la transitivité de $\overline{G}$ sur $(\overline{\mathcal{X}}, \overline{d})$.*

Nous avons défini $\tilde{X} \in B_A$ tel que $\overline{G}^{(k_0+N_v)}\left(\tilde{X}\right) \in B_B$. En fait, pour ce $\tilde{X}$, nous avons : $\overline{G}^{(k_0+N_v)}\left(\tilde{X}\right) = X_B$, ce qui est la forte transitivité.

Cette propriété de transitivité forte renforce les effets et les conséquences de la transitivité, en termes de chaos-sécurité et d'authentification. En fait, en raison de la transitivité forte, l'ensemble des médias tatoués obtenus en utilisant un filigrane fixé une fois pour toute, est potentiellement égal à l'ensemble

de tous les médias. Dans cette situation, Eve ne peut diviser l'ensemble des médias à étudier lors d'une attaque CMA.

b. Le mélange topologique

Nous montrons dans cette section que les techniques d'étalement de spectre possèdent la propriété de mélange topologique, ce qui renforce la transitivité forte. Les conséquences de cette propriété seront discutées un peu plus loin.

On rappelle qu'un système itératif est dit topologiquement mélangeant si et seulement si pour tout couple d'ouverts disjoints $U, V \neq \emptyset$, $n_0 \in \mathbb{N}$ peut être trouvé tel que $\forall n \geqslant n_0, f^{(n)}(U) \cap V \neq \emptyset$ (voir la définition II.31). Nous pouvons énoncer que :

THÉORÈME IV.5 : *Les techniques d'étalement de spectre sont topologiquement mélangeantes sur* $(\overline{\mathcal{X}}, \overline{d})$.

Ce résultat est une conséquence immédiate du lemme 21.1 ci-dessous.

LEMME 21.1. Pour toute boule ouverte B de $\overline{\mathcal{X}}$, un indice $n \in \mathbb{N}$ peut être trouvé tel que $\overline{G}^{(n)}(B) = \overline{\mathcal{X}}$. ◇

PREUVE : *Soit* $B = \mathcal{B}((S,E),\varepsilon)$ *une boule ouverte, dont le rayon peut être considéré comme strictement inférieur à 1. Tous les éléments de* B *présentent donc le même état* E *et sont tel qu'un entier* $k = \lceil -\log_{10}(\varepsilon) \rceil$ *vérifie :*

- *toutes les stratégies de* B *ont les mêmes* k *premiers termes,*
- *après l'indice* k*, toutes les valeurs sont possibles.*

Alors, après k *itérations, le nouvel état du système est* $\overline{G}^{(k)}(S,E)_2$ *et toutes les stratégies sont possibles (tous les points* $(\overline{G}^{(k)}(S,E)_2, \hat{S})$*, avec n'importe quel* $\hat{S} \in \overline{\mathcal{S}}$*, sont accessibles à partir de* B*).*

Nous allons prouver que tous les points de $\overline{\mathcal{X}}$ *sont accessibles à partir de* B*. Soit* $(S',E') \in \overline{\mathcal{X}}$*. Alors le point* $(\check{S},\check{E})$ *de* B *défini par :*

- $\check{E} = E$,
- $\check{S}^i = S^i, \forall i \leqslant k$,
- $\check{S}^{k+i} = -E^i + E'^i, \forall i \leqslant N_v$,
- $\forall i \in \mathbb{N}, S^{k+N_v+i+1} = S'^i$.

est tel que $\overline{G}^{(k+N_v)}(\check{S},\check{E}) = (S',E')$.

Ceci conclut la preuve du lemme et donc de la proposition.

Les conséquences du mélange topologique sont multiples en stéganalyse.

Tout d'abord, dans une situation de mélange topologique, la chaos-sécurité peut être largement améliorée en considérant le nombre d'itérations de la machine de tatouage comme une clé secrète. Un attaquant va atteindre tous les médias possibles lors de ses tentatives d'itérations, même en connaissant tout sauf cette clé.

De plus, l'adversaire ne peut pas espérer bénéficier de la configuration KOA, en faisant une étude approfondie de l'ensemble des médias semblables au média d'origine. Et, comme dans une situation de mélange topologique, il est possible que tout message caché (état initial) soit envoyé dans un seul et unique contenu tatoué fixé (avec cependant un nombre différent d'itérations), l'intérêt d'être dans une configuration KMA est considérablement réduite.

Enfin, comme tous les contenus tatoués sont possibles pour un message caché donné, selon le nombre d'itérations, les attaques CMA vont clairement échouer.

c. Conclusion

Pour résumer, le mélange topologique et la transitivité forte sont nécessaires afin de résister aux attaques KOA et KMA, alors que seul le mélange topologique est fondamental dans la configuration CMA. On peut déduire de l'étude précédente que :

- Seul l'étalement de spectre naturel avec $\eta = 1$ devrait être utilisé dans la configuration WOA, car c'est la seule classe stégo-sûre d'étalement de spectre.
- A contrario, comme toutes les techniques d'étalement de spectre sont chaos-sûres, nous ne pouvons rejeter leur utilisation dans la configuration CMA.

Nous allons maintenant étudier quantitativement le chaos de l'étalement de spectre.

2. Mesures quantitatives

a. La sensibilité aux conditions initiales

L'une des mesures quantitatives les plus célèbres de la théorie du chaos est la constante de sensibilité rappelée à la définition II.37. Intuitivement, quand une fonction f a une constante de sensibilité égale à δ, alors il existe des points aussi proches que l'on veut de x qui *peuvent* s'éloigner de x d'au moins δ après un certain nombre d'itérations de f. De ce fait, une erreur arbitrairement petite sur un état initial *peut* être agrandie d'au moins δ, quand on itère f. Elle peut l'être, mais ne l'est pas forcément : cette incertitude augmente de fait l'imprévisibilité du système.

On a vu que la sensibilité aux conditions initiales est une conséquence de la régularité et de la transitivité dans un espace métrique [BBCS92]. Cependant, la constante de sensibilité δ ne peut être obtenue finalement qu'en démontrant cette propriété sans le théorème de Banks.

THÉORÈME IV.6 : *Les techniques d'étalement de spectre sont sensibles aux conditions initiales sur* $\left(\overline{\mathcal{X}}, \overline{d}\right)$*, et leur constante de sensibilité est au moins égale à* $\dfrac{\mathsf{N}_b}{2}$.

PREUVE : *Soit* $X = (S, E) \in \overline{\mathcal{X}}$, $B = \mathcal{B}(X, r)$ *une boule ouverte centrée sur* X, *et* $k_0 \in \mathbb{Z}$ *tel que* $10^{-k_0} \leqslant r < 10^{-k_0+1}$. *On définit* $\check{X}$ *par :*

- $\check{E} = E$,
- $\check{S}^k = S^k$, $\forall k \in \mathbb{N}$ *tel que* $k \neq k_0 + 1$,
- *si* $S_1^{k_0+1} < \frac{\mathsf{N}_b}{2}$, *alors* $\check{S}_1^{k_0+1} = \mathsf{N}_b$, *sinon* $\check{S}_1^{k_0+1} = 0$,
- $\forall i \in [\![2, N_v]\!]$, $\check{S}_i^{k_0+1} = S_i^{k_0+1}$.

Alors :

$$\overline{d}(X,\check{X}) = d_\infty(E,\check{E}) + \overline{d_S}(S,\check{S}) = 0 + \frac{9}{\mathsf{N}_b}\frac{d_\infty(S^{k_0+1},\check{S}^{k_0+1})}{10^{k_0+1}} \leqslant \frac{9}{\mathsf{N}_b}\frac{\mathsf{N}_b}{10^{k_0+1}} \leqslant \frac{1}{10^{k_0}} \leqslant r,$$

donc $\check{X} \in B$. *Soit maintenant* $\mathcal{E} : \overline{X} \to \overline{X}, (S,E) \mapsto E$. *Alors* $\mathcal{E}\left(\overline{G}^{(k_0+1)}(X)\right)_0 = \mathcal{E}\left(\overline{G}^{(k_0+1)}(\check{X})\right)_0$, *car* $E = \check{E}$ *et* $S^k = \check{S}^k, \forall k \leqslant k_0+1$. *Comme :*

- $\mathcal{E}\left(\overline{G}^{(k_0+2)}(X)\right)_0 = \mathcal{E}\left(\overline{G}^{(k_0+1)}(X)\right)_0 + S_0^{k_0+1}$,
- $\mathcal{E}\left(\overline{G}^{(k_0+2)}(\check{X})\right)_0 = \mathcal{E}\left(\overline{G}^{(k_0+1)}(\check{X})\right)_0 + \check{S}_0^{k_0+1}$,
- $\left|S_0^{(k_0+1)} - \check{S}_0^{(k_0+1)}\right| \geqslant \frac{\mathsf{N}_b}{2}$. □

Nous avons alors :

$$\begin{aligned} d\left(\overline{G}^{(k_0+2)}(X), \overline{G}^{(k_0+2)}(\check{X})\right) &\geqslant d_\infty\left(\mathcal{E}\left(\overline{G}^{(k_0+2)}(X)\right), \mathcal{E}\left(\overline{G}^{(k_0+2)}(\check{X})\right)\right) \\ &\geqslant \left|\mathcal{E}\left(\overline{G}^{(k_0+2)}(X)\right)_0 - \mathcal{E}\left(\overline{G}^{(k_0+2)}(\check{X})\right)_0\right| \\ &\geqslant \frac{\mathsf{N}_b}{2}. \end{aligned}$$

b. L'expansivité

Voyons maintenant une dernière mesure quantitative importante de désordre : l'expansivité. On rappelle (définition II.38) qu'une fonction f possède la propriété d'*expansivité* si :

$$\exists \varepsilon > 0, \forall x \neq y, \exists n \in \mathbb{N}, d(f^{(n)}(x), f^{(n)}(y)) \geqslant \varepsilon.$$

ε est appelée la *constante d'expansivité* de f. Elle vérifie la propriété suivante : une erreur arbitrairement petite sur toute condition initiale est *toujours* amplifiée jusqu'à atteindre, à un moment donné, ε.

Nous allons prouver que :

THÉORÈME IV.7 : *L'étalement de spectre* $\overline{G}$ *n'est pas expansif sur* $(\overline{X}, \overline{d})$.

PREUVE : *Soit* $\varepsilon > 0$. *On pose :*

- $X = (O_{N_v}; (O_{N_v}, O_{N_v}, \ldots, O_{N_v}, \ldots))$,
- $Y = \left(O_{N_v}; \left(\frac{\varepsilon}{2} I_{N_v}, -\frac{\varepsilon}{2} I_{N_v}, \ldots, \frac{(-1)^n \varepsilon}{2} I_{N_v}, \ldots\right)\right)$,

où $O_{N_v} = (0,\ldots,0)$ *est le vecteur nul de taille* N_v *et* I_{N_v} *est le vecteur de taille* N_v *égal à* $(1,0,\ldots,0)$. *Alors, pour ces deux points, nous avons :* $\forall n \in \mathbb{N}, d\left(\overline{G}^{(n)}(X); \overline{G}^{(n)}(Y)\right) < \varepsilon$.

TABLE 21.1 – Évaluation des techniques d'étalement de spectre

	Propriété	Étalement de spectre
CAPACITÉ	Tatouage aveugle	Oui
	Domaine de tatouage	Spatial, fréquentiel
	Authentification	Non
	Chiffrement du filigrane	Non
	Robustesse	Non
	Chaos-sécurité	Étalement de spectre
CHAOS-SÉCURITÉ	Chaos selon Devaney	Oui
	Espace des phases	$\overline{\mathcal{X}} = (\llbracket -\mathsf{N}_b ; \mathsf{N}_b \rrbracket^{N_v})^{\mathbb{N}} \times \mathbb{R}^{N_v}$
	Topologie	Induite par la métrique $\overline{d}$
	Propriétés qualitatives	Étalement de spectre
	Transitivité forte	Oui
	Mélange topologique	Oui
	Propriétés qualitatives	Étalement de spectre
	Sensibilité	Oui. $\delta \geqslant \dfrac{\mathsf{N}_b}{2}$
	Expansivité	Non
	Propriété	Étalement de spectre
STÉGO-SÉCURITÉ	Stégo-sécurité	L'étalement de spectre naturel l'est quand $\eta = 1$

c. Conséquences partielles

Nous avons résumé certains aspects de la sécurité de l'étalement de spectre dans le tableau 21.1, et y avons reporté d'autres propriétés présentes dans la littérature [CMK+97, CB08]. Nous allons maintenant discuter de l'impact du choix de la condition initiale, et de l'absence d'expansivité.

i. Du bon choix de la condition initiale. Tout d'abord, toutes les techniques d'étalement de spectre sont concernées par la propriété de chaos-sécurité. Du point de vue présenté dans ce chapitre, le choix de l'étalement de spectre naturel (NW) ou de l'étalement de spectre amélioré (ISS) n'intervient que dans la sélection de l'état initial des itérations de $\overline{G}$. En fait, la théorie du chaos donne une approche globale du comportement imprévisible d'un système donné, mais n'explique pas comment choisir un « bon état initial ». Par exemple, la suite logistique ($x^0 \in [0,1]$, $x^{n+1} = 4\,x^n\,(1-x^n)$) a un comportement chaotique, mais si nous choisissons $x^0 = 0$, alors $\forall n \in \mathbb{N}$, $x^n = 0$: le sous-ensemble des valeurs initiales autorisées doit sûrement être considéré avec précaution.

La stégo-sécurité peut être utile pour déterminer de bonnes conditions initiales dans certaines configurations WOA. Par exemple, S^0 défini comme dans l'équation IV.5 (étalement de spectre naturel), avec $\eta = 1$, conduit à un étalement de spectre chaos-sûr et stégo-sûr.

ii. De l'importance de l'expansivité. Nous allons maintenant discuter des conséquences du fait que l'étalement de spectre n'est pas expansif. La propriété d'expansivité renforce considérablement les effets de la sensibilité, pour contrer les attaques d'Eve dans les configurations KMA ou KOA. Par exemple, du fait de cette expansivité, il est impossible d'avoir ne serait-ce qu'une quelconque estimation du filigrane, en déplaçant le message (ou l'hôte) comme un curseur : ce curseur sera trop sensible et les modifications

induites seront trop importantes pour qu'Eve puisse en tirer la moindre information.

A contrario, une très grande constante d'expansivité ε n'est pas acceptable. Il faudrait normalement que le filigrane soit indétectable mais, dans cette situation, la couverture sera fortement altérée lors du tatouage. En effet, considérons la même couverture E tatouée par deux filigranes différents S et S'. Alors $\overline{d}(X,Y) < 1$, où $X = (S,E)$ et $Y = (S',E)$. Mais du fait de l'expansivité, $\exists n \in \mathbb{N}, \overline{d}\left(\overline{G}^{(n)}(X); \overline{G}^{(n)}(Y)\right) \geqslant \varepsilon$. Alors, $d_\infty\left(\overline{G}^{(n)}(X)_2; \overline{G}^{(n)}(Y)_2\right) \geqslant \varepsilon - 1$, donc soit $d_\infty\left(E; \overline{G}^{(n)}(X)_2\right) \geqslant \dfrac{\varepsilon-1}{2}$, soit $d_\infty\left(E; \overline{G}^{(n)}(Y)_2\right) \geqslant \dfrac{\varepsilon-1}{2}$: si ε est grand, alors au moins une des deux images tatouées sera très différente de la couverture d'origine.

En conclusion, un algorithme de tatouage chaos-sûr qui n'est pas expansif peut seulement, éventuellement, être utilisé dans les configurations WOA et CMA, alors que les configurations KMA et KOA ont besoin d'expansivité pour assurer un niveau de sécurité satisfaisant (telle que nous la définissons).

3. Conclusion sur les techniques d'étalement de spectre

Commençons par remarquer que nous avons réussi dans ce qui précède à faire l'étude des propriétés chaotiques des techniques d'étalement de spectre. Plus précisément, nous avons prouvé que l'étalement de spectre est chaotique au sens de Devaney, et établi quelques autres propriétés de désordre topologique. Il s'agit là d'un résultat en soi : l'étalement de spectre, étant à peu près la seule technique de dissimulation à avoir été stéganalysée en profondeur, est actuellement utilisée de manière non négligeable. Notre contribution visant à déterminer le comportement topologique de cet étalement, et son caractère foncièrement imprévisible, devrait donc renforcer la confiance que l'on porte dans de telles techniques, pourvu qu'elles soient utilisées en respectant les recommandations du chapitre 18 (expliquant comment préserver le chaos au cours des implantations sur machine). Notre étude permet de plus de conclure que :

1. L'utilisation de l'étalement de spectre, dans le cas « naturel » avec $\eta = 1$, est particulièrement pertinente lorsqu'une technique discrète et sûre est requise dans la configuration WOA.
2. Le cas particulier de la configuration CMA doit être approfondie, afin de vérifier si le caractère imprévisible de ces techniques est suffisamment élevé ou non. Cette étude sera réalisée à terme, notamment en utilisant l'entropie topologique et l'exposant de Lyapunov.
3. Enfin, ces techniques ne devraient pas être utilisées dans les configurations KOA et KMA, en raison notamment de leur manque d'expansivité.

Dans le prochain chapitre, nous présenterons une nouvelle classe de fonctions de tatouage dite dhCI. Contrairement aux techniques d'étalement de spectre, cette nouvelle classe est expansive.

CHAPITRE 22

Les itérations chaotiques pour l'information dissimulée

> Les meilleurs événements sourient en vain à l'homme malheureux. Et il y a plus de volonté qu'on ne croit dans le bonheur.
>
> *Propos sur le bonheur*
> ALAIN

Notre objectif, dans ce chapitre, est de construire un algorithme de tatouage qui serait meilleur que l'étalement de spectre dans les configurations KOA et KMA. Pour ce faire, nous avons cherché un processus chaotique itératif applicable à la dissimulation d'information et qui soit expansif. Ce chapitre présente les résultats de nos recherches.

L'algorithme dhCI (*data hiding based on Chaotic Iterations*) et sa chaos-sécurité ont été publiés dans [BG10b] et [BG10a]. Pour ce qui est de sa stégo-sécurité, elle fait l'objet de la contribution [GFB10]. Enfin, cet algorithme a servi d'exemple dans nos contributions sur les nombres pseudo-aléatoires [WBGF10, BGW10a].

I. NOTRE APPROCHE ET L'EXISTANT

Commençons par situer notre approche par rapport à celles qui ont habituellement cours dans les techniques de tatouage par chaos.

1. L'utilisation actuelle du chaos pour l'information dissimulée

La pertinence de l'utilisation du chaos pour la dissimulation de données a été démontrée ces dernières années [MKP08]. Dans les méthodes existantes, le chaos est principalement utilisé soit pour chiffrer le filigrane, soit pour l'insérer dans l'image hôte, comme l'illustre l'état de l'art suivant.

Les auteurs de [WGW07] et [WG07] proposent de chiffrer la marque avec la suite logistique : on convertit cette dernière en suite de bits, et l'on effectue alors le *ou exclusif* bit à bit avec le filigrane original. Puis, pour chaque bit de la marque chiffrée, on choisit le pixel de l'hôte qui contiendra la marque en itérant le « chat d'Arnold », et l'on itère à nouveau la suite logistique pour déterminer quel

bit de poids faible du pixel choisi sera remplacé par le bit de la marque chiffrée. Cet exemple a déjà été cité en partie au chapitre 18.

Dans [LX07], les auteurs proposent eux-aussi d'utiliser le *ou exclusif* avec une suite logistique pour chiffrer la marque. Ils l'insèrent ensuite dans le domaine fréquentiel, via la transformée en ondelettes, de la manière suivante. Soit A l'image hôte, et :

$$dwt(A) = \begin{bmatrix} cA & cH \\ cV & cD \end{bmatrix}$$

sa transformée en ondelettes discrètes. Les auteurs découpent la marque chiffrée en 4 : $D_1, \ldots, D_4$, et se fixent un facteur de dissimulation K. L'image tatouée est alors la transformée inverse de B', qui est construit comme suit :

$$B' = \begin{bmatrix} cA + K \times D_1 & cH + K \times D_2 \\ cV + K \times D_3 & cD + K \times D_1 \end{bmatrix},$$

et qui est ramenée à des valeurs entières.

Les auteurs de [ZZX+04] proposent de procéder de la manière suivante pour tatouer une image donnée. Tout d'abord, mélanger le filigrane avec le chat d'Arnold. Soit n le nombre de lignes de la marque ; calculer les n premiers termes de la suite logistique, mise sous la forme équivalente suivante : $x^0 = 0,3$, $x^{k+1} = 1 - 2\left(x^k\right)^2$, en arrondissant chaque itérée à la quatrième décimale. Passer cette suite à travers un réseau de neurones de type perceptron multi-couches, à trois couches de n neurones, pour « rendre la clé de chiffrement plus difficile à interpréter » et « le marquage plus robuste ». Faire le *ou exclusif* du vecteur issu du réseau avec chaque vecteur colonne de la marque : on obtient W_d. Alors, transformer en ondelettes discrètes (DWT) l'image originale, et ne considérer que la « troisième composante » (dont le sens précis n'est pas donné). Séparer cette troisième composante en blocs de même taille que W_d, et insérer W_d dans chacun de ces sous-blocs $X_b^i : X_b^{i\prime} = X_b^i + \alpha W_d$, où α est l'intensité du marquage (s'il est grand, la marque sera plus robuste, mais plus visible). Calculez finalement la DWT inverse, après avoir refusionné les sous-blocs.

Les auteurs de [CJQZ06] proposent d'itérer une suite logistique x^k, avec $\mu = 4$ et $x^0 = 0,123$, de se fixer $T \in [0,1]$, et de transformer x^k en suite x'^k de bits : si $x^k < T$, alors $x'^k = 0$, sinon $x'^k = 1$. On doit alors chiffrer la marque en réalisant le *ou exclusif* de x'^k avec cette dernière, les bits étant pris dans un ordre différent que l'ordre naturel, ce qui donne une suite w^i de bits. Les auteurs proposent ensuite de décomposer l'image hôte en ondelettes (Daubechies-1), ce qui fournit une matrice de compression et trois matrices de détails. On doit alors recommencer récursivement, trois autres fois, la décomposition de chacune de ces quatre matrices, et considérer les matrices de détails du niveau 3. Pour chacune des matrices (par exemple, la $i-$ième) de ce niveau 3, on doit augmenter son coefficient de αw^i, où α contrôle l'énergie de la marque insérée : plus α est grand, plus la marque est visible, et plus elle est robuste. Reste alors à recomposer l'image pour obtenir l'hôte tatoué.

Enfin, les auteurs de [CXZ06] proposent un tatouage fragile/robuste, défini de la manière suivante. Générer une marque fragile à l'aide d'une suite logistique, et mélanger une autre marque, dite *robuste* à l'aide d'une autre suite logistique. Puis, découper l'image hôte en N blocs ne se chevauchant pas, et pour chaque bloc :

1. Insérer un bit de la marque robuste en position (1,1) dans la matrice DCT.
2. Insérer un bit de la marque fragile en position (1,2) dans la matrice DCT.
3. Calculer la DCT-II inverse de la matrice résultante, et remplacer le bloc considéré par ce nouveau bloc.

Des approches similaires à ce qui précède peuvent encore être trouvées dans [LYGL07] et [DGW04].

2. Critique de l'existant

L'état de l'art ci-dessus illustre les problèmes signalés au chapitre 18. En effet, dans ces documents, il est implicitement supposé que l'utilisation de suites chaotiques en tant que composants de l'algorithme, suffit à obtenir un comportement globalement chaotique pour l'algorithme (c'est du moins ce qu'annoncent les titres de ces papiers). Toutefois, comme nous l'avions signalé, cette hypothèse générale ne nous semble pas triviale, et devrait être prouvée.

D'autre part, ni la définition du chaos ni son intérêt dans le contexte considéré ne sont donnés. Et ce terme est souvent confondu avec la sensibilité aux conditions initiales. Pour illustrer notre propos, nous donnons ci-dessous les définitions de chaos que l'on peut relever dans ces articles :

- « Chaotic sequence is regarded as a noise-like spread sequence. »
- « Chaotic system is a kind of complicated, nonlinear dynamical system. »
- « Chaos is an aperiodic, non-convergent process, which sensitively depends on the initial state. It is determined but stochastic-like and exists in the nonlinear dynamic system. »
- « Chaotic maps are very sensitive to initial conditions, in the sense that two logistic sequences generated from different initial conditions are uncorrelated statistically. »
- « Chaos means that the system obeys deterministic laws of evolution, but the outcome is highly sensitive to small uncertainties in the specification of the initial state. If a deterministic system is locally unstable and globally mixing, it is said to be chaotic. » (Les termes « locally unstable » et « globally mixing » ne sont pas définis.)

De plus, la question : « Quelles propriétés chaotiques sont nécessaires pour atteindre des objectifs tels que la robustesse, la sécurité ou l'authentification ? » n'est jamais soulevée dans ces documents.

Un autre problème, propre à cette thématique, nous semble être le fait que le chaos est utilisé pour obtenir de la robustesse, alors que nous ne voyons pas vraiment le lien entre ces deux notions : pourquoi chiffrer le filigrane en utilisant un algorithme chaotique, ou insérer la marque de manière chaotique dans le support hôte, rendrait ce tatouage résistant aux attaques aveugles de type redimensionnement, découpage, *etc.* ? Pourtant, de telles affirmations se trouvent dans ces papiers. L'indéterminisme et l'imprévisibilité qu'il pourrait y avoir concernant les lieux de l'hôte utilisés pour l'insertion du filigrane, nous semblent clairement plus utiles pour contrer des attaques intelligentes, malicieuses. Aussi, nous avons proposé au chapitre 20 d'utiliser la théorie du chaos pour étudier, sous un certain angle, la sécurité des méthodes de dissimulation.

Enfin, la propriété du chaos est réduite à la simple utilisation d'une suite logistique ou du chat d'Arnold, soit pour le chiffrement du filigrane, soit pour sélectionner les coefficients à altérer. Cependant, comme nous l'avons signalé au chapitre 18, la suite logistique n'est pas suffisamment sûre pour des applications cryptographiques [AAF08], le chat d'Arnold n'a, à notre connaissance, pas été suffisamment étudié dans ce domaine. De plus, en raison de la finitude de la mémoire d'un ordinateur, seule une sorte de « chaos discret » est généré par ces suites.

Dernier problème, les conséquences de ces faits ne sont jamais discutés.

3. Notre approche

Nous avons tenté d'expliquer au chapitre 18 comment apporter une réponse à chacun des problèmes soulevés ci-dessus. Reprenons nos explications, en les adaptant au contexte particulier de la dissimulation par chaos.

Tout d'abord, les diverses notions de chaos utilisées dans le présent document ont été clairement définies dans la partie II. Ces notions ont été à plusieurs reprises (aux chapitres **??** à 21) reliées aux objectifs que l'on souhaite atteindre dans la dissimulation d'informations, et le seront à nouveau dans ce qui suit. De plus, respectant la méthode que l'on s'est imposée au chapitre 18, l'algorithme de dissimulation que nous allons définir ici va se résumer à effectuer des itérations chaotiques sur le média, cela et rien de plus [8]. On pourra en déduire que cet algorithme possède les différentes propriétés établies dans la partie III, et toutes les conséquences de la chaos-sécurité, évoquées aux chapitres **??** à 21, seront aussi applicables à notre algorithme. Les outils qualitatifs et quantitatifs issus de la théorie du chaos nous permettront de préciser, à la section 22.3, le niveau exact de chaos-sécurité de notre algorithme. La stégo-sécurité et la robustesse seront quant à elles étudiées respectivement à la section 22.3 et au chapitre 23, rendant ainsi possible la comparaison entre notre algorithme et l'étalement de spectre. Rappelons enfin que les itérations chaotiques, l'outil constituant notre algorithme, ne perdent pas leurs propriétés chaotiques quand on les implémente (*c.f.* chapitre 18).

II. L'ALGORITHME DHCI

1. Les médias numériques

a. Définitions

Nous nous intéressons dans ce qui suit à l'ensemble des médias numériques, que nous avons choisi de définir de la manière suivante :

DÉFINITION IV.17 (MÉDIAS NUMÉRIQUES) : L'*ensemble des médias numériques*, noté $\mathfrak{M}$, est l'ensemble des suites de $\mathbb{B}^{\mathbb{N}}$ nulles sauf en un nombre fini d'indices. Un *média numérique* est un élément de $\mathfrak{M}$. ◊

DÉFINITION IV.18 (TAILLE D'UN MÉDIA) : La *taille* d'un média numérique x, notée $|x|$, est l'indice de son dernier terme non nul. ◊

NOTATION IV.2. L'ensemble des médias numériques de taille n sera noté $\mathfrak{M}_n$.

b. Acquisition

Ces médias numériques sont acquis à partir d'un appareil qui capture des objets :

DÉFINITION IV.19 (FONCTION D'ACQUISITION) : Soit $\mathcal{X}$ un ensemble. Nous appellerons *fonction d'acquisition sur* $\mathcal{X}$ toute fonction d'un ensemble $\mathcal{X}$ vers l'ensemble des médias numériques. ◊

Cette fonction n'est pas l'objet de notre étude, nous supposons juste qu'elle existe pour tout ensemble $\mathcal{X}$, ce qui se montre avec l'axiome du choix.

c. Signification

Une fois l'acquisition faite, la description d'un élément de $\mathcal{E}$ sous la forme d'un média numérique peut n'être pas quelconque, mais telle que certaines parties du média numérique « nous parlent plus » que d'autres :

8. Pour être rigoureusement exact, nous proposerons plusieurs variantes à notre algorithme ; la plus élémentaire se réduira aux IC, donc sera clairement chaotique, tandis que la plus complexe proposera notamment le chiffrement et l'authentification, mais cependant s'éloignera de la chaos-sécurité au sens strict.

Définition IV.20 (Fonction de signification) : On appelle *fonction de signification* toute suite de réels tendant vers 0. ◊

Cette fonction de signification permet donc d'attacher un poids éventuellement différent à chaque coefficient d'un média numérique (*i.e.* à chaque terme de la suite), suivant sa position dans la suite de bits et sa signification à nos yeux. La convergence vers 0 est là pour signifier que dans le cas d'un algorithme œuvrant sur un flux de données très grand, les termes les plus éloignés de ce flux ne peuvent pas avoir beaucoup d'importance, ou encore que l'homme n'appréhende pas l'infini.

d. Exemple

Supposons par exemple que nous prenions des photographies en niveaux de gris, avec notre appareil photo numérique, et que l'on récupère des images matricielles (« bitmap », par exemple des fichiers *pgm* binaires). Ce procédé physique se traduit par une fonction d'acquisition qui se conçoit aisément, mais dont la définition exacte, elle, est relativement malaisée à formuler.

Cela étant établi, nous avons quand même bien là un ensemble de médias numériques, et à chaque photographie est associé un élément de cet ensemble. Chaque média numérique correspond à ce à quoi l'on s'attend, à savoir à la description binaire des niveaux de gris de l'image, pixel après pixel, chaque bloc de 8 bits étant associé au niveau de gris d'un pixel donné.

Soit $u^n = 8 - (n \bmod 8)$ si $n < 10^{70}$, et $u^n = 0$ sinon, une fonction de signification. Elle nous dit que dans les premiers termes de chaque média numérique, regroupés par groupes de 8 bits représentant un pixel, le premier bit de chaque pixel a une importance 8, quand le dernier a une importance 1 : dans une description en 256 niveaux de gris de chaque pixel, le changement du premier bit affecte beaucoup plus l'image que le changement du huitième bit.

e. Discussion pratique

Le choix de la fonction de signification, suivant la fonction d'acquisition choisie, va en général de soi dans la plupart des cas pratiques, et ne diffère souvent que d'un facteur d'échelle et de la rapidité à atteindre 0 :

- Dans les images JPEG, ce sont les coefficients supérieur gauche de chaque matrice DCT 8×8 qui ont le plus de poids.
- Dans le JPEG2000, c'est la matrice des coefficients DWT d'approximation.
- Etc.

Il s'agit là de discussions pratiques qui ne sont pas vraiment le propos de ce document : nous nous contenterons de supposer que ces fonctions-là existent, et qu'elles ont été bien choisies.

2. Représentation des contenus

a. Le modèle de représentation

Définition IV.21 (Modèle de représentation) : On appelle *modèle de représentation* tout triplet $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M, m)))$, où :

- $\mathcal{X}$ et $\mathcal{Y}$ sont des ensembles,
- $\mathcal{A}_1$ (resp. $\mathcal{A}_2$) est une fonction d'acquisition sur $\mathcal{X}$ (resp. $\mathcal{Y}$),
- s est une fonction de signification,
- $M, m \in \mathbb{R}^*$. ◊

b. L'hôte et le filigrane

On peut redéfinir les notions d'hôte et de filigrane dans ce modèle de représentation :

Définition IV.22 (Hôtes et filigranes) : Soit $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M, m)))$ un modèle de représentation. Alors :

- L'ensemble des acquisitions $\mathcal{A}_1(\mathcal{X})$ est appelé *ensemble des hôtes*,
- L'ensemble des acquisitions $\mathcal{A}_2(\mathcal{Y})$ est appelé *ensemble des filigranes*. ◊

On peut maintenant définir les coefficients « les plus » et « les moins » significatifs d'un hôte donné.

c. Les coefficients « les plus » et « les moins » significatifs

Définition IV.23 (Signification des coefficients) : Soit $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M, m)))$ un modèle de représentation, et $x \in \mathcal{A}_1(\mathcal{X})$ un hôte. Alors :

- Le vecteur des *coefficients les plus significatifs* de x est $m(x) = \left(x^n \,\middle/\, n \in \mathbb{N} \text{ et } s(n) \geqslant M\right)$.
- Le vecteur des *coefficients les moins significatifs* de x est $l(x) = \left(x^n \,\middle/\, n \in \mathbb{N} \text{ et } s(n) \leqslant m\right)$.
- Le vecteur des *coefficients passifs* de x est $u(x) = \left(x^n \,\middle/\, n \in \mathbb{N} \text{ et } s(n) \in]m; M[\right)$ ◊

Notation IV.3. $m(x)$ et $l(x)$ sont aussi appelés, respectivement, MSCs (*most significant coefficients*) et LSCs (*least significant coefficients*) de x.

En d'autres termes, pour une image donnée,

- Les coefficients les plus significatifs sont des coefficients qui permettent de décrire la partie portant le plus d'informations de l'image, c'est-à-dire sa partie la plus riche, selon la fonction de signification choisie.
- Par coefficients les moins significatifs, nous entendons une traduction de certaines parties insignifiantes d'un média, sous la forme d'une suite de bits (insignifiant peut être compris ainsi : « qui peut être altéré sans dommage apparent »).

La figure 22.1 fournit une illustration de ces notions. Lena, un mannequin célèbre fréquemment utilisé en traitement du signal, est représentée en 22.1(a) sous la forme d'une image bitmap en 256 niveaux de gris. Les figures 22.1(b) et 22.1(c) représentent respectivement des MSCs et des LSCs de cette représentation : la première est constituée des 3 premiers bits de chaque pixel de Lena, quand la seconde est obtenue à partir des 4 derniers bits (les niveaux de gris des LSCs ont été multipliés par 17).

On peut alors décomposer toute image selon ces différents coefficients...

d. Décomposition et recomposition des hôtes

Définition IV.24 (Ensemble des décompositions des hôtes) : Soient un modèle de représentation $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M, m)))$, $\mathfrak{B}$ l'ensemble des suites finies de booléens, et $\mathfrak{N}$ l'ensemble des suites finies d'entiers. On appelle *ensemble des décompositions des hôtes* de $\mathcal{M}$, l'ensemble $\mathcal{D}$ des sextuplets $(m, l, u, \varphi_1, \varphi_2, \varphi_3)$ de $\mathfrak{B} \times \mathfrak{B} \times \mathbb{B}^{\mathbb{N}} \times \mathfrak{N} \times \mathfrak{N} \times \mathbb{N}^{\mathbb{N}}$ tels que :

- $\varphi_1, \varphi_2, \varphi_3$ sont strictement croissantes,
- les suites m et φ_1 ont le même nombre de termes,
- les suites l et φ_2 ont le même nombre de termes. ◊

PROPOSITION IV.8 (DÉCOMPOSITION ET RECOMPOSITION DE L'HÔTE) : *Soient un modèle de représentation* $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M, m)))$ *, et* $x \in \mathcal{A}_1(\mathcal{X})$ *un hôte. Il existe trois suites,* $\varphi_{m(x)}, \varphi_{l(x)}, \varphi_{u(x)}$ *strictement croissantes et à valeurs dans* $\mathbb{N}$*, telles que :*

- $m(x) = \left(x^{\varphi^1_{m(x)}}, x^{\varphi^2_{m(x)}}, \ldots, x^{\varphi^{|m(x)|}_{m(x)}}\right)$,
- $l(x) = \left(x^{\varphi^1_{l(x)}}, x^{\varphi^2_{l(x)}}, \ldots, x^{\varphi^{|l(x)|}_{l(x)}}\right)$,
- $u(x) = \left(x^{\varphi^1_{u(x)}}, x^{\varphi^2_{u(x)}}, \ldots\right)$.

Pour un modèle de représentation donné, et pour tout hôte x *de ce modèle, l'application*

$$dec_{\mathcal{M}} : x \in \mathcal{A}_1(\mathcal{X}) \mapsto \left(m(x), l(x), u(x), \varphi_{m(x)}, \varphi_{l(x)}, \varphi_{u(x)}\right) \in \mathcal{D}$$

est bijective. Sa réciproque est l'application :

$$rec_{\mathcal{M}} : (m, l, u, \varphi_1, \varphi_2, \varphi_3) \in \mathcal{D} \mapsto x = \sum_{i=1}^{|m|} m^i e_{\varphi_1(i)} + \sum_{i=1}^{|l|} l^i e_{\varphi_2(i)} + \sum_{i=1}^{+\infty} u^i e_{\varphi_3(i)} \in \mathcal{A}_1(\mathcal{X}),$$

où e_i *est la suite dont le terme* j *vaut* $\delta(i, j)$*, c'est-à-dire que* $(e_i)_{i \in \mathbb{N}}$ *est la base usuelle du* $\mathbb{R}-$*espace vectoriel* $\left(\mathbb{N}^{\mathbb{N}}, +, .\right)$*. On dit alors que* $dec_{\mathcal{M}}$ *est la* fonction de décomposition des hôtes *associée au modèle de représentation* $\mathcal{M}$*, et* $rec_{\mathcal{M}}$ *en est sa* fonction de recomposition.

REMARQUE. Les vecteurs $(m(x), l(x), u(x))$ contiennent les valeurs des coefficients de x quand les fonctions $(\varphi_{m(x)}, \varphi_{l(x)}, \varphi_{u(x)})$ permettent de ranger ces coefficients dans le bon ordre et au bon endroit, recomposant ainsi x.

e. Utilité des MSCs et des LSCs

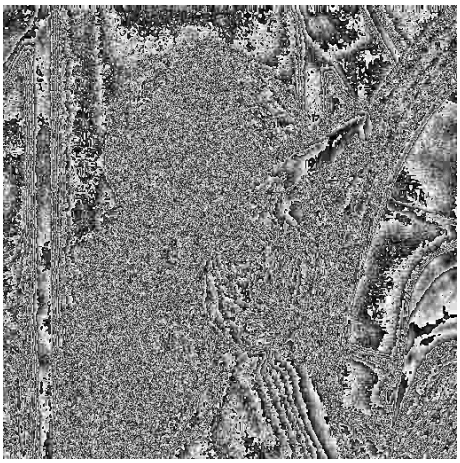
Dans ce qui suivra, les LSCs seront utilisés pendant la phase dite d'embarquement du filigrane : certains LSCs de l'image hôte seront choisis de manière imprévisible, et seront « altérés », d'une manière ou d'une autre, par les bits du filigrane. Les MSCs ne serviront qu'en cas d'authentification ; l'embarquement du filigrane dépendra alors de ces coefficients. Ainsi, un coefficient ne doit pas être défini en même temps comme un MSC et un LSC : les seconds peuvent être altérés, alors que les premiers peuvent s'avérer nécessaires pour extraire le filigrane.

Les LSCs et MSCs peuvent apparaître, de prime abord, comme l'approche inverse de Cox *et al.* [CMM99]. Toutefois nous n'imposons pas que l'ensemble des MSCs soit le complémentaire de l'ensemble des LSCs. En outre, la manière de définir les LSCs et MSCs peut être secrètement partagée avant le tatouage (en utilisant soit un canal secret, soit un système de chiffrement à clé publique), ainsi un pirate ne peut pas concentrer ses attaques uniquement sur les LSCs. D'autre part, la définition des LSCs devrait être telle que :

- Un petit nombre d'altérations ne change pas foncièrement l'image.
- Leur suppression complète détruirait grandement le support hôte.

Enfin, l'ensemble des LSCs devrait être très grand devant le message à cacher. Ces divers aspects devront être étudiés en profondeur au moment de la mise en pratique effective du dhCI.

(a) Léna (b) MSCs : les 3 premiers bits de chaque pixel



(c) LSCs : les 4 derniers bits (facteur 17)

FIGURE 22.1 – Exemple de MSCs et LCSs de Lena.

3. Modes et configurations

a. Le mode

DÉFINITION IV.25 (MODE) : On appelle *mode* toute application qui à $n \in \mathbb{N}$ associe une application de $\mathbb{B}^n$ dans $\mathbb{B}^n$. ◊

Exemple IV.1 : Le *mode négation* sera l'application qui à l'entier n associe la négation vectorielle : $\mathbb{B}^n \to \mathbb{B}^n$.

b. La configuration

DÉFINITION IV.26 (CONFIGURATION) : On appelle *configuration* toute fonction de $\mathbb{N}$ dans l'ensemble des suites entières, qui à un entier n associe une suite de $[\![1, n]\!]^{\mathbb{N}}$ de limite nulle. ◊

Donnons tout de suite deux exemples de configuration, qui nous serviront au chapitre prochain. La première fait appel à la fonction chaotique linéaire par morceaux, que l'on rappelle ci-dessous :

Définition IV.27 (Fonction chaotique linéaire par morceaux) : Soit $p \in]0;0,5[$ un paramètre de contrôle. La *fonction chaotique linéaire par morceaux* est la fonction F définie par [AAF08, SQW+01] :

$$F(x,p) = \begin{cases} \dfrac{x}{p} & x \in [0;p], \\ \dfrac{x-p}{\frac{1}{2}-p} & x \in [p;\frac{1}{2}], \\ F(1-x,p) & x \in [\frac{1}{2};1]. \end{cases}$$

Définition IV.28 (Configuration CIIS) : Soient $(K,M,p,N) \in [0,1] \times [0,1] \times]0;0,5[\times \mathbb{N}$. La configuration *CIIS* (*Chaotic Iterations with Independent Strategy*) de paramètres (K,M,p,N) est la fonction qui à N associe la suite $(S^n)_{n\in\mathbb{N}}$ définie par :

- $K^0 = M \oplus K$: K^0 est le réel dont l'écriture binaire s'obtient en réalisant le *ou exclusif* bit à bit entre les écritures binaires de M et de K,
- $\forall n \leqslant N, K^{n+1} = F(K^n, p)$,
- $\forall n \leqslant N, S^n = \lfloor \mathsf{N} \times K^n \rfloor + 1$,
- $\forall n > N, S^n = 0$. ◊

Remarque. La configuration CIIS semble un peu artificiellement compliquée. Ses paramètres seront directement reliés au problème de la dissimulation, justifiant la raison pour laquelle cette configuration a cette forme. Elle sera fondamentale dans l'étude de sécurité à venir.

Définition IV.29 (La configuration CIDS) : La configuration *CIDS* (*Chaotic Iterations with Dependent Strategy*) de paramètres $(N,X) \in \mathbb{N} \times \mathbb{B}^{\mathbb{N}}$ est la fonction qui à $\mathsf{N} \in \mathbb{N}$ associe la suite $(S^n)_{n\in\mathbb{N}}$ définie comme suit :

- $\forall k \leqslant N$, si $k \leqslant \mathsf{N}$ et $X^k = 1$, alors $S^k = k$, sinon $S^k = 1$.
- $\forall k > N, S^k = 0$. ◊

4. L'algorithme dhCI

a. L'embarquement des données

Définition IV.30 (Embarquement) : Soit $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M,m)))$ un modèle de représentation, $x \in \mathcal{A}_1(\mathcal{X})$ un hôte, et y un média numérique de taille $|l(x)|$. On dit que z est le résultat de l'*embarquement* de y dans x lorsque :

$$\forall n \in \mathbb{N}, z^n = \begin{cases} x^n & \text{si } s(n) > m, \\ y^n & \text{sinon.} \end{cases}$$

En d'autres termes, l'embarquement est la fonction qui à tout x de $\mathcal{A}_1(\mathcal{X})$ et tout y de $\mathfrak{M}_{|l(x)|}$ associe l'élément $rec_{\mathcal{M}}(m,y,u,\varphi_1,\varphi_2,\varphi_3)$ de $\mathfrak{M}$, où $(m,l,u,\varphi_1,\varphi_2,\varphi_3) = dec_{\mathcal{M}}(x)$. ◊

Pour faire simple, l'embarquement consiste à remplacer les LSCs $l(x)$ de x par y. Il peut être authentifié :

Définition IV.31 (Embarquement authentifié) : Soit $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M,m)))$ un modèle de représentation, $x \in \mathcal{A}_1(\mathcal{X})$ un hôte, et y un média numérique de taille $|l(x)|$. On parle d'*embarquement authentifié dans* x lorsque le y que l'on embarque est fonction de $m(x)$. ◊

En d'autres termes, il y a authentification lorsque le filigrane que l'on embarque dans x dépend des coefficients les plus significatifs de x.

b. Le dhCI

Nous avons maintenant défini tout ce qu'il nous fallait pour donner notre algorithme de dissimulation dhCI :

Définition IV.32 (Le dhCI) : Soit $\mathcal{M} = (\mathcal{X} \times \mathcal{Y}, (\mathcal{A}_1, \mathcal{A}_2), (s, (M, m)))$ un modèle de représentation. La *dissimulation dhCI de mode f et de configuration $S(p)$*, dépendante d'un paramètre p, est l'application qui à tout couple $(x, y) \in \mathcal{A}_1(\mathcal{X}) \times \mathcal{A}_2(\mathcal{Y})$ associe le média numérique z, résultant de l'embarquement de $\overline{y}$ dans x, où $\overline{y}$ est le $|S(y)^{|l(x)|}|$−ième terme des itérations chaotiques $G_{f(|l(x)|)}$ de condition initiale $\left(S(y)^{|l(x)|}, l(x)\right)$. ◊

Plus clairement peut-être, mais sûrement moins rigoureusement : on prend les coefficients les moins significatifs de notre hôte, on réalise dessus des itérations chaotiques (le mode fournit la fonction d'itération, la configuration en donne la stratégie), et on réinjecte la dernière itérée obtenue à la place des coefficients les moins significatifs de l'hôte.

La difficulté dans la formalisation provient du fait que, comme on ne souhaite pas fixer la taille des médias (sans quoi on perdrait notre chaos), on a donc un nombre variable de LSCs. Or, ces derniers fournissent le système sur lequel itérer : la fonction d'itération et les termes de la stratégie doivent donc s'adapter à cette taille, d'où les définitions IV.25 et IV.26.

III. Chaos et stégo-sécurité du dhCI

Un système de tatouage qui n'est pas chaos-sûr ne doit pas être utilisé dans des configurations KMA et KOA, vu qu'il est prévisible. Toutefois, prouver qu'un algorithme est chaos-sûr n'est, on l'a déjà signalé, que le début de l'étude. La prochaine étape consiste à évaluer la qualité de son comportement chaotique, à l'aide des nombreux outils qualitatifs et quantitatifs proposés par la théorie du chaos, comme cela a été fait en partie au chapitre 21.

Ces outils vont nous permettre de comparer l'étalement de spectre à la technique dhCI, contribuant ainsi à décider lequel des deux systèmes peut être utilisé dans une configuration donnée. Pour réussir cela, les propriétés qualitatives et quantitatives de chaos rappelées à la partie II vont être utilisées une fois encore, pour évaluer l'imprévisibilité des itérations chaotiques G_{f_0}. Ces propriétés seront possédées par notre algorithme, lorsque la négation vectorielle est choisie.

1. Chaos-sécurité

Nous reprenons dans cette section l'ensemble des propriétés chaotiques des itérations du même nom. Elle permettent d'évaluer la chaos-sécurité du dhCI.

a. Étude qualitative

Pour commencer, comme les itérations chaotiques sont du chaos selon Devaney (théorème III.5), on peut en déduire que :

Théorème IV.8 : *La technique dhCI de dissimulation d'information est chaos-sûre.*

L'étude menée dans la partie III permet de préciser le niveau de chaos-sécurité du dhCI. Commençons par l'étude qualitative de sa chaos-sécurité :

PROPOSITION IV.9 : *La technique dhCI de dissimulation de l'information possède les qualités de transitivité forte et de mélange topologique.*

Ces propriétés de chaos-sécurité, provenant de la proposition III.16 et du théorème III.10, sont partagées avec l'étalement de spectre. Leurs conséquences concernant la capacité à faire face à diverses attaques, ont été discutées notamment à la section 21.3.1 dans le cadre de l'étalement de spectre.

Nous énonçons maintenant d'autres propriétés qualitatives de chaos-sécurité, qui peuvent se déduire de l'étude menée à la partie III (théorèmes III.11 et III.12) :

PROPOSITION IV.10 : *La dissimulation dhCI embarque le tatouage au sein de son hôte de manière imprévisible, suivant l'acception de cette notion dans les théories de Knudsen et de Li-Yorke.*

REMARQUE. Nous n'avons pas encore eu le temps de regarder précisément quelles pouvaient être les conséquences concrètes de ces qualités, et nous ne savons pas encore si l'étalement de spectre les possède.

b. Étude quantitative

Venons-en à l'évaluation quantitative de la chaos-sécurité du dhCI, en commençant par sa sensibilité :

PROPOSITION IV.11 : *La dissimulation dhCI est sensible aux conditions initiales, et sa constante de sensibilité est égale à* $\mathsf{N}-1$.

Cela provient de la proposition III.17. Enfin, contrairement aux techniques d'étalement de spectre, le dhCI est expansif (théorème III.9) :

PROPOSITION IV.12 : *La dissimulation dhCI est expansive. Sa constante d'expansivité vaut 1.*

Pour résumer, la dissimulation dhCI est chaos-sûre, avec les propriétés supplémentaires de transitivité forte et de mélange topologique, comme c'est le cas des techniques d'étalement de spectre. Toutefois, contrairement à ces dernières, les itérations chaotiques ont une bonne constante de sensibilité, et sont expansives (avec une constante d'expansivité égale à 1). Nous pouvons donc en conclure que les itérations chaotiques sont plus sûres que l'étalement de spectre dans les configurations KOA et KMA. D'autres propriétés de chaos-sécurité ont été établies dans la partie III, telles que l'entropie infinie et l'exposant de Lyapunov égal à $\ln(\mathsf{N})$. Nous ne nous étendrons pas dessus, bien que ces résultats soient importants : nous n'avons pas encore eu le temps de les exploiter.

2. Stégo-sécurité

a. Le cadre

Pour réaliser l'étude de stégo-sécurité du dhCI, nous avons besoin de rappeler les paramètres dont ce dernier dépend. Soient :

- $(K, N) \in [0;1] \times \mathbb{N}$ les clés de tatouage,
- $X \in \mathbb{B}^{\mathsf{N}}$ les N coefficients les moins significatifs (LSCs) d'une couverture C donnée,

- $(S^n)_{n\in\mathbb{N}} \in [\![1,\mathsf{N}]\!]^{\mathbb{N}}$ une stratégie (la configuration), qui dépend du message à cacher $M \in [0;1]$, et de K,
- $f_0 : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ la négation vectorielle (le mode).

Alors l'image tatouée est C, dont les LSCs ont été remplacé par $Y_K = X^N$, où :

$$\begin{cases} X^0 = X \\ \forall n < N, X^{n+1} = G_{f_0}\left(X^n\right). \end{cases}$$

b. La stégo-sécurité

Nous allons maintenant étudier la stégo-sécurité de la dissimulation dhCI. Nous allons prouver que :

THÉORÈME IV.9 : *La dissimulation dhCI, en mode négation et en configuration CIIS, est stégo-sûre.*

PREUVE : *Supposons que X suit une loi uniforme sur $\mathbb{B}^{\mathsf{N}}$: $X \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$, c'est-à-dire que $\forall e \in \mathbb{B}^{\mathsf{N}}, p(X = e) = \dfrac{1}{2^{\mathsf{N}}}$, pour le dhCI en configuration CIIS. Nous allons prouver par récurrence que $\forall n \in \mathbb{N}, X^n \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$.*

- *L'initialisation est immédiate, puisque $X^0 = X \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$.*
- *Supposons que l'hypothèse $X^n \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$ est vrai pour un certain n. Soit $e \in \mathbb{B}^{\mathsf{N}}$ et $\mathbf{B}_k = (0,\cdots,0,1,0,\cdots,0) \in \mathbb{B}^{\mathsf{N}}$ (le chiffre 1 est en position k). Alors $p\left(X^{n+1} = e\right) = \sum_{k=1}^{\mathsf{N}} p\left(X^n = e + \mathbf{B}_k, S^n = k\right)$. Ces deux événements sont indépendants dans la configuration CIIS, donc :*

$$p\left(X^{n+1} = e\right) = \sum_{k=1}^{\mathsf{N}} p\left(X^n = e + \mathbf{B}_k\right) \times p\left(S^n = k\right).$$

Selon l'hypothèse de récurrence : $p\left(X^{n+1} = e\right) = \frac{1}{2^{\mathsf{N}}} \sum_{k=1}^{\mathsf{N}} p\left(S^n = k\right)$. L'ensemble d'événements $\{S^n = k\}$ pour $k \in [\![1;\mathsf{N}]\!]$ est une partition de l'univers des possibles, donc $\sum_{k=1}^{\mathsf{N}} p\left(S^n = k\right) = 1$. Finalement, $p\left(X^{n+1} = e\right) = \frac{1}{2^N}$, ce qui conduit à $X^{n+1} \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$.

Ce résultat est vrai $\forall n \in \mathbb{N}$, donc pour N : nous avons prouvé que, $\forall K \in [0;1], Y_K = X^N \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$ quand $X \sim \mathbf{U}\left(\mathbb{B}^{\mathsf{N}}\right)$. D'où le résultat.

Nous allons maintenant prouver que,

THÉORÈME IV.10 : *La dissimulation dhCI, en mode négation et en configuration CIDS, n'est pas stégo-sûre.*

TABLE 22.1 – Évaluation de la dissimulation dhCI en mode négation

CAPACITÉ	Tatouage aveugle	Non
	Domaine	Spatial, fréquentiel
	Authentification	Capable
	Chiffrement du filigrane	Oui
	Robustesse	Voir chapitre 23
CHAOS-SÉCURITÉ	Chaos selon Devaney	Oui
	Espace des phases	$\mathcal{X}' = [\![1;\mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$
	Topologie	d'
	Transitivité forte	Oui
	Mélange topologique	Oui
	Sensibilité	Oui. $\delta = \mathsf{N} - 1$
	Expansivité	Oui. $\varepsilon = 1$
STÉGO-SÉCURITÉ	Stego-sécurité	Oui, en configuration CIIS

PREUVE : *Du fait de la définition de la configuration CIDS, nous avons* $P(Y_K = (1,1,\cdots,1)) = 0$. *Donc il n'y a pas répartition uniforme pour les stégo-contenus* Y_K.

Les aspects de sécurité du dhCI sont résumés dans le tableau 22.1.

IV. L'EXTRACTION

La définition IV.32 de la dissimulation dhCI ne traite que de l'embarquement du filigrane, c'est-à-dire uniquement de l'étape de tatouage. C'est notre unique sujet d'étude : nous ne nous intéressons qu'à la sécurité du tatouage, et nous verrons au chapitre suivant qu'elle est bien présente. Cela dit, dissimuler de l'information de manière sûre, c'est bien, pouvoir en faire quelque chose, c'est encore mieux.

Nous donnons ici deux pistes permettant d'arriver à tirer parti de l'information que l'on a dissimulée, suivant la manière d'utiliser la dissimulation dhCI.

Considérons tout d'abord le cas où le mode est la négation vectorielle. On se doute qu'alors la dissimulation sera très sûre, vues les propriétés des itérations chaotiques G_{f_0}, mais le problème est que l'on ne peut pas faire marche arrière. Pour exploiter cette dissimulation, il faut donc posséder l'hôte et le filigrane d'origine : on embarque notre filigrane dans l'objet tatoué, et l'on doit retrouver notre hôte d'origine. On pourrait de prime abord penser qu'un tel tatouage ne sert pas à grand chose, mais il n'en est rien :

- On peut ainsi prouver être à l'origine d'un document circulant sur internet, embarquer sa propre signature (problème de la détection du filigrane).
- On peut ainsi créer un canal caché et transmettre des informations binaires. Par exemple, Alice et Bob surveillent une image sur Internet, regardant périodiquement si le tatouage (le signal) n'est pas apparu dans l'image. Comme ils partagent le filigrane tel une clé secrète, ils ne craignent pas les faux signaux.
- Un média numérique étant une suite de bits, le procédé ci-dessus peut être reproduit plusieurs fois, avec des filigranes différents...
- Ce tatouage non-aveugle peut aussi être utile dans les réseaux, pour la détection d'intrusion ou l'anonymat [HKB09].

- Un autre exemple d'utilisation d'un tel tatouage est donnée dans [GTOMD05], afin de protéger des données numériques transitant par internet.

La deuxième solution consiste à placer directement son message dans les LSCs de l'hôte, puis à réaliser la dissimulation dhCI sur l'hôte ainsi enrichi. Si l'on utilise des modes inversibles, il suffira d'inverser la configuration pour retrouver le message d'origine. Nous n'entrerons pas plus dans les détails, laissant la pleine exploitation de la dissimulation dhCI pour plus tard.

CHAPITRE 23

Une étude pratique, avec variantes, du dhCI

> Je préfère la lumière du couchant sur mes mains à l'or que sa majesté le propose.
>
> *Tous les matins du monde*
> PASCAL QUIGNARD

Dans ce chapitre, notre but est de donner quelques réalisations concrètes de la dissimulation dhCI, d'en présenter quelques variantes accroissant les possibilités offertes par cet algorithme (au prix d'une perte éventuelle de son haut niveau de sécurité), et de finir par les prémices d'une étude de robustesse. Il ne s'agira pas à proprement parler de preuves de robustesse dans les règles de l'art, mais juste d'une première « prise de contact » : une dissimulation sûre incapable de résister à la moindre altération du support serait en effet d'un intérêt assez mince. Les quelques tests que l'on a effectué ici semblent indiquer que la dissimulation dhCI n'est pas dans ce cas. Ce chapitre est en partie constitué des résultats publiés dans [BG10b] et [BG10a].

I. ÉTAPES DE L'ALGORITHME

Nos simulations pratiques vont se décomposer en trois étapes (voir figure 23.1) :

1. le chiffrement du filigrane,
2. son embarquement dans l'hôte,
3. enfin, son extraction.

La première étape ne nous fait pas vraiment sortir du cadre de la dissimulation dhCI : on embarque un filigrane, par dhCI, qui n'est autre que le chiffre du filigrane originel. On change de condition initiale, mais on applique toujours le même algorithme de dissimulation, qui est chaotique. Dit autrement, on n'agit que sur la fonction d'acquisition $\mathcal{A}_2$ des filigranes, en amont de la dissimulation.

Cette étape de chiffrement n'est pas toujours requise, cela dépend des raisons à l'origine de la dissimulation. Cette variante est surtout pour nous l'occasion de montrer qu'avec les itérations chaotiques, on peut aussi chiffrer de manière chaotique le filigrane.

1. Chiffrement du filigrane

Une façon courante de chiffrer le filigrane consiste à réaliser le *ou exclusif* bit à bit entre le filigrane, et une suite binaire donnée d'allure aléatoire générée par une clé. Pour le déchiffrement, il suffira alors de regénérer cette suite en utilisant la même clé, puis de refaire le ou exclusif bit à bit avec le cryptogramme.

Dans ce chapitre, nous introduisons un nouveau crypto-système basé sur les itérations chaotiques. Sa stratégie chaotique sera très sensible aux MSCs, dans le cas d'un tatouage authentifié. Pour le détail de cette étape, voir la section 23.2 ci-dessous.

2. Embarquement du filigrane

Nous avons vu que la dissimulation dhCI de la définition IV.32 consiste à altérer certains LSCs de l'hôte. Dans le mode négation, cette altération consiste en la négation de certains bits, le choix de ces bits dépendant du filigrane à cacher.

Dans ce qui suit, certains LSCs seront soit niés, soit remplacés par les bits du filigrane éventuellement chiffré (voir figure 23.1). Les raisons et conséquences de cette variante sont expliqués ci-dessous. Pour choisir la suite des LSCs à modifier, une suite chaotique $\left(U^k\right)_k$ d'entiers inférieurs ou égaux au nombre N de LSCs, est produite à partir de la stratégie chaotique utilisée lors du chiffrement. Ainsi, le coefficient LSC en position U^k est soit nié, soit substitué par le $k-$ième bit du filigrane. En cas d'authentification, une telle procédure conduit à un choix des LSCs, qui sont fortement dépendant des MSCs.

1. Lorsque la négation est choisie, l'image tatouée est obtenue à partir de l'image d'origine, en remplaçant ses LSCs $L = \mathbb{B}^{\mathsf{N}}$ par le résultat d'itérations chaotiques.

   Dans ce cas, la fonction d'itération est la négation booléenne (vectorielle), l'état initial est L, et la stratégie vaut $\left(U^k\right)_k$. Le tatouage se résume à effectuer des itérations chaotiques, et on peut en déduire que l'algorithme possède la propriété de chaos-sécurité. Toutefois, le support d'origine est nécessaire pour extraire le filigrane.

2. D'autre part, lorsque les LSCs sélectionnés sont remplacés par le filigrane, l'extraction peut se faire sans la couverture d'origine (stéganographie aveugle). Dans ce cas, la sélection des LSBs reste chaotique, à cause de l'utilisation d'une suite chaotique, mais l'ensemble du processus ne satisfait pas la propriété de chaos topologique. L'utilisation d'itérations chaotiques est réduite au mélange du filigrane, et l'embarquement du filigrane ne peut être considéré chaos-sûr. Cette variante est avant tout donnée pour avoir une vision graphique, plus parlante, de la robustesse du dhCI, et illustrer ainsi graphiquement les probabilités de présence après attaque. Cette variante n'est pas à utiliser en pratique, du moins tant que son étude de sécurité n'aura pas été menée. Voir la section 23.2 pour plus de détails.

3. Extraction

La stratégie chaotique peut être regénérée, même dans le cas d'un tatouage authentifié, car les MSCs n'ont pas été modifiés pendant le tatouage. Ainsi, les quelques LSCs modifiés peuvent être retrouvés, le filigrane chiffré peut être reconstruit, et déchiffré. Dans le cas d'une négation, le résultat d'itérations chaotiques (effectuées une seconde fois) devraient redonner l'image d'origine. La probabilité d'avoir été tatoué diminue lorsque le nombre de différences entre l'image d'origine et cette nouvelle image augmente. Une courbe ROC peut aussi être utilisée pour déterminer si l'image est tatouée ou pas.

En cas d'authentification, si l'image tatouée est attaquée, les MSCs vont changer. Par conséquent, en raison de la forte sensibilité de l'opération d'embarquement (due à la sensibilité, l'expansivité et la

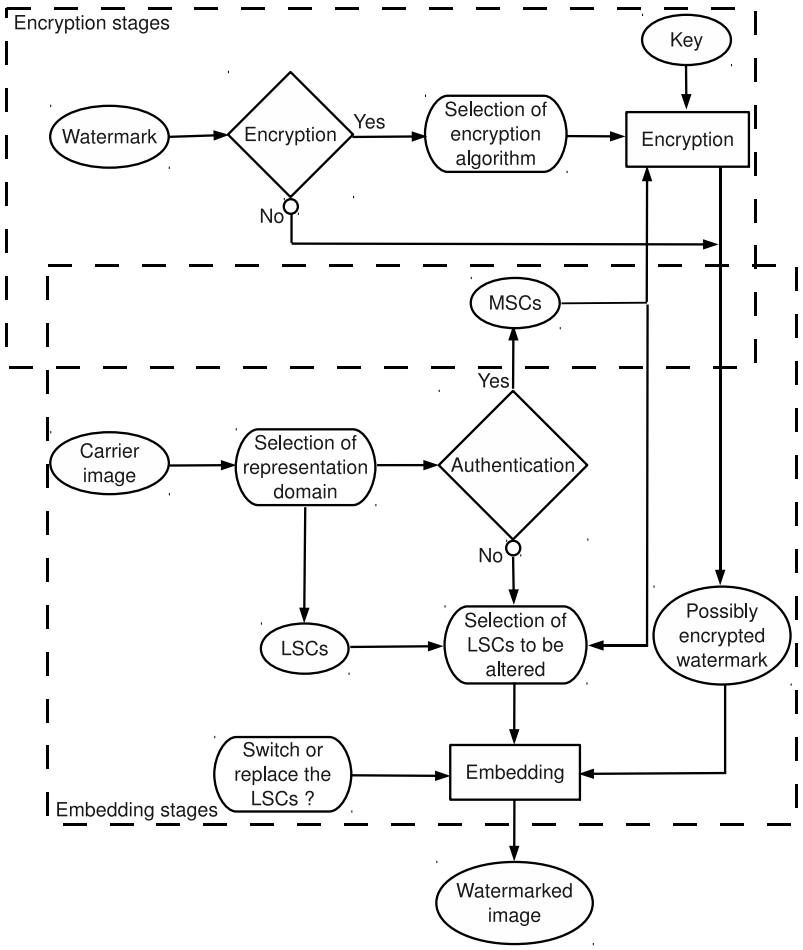


FIGURE 23.1 – L'algorithme dhCI.

constante de Lyapunov des IC), les LSCs destinés à recevoir le filigrane seront complètement différents. Par conséquent, le résultat de l'extraction n'aura aucune similitude avec le filigrane original.

Dans les paragraphes suivants, deux exemples d'application de notre méthode sont donnés, à savoir dans les domaines spatial et fréquentiel. Le but n'est pas de proposer une application finalisée et prête à l'emploi. Cette étude va seulement nous permettre de préciser les détails de l'algorithme, et de les expliquer avec quelques illustrations.

On donne dans ce qui suit deux illustrations de l'algorithme de dissimulation dhCI, la première dans le domaine spatial, la seconde dans le domaine ondelettes.

II. Exemple d'un tatouage spatial

1. Description des images

Le support hôte sera Lena, qui est ici une image de 256 niveaux de gris, de taille 256×256 (voir figure 23.2(b)). Le filigrane est l'image en noir et blanc, de taille 64×64, représenté dans la figure

23.2(a). Notons que l'hôte et le filigrane peuvent être autre chose que des images : notre algorithme est compatible avec toute donnée numérique.

Le domaine d'insertion sera le domaine spatial. Les MSCs seront les quatre bits de poids fort, et les LSCs les trois derniers bits de chaque pixel (un pixel donné sera, au plus, modifié par quatre niveaux de gris au cours d'une itération, voir figure 22.1). Avant son embarquement dans l'hôte, le filigrane est chiffré par itérations chaotiques. Le système à itérer, la stratégie chaotique S^n, et la fonction d'itération sont définis ci-après.

(a) Filigrane (b) Lena (c) Lena tatouée (d) Différences avec l'original

FIGURE 23.2 – Filigrane, Lena et différences.

2. Chiffrement du filigrane

a. Le chiffrement

Commençons par expliquer comment chiffrer le filigrane en utilisant les itérations chaotiques définies en I.31. L'état initial x^0 est le filigrane, vu comme un vecteur booléen. La fonction d'itération est la négation vectorielle f_0, définie en I.38.

On commence par engendrer un vecteur booléen chaotique $(B^k)_{k\leqslant T}$ par T itérations de la fonction chaotique linéaire par morceaux F (*c.f.* définition IV.27), comme suit :

- si $x^k > \mathcal{Y}$, alors $B^k = 1$,
- sinon $B^k = 0$,

où $\mathcal{Y} \in]0,1[$ est une valeur frontière, $x^0 \in [0;1]$, et $x^{n+1} = F(x^n, p)$. Une autre manière acceptable de définir B^k consiste à utiliser le générateur de nombres pseudo-aléatoires chaotique étudié dans [BGW09], [BGW10b] et [WBGF10], qui est cryptographiquement sûr.

Dernier élément à définir pour le chiffrement par itérations chaotiques, la stratégie chaotique $(S^k)_{k\in\mathbb{N}}$ dépendra du fait que l'on ait choisi un tatouage authentifié ou non, de la manière suivante :

- En cas de tatouage non authentifié, les bits du vecteur booléen chaotique (B^k) sont regroupés douze par douze, pour obtenir une suite $(S^k)_{k\in\mathbb{N}}$ d'entiers inférieurs à 4096, qui constituera la stratégie chaotique.
- En cas d'authentification, le *ou exclusif* bit à bit (XOR) est calculé entre le vecteur booléen chaotique (B^k) et les MSCs. Le résultat est converti en une stratégie chaotique en regroupant les bits comme ci-dessus.

Le filigrane chiffré est le dernier vecteur booléen obtenu par les itérations chaotiques décrites dans cette section.

b. Taille de l'espace des clés

Les clefs de cet algorithme de chiffrement dépendent des données suivantes : p, $\mathcal{Y}$, x^0 et T. Si nous supposons que les nombres machine sont codés avec N bits, alors une première approximation de la taille de l'ensemble des clés est de $2^{4\times N}$.

Toutefois, cet algorithme peut être formulé de manière plus générale. En fait, ce cryptosystème est fait d'itérations chaotiques sur $\mathcal{X}' = [1, \mathsf{N}]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$. Dans cette situation, une clé privée est constituée par la définition :

- d'une suite d'entiers $\leqslant \mathsf{N}$,
- d'un vecteur booléen de taille N,
- et d'un nombre T d'itérations.

Donc, avec la même hypothèse que ci-dessus, la taille de l'ensemble des clés est de $2^{\mathsf{N}} \times N$, multiplié par le cardinal de l'ensemble des suites qui ont une complexité de Kolmogorov inférieure à une frontière donnée. Enfin, le choix de la fonction d'itération peut être intégré dans l'ensemble des clés, si nécessaire.

3. L'embarquement du filigrane

Pour embarquer le filigrane, la configuration du dhCI doit être précisée ; dans ce contexte précis, il s'agit de la suite $(U^k)_{k\in\mathbb{N}}$ des bits à altérer parmi les LSCs, qui doit être définie. Pour ce faire, la stratégie $(S^k)_{k\in\mathbb{N}}$ de l'étape de chiffrement, est réutilisée de la manière suivante :

$$\left\{ \begin{array}{lcl} U^0 & = & S^0, \\ U^{n+1} & = & S^{n+1} + 2 \times U^n + n \pmod{\mathsf{N}}, \end{array} \right. \tag{23.1}$$

ce qui permet d'obtenir le résultat de la figure 23.2(c). Dans cet exemple, les LSCs désignés par la suite U^k sont remplacés par les bits du filigrane.

La fonction $\theta \mapsto 2\theta$ du tore, qui est le doublement de l'angle (définition II.39), a été choisie dans notre configuration pour rendre $(U^k)_{k\in\mathbb{N}}$ fortement sensible à la stratégie chaotique $(S^k)_{k\in\mathbb{N}}$. En conséquence de quoi, la suite $(U^k)_{k\in\mathbb{N}}$ est très sensible à l'altération des MSCs : en cas d'authentification, toute modification significative de l'image tatouée conduira à un filigrane extrait complètement différent de celui attendu.

4. Première approche de la robustesse dans le domaine spatial

Dans ce qui suit, on embarque un filigrane en noir et blanc dans une image en niveaux de gris. Nous avons utilisé la suite logistique $X^{n+1} = \mu X^n(1 - X^n)$, avec les paramètres $\mu = 4$ et $X_0 = 0,61$, transformée, pour chiffrer le filigrane par la méthode décrite à la figure 23.1, en une suite de bits b de la manière suivante : si $X^n > 0,5$, alors $b^n = 1$, sinon $b^n = 0$.

Le tatouage a été fait en utilisant l'algorithme dhCI dans la configuration décrite au chapitre 22. Le domaine de tatouage est le domaine spatial, les MSCs étant les quatre premiers bits de chaque niveau de gris de chaque pixel, tandis que les LSCs sont les trois derniers bits de chaque pixel (le bit restant n'est donc pas utilisé). On rappelle que les MSCs ne servent qu'en cas d'authentification.

Pour vérifier si notre algorithme résiste ne serait-ce qu'un petit peu à des attaques involontaires, nous avons appliqué à notre image tatouée différentes attaques. À chaque fois, un pourcentage de similarité entre le filigrane extrait après attaque, et le filigrane d'origine, est donné. Ce pourcentage est égal au nombre de bits égaux entre le filigrane d'origine et le filigrane extrait (rapporté à un pourcentage). Remarquons qu'un résultat inférieur ou égal à 50% implique que l'image n'a probablement pas été tatouée. Cette étude était le sujet de la contribution [BG10b].

NON-AUTHENTIFICATION		AUTHENTIFICATION	
Taille (pixels)	Similarité	Taille (pixels)	Similarité
10	99,08%	10	89,81%
50	97,31%	50	54,54%
100	92,43%	100	52,24%
200	70,75%	200	51,87%

TABLE 23.1 – Attaque par annulation de pixels

a. Attaque par annulation de pixels

Dans cette catégorie d'attaques, on annule les pixels d'une partie d'une image tatouée, comme dans la figure 23.3(a). Dans ce cas, les résultats du tableau 23.1 ont été obtenus.

Dans la figure 23.4, on montre les filigranes extraits et déchiffrés après l'annulation de tous les pixels d'un carré de côté 50 pixels partant du coin supérieur gauche de l'image tatouée, ainsi que dans le cas d'un carré de taille 10 pixels pour un tatouage authentifié. On rappelle que pour obtenir ces images, on n'a pas nié, mais remplacés les LSCs par ceux de la marque, ceci uniquement à des fins d'illustration.

En comparant les pourcentages de similarité entre l'image tatouée après attaque et l'image d'origine, on peut conclure qu'en cas de non-authentification, le filigrane est encore présent, et exploitable, après une attaque d'annulation de pixels. On peut de plus constater que ces pourcentages sont à peu près proportionnels à la taille du carré qui a été, de facto, supprimé.

Par contre, dans le cas authentifié, même un petit changement sur l'image hôte (une annulation d'un carré de 10×10 pixels) conduit à une présence de filigrane extrait complètement différente : toute tentative d'altération de l'image sera bien signalée, et l'authentification souhaitée est obtenue.

b. Attaque par rotation

Soit r_θ la rotation d'angle θ autour du centre $(128, 128)$ de l'image hôte. La transformation $r_{-\theta} \circ r_\theta$ est appliquée à l'image tatouée, qui est alors altérée comme dans la figure 23.3(b). Les résultats de la dissimulation dhCI face à cette attaque ont été résumés dans le tableau 23.2.

Les mêmes conclusions que ci-dessus peuvent être tirées : la dissimulation dhCI dans le domaine spatial résiste, au moins partiellement, à ce genre d'attaques.

(a) Attaque par annulation de pixels (100 pixels)

(b) Attaque par rotation (25 degrés)

FIGURE 23.3 – Lena tatouée puis attaquée

(a) Non-authentification (100×100).

(b) Authentification (100×100).

(c) Authentification (10×10).

FIGURE 23.4 – Filigranes extraits après des attaques d'annulation de pixels.

NON-AUTHENTIFICATION		AUTHENTIFICATION	
Angle (degré)	Similarité	Angle (degré)	Similarité
2	97,51%	2	70,01%
5	94,67%	5	59,47%
10	91,30%	10	54,51%
25	80,85%	25	50,21%

TABLE 23.2 – Attaques par rotation

c. Attaque par compression JPEG

Dans ce type d'attaque, une compression JPEG est appliquée à l'image tatouée. Cette attaque conduit à un changement de domaine (du spatial au domaine DCT : cosinus discret). Face à cette attaque, les résultats du tableau 23.3 ont été obtenus.

On peut considérer que les résultats sont bons face à ce genre d'attaques, car on a tatoué dans le domaine spatial, et que la compression JPEG utilise le domaine fréquentiel DCT. L'authentification est très bonne, la moindre compression enlève complètement le filigrane. Dans le cas non-authentifié, on parvient encore à s'assurer de la présence du filigrane, malgré des taux de compression de l'ordre de 10%.

d. Attaque par bruitage gaussien

L'image tatouée peut aussi être attaquée en ajoutant du bruit, en espérant ainsi brouiller le tatouage. Ce genre de bruit est en général gaussien, et dépend d'un ecart-type donné (fixant de combien on souhaite s'écarter, en moyenne, du niveau de gris originel du pixel considéré). Face à cette attaque, nous avons

NON-AUTHENTIFICATION		AUTHENTIFICATION	
Compression	Similarité	Compression	Similarité
2	82,95%	2	54,39%
5	65,23%	5	53,46%
10	60,22%	10	50,14%
20	53,17%	20	48,80%

TABLE 23.3 – Attaque par compression JPEG

NON-AUTHENTIFICATION		AUTHENTIFICATION	
Ecart-type	Similarité	Ecart-type	Similarité
1	74,26%	1	52,05%
2	63,33%	2	50,95%
3	57,44%	3	49,65%
5	51,26%	5	49,43%

TABLE 23.4 – Attaque par bruit gaussien

obtenus les résultats du tableau 23.4.

Une fois encore, on peut remarquer que les résultats ne sont pas si mauvais que ça, en particulier du fait que notre domaine de tatouage est spatial, et donc n'est pas forcément idéal pour contrer ce genre d'attaques.

III. EXEMPLE D'UN TATOUAGE DANS LE DOMAINE ONDELETTES

1. Détail de la méthode

On détaille dans cette section l'utilisation de notre algorithme dans le domaine ondelette (DWT), qui a été l'un des objets de la contribution [BG10a]. Les mêmes étapes que dans la section 23.2 seront suivies pour atteindre un objectif identique. La seule différence concerne la manière de définir les LSCs et les MSCs.

L'image hôte et le filigrane sont les mêmes que dans le premier exemple, mais Lena est maintenant constituée de 512×512 pixels, pour agrandir dans notre exemple la place pour le tatouage : le domaine fréquentiel a « moins de place », à taille égale, que le domaine spatial. Le domaine de tatouage est le domaine ondelettes (DWT). Dans cet exemple, nous avons choisi la famille d'ondelettes Daubechies : le filigrane est chiffré par des itérations chaotiques, puis embarqué dans des coefficients Daubechies DWT-1 de Lena. Le système à itérer, la stratégie chaotique S^n, et la fonction d'itération sont définis comme précédemment.

L'algorithme dépend d'un niveau de décomposition et d'une matrice de coefficients (voir figure 23.5) : *LL* signifie coefficients d'approximation, quand *HH, LH, HL* veulent dire coefficients de détail respectivement diagonal, vertical, et horizontal. Par exemple, le coefficient HH2 est la matrice des coefficients de détails diagonaux du second niveau de décomposition de Lena.

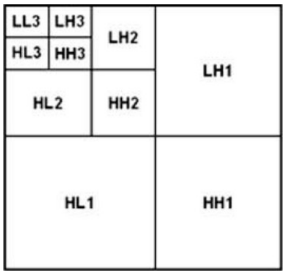


FIGURE 23.5 – Coefficients ondelettes.

Dans l'exemple de la figure 23.6, le filigrane est inséré dans les coefficients de détail diagonal HH2 (une matrice de réels de taille 128×128). Des itérations chaotiques sont faites pour insérer le filigrane :

- le système à itérer est le vecteur booléen de taille 128^2, constitué des LSCs de Lena – qui sont ici le second bit le moins significatif de chaque valeur entière de HH2,
- la fonction d'itération est la négation vectorielle,
- la stratégie chaotique est définie comme dans l'équation 23.1, avec $U^0 = 1$ et $\mathsf{N} = 256^2$.

(a) Lena originale.

(b) Lena tatouée.

FIGURE 23.6 – Dissimulation dans le domaine DWT

2. Résultats

a. Méthode d'évaluation

On commence par introduire quelques grandeurs permettant d'évaluer notre méthode (et de la comparer à des méthodes existantes).

DÉFINITION IV.33 (PSNR) : Le *PSNR* (*Peak Signal to Noise Ratio*), ou rapport signal sur bruit, est la mesure de distorsion suivante :

$$PSNR = 10 \cdot \log_{10}\left(\frac{d^2}{MSE}\right)$$

où

- d est la *dynamique du signal*. Dans le cas standard d'une image où les composantes d'un pixel sont codées sur 8 bits, d = 255.
- *MSE* (*Mean Squared Error*) est l'*erreur quadratique moyenne*. Elle est définie, pour deux images I_o et I_r de taille $m \times n$, comme étant égale à :

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(I_o(i,j) - I_r(i,j)\right)^2$$

DÉFINITION IV.34 (MOYENNE QUADRATIQUE) : La *moyenne quadratique*, ou *RMS* (pour Root Mean Square), est la racine carrée de la *MSE*. ◊

Ces deux mesures de distorsion sont habituellement utilisée en information dissimulée pour évaluer la discrétion d'un signal.

TABLE 23.5 – RMS pour un embarquement dans le coefficient HH2

	μ	U^0	Itérations	Authentification	RMS
Chiffrement	3,99987	0,65	20000	MSB = [5,6,7,8]	1,131
	3,999999	0,64	20000	MSB = [5,6,7,8]	1,129
	3,999999	0,65	19950	MSB = [5,6,7,8]	0,796
	3,999999	0,65	20000	MSB = [5,6,7]	1,122
	Coefficient	S^0	LSB	RMS	
Embarquement	HH1	1	[2]	253,65	
	HH2	2	[2]	0,653	
	HH2	1	[1]	0,983	

b. Évaluation de la méthode dans le domaine ondelettes

Pour la méthode présentée à la section 23.3.1, le PSNR vaut 53,45 dB. Les valeurs des pixels ont été modifiées d'au plus un niveau de gris. La valeur moyenne des différences est de 0,294, lorsque la RMS est égale à 0,542. L'altération peut donc être considérée comme indiscernable.

Pour l'extraction, on réalise à nouveau des itérations chaotiques. Le système à itérer est constitué du deuxième bit le moins significatif de chaque valeur entière comprise dans la matrice HH2 de Lena tatouée. La fonction d'itération est une fois encore la négation vectorielle, et la stratégie chaotique est calculée comme ci-dessus. Le résultat est alors comparé au coefficient HH2 de la Lena d'origine, et l'on trouve une RMS égale à 0,129. Cette RMS serait beaucoup plus élevée si l'on avait changé un quelconque paramètre pour notre extraction, comme le montre le tableau 23.5. Ce tableau correspond à de mauvaises extractions, c'est-à-dire à des extractions avec de mauvais paramètres.

On constate que la plus faible moyenne quadratique est bien obtenue pour une extraction avec les bons paramètres (ceux utilisés lors de l'embarquement du filigrane). Remarquons pour finir que si l'on avait essayé d'extraire le filigrane à partir de la Lena d'origine (non tatouée), la RMS obtenue serait alors deux fois plus grande que 0,127.

IV. PROPOSITION D'AMÉLIORATION PRATIQUE POUR LA ROBUSTESSE

Nous avons aussi proposé dans [BG10a] d'utiliser les codes correcteurs d'erreurs de Reed-Solomon [Cha85], pour améliorer la robustesse de notre méthode dans le cadre d'une stéganographie dans le domaine spatial. L'objectif visé était de proposer une méthode d'embarquement de description des médias directement dans leurs propres données numériques, cette description étant issue de sites collaboratifs tels Frick [Fri] et Delicious [Del]. Des moteurs de recherche « sémantiques » seraient capables de récupérer ces descriptions embarquées, pour renvoyer des résultats plus pertinents. Augmenter la robustesse de l'embarquement des descriptions permettrait de faire en sorte que l'information soit toujours présente après modifications du média par l'utilisateur (rotation, redimensionnement,...), modifications qui devraient pouvoir être permises dans ce contexte. Nous expliquons cette amélioration au travers d'un exemple illustratif basique, qui sera pour nous l'occasion de :

- Donner un exemple de chiffrement par itérations chaotiques.
- Montrer que les codes correcteurs peuvent participer au renforcement de la robustesse du dhCI.
- Présenter l'utilisation du dhCI dans un contexte de faible sécurité, dans lequel le remplacement peut être préféré à la négation.

Bref, il ne s'agira là que d'une illustration présentant quelques améliorations pour notre technique, et non d'une preuve de robustesse en tant que telle. Dans cet exemple illustratif, la description suivante va être insérée dans une image de Lena.

```
Lena (Soderberg), a standard test image originally cropped
from the November 1972 issue of Playboy magazine.
```

L'image de couverture sera la Lena de la figure 23.2(b), qui est une image en niveaux de gris de taille 256×256. Le texte à embarquer est converti en une séquence de 756 bits à l'aide de la table ASCII : chacun des 108 caractères est encodé sur 7 bits, ce qui nous donne le système suivant :

```
100110011001011101110110000101000000101000101001111011
111100100110010111100101100010110010111100101100111...
```

20000 bits sont calculés à partir d'une suite logistique[9] de paramètres $\mu = 3,999999$, $x^0 = 0,65$, et ces bits sont regroupés 10 par 10 ($10 = \lceil \log_2(756) \rceil$) pour obtenir une suite S d'entiers inférieurs ou égaux à 756. Alors, des itérations chaotiques sont appliquées au système ci-dessus, avec pour stratégie S et pour fonction d'itérations f_0, pour obtenir le cryptogramme suivant :

```
001000111110001110001101110111111000011011010011000101
001011110000110110011010010001110101101100010110101...
```

Nous utilisons alors deux couches de codes correcteurs de Reed-Solomon, respectivement de paramètres (32,24) et (24,16), séparés par une opération d'entrelacement des bits, pour reproduire le schéma du compact disque (le CIRC : « Cross-Interleaved Reed Solomon Coding »). Le message que l'on embarquera sera le résultat de ce CIRC, à savoir une suite de 2112 bits, qui commence de la manière suivante :

```
010110100101100000100001000111000010011100111111010001
110111100000010110001101010111011000010011001001110...
```

Ces 2112 bits seront embarqués dans Lena, qui possède $256 \times 256 \times 8 = 524288$ bits (8 bits par pixel). Les LSCs seront les deux derniers bits de chaque pixel, dont certains seront *remplacés* par la séquence CIRC. Pour sélectionner les bits à remplacer, la stratégie S de l'étape de chiffrement est utilisée une fois encore, pour engendrer une suite de triplets $(x^n, y^n, z^n)_{n\in\mathbb{N}}$ de telle sorte que $x^n, y^n \in [\![0; 255]\!]^{\mathbb{N}}$, et $z^n \in \{1; 2\}^{\mathbb{N}}$. Cette génération est réalisée de la manière suivante :

$$\left\{ \begin{array}{lcl} x^0 & = & 11, \\ y^0 & = & 23, \\ z^0 & = & 1, \end{array} \right.$$

et, pour tirer profit du doublement de l'angle :

$$\left\{ \begin{array}{lcll} x^{n+1} & = & 2x^n + S^{3n} + n & (\text{mod } 255), \\ y^{n+1} & = & 2y^n + S^{3n+1} + n & (\text{mod } 255), \\ z^{n+1} & = & 2z^n + S^{3n+2} + n & (\text{mod } 2). \end{array} \right.$$

Alors le $n-$ième bit de la séquence CIRC sera insérée dans le z^n bit le moins significatif du pixel (x^n, y^n) de Lena, pour obtenir la Lena tatouée de la figure 23.7(a). Dans la figure 23.7(b), nous montrons les différences entre la Lena d'origine et la Lena tatouée. Cette image illustre entre autre le fait que les LSCs remplacés ont été choisis chaotiquement et sont distribués uniformément [BG10c].

9. Une fonction chaotique linéaire par morceaux serait préférable à ce stade, mais nous avons préféré opter pour une reproduction exacte de l'application du papier [BG10a].

(a) Lena tatouée (b) Différences avec Lena (c) Lena attaquée

FIGURE 23.7 – Exemple d'utilisation de codes correcteurs.

Dans la section 23.2.4, nous avions prouvé que notre algorithme d'origine (dissimulation dhCI) possédait une certaine robustesse face à un certain nombre d'attaques classiques. Dans ce contexte, on effectuait bien le dhCI (négation, et non remplacement), ce qui avait pour avantage de garantir un bon niveau de sécurité, et pour inconvénient de ne pouvoir embarquer que peu d'information. De plus, après attaque, on ne retrouvait pas *exactement* l'information embarquée, mais quelque chose d'approchant. Dans la présente section, nous exigeons que le message extrait soit exactement le message d'origine, même après un certain nombre d'attaques.

Pour illustrer le fait que les CIRC peuvent nous permettre de parvenir à nos fins, la Lena tatouée de l'image 23.7(a) est attaquée par une mise à zéro de pixels : un carré de 40×40 pixels est supprimé au centre de l'image, comme on peut le voir à la figure 23.7(c). Alors, le message est extrait de l'image attaquée : la stratégie S est regénérée à partir d'une suite logistique ayant les mêmes paramètres que ci-dessus. Les suites x^n, y^n et z^n peuvent être regénérées elles-aussi, et les bits embarqués peuvent donc être extraits. Ces bits sont décodés par le procédé inverse : décodage Reed-Solomon de paramètre (24,16), désentrelacement, et décodage Reed-Solomon (32,24). Enfin, la suite de bits résultante est déchiffrée, les bits sont regroupés 7 par 7, et convertis en caractères à l'aide d'une table ASCII, pour obtenir finalement :

```
Lena (Soderberg), a standard test image originally cropped
from the November 1972 issue of Playboy magazine.
```

Notons pour finir que dans un contexte de pure dissimulation dhCI (*i.e.* avec négation de bits), nous aurions obtenu un score de 100% de similarités grâce au CIRC.

V. CONCLUSION

Commençons par rappeler une fois encore que le sujet de notre recherche n'est pas forcément de proposer un nouvel algorithme de tatouage, mais d'apporter un nouveau cadre théorique pour la stéganalyse des configurations KMA, KOA et CMA. L'algorithme présenté ci-dessus n'est pas, à l'heure actuelle, un candidat possible pour parvenir à un tatouage robuste. Ce n'est qu'un deuxième exemple d'une évaluation de sécurité, telle que nous l'entendons. Les résultats empiriques comparant le schéma proposé avec l'état de l'art en termes, par exemple, de taux d'erreur, ou de résistance contre les attaques de sensibilité, ne seront abordés que dans un prochain travail de recherche.

Dans cette partie, notre but était avant tout de proposer un nouveau concept de sécurité pour la stéganalyse, une approche complémentaire au cadre existant. Cette nouvelle notion de sécurité, dite « chaos-sécurité », devrait contribuer à renforcer la confiance dans les algorithmes existants. En outre, l'étude

de sécurité dans les configurations KMA, KOA, et CMA est dorénavant réalisable grâce à cette approche. Enfin, pour nous, un algorithme est sûr si il est imprévisible. Son processus itératif doit satisfaire la propriété de chaos selon Devaney et son niveau de sécurité augmente avec le nombre de propriétés satisfaites.

Nous avons montré dans cette partie que l'intersection entre les deux ensembles de stego-sécurité et de chaos-sécurité n'est pas vide, en raison de l'étalement de spectre et de la dissimulation dhCI qui vérifient tous deux ces propriétés. Ce faisant, nous avons établi un premier lien entre ces deux approches pour la stéganalyse. À l'avenir, nous étudierons ce lien davantage, afin de mieux comprendre les interactions entre ces deux cadres. La comparaison entre l'étalement de spectre et notre dissimulation dhCI sera étendue. De plus, de nouveaux outils tirés de la théorie du chaos seront introduits pour enrichir la chaos-sécurité. Tous ces outils seront comparés à l'information de Fisher et d'autres outils issus de la théorie de la mesure. La chaos-sécurité d'autres algorithmes existants sera étudiée, et ces derniers seront comparés à l'étalement de spectre et à la dissimulation dhCI. Nous mesurerons quantitativement l'impact de l'expansivité sur la chaos-sécurité (en la reliant à la théorie de l'information). La façon de choisir, dans la pratique, les MSCs et LSCs sera discutée, et le fait que les MSCs peuvent être corrélés entre eux sera étudié en détail. La distorsion due au tatouage sera regardée précisément. Enfin, la façon de comprendre ces nouveaux outils en termes de stéganographie et tatouage sera approfondie : cette étude est nécessaire pour rendre la chaos-sécurité vraiment utile en pratique.

Cinquième partie

Application aux fonctions de hachage

CHAPITRE 24

Quelques rappels concernant les fonctions de hachage

Dans la vie, on ne doit pas s'encombrer.

LUDWIG WITTGENSTEIN

Dans ces parties applicatives, nous souhaitons prouver que nos itérations chaotiques peuvent enrichir le monde de la sécurité informatique, notamment par la possibilité offerte, par elles, d'écrire des programmes au comportement réellement chaotique.

L'un de nos angles d'attaques a été les fonctions de hachage, pour au moins deux raisons : ce domaine est en profonde remise en question (les fonctions actuellement utilisées ne conviennent plus), et les exigences dans ce domaine nous rappellent grandement diverses propriétés de chaos topologique. Citons simplement l'effet avalanche : il nous semble qu'il est assuré par une constante de Lyapunov positive.

Le lecteur voulant en savoir plus sur ces fonctions de hachage pourra se référer au livre de Douglas Stinson [Sti02].

I. FONCTIONS DE HACHAGE

1. Définitions

Commençons par un petit rappel préliminaire :

DÉFINITION V.1 (ALPHABET, MOT) : Soit Σ un ensemble, appelé *alphabet*, dont les éléments sont appelés des *lettres*. Un *mot* de longueur n est un n-uplet de lettres. On note alors Σ^n l'ensemble des mots de longueur n, et $\Sigma^* = \cup_{n\in\mathbb{N}}\Sigma^n$ l'ensemble des mots de longueur quelconque. ◊

Nous pouvons maintenant donner la définition d'une fonction de hachage.

DÉFINITION V.2 (FONCTION DE HACHAGE) : Soit Σ un alphabet. On définit une *fonction de hachage* comme une application :

$$h : \Sigma^* \longrightarrow \Sigma^n$$

où $n \in \mathbb{N}$ est un entier préalablement fixé. L'image $h(x)$ est alors appelée le *résumé*, le *condensé*, ou encore la *valeur hachée* de x, et x est l'*antécédent*, ou la *pré-image* de $h(x)$. ◊

Les fonctions de hachage associent donc, à des chaînes de caractères de longueur quelconque, d'autres chaînes de longueur fixe. Remarquons dès à présent qu'elles ne peuvent être injectives.

Exemple V.1 : L'application envoyant le mot binaire $b_1 b_2 \dots b_n \in \{0;1\}^*$ sur $b_1 \oplus b_2 \oplus \dots \oplus b_n$ est une fonction de hachage. Elle envoie le mot 01101 en 1, et transforme une chaîne de caractères b en 1 si, et seulement si le nombre de 1 dans b est impair.

Quand de telles fonctions sont utilisées en cryptographie, il faut s'assurer qu'un certain nombre de propriétés soient satisfaites. Décrivons ces propriétés...

2. Propriétés exigées des fonctions de hachage

Soit h une fonction de hachage d'ensemble de départ D.

a. Les fonctions de hachage sont à sens unique

Tout d'abord, l'élément $h(x)$ devrait être facile à calculer, pour tout x de D. Et si h est utilisée à des fins cryptographiques, alors elle doit être à sens unique :

DÉFINITION V.3 (FONCTION À SENS UNIQUE) : Une fonction h est dite *à sens unique* s'il est infaisable de déterminer le ou les antécédent(s) d'une image par h donnée. On parle aussi de *résistance à la première pré-image*. ◊

Autrement dit, une fonction h est à sens unique si le calcul des x tel que $h(x) = s$, à partir de s, est impossible en pratique. Il faut définir ce que signifie *infaisable* dans ce contexte, mais cette idée est assez difficile à mettre en forme. Pour bien faire, il faudrait user de concepts issus de la théorie de la complexité, ce qui alourdirait ce document. Nous nous restreindrons donc à une idée intuitive, sachant que cette dernière se définit rigoureusement :

DÉFINITION V.4 : Nous dirons qu'une fonction h est *à sens unique* lorsque les programmes qui essaient, à partir d'une donnée s, de calculer x tel que $h(x) = s$, échouent *presque toujours*, par manque de temps ou d'espace mémoire. ◊

REMARQUE. On ne connaît à l'heure actuelle aucune fonction à sens unique, et l'on ne sait même pas s'il en existe ne serait-ce qu'une seule. Cependant, il existe des fonctions faciles à évaluer, mais dont aucun algorithme efficace d'inversion n'est connu. À défaut de mieux, on les utilise comme fonction à sens unique.

Exemple V.2 : Soit p un nombre premier de 1024 bits (choisi au hasard), et g une racine primitive modulo p. Alors la fonction de $[\![0, p-1]\!]$ dans lui-même, définie par $x \longmapsto g^x \bmod p$, est facile à calculer (par exponentiation rapide). Mais on ne sait pas, dans un certain nombre de cas, déterminer son inverse efficacement. Il peut en effet être très difficile de calculer le logarithme discret x en base g de $X = g^x \bmod p$ (le cryptosystème El Gamal est fondé sur cette constatation). Cette application peut donc, à défaut de mieux, servir de fonction à sens unique.

Les fonctions de hachage ne doivent pas seulement être des fonctions à sens unique. Elles doivent aussi être résistantes aux collisions.

b. Les fonctions de hachage sont résistantes aux collisions

i. Les collisions. Commençons par définir ce qu'est une collision.

DÉFINITION V.5 (COLLISION) : On appelle *collision* de h tout couple (x, x') de D^2 tel que $x \neq x'$ et $h(x) = h(x')$. ◊

Exemple V.3 : Une collision de l'exemple V.1 est (111,100), ou tout couple formé de deux chaînes de caractères distinctes, ayant toutes les deux, ou n'ayant ni l'une ni l'autre, un nombre pair de 1.

Ces collisions sont inévitables, vu que les fonctions de hachage partent d'un ensemble infini, et arrivent toujours dans un ensemble fini. Cependant, on ne souhaite pas que ces collisions soient « trouvables » par un adversaire, ce qui aboutit à la définition de résistance « faible et forte » aux collisions.

ii. Fonctions faiblement résistantes aux collisions.

DÉFINITION V.6 (FONCTIONS FAIBLEMENT RÉSISTANTES) : h est dite *faiblement résistante aux collisions* si pour tout $x \in D$, il est infaisable (en pratique) de trouver x' tel que (x, x') soit une collision de h. On parle encore de *résistance à la seconde pré-image*. ◊

REMARQUE. La résistance à la seconde pré-image veut donc dire qu'étant donnée une entrée x, fixée à l'avance, et son haché $y = h(x)$, il doit être impossible pour un adversaire de trouver une seconde entrée x' telle que $h(x') = y$ avec $x \neq x'$.

Pour illustrer cette définition et mieux la comprendre, donnons l'exemple d'une situation où une fonction faiblement résistante aux collisions est nécessaire.

Exemple V.4 : Supposons qu'Alice souhaite protéger un programme de chiffrement x enregistré sur son disque dur. À l'aide d'une fonction de hachage $h : \Sigma^* \longrightarrow \Sigma^n$, elle calcule la valeur hachée $y = h(x)$ du programme, et elle mémorise y.

Le lendemain, Alice retourne sur son ordinateur et, avant d'utiliser son programme de chiffrement, elle regarde s'il n'a pas été modifié par un tiers. Pour cela, Alice vérifie que la valeur hachée actuelle du programme est celle qu'elle a en mémoire.

En toute rigueur, ce test n'est valable qu'à partir du moment où la fonction de hachage h est bien faiblement résistante aux collisions. Sans quoi, un adversaire pourrait éventuellement arriver à remplacer le programme x par un autre, tout en faisant en sorte que ce programme altéré possède la même valeur hachée que x.

Cet exemple illustre une utilisation habituelle des fonctions de hachage résistantes aux collisions : l'intégrité d'un document donné est réduite à celle de son résumé, qui est une chaîne de caractères généralement beaucoup plus petite, donc plus facile à mémoriser, à stocker, et à vérifier.

iii. La résistance forte. Dans la définition de la résistance faible, on partait d'un x donné, et l'on ne souhaitait pas qu'il soit possible de trouver un x' ayant le même condensé que x. Une définition plus forte de résistance est donnée ci-dessous :

DÉFINITION V.7 (FONCTION FORTEMENT RÉSISTANTE) : On dit que la fonction de hachage h est *fortement résistante aux collisions* si le calcul d'une quelconque collision (x, x') de h est infaisable. ◊

REMARQUE. Le terme « résistant aux collisions », sans précision supplémentaire, signifiera par la suite cette résistance forte.

L'utilisation de fonctions de hachage fortement résistantes aux collisions s'impose souvent en cryptographie, par exemple dans le cas des signatures électroniques, et ce pour des raisons évidentes de sécurité. On peut montrer que :

PROPOSITION V.1 : *Les fonctions de hachage fortement résistantes aux collisions sont des fonctions à sens unique.*

iv. L'attaque dite « des anniversaires ». Soit h une fonction de hachage de Σ^* dans Σ^n. Nous décrivons dans ce qui suit une attaque simple à la résistance forte aux collisions d'une fonction de hachage, appelée l'*attaque des anniversaires*. L'attaque consiste à calculer autant de valeurs hachées que le temps et l'espace le permettent, et à les ranger avec leur pré-images, jusqu'à tomber sur une collision. Elle permet de fixer une taille minimale aux condensés.

En effet, cette attaque nous dit combien de valeurs hachées il est nécessaire de calculer, pour avoir plus d'une chance sur deux de tomber sur une collision [Sti02] :

PROPOSITION V.2 : *Si k chaînes de caractères sont choisies dans Σ^*, avec*

$$k \geqslant \frac{1 + \sqrt{1 + (8 \ln 2)|\Sigma|^n}}{2},$$

où $|\Sigma|$ désigne le cardinal de Σ, alors la probabilité que deux valeurs hachées de ces k chaînes soient égales, est supérieure à $\frac{1}{2}$.

L'intérêt de la proposition V.2 est de nous renseigner sur la taille minimale nécessaire que la valeur hachée doit avoir, pour résister à ce genre d'attaques systématiques, que l'on pourrait qualifier de type « force brute ». Nous supposerons pour simplifier que $\Sigma = \{0, 1\}$. Alors :

$$k \geqslant f(n) = \frac{1 + \sqrt{1 + (8 \ln 2) \times 2^n}}{2}$$

est suffisant pour que le calcul de k valeurs hachées conduit à plus d'une chance sur deux d'avoir une collision.

Le tableau suivant donne les valeurs de $\log_2 f(n)$ pour des tailles typiques de n.

n	50	100	150	200
$\log_2 f(n)$	25,24	50,24	75,24	100,24

Bref, en calculant plus de $2^{\frac{n}{2}}$ valeurs hachées, l'*attaque des anniversaires* a plus d'une chance sur deux de trouver une collision. Pour se prémunir contre une telle attaque, n doit être choisi de sorte que le calcul de $2^{\frac{n}{2}}$ valeurs hachées soit infaisable.

Il est recommandé de prendre $n \geqslant 128$, voire $n \geqslant 160$. Ces recommandations sont cependant relatives aux performances actuelles des ordinateurs ; elles ne sont que provisoires et ne correspondent en aucun cas à un seuil théorique d'« infaisabilité ». Aussi, nous prendrons $n = 256$ pour notre fonction de hachage, présentée au chapitre 25.

fonction de hachage	longueur des blocs	vitesse relative
MD4	128	1,00
MD5	128	0,68
RIPEMD-128	128	0,39
SHA-1	160	0,28
RIPEMD-160	160	0,24

TABLE 24.1 – Quelques fonctions de hachage classiques

3. Exemples de fonctions de hachage

Le tableau 24.1 liste certaines fonctions de hachage célèbres, et en donne quelques caractéristiques, dont la taille du haché. Le SHA-1 est détaillé précisément en annexe A.

Question vitesse, toutes ces fonctions sont très efficaces. La fonction MD4 ne peut plus être considérée comme résistante aux collisions : Dobbertin a trouvé une collision en calculant 2^{20} valeurs hachées [Dob96]. De même, MD5 n'est plus totalement sûre. Enfin, compte tenu de l'attaque des anniversaires, les 160 bits de SHA-1 impliquent une résistance à 2^{80} essais pour l'attaque brutale, alors que l'on exige dorénavant une résistance à 2^{128} essais pour garantir un niveau convenable de sécurité : le SHA-1 ne convient donc lui non plus, comme tous les exemples classiques de la table 24.1.

II. UTILISATION DES FONCTIONS DE HACHAGE

L'utilisation des fonctions de hachages est fréquente en informatique, dans différents cas de figure, certains étant rapportés ci-dessous. Ces exemples d'utilisation permettent de mieux comprendre pourquoi on impose les contraintes de la section 24.1 aux fonctions de hachage.

1. Le contrôle d'accès

Il est évident qu'un mot de passe ne doit pas être stocké en clair sur une machine. Pour éviter cela, on conserve uniquement sa valeur hachée (obtenue par SHA-1 ou MD5, habituellement). L'ordinateur comparera alors, lors de l'identification d'un utilisateur, l'empreinte du mot de passe stocké avec l'empreinte de celui saisi au moment de ladite authentification.

2. Tables de hachage et structures de données

On utilise aussi les fonctions de hachage pour établir des « tables de hachage » de bases de données : les empreintes sont utilisées comme index des « cases » de la table.

Par exemple, supposons que l'on ait constitué un annuaire numérique des habitants de la métropole, contenant les noms, prénoms, adresses et numéros de téléphone de ces derniers. La fonction de hachage permettrait de transformer un nom en sa ligne dans l'annuaire, et la table de hachage contiendrait chacune de ces associations.

Ces empreintes sont des nombres entiers obtenus en hachant la « clé » des objets à stocker, qui est souvent une chaîne de caractères. On peut ensuite retrouver un objet à partir de sa clé : il suffit de lire dans le tableau la case dont l'index est l'empreinte de cette clé.

Toutefois, des collisions existent, car il existe moins d'empreintes possibles que de valeurs possibles pour la clé. Des techniques destinées à lever ces conflits sont nécessaires, telles que le principe de chaî-

nage : chaque case de la table constitue le début d'une liste chaînée. Si deux clés fournissent le même condensé et donc accèdent à la même portion de la table, on doit alors parcourir la liste chaînée jusqu'à atteindre l'élément correspondant à la clé donnée.

3. Codes d'authentification de message

On a vu dans l'exemple V.4 que l'on pouvait utiliser des fonctions cryptographiques de hachage pour tester si un fichier a été modifié : la valeur hachée du fichier est stockée séparément, et l'intégrité du fichier est testée en comparant la valeur hachée du fichier actuel à la valeur hachée stockée.

Pour prouver l'intégrité d'un document *et son authenticité*, on peut utiliser des fonctions de hachage paramétrées.

DÉFINITION V.8 (FONCTION DE HACHAGE PARAMÉTRÉE) : Une *fonction de hachage paramétrée* est une famille $\{h_k, k \in \mathcal{K}\}$ de fonctions de hachage, où $\mathcal{K}$ est un ensemble appelé l'*espace des clés* de h.

Une fonction de hachage paramétrée s'appelle aussi un *code d'authentification de message* (on parle encore de *MAC* : *Message Authentication Code*). ♢

Exemple V.5 : Soit une fonction de hachage $g : \mathbb{B}^* \longrightarrow \mathbb{B}^4$. On peut construire un MAC ainsi [10] :

$$\begin{array}{rcl} h_k : \mathbb{B}^* & \longrightarrow & \mathbb{B}^4 \\ x & \longmapsto & g(x) \oplus k \end{array}$$

qui admet $\mathbb{B}^4$ pour espace des clés.

Dans ce qui suit, on illustre comment utiliser un MAC :

Exemple V.6 : Alice, un professeur, envoie par mail au service de la scolarité la liste des notes de ses étudiants. Ce service doit pouvoir être assuré de l'authenticité de ces notes. Pour ce faire, un MAC $\{h_k, k \in \mathcal{K}\}$ est utilisé : Alice et le service des examens échangent une clé secrète $k \in \mathcal{K}$ et, avec sa liste x, Alice envoie aussi la valeur hachée $y = h_k(x)$. Bernard, le responsable du service, réceptionne (x, y), calcule à son tour $h_k(x)$ avec la clé k qu'il possède lui-aussi. S'il retrouve y, alors il accepte x.

Le protocole de l'exemple précédent prouve l'authenticité, à condition qu'il soit impossible de calculer le couple $(x, h_k(x))$ sans connaître k.

10. $\oplus$ désigne le *ou exclusif* bit à bit.

CHAPITRE 25

Notre fonction de hachage

Plonger au fond du gouffre, Enfer ou Ciel, qu'importe ?
Au fond de l'Inconnu pour trouver du nouveau !

Le voyage
CHARLES BAUDELAIRE

L'utilisation de fonctions chaotiques dans des algorithmes de hachage a connu plusieurs évolutions ces dernières années. Dans [FSSM05] par exemple, les auteurs proposent de combiner un algorithme de signature numérique basée sur des courbes elliptiques, avec une fonction chaotique, permettant ainsi d'obtenir un nouvel algorithme aux meilleures propriétés. D'autres exemples de génération d'une fonction de hachage utilisant des fonctions chaotiques peuvent être trouvés dans [WZZ03], [XLW09], et [PQL05].

Nous définissons pour notre part une nouvelle façon de construire des fonctions de hachage, basée sur l'utilisation des itérations chaotiques, qui a fait l'objet des publications [GB10, BG10c]. Du fait de la théorie présentée dans les parties précédentes, les fonctions de hachage que l'on va ainsi générer vont satisfaire les propriétés de chaos topologique rappelées dans la partie II, du moins est-ce le but visé. De ce fait, différents comportements recherchés dans ce domaine seront garantis par notre approche (par exemple, l'effet avalanche est étroitement lié à la propriété de sensibilité). Nous illustrerons cela à l'aide d'un exemple, et nous chercherons par la suite à le vérifier plus rigoureusement.

I. LES ITÉRATIONS CHAOTIQUES VUES COMME FONCTIONS DE HACHAGE

Dans cette section, nous présentons une nouvelle manière de définir et d'obtenir des valeurs hachées à partir d'un média numérique donné. Cette méthode est basée sur les itérations chaotiques, à la normalisation du message près (*i.e.*, sa transformation en une condition initiale pour les itérations chaotiques) et devrait donc satisfaire les diverses propriétés de chaos topologique de la partie II, pourvu que la normalisation n'impacte pas cela. Ce point délicat sera discuté au chapitre suivant.

La valeur hachée sera le dernier état d'une suite d'itérations chaotiques donnés. Il nous faut donc définir l'état initial X_0, la stratégie finie S et la fonction d'itérations qui doivent être utilisés, et qui dépendront du média à hacher.

1. L'état initial des itérations

La condition initiale $X_0 = (S, E)$ est composée d'une suite de N bits, notée E, qui est de la forme $2^{\mathsf{n}}, \mathsf{n} > 2$, et d'une stratégie chaotique S. Dans les deux paragraphes suivants, nous décrivons en détail la façon d'obtenir cette condition initiale à partir du support original.

Remarque. Des valeurs $\mathsf{N} \geqslant 256$, *i.e.* $n \geqslant 8$, devraient être choisies suite aux conséquences de l'attaque dite des anniversaires (proposition V.2).

a. Comment obtenir E (la fonction de compression)

La première étape de notre algorithme consiste à transformer le message d'origine en une suite « normalisée » de N bits, notée E. Cette normalisation s'inspire fortement de la normalisation du SHA-X (voir annexe A). Elle est l'équivalent de la fonction d'acquisition de la définition IV.19. D'autres « acquisitions » sont possibles. L'impact de ce choix sera discuté au chapitre suivant.

Afin d'illustrer cette étape de normalisation, nous supposerons que le texte d'origine est un extrait du *Corbeau* d'E.A.Poe [Poe39] :

```
And my soul from out that shadow that lies floating on the floor
Shall be lifted---nevermore!
```

et que $\mathsf{n} = 8$ ($\mathsf{N} = 256$).

Chaque caractère de cette chaîne est remplacé par son code ASCII (sur 7 bits). Puis, nous ajoutons un 1 à la fin de cette chaîne.

```
1000001 1101110 1100100 0100000 1101101 1111001 0100000 1110011 1101111
1110101 1101100 0100000 1100110 1110010 1101111 1101101 0100000 1101111
1110101 1110100 0100000 1110100 1101000 1100001 1110100 0100000 1110011
1101000 1100001 1100100 1101111 1110111 0100000 1110100 1101000 1100001
1110100 0100000 1101100 1101001 1100101 1110011 0100000 1100110 1101100
1101111 1100001 1110100 1101001 1101110 1100111 0100000 1101111 1101110
0100000 1110100 1101000 1100101 0100000 1100110 1101100 1101111 1101111
1110010 0001010 1010011 1101000 1100001 1101100 1101100 0100000 1100010
1100101 0100000 1101100 1101001 1100110 1110100 1100101 1100100 0101101
0101101 0101101 1101110 1100101 1110110 1100101 1110010 1101101 1101111
1110010 1100101 0100001 1
```

Alors, comme dans le SHA-1, la valeur binaire (10100101111) de la longueur de cette chaîne (652 bits) est ajoutée, suivie d'un autre 1 :

```
1000001 1101110 1100100 0100000 1101101 1111001 0100000 1110011 1101111
1110101 1101100 0100000 1100110 1110010 1101111 1101101 0100000 1101111
1110101 1110100 0100000 1110100 1101000 1100001 1110100 0100000 1110011
1101000 1100001 1100100 1101111 1110111 0100000 1110100 1101000 1100001
1110100 0100000 1101100 1101001 1100101 1110011 0100000 1100110 1101100
1101111 1100001 1110100 1101001 1101110 1100111 0100000 1101111 1101110
0100000 1110100 1101000 1100101 0100000 1100110 1101100 1101111 1101111
1110010 0001010 1010011 1101000 1100001 1101100 1101100 0100000 1100010
1100101 0100000 1101100 1101001 1100110 1110100 1100101 1100100 0101101
0101101 0101101 1101110 1100101 1110110 1100101 1110010 1101101 1101111
1110010 1100101 0100001 1101001 011111
```

La chaîne entière est recopiée, mais dans l'autre sens. Ce qui donne :

```
1000001 1101110 1100100 0100000 1101101 1111001 0100000 1110011 1101111
1110101 1101100 0100000 1100110 1110010 1101111 1101101 0100000 1101111
1110101 1110100 0100000 1110100 1101000 1100001 1110100 0100000 1110011
1101000 1100001 1100100 1101111 1110111 0100000 1110100 1101000 1100001
1110100 0100000 1101100 1101001 1100101 1110011 0100000 1100110 1101100
1101111 1100001 1110100 1101001 1101110 1100111 0100000 1101111 1101110
0100000 1110100 1101000 1100101 0100000 1100110 1101100 1101111 1101111
1110010 0001010 1010011 1101000 1100001 1101100 1101100 0100000 1100010
1100101 0100000 1101100 1101001 1100110 1110100 1100101 1100100 0101101
0101101 0101101 1101110 1100101 1110110 1100101 1110010 1101101 1101111
1110010 1100101 0100001 1101001 0111111 0000011 1011101 1001000 1000001
1011011 1110010 1000001 1100111 1011111 1101011 1011000 1000001 1001101
1100101 1011111 1011010 1000001 1011111 1101011 1101000 1000001 1101001
1010001 1000011 1101000 1000001 1100111 1010001 1000011 1001001 1011111
1101110 1000001 1101001 1010001 1000011 1101000 1000001 1011001 1010011
1001011 1100110 1000001 1001101 1011001 1011111 1000011 1101001 1010011
1011101 1001110 1000001 1011111 1011100 1000001 1101001 1010001 1001010
1000001 1001101 1011001 1011111 1011111 1100100 0010101 0100111 1010001
1000011 1011001 1011000 1000001 1000101 1001010 1000001 1011001 1010011
1001101 1101001 1001011 1001000 1011010 1011010 1011011 1011101 1001011
1101101 1001011 1100101 1011011 1011111 1100101 1001010 1000011 1010010
11111
```

On recopie alors suffisamment cette chaîne, jusqu'à ce que sa taille dépasse le prochain multiple M de $2\times\mathsf{N}$, et on la tronque au $M-$ième bit. Cette chaîne, qui contient l'intégralité du message d'origine et qui a pour taille un multiple de $2\times\mathsf{N}$, sera désignée par la lettre D dans ce qui suit. D sera à l'origine de la stratégie, ce qui justifie pourquoi nous avons fait l'étape de recopiage dans l'autre sens : tout changement d'un quelconque bit dans la chaîne considérée interviendra, au plus tard, au milieu de D, et donc tôt dans la stratégie.

Enfin, nous scindons cette chaîne en $2\times\frac{M}{2\times\mathsf{N}}$ blocs de N bits, sur lesquels nous appliquons le *ou exclusif* bloc à bloc, obtenant ainsi une séquence de N bits, qui n'est autre que l'état initial E :

```
01100001 00111101 10110011 11111001 00000101 10001110 11000110 01010111
01101011 00001110 11110001 10110100 10111100 01001111 01000011 01011001
11001010 11110000 10111100 11000111 00110011 00001010 01101100 10111001
11001010 01000100 11101101 00111110 01110001 00111011 01111101 10000100
```

On pourra noter que beaucoup de textes ont la même chaîne E. Ce n'est pas un problème, car il nous faut encore définir la stratégie, qui elle aussi dépendra du texte. Justement, expliquons comment construire la stratégie S.

b. Comment choisir S

Pour obtenir la stratégie S, une suite intermédiaire (u^n) est construite à partir de D comme suit :

- On commence par agrandir D, pour faire en sorte d'itérer suffisamment pour que le chaos se fasse ressentir pleinement.

Pour ce faire, on rajoute à D sa version décalée d'un bit vers la gauche, puis de deux bits vers la gauche, *etc.*, jusqu'à 7 bits vers la gauche (ce sont des rotations circulaires, sans perte : le premier bit devient dernier) : on retrouve les décalages du SHA-X. Il est à noter que l'opération de décalage est chaotique selon Devaney.

- La chaîne ainsi obtenue est découpée en blocs de n bits.
- u^n est alors la valeur décimale du $n-$ième bloc de n bits de cette chaîne.

Il est maintenant possible de construire la stratégie S, de la manière suivante :

$$\begin{cases} S^0 = u^0 \\ S^n = u^n + 2 \times S^{n-1} + n \pmod{\mathsf{N}}. \end{cases}$$

Notre stratégie n'est donc pas u, mais une version transformée utilisant le doublement de l'angle (qui est lui aussi chaotique selon Devaney) : S sera fortement sensible à tout changement du texte d'origine.

2. Comment construire le condensé (la fonction de hachage)

Pour construire le condensé, des itérations chaotiques sont réalisées, avec pour état initial E, pour fonction d'itération la négation vectorielle :

$$\begin{array}{rccc} f : & \mathbb{B}^{\mathsf{N}} & \longrightarrow & \mathbb{B}^{\mathsf{N}} \\ & (E_0, \ldots, E_{\mathsf{N}-1}) & \longmapsto & (\overline{E_0}, \ldots, \overline{E_{\mathsf{N}-1}}), \end{array}$$

et pour stratégie chaotique S.
Le résultat de ces itérations est un vecteur de N bits. Ses bits sont pris 4 à 4 et traduits en chiffres hexadécimaux, afin d'obtenir la valeur hachée :

```
9ED289DC85087E36678BF539509597FD727922AD124AF30E911A46D0654E34F6
```

En guise de comparaison, si au lieu d'utiliser le texte « *nevermore* », nous avions pris « *evermore* » (remplacement d'un *n* par un espace), la fonction de hachage aurait renvoyé :

```
9EFE8CE68D17C416080F4D9CDA16DAB1EF3A622A0261EAA7DAAD3482274CA003
```

Remarque. Dans cette illustration, la valeur hachée est obtenue en utilisant la négation vectorielle f_0. Néanmoins, notre procédure est générale, et la négation vectorielle peut être remplacée par toute fonction $f \in C$.

Dans le paragraphe suivant, un exemple complet d'utilisation de notre procédure, sur des images, est donné.

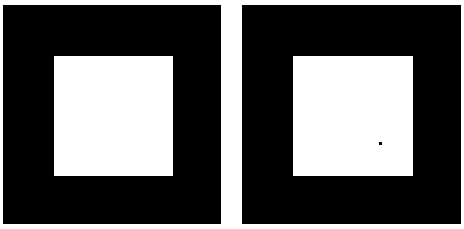
II. Quelques valeurs hachées en guise d'exemple

Considérons les deux images noir et blanc de taille 64×64 de la figure 25.1, dans lesquelles la seule différence est le pixel en position (40,40).

Notre fonction de hachage renvoie :

```
34A5C1B3DFFCC8902F7B248C3ABEFE2C9C9538E5104D117B399C999F74CF1CAD
```

pour la figure 25.1(a) et



(a) Image originale. (b) Image modifiée.

FIGURE 25.1 – Hachage d'images en noir et blanc.

```
5E67725CAA6B7B7434BE57F5F30F2D3D57056FA960B69052453CBC62D9267896
```

pour la figure 25.1(b). Considérons maintenant les deux images de Lena en 256 niveaux de gris et 256×256 pixels de la figure 25.2, pour lesquels le niveau de gris du pixel en position $(50, 50)$ est passé de 93 (figure 25.2(a)) à 94 (figure 25.2(b)).

(a) Léna originale. (b) Léna modifiée.

FIGURE 25.2 – Hachage d'images en niveaux de gris.

Dans ce cas, notre fonction de hachage renvoie :

```
FA9F51EFA97808CE6BFF5F9F662DCD738C25101FE9F7F427CD4E2B8D40331B89
```

pour la Léna de gauche, et :

```
BABF2CE1455CA28F7BA20F52DFBD24B76042DC572FCCA4351D264ACF4C2E108B
```

pour la Léna de droite. On trouve là une illustration de l'effet avalanche souhaité (illustration, qui en soi, ne prouve rien). Le chapitre suivant propose une discussion autour de cette fonction de hachage.

CHAPITRE 26

Première évaluation de notre fonction de hachage

N'a de conviction que celui qui n'a rien approfondi.

De l'inconvénient d'être né.
EMIL CIORAN

I. PRÉCISION CONCERNANT NOTRE APPROCHE

L'objectif principal de cette application est de montrer qu'il est facile de tirer partie des propriétés de désordre des itérations chaotiques en sécurité informatique. Nous avons étudié le comportement topologique de ces dernières dans la partie III, et il s'est révélé très riche. Comme ces itérations chaotiques ne manipulent que des entiers, elles ne perdent pas leurs propriétés une fois implémentées sur machine, pourvu que l'on s'y prenne bien – c'est-à-dire principalement que l'ensemble sur lequel on itère soit la mémoire de l'ordinateur couplée avec l'ensemble de tous les médias. De plus, la fonction d'itération peut être élémentaire (négation booléenne dans le cas que l'on a le plus étudié), et les itérations résultantes de faible complexité. On peut donc alors obtenir, de cette manière, des programmes réellement imprévisibles et efficaces.

L'idée qui nous est alors venue à l'esprit est d'exploiter ce désordre pour apporter des solutions à des problèmes issus de la sécurité informatique : notre but est ici de montrer que ces objets aux propriétés intéressantes s'utilisent aisément. Il s'agit d'exemples de faisabilité, d'illustrations ; nous ne prétendons pas offrir de solutions « magiques » aux problèmes auxquels font face les communautés scientifiques étudiant, parfois depuis plusieurs dizaines d'années, les objets que nous avons présenté dans ces chapitres applicatifs. Nous voulons juste montrer que ces itérations chaotiques s'exploitent facilement, donner des sortes de cas d'école, d'exemples-jouets, dans des domaines qui s'y prêtent bien. Nous souhaitons aussi montrer que la théorie mathématique du chaos peut s'utiliser autrement, peut-être plus rigoureusement, que ce qui a été fait jusqu'à présent en sécurité informatique. Mais en aucune manière nous jugeons, à l'heure actuelle, définitivement valables les exemples illustratifs donnés dans ces parties applicatives (dissimulation dhCI, fonction de hachage). Nous savons que, pour chacune des applications proposées, toute l'étude reste à faire, et que tout ceci ne constitue au mieux que des travaux préliminaires, ayant sûrement au final peu de chance d'aboutir sur quelque chose de concrètement exploitable.

Dit autrement, il ne s'agit pas d'une fin, mais d'un commencement. Il ne s'agit que, d'une part, d'une illustration sans ambition ni finalité de notre théorie, et d'autre part des toutes premières interrogations concernant l'intérêt effectif de notre approche dans le domaine des fonctions de hachage, comme dans le domaine du tatouage. Cependant, certaines raisons évoquées ici nous semblent justifier l'intérêt qu'il peut y avoir de partir du chaos pour obtenir ensuite une fonction de hachage. Puisque l'on exige des propriétés pour de telles fonctions, pourquoi ne pas se placer dès le commencement dans un contexte favorable, en partant d'objets que l'on sait posséder des propriétés au moins similaires ? Alors, bien sûr, une fois encore, tout le travail théorique reste à faire, et les preuves sont à fournir, sans aucune certitude de réussite. Mais l'on ne part pas complètement de rien, dans l'inconnu, et l'on peut être un minimum confiant. De plus, les systèmes que l'on manipule ont une formulation simple (bien qu'une évolution complexe), ce qui augmente les chances de réussir à écrire les preuves de sécurité requises. Aussi, nous pensons raisonnable de creuser cette voie, qui pourrait peut-être s'avérer à terme intéressante.

Dans ce qui suit, en guise d'introduction véritable à l'étude de notre fonction de hachage, nous allons discuter de la complexité de nos opérations (ces fonctions doivent se calculer rapidement) et des raisons topologiques nous incitant à penser que notre fonction de hachage pourrait avoir certaines des propriétés attendues dans ce domaine.

II. Fonction à sens unique

Rappelons avant toute chose que notre algorithme de hachage peut se résumer de la manière suivante :

Initialisation. Construire le couple (S, E) à partir du média à hacher, où E est un vecteur de 256 bits et S est une suite finie. C'est la *normalisation* de la chaîne à hacher.

Itérations. Réaliser des itérations chaotiques sur l'état initial E, avec la stratégie S et la négation vectorielle.

Rappelons de plus qu'une fonction de hachage doit être telle que toute valeur hachée est facile et rapide à calculer, mais que la fonction doit être résistante à la première préimage (il doit être difficile de retrouver un antécédent d'une valeur hachée donnée).

1. Complexité de notre algorithme

Nous évaluons dans ce qui suit la complexité de notre algorithme, pour montrer que les valeurs hachées sont faciles et rapides à calculer par notre méthode. Commençons par évaluer le coût de la première étape :

Proposition V.3 : *Réaliser l'étape d'initialisation nécessite de l'ordre de $O(x)$ opérations élémentaires, pour un média de x bits.*

Preuve : *En effet, cette première étape n'effectue qu'un nombre fini d'opérations linéaires sur des tableaux de bits : copie, renversement, et décalage circulaire.*

Venons-en à la seconde étape :

Proposition V.4 : *L'étape d'itérations requiert moins de $2x + 2\log_2(x+1) + 515$ opérations élémentaires, pour un média de x bits.*

PREUVE : *Le coût d'une itération est réduit à la négation d'un bit, soit une opération élémentaire. La deuxième étape s'effectue donc en n opérations élémentaires, où n est le nombre de termes de la stratégie chaotique S. Or, S a le même nombre de termes que u, et ce dernier a le même nombre de termes que D, vu que pour constituer u, on a recopié 8 fois D, et que l'on a ensuite regroupé les bits 8 par 8 pour obtenir les termes de u. On renvoie au chapitre 25 pour le sens de ces notations.*

Il reste donc à évaluer la taille de D, qui correspondra au nombre d'opérations élémentaires total. Soit x la taille du média en bits. Pour obtenir D, on rappelle que l'on effectue les opérations suivantes :

1. *On ajoute le bit 1 : D possède alors $x+1$ bits.*
2. *La valeur binaire de la longueur est ajoutée, suivie d'un bit : D possède alors $x+2+\log_2(x+1)$ bits.*
3. *Cette chaîne est recopiée, mais dans l'autre sens : D possède alors $2\times\left(x+2+\log_2(x+1)\right)$ bits.*
4. *On la recopie alors jusqu'au prochain multiple de 512 : on rajoute au pire 511 bits, ce qui fait que D aura au pire $2x+2\log_2(x+1)+515$ bits.* □

On peut donc en conclure que :

THÉORÈME V.1 : *Le coût du calcul d'une valeur hachée est linéaire dans l'algorithme du chapitre 25.*

2. Résistance à la première image

Réfléchissons maintenant à la résistance possible de notre fonction de hachage au calcul de la première préimage. Nous n'allons pas en donner de preuve formelle, vu que nos travaux de recherche n'en sont, dans ce domaine, qu'à leurs balbutiements. Nous allons juste tenter d'expliquer les raisons pour lesquelles nous sommes un minimum confiant dans le fait qu'une telle résistance doit avoir lieue.

Soient m le message à hacher, (S,E) le message normalisé (*i.e.* l'état initial des itérations chaotiques) et M la valeur hachée de m par notre méthode. Les itérations chaotiques de condition initiale (S,M) et de fonction d'itérations f_0 ont pour état final E. Ainsi, il est inutile d'espérer « inverser » le processus de hachage, pour obtenir le message normalisé à partir du haché : cela reviendrait à essayer de prévoir l'évolution future des itérations chaotiques à partir d'une connaissance partielle de la condition initiale. En effet, seule M est connue, pas S, ce qui revient en termes topologiques à avoir une incertitude sur la condition initiale, correspondant à savoir uniquement que cette dernière se situe dans une boule ouverte de centre M et de rayon 1. Cette boule possède une infinité de termes. Partant d'une telle incertitude sur la condition initiale, et vue les nombreuses propriétés de chaos satisfaites par les IC (notamment, la sensibilité de 255, l'expansivité de 1 et l'exposant de Lyapunov de $\ln 256$), cette prévision est impossible. De plus, en partant de cette boule, on atteint tous les points possibles de notre espace des phases au bout de 256 itérées ; or, l'on itère au minimum 519 fois pour obtenir notre valeur hachée (*c.f.* proposition V.4).

III. L'EFFET AVALANCHE

L'équivalence la plus simple à établir entre les définitions topologiques et les propriétés attendues pour les fonctions de hachage a déjà été évoquée, il s'agit de la forte ressemblance entre l'effet avalanche

d'une part, et la sensibilité, l'expansivité, et l'exposant de Lyapunov d'autre part. Cet effet avalanche exige qu'un bit différent entre les médias conduit à des valeurs hachées complètement différentes. À vrai dire, nous n'avons pas rencontré de définition rigoureuse de cette notion. Il nous semble cependant que nous ne pouvons la comprendre que de l'une des manières suivantes :

1. Deux médias proches conduisent toujours à des hachés très éloignés.
2. Deux médias proches peuvent conduire à des hachés très différents, et le font très souvent.
3. Il sera toujours possible que deux médias proches conduisent à des hachés différents.

Le troisième point proposé ci-dessus pour la notion d'effet avalanche devrait pouvoir, une fois formalisé, être prouvé équivalent à la sensibilité aux conditions initiales. Le deuxième point devrait être équivalent à l'expansivité, ou du moins à une version plus forte de cette dernière dans un espace probabilisé. Enfin, la première version de l'effet avalanche nous semble la traduction d'un exposant de Lyapunov plus grand que 1.

IV. Conclusion et perspectives

MD5 et SHA-0 ont été cassé en 2004, et l'équipe de Wang a découvert (conférence CRYPTO-2005) une attaque de SHA-1 en 2^{69} opérations, c'est-à-dire plus rapide d'un facteur 2000 par rapport à l'attaque par force brute en 2^{80}. Si 2^{69} opérations reste à l'heure actuelle hors de portée du commun des ordinateurs, cette attaque, basée sur une précédente attaque de SHA-0, reste un résultat très important dans le domaine de la cryptanalyse.

Le nombre de fonctions de hachage cryptographiquement sûres commence à se réduire, d'autant que l'on suspecte SHA-2 de n'être plus aussi sécurisé qu'on a pu le croire. Le monde de la cryptographie en est à se demander s'il est réellement possible de construire un algorithme de hachage suffisamment solide. Aussi, le NIST (l'organisme de normalisation des standards et de la technologie aux USA) a annoncé le 23 janvier 2007 l'ouverture d'un processus de développement de nouveaux algorithmes de hachage standard, illustrant les craintes pesant sur les fonctions actuelles. Nous nous sommes intéressés à ce domaine trop tard pour pouvoir participer à ce processus, et nos réflexions ne sont pas à l'heure actuelle suffisamment abouties pour espérer vouloir jouer un rôle dans ce processus. Nous le suivons cependant de près, y trouvant là une source d'inspiration pour nos réflexions à venir.

Sixième partie

Application aux réseaux de capteurs

CHAPITRE 27

L'agrégation sécurisée de données au sein de réseaux de capteurs sans fil

> Par suite d'un concours de circonstances, il est vrai, mais il y a toujours des circonstances.
>
> *La chute*
> ALBERT CAMUS

Les réseaux de capteurs sont des systèmes collaboratifs/itératifs : le réseau est le système, chaque capteur représente une cellule, et ces cellules communiquent entre elles suivant une règle pré-établie, qui peut être vue comme la fonction d'itérations du système. Les problèmes de sécurité étant sensibles dans les réseaux de capteurs, nous y avons vu là une possible application à notre théorie.

On explique dans cette introduction en quoi consiste les réseaux de capteurs sans fil, quelles sont les contraintes inhérentes à ce type de réseaux, ce qui signifie dans ce contexte l'agrégation sécurisée des données. On rappelle les travaux existants et l'on introduit nos deux solutions.

I. INTRODUCTION

1. Les réseaux de capteurs

Un *réseau de capteurs* est constitué d'un nombre éventuellement grand de *capteurs* (des « nœuds ») déployés aléatoirement, éventuellement sur une grande échelle. La fonction principale de ces capteurs est de mesurer certaines grandeurs physiques (la température, l'humidité, *etc.*) à l'aide d'une sonde, et de transmettre de proche en proche ces mesures en direction d'une station terminale, appelée la *base* ou le *puits*. Ces nœuds sont limités en puissance, en capacité de stockage, de communication et de calcul : la seule source d'énergie de ces nœuds est leur batterie qui a une capacité limitée. C'est pourquoi tout doit être fait pour limiter au mieux leur consommation d'énergie, afin d'augmenter la durée de vie du réseau.

Les réseaux de capteurs ont reçu beaucoup d'attention ces dernières années, du fait de l'étendue de leurs applications potentielles, touchant l'écologie, l'environnement, les transports, l'armée, *etc.*

2. L'agrégation sécurisée des données dans les réseaux sans fil

a. L'agrégation des données

Les mesures des capteurs doivent être analysées à un moment ou à un autre, et le résultat de cette analyse peut conduire par la suite à certaines actions entendues. La plupart du temps, ces analyses supposent des calculs de maximum, minimum, moyenne, *etc.* Ces calculs peuvent être faits soit au niveau du puits, soit par les nœuds eux-mêmes, ce qui conduirait alors à un réseau hiérarchisé possédant plusieurs couches de capteurs : les uns, dits « terminaux », mesurant l'environnement, et les autres en charge des calculs et de la transmission des données au puits.

Afin de réduire la quantité de données transmise au puits et donc la consommation d'énergie, il s'avère souvent intéressant de préférer l'approche dans laquelle l'analyse des données est réalisée, au moins en partie, au sein même du réseau de capteurs. Pour cela, les données devront être *agrégées* le long de leur parcours vers le puits : les capteurs terminaux envoient leurs mesures vers des nœuds spécifiques, appelés *agrégateurs*. Ces agrégateurs peuvent être, au choix, des capteurs particuliers (possédant plus de puissance ou d'énergie), ou bien des capteurs classiques à qui on attribue un rôle spécifique. Ces derniers « compriment », « agrègent » les données reçues avant de les transmettre à un autre niveau d'agrégateurs ou directement au puits. On peut ainsi économiser l'énergie des capteurs terminaux et réduire le coût de communication, pourvu que le processus d'agrégation ne soit pas trop gourmand.

b. La sécurité des réseaux

D'autre part, les réseaux de capteurs sont souvent déployés en des lieux publics, voire en des milieux hostiles, ce qui conduit à de nombreux problèmes de sécurité : authentification des données, chiffrement de ces dernières et gestion subséquente des clés utilisées, contrôle d'accès, *etc.* C'est pourquoi la sécurisation des données au sein des réseaux de capteurs est une préoccupation récurrente dans ce domaine, se traduisant par de nombreux développements au cours de ces dernières années.

c. Agrégation sécurisée

Les deux contraintes évoquées ci-dessus, à savoir économiser les ressources et garantir un niveau raisonnable de sécurité, sont toutes deux importantes dans les réseaux de capteurs sans fil. Elles mènent naturellement à la recherche de solutions permettant une *agrégation sécurisée* des données au sein de ces réseaux.

Dans ce contexte les capteurs terminaux chiffrent ou authentifient les valeurs mesurées, et transmettent ces résultats à une deuxième couche de capteurs. Cette couche doit pouvoir agréger les données sécurisées reçues pour réduire la consommation d'énergie totale du réseau. Cette agrégation doit pouvoir se faire en économisant les ressources des agrégateurs, et sans introduire de problèmes de sécurité. Une fois son travail achevé, chaque agrégateur transmet alors le résultat ainsi obtenu en direction du puits (soit directement à ce dernier, soit en passant par d'autres couches intermédiaires d'agrégateurs).

II. La sécurité des données

1. Le chiffrement

a. Présentation

Le chiffrement des données est une nécessité quand la confidentialité des données transmises doit être garantie [CBS06].

Le plus fréquemment, le protocole appliqué est le suivant : les nœuds terminaux chiffrent leurs valeurs mesurées et les envoient aux agrégateurs, qui possèdent les clefs de déchiffrement de tous les nœuds auxquels ils sont associés. Les agrégateurs déchiffrent alors les cryptogrammes reçus, agrègent l'ensemble des valeurs déchiffrées, puis chiffrent à nouveau les valeurs agrégées avant de les transmettre en direction du puits.

Bien que cette approche soit viable, elle est très coûteuse et compliquée, du fait du déchiffrement de chaque valeur reçue au niveau de chaque agrégateur. De plus, la gestion des clés pose problème : un chiffrement symétrique n'est pas envisageable, car cela signifie que la clé de déchiffrement est embarquée dans les nœuds, qui peuvent alors être la cible d'attaques. Quant au chiffrement asymétrique, il nécessite pour bien faire le remplacement fréquent des clés, ce qui pose de nouveaux problèmes de sécurité et un surcroît de communications. Enfin, la présence des données déchiffrées au sein des agrégateurs soulève d'autres problèmes de sécurité : un adversaire pouvant les observer aurait alors accès aux données transitant dans le réseau.

Le problème est donc le suivant : le chiffrement de données permet de résoudre les problèmes de sécurité lors des transmissions, mais comment garantir la confidentialité lors des opérations d'agrégation ? La seule solution viable est de parvenir à réaliser ces opérations directement sur les cryptogrammes, ce qui nous conduit à introduire l'homomorphisme.

b. L'homomorphisme

Récemment, certains auteurs [Cas05, GSW04, AGW05] ont proposé d'utiliser des chiffrements homomorphes afin de parvenir à réaliser différentes fonctions d'agrégation directement sur les cryptogrammes (*i.e.* sans déchiffrement), d'où la définition [YLP06] :

DÉFINITION VI.1 (CHIFFREMENT HOMOMORPHE) : Un chiffrement est dit *homomorphe* si à partir des cryptogrammes $Enc(a)$ et $Enc(b)$, il est possible de calculer $Enc(f(a,b))$, où f peut être, par exemple, l'addition, le produit, ou le *ou exclusif* (XOR), et ce sans le recours à la clé privée.

Un chiffrement homomorphe est dit *total* s'il est compatible avec l'addition *et* la multiplication, et *partiel* dans le cas contraire. ♢

Exemple VI.1 : Les cryptosystèmes RSA et ElGamal sont des chiffrements homomorphes partiels : on peut obtenir le produit à partir des cryptogrammes, mais pas la somme.

REMARQUE. Bien que de tels cryptosystèmes soient recherchés depuis 1978, il n'existe pas à l'heure actuelle de cryptosystème totalement homomorphe suffisamment étudié, et qui soit exploitable dans les réseaux de capteurs sans fil. Le premier cryptosystème totalement homomorphe a été proposé en 2009 par Craig Gentry [Gen09], mais ce dernier reste trop théorique, et nécessite des calculs beaucoup trop lourds pour les objectifs que l'on vise. Il a depuis été amélioré [vDGHV10] et s'approche petit à petit d'une solution utilisable en pratique.

Les méthodes de chiffrement homomorphe qui ont été utilisées jusqu'à présent dans le but de parvenir à une agrégation sécurisée des données, nécessitent beaucoup de ressources pour mener à bien les opérations de chiffrement et d'agrégation dans les réseaux de capteurs sans fil. Ainsi, l'utilisation de cryptosystèmes basés sur RSA a été proposé dans [WSL, LN08], cependant ce chiffrement requiert de la puissance de calcul et de l'espace mémoire relativement conséquents, vu les ressources disponibles. De plus, ces cryptosystèmes produisent des cryptogrammes de grande taille (même si la taille des données d'origine est petite), ce qui augmente d'autant le coût de transmission. Cependant, les capteurs sont limités et n'ont pas assez de ressources pour réaliser longtemps de telles opérations, la durée de vie des réseaux ainsi construits est de ce fait beaucoup trop courte. Tout cela sera détaillé dans le chapitre suivant.

c. Notre première approche

Nous avons proposé dans [BGM10a] d'utiliser un chiffrement sur courbes elliptiques [BGN05] qui manipule des clés de chiffrement de taille très petite tout en conservant un fort niveau de sécurité. Le cryptosystème de Boneh *et al.* [BGN05], que l'on a adapté au cas d'un réseau de capteurs sans fil, permet de faire N additions et un produit sur les cryptogrammes : son homomorphisme n'est pas réduit à une seule opération, ce chiffrement est *presque* totalement homomorphe. De plus, parmi les quelques cryptosystèmes homomorphes qui sont compatibles avec l'addition et la multiplication, ce cryptosystème est le seul à avoir été prouvé sûr, à être utilisable en pratique, et à n'avoir pas été cryptanalysé jusqu'à présent. Cette contribution est présentée en détail au chapitre 28.

Pour prouver le caractère réalisable et efficace de notre technique, nous avons simulé un réseau de capteurs embarquant ce cryptosystème. Nous avons comparé notre approche avec une agrégation sécurisée basée sur un cryptosystème RSA. Les résultats obtenus montrent que l'on peut ainsi réduire significativement les coûts de calcul et de communication, sans réduire pour autant la sécurité du réseau, ce qui montre que notre méthode d'agrégation sécurisée des données peut être implémentée effectivement.

2. L'authentification

Une deuxième approche a ensuite été proposé dans [BGM10b]. Nous avons expliqué comment étendre le nombre de fonctions d'agrégations réalisables au sein des réseaux de capteurs sans fil, quand le niveau de sécurité requis est relativement moins élevé que ce qui précède. En d'autres termes, lorsque l'authentification des données peut suffire, et quand l'on n'a pas forcément besoin de les chiffrer. Pour ce faire, nous avons repris et amélioré l'idée des auteurs de [ZLDD08], qui consiste à utiliser la dissimulation de l'information pour authentifier les données mesurées par les capteurs terminaux. Les opérations d'agrégation s'effectuent donc sur des valeurs tatouées, et l'enjeu consiste à faire en sorte que la marque d'authentification reste après agrégation. C'est possible, quand la méthode de tatouage utilisée est robuste, mais l'algorithme que les auteurs de [ZLDD08] ont choisi ne l'est pas. Nous avons pour notre part porté une plus grande attention sur la robustesse de la méthode choisie. C'est pour cela que dans [BGM10b] nous avons proposé d'utiliser notre algorithme de dissimulation dhCI, ou l'étalement de spectre naturel avec $\eta = 1$, en transposant ces techniques dans le domaine de l'agrégation sécurisée des données dans les réseaux de capteurs sans fil. De plus, nous avons porté les notions de sécurité présentes dans le domaine de la dissimulation d'information, que l'on a rappelées dans la partie IV, au domaine des réseaux de capteurs. Le chapitre 29 comprend le détail de cette contribution.

CHAPITRE 28

Utilisation d'un cryptosystème homomorphe sur courbes elliptiques

C'est incroyable comme les circonstances obligent souvent les gens à faire ce qu'ils ont envie de faire.

La mort dans les nuages
AGATHA CHRISTIE

Dans ce chapitre, nous sommes avant tout intéressés par la préservation de la confidentialité des données dans les réseaux de capteurs sans fil. Notre but est d'empêcher un quelconque adversaire d'obtenir la moindre information concernant les mesures effectuées par les capteurs terminaux.

Cependant, quand des agrégations sont utilisées dans un réseau, faire en sorte que les données n'apparaissent jamais en clair au cours du trajet entre les nœuds terminaux et le puits est problématique. L'un des principal problème est que la plupart des cryptosystèmes ne sont pas additivement homomorphes : la somme des cryptogrammes n'est pas le cryptogramme de la somme. Cela est problématique, ne serait-ce que lors d'un simple calcul de moyenne : sans cette propriété, on est obligé de déchiffrer les cryptogrammes si l'on souhaite obtenir la moyenne des valeurs mesurées. De plus, les homomorphismes actuellement utilisés ont des temps de calcul exponentiels [Pet07].

Pour résoudre ces problèmes, nous proposons un modèle de sécurité basé sur un cryptosystème *presque* complètement homomorphe sur courbes elliptiques (il s'agit, avec le chapitre suivant, de nos contributions [BGM10a]). Nous montrerons que ce modèle permet de nombreuses agrégations différentes, et qu'il ne nécessite pas beaucoup d'énergie.

I. ÉTAT DE L'ART ET CONTRIBUTION

1. L'état de l'art

De nombreux auteurs ont étudié les vulnérabilités des réseaux de capteurs sans fil, les menaces auxquels ils font face, le besoin de sécuriser les données même durant leur agrégation, et différentes solutions ont d'ores et déjà été proposées dans la littérature. Nous présentons dans ce qui suit un état de l'art, le plus

exhaustif possible, des techniques d'agrégation sécurisée (AS) des données dans les réseaux de capteurs sans fil (RCF).

La première proposition d'utilisation des cryptosystèmes homomorphes pour résoudre les problèmes d'AS dans les RCF remonte à l'année 2004, avec le « concealed data aggregation » (CDA) de Joao Girao *et al.* [GSW04, WGA06]. Les avantages d'une telle approche sont de renforcer la sécurité (confidentialité) des données durant les transmissions et les agrégations, tout en diminuant les dépenses énergétiques. La sécurité augmente, car à aucun moment on ne déchiffre les données : ces dernières sont chiffrées sitôt les mesures prises par les nœuds terminaux, et déchiffrées uniquement au niveau du puits. La méthode sera moins gourmande, car on gagne l'étape de déchiffrement et rechiffrement au niveau de chaque agrégateur. Il ne reste qu'à s'occuper de l'opération d'agrégation à proprement parler, d'où des calculs réduits à leur strict minimum. Dans ces articles, Girao *et al.* ne se contentent pas d'avancer l'idée d'utiliser les cryptosystèmes homomorphes pour l'AS dans les RCF, ils proposent aussi une réalisation concrète (le CDA), qui permet d'effectuer une addition des cryptogrammes au niveau des agrégateurs. Pour ce faire, les auteurs utilisent le cryptosystème symétrique de Domingo-Ferrer [DF02]. Bien que ce dernier soit compatible avec l'addition et la multiplication, les auteurs n'utilisent que la première de ces opérations. Ils appliquent leur CDA au calcul de moyenne et à la détection de mouvements. On peut surtout reprocher à cette approche d'utiliser un cryptosystème qui a été maintes fois cryptanalysé dès 2003 [CN03, Wag03, YLP06]. De plus, ce cryptosystème est symétrique, ce qui pose le problème de la gestion des clés privées : chaque nœud terminal doit posséder ses clés privées, ce qui pose de nombreux problèmes de sécurité [Pet07]. Enfin, le cryptosystème de Domingo-Ferrer est très gourmand en ressources, produit des cryptogrammes très grands devant la taille des données, et donc est incompatible avec les ressources limitées des RCF [Pet07].

Les auteurs du CDA ont poursuivi leur travaux dans [AGW05], en s'intéressant alors à l'opération de comparaison ($<$). Leur but est de pouvoir effectuer des calculs de médiane, maximum, minimum et de comparaison au niveau des agrégateurs, sans avoir à déchiffrer. Cela semble être le premier travail d'AS envisageant la comparaison pour réaliser l'agrégation. Pour parvenir à leurs fins, ils n'utilisent pas le cryptosystème de Domingo-Ferrer, mais l'OPES [AKSX04], un cryptosystème homomorphe proposé par Agrawal *et al.*, qui préserve la comparaison. Cependant, les auteurs rappellent dès leur introduction le point suivant : Rivest *et al.* ont montré que si un cryptosystème était compatible avec la comparaison, alors il n'était pas sûr, même face aux attaques les plus élémentaires (« cyphertext only attack » : attaques basées uniquement sur la connaissance des cryptogrammes). Nous refuserons donc d'envisager la comparaison, puisque nous tenons à proposer une agrégation ayant un bon niveau de sécurité.

Les auteurs de [HST10] cherchent à mettre à l'écart les données redondantes reçues au niveau des agrégateurs : les nœuds terminaux chiffrent leurs données avant de les envoyer à leurs agrégateurs, le rôle de ces derniers étant de supprimer les redondances éventuelles. À cette fin, ils utilisent un cryptosystème symétrique (à base de *ou exclusif*), une fonction de hachage et un générateur de nombres aléatoires. Chaque nœud terminal possède sa propre clé secrète. Il chiffre sa mesure, génère un nombre pseudo-aléatoire K_i, hache ce nombre, effectue le *ou exclusif* entre la mesure et la valeur hachée de K_i, puis entre le résultat obtenu et K_i. À la suite de quoi, ce nœud doit calculer le *ou exclusif* entre K_i et sa clé secrète, et concaténer le tout pour finalement obtenir le cryptogramme à transmettre. Un même nombre d'opérations est requis au niveau des agrégateurs. Ce grand nombre d'opérations et la taille des cryptogrammes sont incompatibles avec l'économie des ressources du réseau. De plus, les capteurs embarquent leurs clés privées, ce qui soulève un certain nombre de problèmes de sécurité : d'une part, ces capteurs risquent d'être la cible d'attaques et ne sont pas armés pour lutter, et d'autre part pour qu'un chiffrement de type *ou exclusif* (dit encore masque jetable) soit sûr, chaque clé secrète ne doit être utilisée qu'une fois et doit être de la taille du plus grand des messages possibles. Donc, comment mettre à jour les clés secrètes, et comment le faire sans que cela ne soit pénalisant en terme de communication ? Autre point négatif, les

agrégateurs doivent posséder tous les $K_i \oplus K_j$ de leurs capteurs (soit $\frac{n(n-1)}{2}$ valeurs, chacune de la taille des clés), et doivent, pour chaque couple de valeurs cryptées reçues, calculer un grand nombre de XOR, et comparer enfin chacune des $\frac{n(n-1)}{2}$ valeurs intermédiaires ainsi obtenues, afin d'écarter les doublons. Cela ne nous semble pas vraiment réaliste pour un RCF, d'autant que le nombre de contraintes imposées est grand : le réseau est forcément constitué d'un puits connecté à des agrégateurs, eux-mêmes connectés aux nœuds terminaux (il s'agit d'un arbre de profondeur 2, et il n'y a pas d'autre topologie possible), et les nœuds terminaux sont forcément reliés à un agrégateur bien défini qu'ils ne peuvent changer (topologie statique). Le prix à payer est trop élevé, surtout pour ne pouvoir faire que des recherches de redondances en guise d'agrégation.

Les auteurs de [LC09b] proposent eux d'utiliser les courbes elliptiques pour assurer la sécurité dans les RCF. L'avantage de telles courbes est que les tailles des clés sont 8 à 20 fois plus petites que dans les autres algorithmes classiques (RSA...), ce qui se traduit par des opérations de chiffrement qui ne sont pas coûteuses (gain de mémoire et d'énergie). La solution de ces auteurs consiste à effectuer des mesures au niveau des nœuds terminaux, d'y ajouter le MAC (voir définition V.8) de la valeur mesurée et l'identifiant du nœud concerné, et de chiffrer le tout avant de l'envoyer à son agrégateur. Ce dernier déchiffre, agrège les différentes données qu'il possède, rechiffre le tout comme ci-dessus, et envoie le résultat à la prochaine couche d'agrégateurs. Du fait de l'utilisation des courbes elliptiques, leur algorithme est efficace, rapide et donc utilisable dans les RCF. Cependant, leur cryptosystème n'est pas homomorphe, ce qui les conduit à devoir déchiffrer avant d'agréger, ce qui est une faille de sécurité (les valeurs apparaissent en clair dans les agrégateurs), qui conduit de plus à des calculs supplémentaires inutiles.

La méthode de [Cas05], dite CTM, propose le cryptosystème suivant : choisir un grand entier M et un nombre aléatoire k dans $[\![0; M-1]\!]$, le cryptogramme c du message m est alors $c = m + k \pmod{M}$. Pour déchiffrer, on fait $c - k \pmod{M}$. Il faut donc que le k, tiré aléatoirement au niveau des nœuds terminaux, puisse être retrouvé par le puits, ce qui nécessite que le générateur du puits et celui du nœud considéré soient synchronisés, et que le puits sache d'où provient ladite valeur pour retrouver la bonne clé, ce qui est contradictoire avec la notion d'agrégation des données. Il faut de plus que les valeurs que l'on agrège aient été chiffrées avec la même clé, le même générateur, ce qui conduit à de multiples failles de sécurité, rendant par exemple possible une attaque par analyse de fréquence. De plus, ce cryptosystème n'est compatible qu'avec l'addition. Enfin, on a à nouveau affaire à un chiffrement symétrique, et donc au délicat problème de la gestion des clés secrètes et au caractère figé du réseau.

Enfin, les auteurs de [Pet07] rappellent les défauts des cryptosystèmes du CDA et du CTM, étudient diverses attaques qui peuvent survenir, et constatent qu'aucune de ces méthodes ne permet de faire face à tous les types d'attaques. Ils proposent donc, pour chiffrer, de réaliser la première méthode, puis la seconde, ce qui renforcerait à leur sens la sécurité du tout et permettrait de contourner les problèmes de sécurité du cryptosystème de Domingo-Ferrer. Cependant, une telle affirmation n'est pas évidente, et n'est pas prouvée. De plus, les coûts de calcul s'accumulent, rendant le tout sûrement irréalisable sur RCF. En outre, on reste dans du chiffrement symétrique, avec les problèmes qui y sont associés. Enfin, comme le CTM ne permet que l'addition sur les cryptogrammes, les auteurs perdent la possibilité de faire le produit de cryptogrammes, ce qui réduit d'autant le nombre d'agrégations possibles.

2. Notre contribution

Nous avons essayé pour notre part de résoudre les problèmes restant ouverts dans l'état de l'art, en proposant l'utilisation d'un cryptosystème presque totalement homomorphe sur courbes elliptiques, conduisant ainsi à une agrégation sécurisée de données réalisable sur réseaux de capteurs sans fil, et sans faille de sécurité. Les points-forts de notre contribution sont :

- Tout d'abord, notre schéma se base sur un cryptosystème qui a été prouvé sûr et n'a jusqu'à présent pas été cryptanalysé. Il est à ce jour le seul cryptosystème homomorphe sûr permettant plusieurs types d'opérations sur les cryptogrammes.
- De plus, deux textes clairs identiques conduisent à deux cryptogrammes différents, du fait du tirage d'un nombre aléatoire dans l'opération de chiffrement.
  Une telle propriété renforce la sécurité au sein du réseau de capteurs, en réduisant fortement l'intérêt qu'un adversaire peut avoir à espionner les échanges : il ne peut pas espérer tirer partie d'une analyse des fréquences d'apparition des différents cryptogrammes possibles. Cette propriété est fondamentale dans les réseaux de capteurs [BGN05, HST, LC09a].
- En plus de ces diverses propriétés, notre approche permet une utilisation optimisée des ressources et préserve au mieux l'énergie des capteurs, car le cryptosystème considéré œuvre sur des courbes elliptiques, donc manipule de très petites clés de chiffrement, de petits cryptogrammes, et ne nécessite qu'une faible puissance de calcul.
- Enfin, l'homomorphisme du cryptosystème que l'on utilise est le plus complet qui soit à l'heure actuelle : on peut faire autant d'additions que souhaité, ainsi qu'une multiplication. Cet homomorphisme *presque* total permet une grande variété de fonctions d'agrégation.

Dans ce qui suit, nous détaillerons notre contribution, qui est une étude d'adaptation du travail de Boneh *et al.* [BGN05] dans les réseaux de capteurs sans fil.

II. Le modèle

Cette section contient une courte introduction concernant la cryptographie sur courbes elliptiques, suivie du détail du cryptosystème de Boneh *et al.*. Le lecteur pourra se référer à [Sti02] ou [HMV04] pour de plus amples détails.

1. Opérations sur les courbes elliptiques

Les courbes elliptiques utilisées en cryptographie sont généralement définies sur deux types de corps finis : les corps $\mathbb{F}_p$, avec p un grand nombre premier, et les extensions de corps de la forme $\mathbb{F}_{2^m}$ [CTLC05]. Dans ce document, on s'intéresse aux courbes elliptiques sur $\mathbb{F}_p$. Soit $p > 3$, alors une courbe elliptique sur $\mathbb{F}_p$ d'équation $y^2 = x^3 + ax + b$ est l'ensemble

$$\mathcal{E} = \left\{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p, y^2 \equiv x^3 + ax + b \pmod p\right\},$$

où $a, b \in \mathbb{F}_p$ sont des constantes vérifiant $4a^3 + 27b^2 \not\equiv 0 \pmod p$. Une courbe elliptique sur $\mathbb{F}_p$ est constituée de l'ensemble des points de $\mathcal{E}$ auxquels on ajoute un point à l'infini $\mathcal{O}$.

L'addition entre deux points, et donc le multiple d'un point, est définie de la manière suivante, les opérations arithmétiques se réalisant dans $\mathbb{F}_p$ [Sti02] :
Soient $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ deux points de $\mathcal{E}$. Alors :

$$P + Q = \begin{cases} \mathcal{O} & \text{si} \quad x_2 = x_1 \text{ et } y_2 = -y_1, \\ (x_3, y_3) & \text{sinon,} \end{cases}$$

où :

- $x_3 = \lambda^2 - x_1 - x_2$,
- $y_3 = \lambda \times (x_1 - x_3) - y_1$,

et λ est donnée par :

$$\lambda = \begin{cases} (y_2 - y_1) \times (x_2 - x_1)^{-1} & \text{si} \quad P \neq Q\,, \\ (3x_1^2 + a) \times (2y_1)^{-1} & \text{si} \quad P = Q. \end{cases}$$

Finalement, on pose $P + O = O + P = P, \forall P \in \mathcal{E}$, ce qui conduit à une structure de groupe abélien pour $(\mathcal{E}, +)$. On rappelle que, dans ce cas, la multiplication $n \times P$ signifie $P + P + ... + P$ (n fois), et que $-P$ est le symétrique de P pour la loi + définie ci-dessus (c'est-à-dire l'unique point dont la somme avec P redonne l'élément neutre O).

2. Génération des clés publique et privée

Dans cette section, on explique comment générer les clés publique et privée, selon le cryptosystème proposé par Boneh *et al.* [BGN05]. L'étude de la complexité de cette génération sera initiée à la section 28.4.

Soit $\tau > 0$ un « paramètre de sécurité », qui n'est rien d'autre qu'un entier. Afin de générer les clés publique et privée, on doit tout d'abord calculer deux nombres premiers de τ-bits. Pour ce faire, un générateur pseudo-aléatoire peut être utilisé, pour obtenir deux vecteurs de τ bits, que l'on notera q_1 et q_2. Le test de Miller-Rabin peut alors être appliqué pour décider de la primalité de q_1 et q_2.

Posons $n = q_1 \times q_2$, et notons l le plus petit entier positif tel que $p = l \times n - 1$. On peut montrer que l est un nombre premier quand $p = 2 \pmod 3$.

Définissons maintenant les clés publique et privée. On note H le groupe des points de la courbe elliptique super-singulière $y^2 = x^3 + 1$ sur $\mathbb{F}_p$. Ce dernier contient $p + 1 = n \times l$ points, donc il a un sous-groupe d'ordre n, que l'on note G. On calcule alors deux générateurs g et u de G, et l'on pose $h = q_2 \times u$. Alors la clé publique du cryptosystème de Boneh *et al.* sera (n, G, g, h), et la clé privée sera simplement q_1.

3. Chiffrement et déchiffrement

Maintenant que nous savons générer les clés, nous pouvons nous lancer dans la description des algorithmes de chiffrement et de déchiffrement :

- **Chiffrement :** Supposons que notre espace de messages soit constitué d'entiers appartenant à l'ensemble $[\![0; T]\!]$, où $T < q_2$, et désignons par m le message (entier) à chiffrer. Commençons par tirer un entier r au sort dans l'intervalle $[\![0; n-1]\!]$. Alors le cryptogramme de m est le point C de G défini par :
$$C = m \times g + r \times h,$$
où + et × sont les lois définies précédemment.
- **Déchiffrement :** Une fois que le message C est arrivé à destination, pour le déchiffrer, nous utiliserons la clé privée q_1 et le logarithme discret de $(q_1 \times C)$ en base $q_1 \times g$, pour retrouver m de la manière suivante :
$$m = \log_{q_1 \times g}(q_1 \times C).$$
La complexité temporelle est en $O\left(\sqrt{T}\right)$ si l'on utilise la méthode lambda de Pollard. Signalons de plus que ce déchiffrement peut être accéléré en calculant à l'avance une table de puissances de $q_1 \times g$.

4. Propriétés d'homomorphisme

Comme nous l'avons mentionné auparavant, notre utilisation du cryptosystème de Boneh *et al.* nous assure des opérations de chiffrement et déchiffrement de faible complexité, qui ne nécessitent pas beaucoup de ressources : nous sommes sur courbes elliptiques.

En outre, ce cryptosystème possède des propriétés d'homomorphisme très intéressantes, permettant de toujours être en mesure de pouvoir réaliser diverses opérations au niveau des agrégateurs, et ce même si les valeurs reçues sont chiffrées. De manière plus explicite, le cryptosystème utilisé permet d'effectuer N additions et une multiplication directement sur les cryptogrammes.

Cela nous permet d'éviter le déchiffrement au niveau des agrégateurs, préservant ainsi la sécurité et l'énergie des nœuds.

a. Addition de deux cryptogrammes

L'addition des cryptogrammes se fait de la manière suivante.

Soient m_1 et m_2 deux messages, et C_1, C_2 leurs cryptogrammes respectifs. Alors la somme de C_1 et C_2, que nous noterons C, est égale à $C = C_1 + C_2 + r \times h$, où r est un entier tiré aléatoirement dans $[\![0, n-1]\!]$, et $h = q_2 \times u$ est l'un des termes de la clé publique. Cette opération sur les cryptogrammes est telle que le déchiffrement de C est égal à $m_1 + m_2$.

Remarque. Cette opération correspondant à l'addition peut être faite à plusieurs reprises : des sommes de sommes cryptées peuvent être obtenues.

b. Multiplication et déchiffrement de deux cryptogrammes

La multiplication de deux valeurs chiffrées, et le cas particulier de son déchiffrement, se font de la manière suivante.

Soient g et h les points de G définis ci-dessus, et E le *couplage de Weil modifié*, obtenu à partir du célèbre couplage de Weil e [BGN05], [BF03] par la formule : $E(P,Q) = e(x \times P, Q)$, où x est une racine du polynôme $X^3 - 1$ sur $\mathbb{F}_{p^2}$. Alors, la multiplication de deux cryptogrammes C_1 et C_2 s'obtient ainsi :

$$C_m = E(C_1, C_2) + r \times h_1,$$

où $h_1 = E(g,h)$, et r est un entier tiré aléatoirement dans $[\![0; n-1]\!]$.

Pour déchiffrer C_m, on calcule le logarithme discret de $q_1 \times C_m$ dans la base $q_1 \times g_1$, et l'on retrouve alors bien le produit des textes clairs :

$$m_1 m_2 = \log_{q_1 \times g_1}(q_1 \times C_m), \text{ où } g_1 = E(g,g).$$

III. Proposition d'agrégation sécurisée pour les réseaux de capteurs

1. Présentation

Comme la majorité des applications des réseaux de capteurs sans fil nécessitent un certain niveau de sécurité, le chiffrement des données mesurées est fréquemment nécessaire avant leur transmission. De plus, pour préserver la confidentialité des données, il est préférable de ne les déchiffrer qu'au niveau

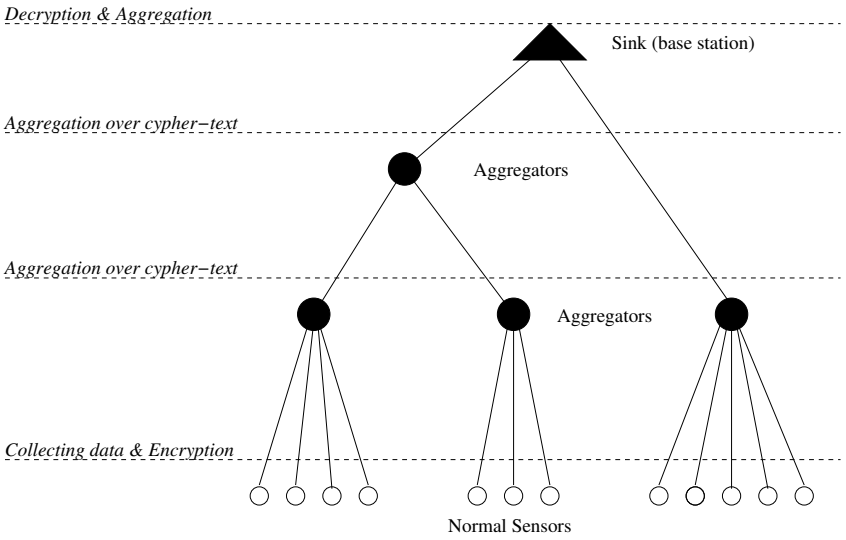


FIGURE 28.1 – Agrégation sécurisée de données dans les réseaux de capteurs

de la station de base (le puits). Dans notre travail, nous adoptons le scénario de la figure 28.1 : après la collecte d'informations, chaque capteur chiffre ses données en utilisant le cryptosystème sur courbe elliptique précédemment rappelé (en section 28.2.1), et l'envoie au plus proche agrégateur.

Les agrégateurs résument les cryptogrammes reçus, sans les déchiffrer. Pour calculer ces résumés, ou valeurs agrégées, ils ont le droit d'utiliser autant d'additions qu'ils veulent, ainsi qu'une et une seule multiplication. Enfin, les agrégateurs envoient le résultat de leurs calculs au puits, qui déchiffre alors les données agrégées.

Nous présentons dans ce qui suit quelques exemples de calcul de résumés, rendus possibles par notre méthode.

2. Exemples d'utilisation

Pour obtenir la moyenne arithmétique des mesures, les agrégateurs peuvent calculer la somme des cryptogrammes, et envoyer au puits cette somme *et* le nombre de valeurs agrégées. Plus précisément, chaque capteur chiffre ses données x_i pour obtenir cx_i. Le capteur transmet alors cx_i à son père, qui agrège les n cryptogrammes cx_j reçus de ses k enfants, simplement en faisant l'addition définie ci-dessus. La valeur ainsi obtenue est ensuite transmise vers le puits, accompagnée de n (éventuellement chiffré). Le puits reçoit finalement les valeurs n et $Cx = \sum_{i=1}^{n} cx_i$. Il peut déchiffrer Cx et diviser le résultat par n pour en déduire la moyenne recherchée.

Une autre agrégation habituelle consiste à calculer la variance des valeurs mesurées. Notre schéma peut aussi être utilisé pour effectuer ce calcul. Dans ce cas, chaque capteur i doit calculer $y_i = x_i^2$, où x_i est la valeur qu'il a mesuré, puis il doit chiffrer y_i et ainsi obtenir cy_i. x_i doit aussi être chiffré, en utilisant la manière exposée dans la section précédente. Le capteur transfert alors cy_i et cx_i à son nœud père. Ce nœud père agrège tous les cy_j de ses k nœuds fils, simplement en les additionnant. Il agrège aussi séparément les cx_j, comme ci-dessus. Les deux valeurs obtenues sont ensuite transmises.

Le puits reçoit alors $Cx = \sum_{i=1}^{n} cx_i$ et $Cy = \sum_{i=1}^{n} cy_i$. Cx est utilisé pour calculer la moyenne m, tandis que Cy sert à la variance : $Var = \frac{Vy}{n} - m^2$, où Vy est le déchiffrement de Cy.

Supposons maintenant que chaque agrégateur i de la première couche d'agrégation ait calculé la

moyenne x_i des valeurs chiffrées provenant de ses capteurs fils. Supposons en outre que ces agrégateurs soient pondérés en fonction de leur importance. Pour des raisons de sécurité, ce poids est également crypté, et la valeur chiffrée est notée w_i. Ce w_i peut être proportionnel au nombre de capteurs reliés à l'agrégateur i. Il peut aussi illustrer le fait que deux régions n'ont pas la même importance. Pour obtenir la moyenne pondérée, chaque agrégateur multiplie sa moyenne chiffrée x_i avec son poids chiffré w_i, comme on l'a expliqué précédemment. La valeur ainsi obtenue est ensuite transmise au puits, qui peut déchiffrer $w_i \times x_i$, puis sommer toutes ces valeurs déchiffrées, pour obtenir la moyenne pondérée définie ci-dessus.

IV. Évaluation de l'approche homomorphe

1. Présentation

Dans cette section, nous présentons une courte étude de sécurité dédiée aux réseaux de capteurs sans fil. Nous introduirons les attaques principales auxquels les réseaux de capteurs sans fil peuvent faire face, et nous expliquerons comment notre approche peut aider à les contrecarrer. Enfin, nous évoquerons quelques aspects pratiques susceptibles d'améliorer la sécurité de notre méthode et son coût.

Cette étude de sécurité sera suivie de résultats expérimentaux : nous avons simulé un réseau de capteurs sans fil sur une machine de type Intel Core2 Duo, en y embarquant, à tour de rôle, les cryptosystèmes RSA, et de Boneh *et al.* Les résultats de cette comparaison sont donnés à la fin de ce chapitre.

2. Étude de sécurité

Protéger les informations sensibles transmises par des nœuds au puits est une tâche difficile, vu les caractéristiques et exigences propres aux réseaux de capteurs. C'est une tâche nécessaire du fait de l'existence d'environnements hostiles. D'autre part, les réseaux sans fils connaissent des problèmes de sécurité auxquels les réseaux traditionnels ne sont pas confrontés.

a. Résultats face aux attaques usuelles

On convient habituellement que des adversaires peuvent principalement réaliser les attaques suivantes sur des réseaux sans fil :

Attaque par connaissance du texte clair : L'adversaire peut tirer partie du fait qu'une seule et unique clé de chiffrement est utilisée, dans le cas où deux textes clairs donnés conduisent au même cryptogramme. En utilisant des capteurs à proximité qu'ils auraient sous leur contrôle, les adversaires peuvent ainsi réussir une attaque par texte clair connu.

Attaque par texte clair choisi : Les adversaires peuvent agir sur les capteurs, et les forcer à utiliser des valeurs prédéterminées.

Attaque homme-du-milieu : Les adversaires peuvent injecter de fausses valeurs, ou renvoyer des valeurs déjà envoyées par des capteurs autorisés, afin de manipuler le processus d'agrégation des données.

Dans notre méthode, comme les données sont chiffrées à l'aide d'un cryptosystème à clés publiques, et comme ces clés publiques sont envoyées par le puits uniquement aux capteurs authentifiés, le réseau sans fil n'est pas vulnérable à une attaque homme-du-milieu. D'autre part, notre approche garantit que deux valeurs similaires conduisent à deux cryptogrammes différents, ce qui empêche les attaques de type connaissance du texte clair. Enfin, le schéma proposé utilise un cryptosystème homomorphe, donc

l'agrégation des données peut se faire sans déchiffrement, empêchant un adversaire de tirer parti d'une attaque par texte clair choisi.

b. Aspects pratiques

Dans cette section, nous présentons quelques aspects pratiques liés à notre modèle de sécurité. Nous allons commencer par étudier les tailles des clés de chiffrement, et nous les comparerons à d'autres approches existantes. Ensuite, nous expliquerons comment optimiser la taille des cryptogrammes, afin d'économiser plus d'énergie au niveau des capteurs par réduction des coûts de communication.

i. La taille des clés. Les cryptogrammes sont des points de la courbe elliptique $\mathcal{E}$. Ils sont constitués de couples de coordonnées entières inférieures ou égales à $p = lq_1q_2 - 1$. Pour garantir un niveau de sécurité convenable jusqu'aux années 2020, un cryptosystème [BR10], [LV01] :

- doit satisfaire $p \approx 2^{1886}$ dans les cas «classiques» sur $\mathbb{F}_p$, tels que le RSA ou ElGamal,
- doit avoir $p \approx 2^{161}$, quand il est défini sur courbes elliptiques sur $\mathbb{F}_p$.

Ainsi, pour un même niveau de sécurité, utiliser un cryptosystème sur courbes elliptiques ne requiert pas de grandes clés, contrairement au RSA ou ElGamal sur $\mathbb{F}_p$. Or, l'utilisation de petites clés et de courbes elliptiques conduisent à des cryptogrammes de petites tailles, à des opérations rapides, donc à moins de dépense d'énergie.

ii. Comment réduire plus encore la taille des cryptogrammes. Dans cette section, nous expliquons comment il est possible de diviser la taille des cryptogrammes par deux, dans le cryptosystème particulier que l'on a choisi d'utiliser, cette réduction étant primordiale pour limiter la consommation d'énergie lors de la transmission des données. On rappelle que les messages sont codés avec q_2 bits, ce qui conduit à des cryptogrammes ayant en moyenne 160 bits.

Comme le cryptogramme est un élément (x, y) de $\mathcal{E}$, qui est définie par $y^2 = x^3 + 1 \pmod p$, on peut donc compresser ce cryptogramme (x, y) en $(x, y \bmod 2)$, avant de l'envoyer à l'agrégateur. En effet, on peut tirer partie du fait que la valeur de y^2 se déduit de x et de la formule définissant $\mathcal{E}$, en n'envoyant que le bit $y \bmod 2$. Dans cette situation, on obtient des cryptogrammes de 81 bits en moyenne, pour des messages entre 20 et 40 bits.

Pour décompresser le cryptogramme (x, i), l'agrégateur doit calculer $z = x^3 + 1 \bmod p$, puis $y = \sqrt{z} \bmod p$. Cette racine carrée modulo p s'obtient par la formule suivante : $y = z^{(p+1)/4} \bmod p$. Alors :

- si $y \equiv i \pmod 2$, c'est que la décompression de (x, i) est (x, y),
- sinon la décompression est égale à $(x, p - y)$.

Ainsi donc, on peut réduire la taille des cryptogrammes de moitié, au prix de quelques opérations supplémentaires au niveau des agrégateurs. Il s'agit alors d'un compromis, qui peut être étudié au cas par cas.

3. Résultats expérimentaux

a. Remarque préliminaire

Signalons pour commencer que mettre en œuvre le cryptosystème de Boneh *et al.* sur un réseau de capteurs n'est pas élémentaire.

Ce cryptosystème est à l'heure actuelle le seul permettant de résoudre tous les problèmes rencontrés dans le cadre de l'agrégation sécurisée des données dans les réseaux de capteurs sans fil. Cependant, malgré sa célébrité, son utilisation pour le problème d'agrégation n'avait jusqu'à présent pas été proposée par les membres de la communauté. Nous attribuons cela au fait que ce cryptosystème n'est pas classique, relativement difficile à appréhender, et qu'un long cheminement sépare sa formulation théorique d'une réalisation pratique sur réseaux de capteurs.

Les efforts fournis pour arriver à la pleine compréhension du cryptosystème n'ont pas été quelconques. De même, la réalisation pratique, le passage de la formulation mathématique au programme informatique, ont été conséquents. Il existe bien sûr des outils permettant de simuler des réseaux de capteurs sans fil et d'en évaluer leurs performances. Cependant, ces simulateurs ne permettent pas de tester l'utilisation du cryptosystème de Boneh *et al.* : les outils de développement qu'ils offrent sont beaucoup trop rudimentaires, et leur code fermé ne permet pas de les enrichir de nouvelles fonctionnalités. Il nous a donc fallu redévelopper notre propre réseau de capteurs, comprenant l'étape de déploiement, la distinction entre les différents types de nœuds, la consommation d'énergie, la puissance de calcul et l'espace mémoire de chaque nœud, les communications et leurs coûts, *etc.* Il nous a aussi fallu incorporer le cryptosystème, la génération et la gestion des clés, ainsi que les fonctions d'agrégation. Enfin, la fin de vie des capteurs et du réseau devait être gérée, l'évaluation des résultats devait apparaître sous forme de tableaux de valeurs et de graphes, *etc.*

Nous avons choisi de réaliser ce réseau en python, du fait de sa portabilité, de notre bonne connaissance de ce langage de programmation, de l'aisance avec laquelle on peut y manipuler simplement des objets complexes, et parce que la puissance de calcul ne devait pas entrer en considération (les capteurs ont de faibles ressources). Une fois le réseau déployé, nous avons développé une classe pour les courbes elliptiques sur $\mathbb{F}_p$, intégrant les opérations algébriques définies sur ce genre d'objets, et avons commencé à mettre en place certaines briques du cryptosystème de Boneh (chiffrement, somme des cryptogrammes, Miller-Rabin,...) Nous n'avons pas pu aller jusqu'au bout par nos propres moyens, la programmation du couplage de Weil modifié et de l'extraction d'une racine d'un polynôme défini sur $\mathbb{F}_{p^2}$ nous posant notamment problème. Pour parvenir à nos fins, nous avons eu recours à la bibliothèque de courbes elliptiques contenue dans le projet SAGE pour finaliser la génération des clés, la multiplication des cryptogrammes (le couplage de Weil) et le déchiffrement (logarithme discret).

Ces différents éléments mis en place, nous avons ensuite conçu une deuxième version du réseau de capteurs, dans lequel nous avons reprogrammé RSA. Ce second réseau est assez différent du premier : les cryptogrammes ne sont pas des points d'une courbe elliptique, les fonctions de génération de clés, de chiffrement et de déchiffrement doivent être réécrites, et enfin l'agrégation est à revoir complètement : les fonctions condensant l'information sont à réécrire et une étape de déchiffrement/rechiffrement est à rajouter.

b. Les résultats

Pour montrer l'efficacité de notre approche, nous avons mené une série de simulations, comparant notre méthode à celle basée sur le cryptosystème RSA. Nous avons considéré un réseau formé de 500 nœuds terminaux, chacun étant équipé d'une batterie de 100 unités. Nous avons de plus supposé que la consommation d'énergie E d'un nœud est proportionnelle au temps de calcul (t, *i.e.* $E = kt$). Le même coefficient de proportionnalité k est pris dans les deux scénarios (le nôtre, et RSA). Les nœuds terminaux sont connectés à 50 agrégateurs, cette connexion se faisant d'une manière aléatoire.

Le fonctionnement de chaque simulation est le suivant :

1. Chaque nœud terminal tire une valeur aléatoire, la chiffre en utilisant l'une des deux méthodes de chiffrement proposées, puis l'envoie à son agrégateur.

Taille p de la clé	Énergie E (unités de batterie)
85	0,05%
125	0,07%
167	0,10%

Notre approche

Taille de la clé	Énergie E (unités de batterie)
945	0,53%
1416	1,63%
1891	3,63%

Chiffrement RSA

TABLE 28.1 – Comparaison au niveau des capteurs terminaux

Taille p de la clé	E (unités de batterie)
85	0,04%
125	0,07%
167	0,10%

Notre approche

Taille de la clé	E (unités de batterie)
945	8,09%
1416	24,04%
1891	56,27%

Chiffrement RSA

TABLE 28.2 – Comparaison au niveau des agrégateurs

2. Les agrégateurs attendent d'avoir reçus 20 valeurs (d'autres valeurs sont possibles), puis calculent la somme des données chiffrées reçues et envoient cette somme au puits (agrégation élémentaire).

Dans les deux cas, nous avons évalué la consommation d'énergie du réseau. Les tables 28.1 montrent l'énergie consommée par les nœuds lors de leurs opérations de chiffrement, respectivement pour le cryptosystème de Boneh *et al.*, et pour le RSA. Nous avons fait varier la taille des clés, et donc les niveaux de sécurité. On remarque que, pour un même niveau de sécurité, nous avons utilisé des clés plus petites, et économisé plus d'énergie avec le cryptosystème de Boneh *et al.* Par exemple, dans le cas d'un niveau de sécurité élevé, un nœud utilise une clé de 167 bits dans notre approche et 1891 bits pour le RSA. De plus, nous consommons 0,1% de notre batterie pour réaliser un chiffrement, quand le RSA en consomme 3,63%.

4. Étude de sécurité

Les tables 28.2 donnent la consommation d'énergie E lors de l'étape d'agrégation. Les mêmes hypothèses que ci-dessus ont été faites, la seule différence étant que les nœuds d'agrégation possèdent une batterie de 1000 unités. On peut constater que l'énergie utilisée par les agrégateurs est entre 50 et 500 fois plus importante pour le RSA, et ce en assurant un même niveau de sécurité.

La figure 28.2 contient une comparaison entre le RSA et les courbes elliptiques, concernant la consommation moyenne d'énergie d'un réseau de capteurs sans fil réalisant l'agrégation des données. On peut remarquer que notre approche permet d'économiser beaucoup plus d'énergie que le RSA.

Enfin, signalons que non seulement la quantité d'énergie nécessaire pour le chiffrement et l'agrégation est fortement réduite, mais en plus le puits reçoit dans notre schéma beaucoup plus de valeurs sur un intervalle de temps donné. En effet, notre chiffrement et notre agrégation sont bien plus rapides (ce qui se retrouve dans les tableaux traitant de la consommation d'énergie : on l'a supposée proportionnelle au temps de calcul).

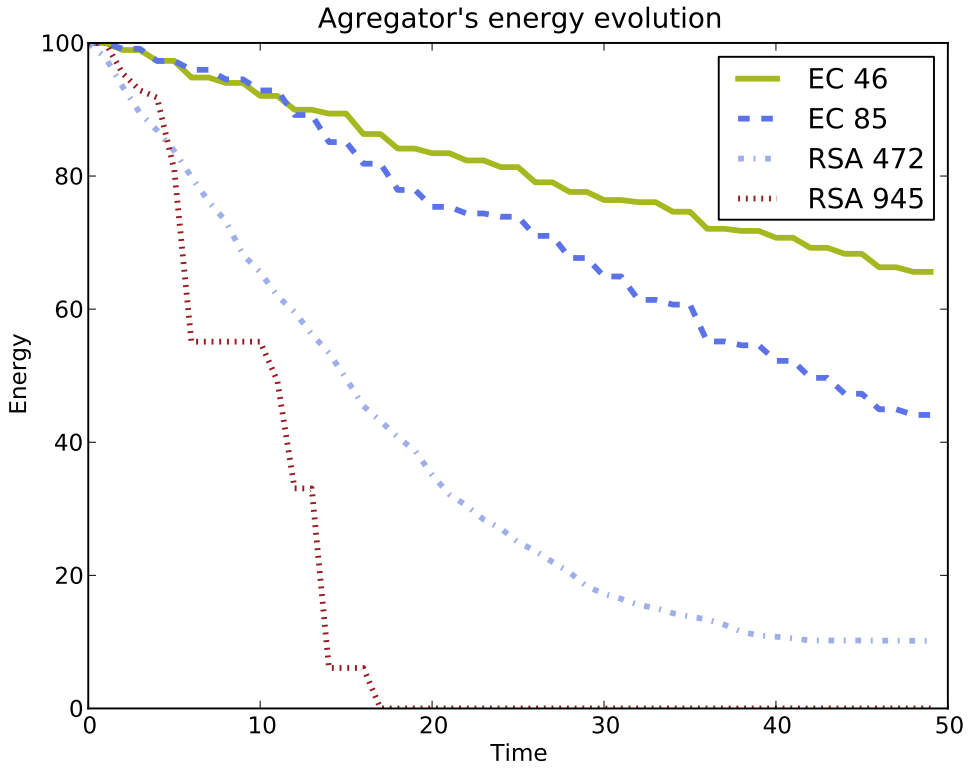


FIGURE 28.2 – Comparaison de la consommation d'énergie entre les cryptosystèmes RSA et de Boneh

CHAPITRE 29

Utilisation de la dissimulation d'information pour l'authentification dans les réseaux de capteurs

La connaissance des secrets d'autrui est un pouvoir enivrant.

Le poète
MICHAEL CONNELY

Dans le précédent chapitre, nous avons proposé d'utiliser un cryptosystème homomorphe pour réaliser des traitements dans le réseau, tout en préservant la confidentialité des données sensibles. Contrairement à d'autres schémas d'agrégation également basés sur du chiffrement homomorphe, notre méthode s'appuyait sur un cryptosystème efficace et sûr, c'est-à-dire qui n'a pas été cryptanalysé.

De plus, comme on peut réaliser sur les cryptogrammes le nombre d'additions souhaité, ainsi qu'une (unique) multiplication, notre méthode permet d'augmenter le nombre et la diversité des opérations d'agrégation permises au sein du réseau. Cependant, les cryptosystèmes homomorphes n'autorisent qu'un nombre restreint de fonctions d'agrégation : celles basées sur l'addition et la multiplication.

Une autre façon de parvenir à agréger des données de manière sûre dans les réseaux de capteurs sans fil consiste à authentifier les valeurs relevées, de sorte que l'authentification se « transmette » aux valeurs agrégées. La sécurité est un peu plus faible dans ce cas, mais le nombre de fonctions d'agrégation peut être augmenté. Ce chapitre reprend la contribution [BGM10b].

I. AUTHENTIFICATION DES DONNÉES PAR TATOUAGE NUMÉRIQUE

1. L'idée générale

Réussir l'agrégation de données authentifiées dans les réseaux sans fil, sans bien entendu perdre l'authentification, n'est évidemment pas une tâche aisée. Les schémas d'authentification actuels, qui s'appuient principalement sur les MAC (Message Authentication Code, voir chapitre V.8), ne sont pas pleinement adaptés, parce qu'un calcul de MAC est une opération consommant beaucoup d'énergie. De plus, même une infime modification des données infirme le MAC (ce qui est normal, et souhaité), donc comment espérer réaliser des opérations d'agrégation préservant l'authentification avec de tels outils ?

Dans [ZLDD08], une nouvelle façon de réaliser l'authentification des données au sein de réseaux de capteurs sans fil a été introduite par Zhang *et al.* Leur technique est basée sur le tatouage numérique (*c.f.* partie IV). Elle offre une approche statistique à l'authentification des données dans les réseaux de capteurs, tout en permettant certaines agrégations de données au sein dudit réseau.

Dans ce schéma, l'information d'authentification est modulée et insérée telle un filigrane dans les données prélevées par les nœuds. L'idée présentée dans [ZLDD08] est de considérer que les mesures produites par les capteurs à un temps donné constituent l'image de la grandeur mesurée. Chaque nœud terminal est assimilé à un pixel, la valeur que le capteur a mesuré est ainsi traduite en un niveau de gris pour le pixel correspondant.

Cette équivalence étant établie, les techniques de dissimulation d'information peuvent alors être utilisées pour authentifier les données transitant par le réseau. Pour bien faire, les données tatouées devraient pouvoir être agrégées par les nœuds intermédiaires sans requérir de contrôles supplémentaires. Et lors de la réception finale des données au niveau du puits, celui-ci devrait pouvoir vérifier l'authenticité de ces dernières, suivant la présence ou l'absence du filigrane.

Dans cette manière de faire, l'information d'authentification n'est ajoutée qu'au niveau des nœuds terminaux, et n'est vérifiée qu'au niveau du puits – sans aucune participation des nœuds intermédiaires.

2. La méthode de Zhang *et al.* dans les détails

Pour réaliser cette authentification, les auteurs de [ZLDD08] proposent d'utiliser un schéma de dissimulation d'information basé sur des techniques d'étalement de spectre.

Dans leur proposition, chaque nœud terminal intègre une partie du filigrane dans chacune de ses données mesurées. La détection des filigranes, seule opération lourde dans cette affaire, est alors uniquement confiée au puits.

De plus, leur méthode permet d'agréger les données le long du réseau, de la manière suivante : on considère un algorithme de tatouage numérique possédant une certaine robustesse par rapport à une compression donnée (JPEG, par exemple), mais qui d'un autre côté est un tatouage fragile la plupart du temps. En d'autres termes, l'algorithme choisi doit être un tatouage fragile, sauf pour une compression donnée. Alors l'agrégation peut être faite dans le réseau : on compressera les données reçues de la même manière qu'une image est compressée. Le tatouage sera préservé.

Par contre, si un adversaire attaque d'une quelconque manière le réseau, alors la marque sautera, du fait du caractère fragile du tatouage.

II. Notre contribution

1. Les problèmes de la solution existante

L'idée générale proposée par Zhang *et al.* est très intéressante nous semble-t-il, mais elle souddre de différents manquements, que l'on répertorie ci-dessous.

Pour commencer, l'étalement de spectre n'est pas robuste, surtout face aux attaques de type compression, ce qui va à l'encontre de leurs objectifs. Leur schéma semble supporter un certain degré de distorsion, ce qui est assez étonnant vu l'algorithme utilisé. Nous ne doutons pas de la bonne foi des auteurs, mais nous pensons que de tels résultats seraient difficiles à reproduire dans des situations plus concrètes, et qu'il vaudrait mieux utiliser des algorithmes ayant fait leur preuve face aux compressions.

Plus important, il nous semble que les problèmes que rencontrent Zhang *et al.* à définir clairement les propriétés requises par leur algorithme de tatouage viennent du fait qu'ils ne connaissent pas les notions de sécurité dans l'information dissimulée. Nous pensons que l'algorithme qu'il leur faut doit être robuste face aux compressions JPEG, et sûr. C'est-à-dire que des adversaires doués d'intelligence ne devraient rien pouvoir faire contre l'authentification. Or, les techniques à étalement de spectre ne sont stégo-sûres que dans la situation de tatouage dite naturelle avec $\eta = 1$, comme on l'a rappelé précédemment (théorème IV.1). La classe d'étalement de spectre utilisée dans [ZLDD08] est reliée à la situation dite « d'étalement de spectre classique », *i.e.* avec la modulation BPSK (voir définition IV.16). Cette sous-classe de techniques n'est pas stégo-sûre, et a un faible niveau de chaos-sécurité, comme cela a été rappelé dans la partie IV.

Une des conséquences de cette faille de sécurité est qu'un adversaire qui observe le réseau peut accéder à la clé secrète servant à insérer le filigrane, et ce dans toutes les situations suivantes : Attaque de l'image tatouée [11] (WOA), Attaque du message connu (KMA), Attaque de l'original connu (KOA), et Attaque du message constant (CMA).

2. Proposition

Pour améliorer la sécurité du réseau dans la configuration WOA, on sait que l'utilisation de l'étalement de spectre dit naturel avec $\eta = 1$ est nécessaire : cette sous-classe est la seule technique d'étalement de spectre à être stego-sûre, et donc la seule pouvant faire face à des attaques de type WOA.

Toutefois, l'étalement de spectre naturel est moins chaos-sûr que notre algorithme dhCI. Cet algorithme, basé sur les itérations chaotiques, est capable de faire face aux attaques des configurations KMA, KOA et CMA, comme on l'a établi dans la partie IV. En outre, on a vu aussi que notre technique avait une certaine robustesse (chapitre 23) que n'ont pas les techniques d'étalement de spectre.

Pour résumer, l'utilisation de notre schéma de tatouage, au lieu des techniques d'étalement de spectre, améliore la sécurité de la solution proposée par Zhang *et al.*, tout en garantissant une certaine robustesse nécessaire à l'agrégation (compression) des données authentifiées. Nous avons publié ce résultat, issu de nos réflexions sur la dissimulation d'information et les réseaux de capteurs, à la fin de notre thèse [BGM10b], mais nous n'avons pas eu le temps d'illustrer ce dernier par des simulations exhaustives. Nous le ferons dès que possible, pour valider par la pratique ce que la théorie prédit.

III. Conclusion et perspectives

Dans cette partie, nous avons étudié le problème de l'agrégation sécurisée des données au sein des réseaux de capteurs sans fil. Nous avons présenté deux méthodes permettant d'atteindre cet objectif.

La première méthode consiste à utiliser un cryptosystème asymétrique presque totalement homomorphe sur courbes elliptiques. Dans ce contexte, la sécurité est élevée, vu que les valeurs sont chiffrées au niveau des capteurs terminaux, et ne sont déchiffrées qu'au niveau du puits. L'agrégation est rendue possible, car l'opération de chiffrement est compatible avec l'addition et la multiplication. Ainsi, on évite de déchiffrer les données au niveau des agrégateurs, et l'on économise donc l'énergie des nœuds, tout en préservant un niveau de sécurité et une variété d'agrégation élevés.

D'autres solutions semblables à la nôtre ont déjà été proposées, mais aucune ne parvenait à satisfaire tous les pré-requis : ne pas faire d'opération coûteuse, transmettre de petites données, permettre un grand

11. Ces attaques, dont la définition a été rappelée au chapitre 19 de la partie IV dans le cadre des techniques de dissimulation, s'étendent sans problèmes à l'authentification des données par tatouage dans les réseaux sans fil.

nombre de fonctions d'agrégations, le tout sans jamais déchiffrer les données ni présenter d'autres failles de sécurité. Nous avons vérifié expérimentalement les prévisions théoriques : le constat est une forte réduction des coûts de chiffrement et d'agrégation des données, conduisant à un plus grand nombre de valeurs mesurées, sur une plus grande durée (optimisation de la durée de vie du réseau).

Nous avons aussi présenté dans ses grandes lignes une seconde approche, basée sur de la dissimulation d'information. Dans cette approche, l'agrégation pourrait être plus riche et variée, mais la sécurité est « réduite » à l'authentification des données. Les capteurs terminaux insèrent un filigrane dans leurs données mesurées, et les agrégateurs compressent leurs valeurs tatouées reçues. Le choix d'un algorithme de tatouage robuste permet d'arriver à conserver l'authentification malgré la compression. Nous avons repris une solution existante, détaillé ses problèmes de sécurité, et proposé des corrections rendant leur méthode sûre et viable.

Nous aimerions par la suite approfondir ces approches, notamment en listant les opérations d'agrégation rendues possibles et la manière de les réaliser suivant l'approche considérée, en réalisant des simulations les validant expérimentalement. Nous aimerions de plus porter de manière plus systématique les études de sécurité de la dissimulation d'information vers l'authentification des données dans les réseaux de capteurs. Une approche hybride de l'agrégation sécurisée de données dans les réseaux de capteurs sans fil pourrait être envisagée, et étudiée en détail : elle peut être obtenue en combinant le chiffrement homomorphe et l'authentification basée sur le tatouage, comme on le résume à la figure 29.1.

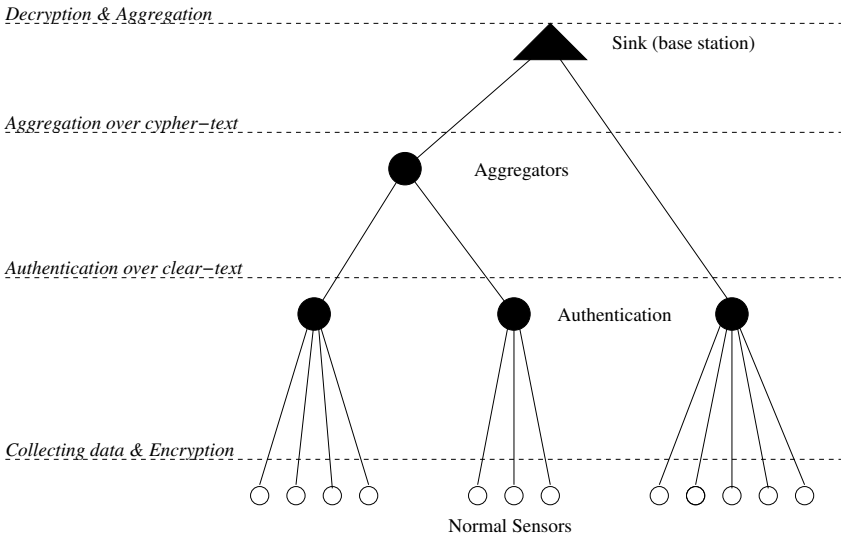


FIGURE 29.1 – Agrégation sécurisée de données dans les réseaux de capteurs sans fil

Enfin, nous voudrions pouvoir porter nos résultats des systèmes dynamiques discrets chaotiques vers les réseaux de capteurs sans fil. Ces réseaux peuvent être vus comme des systèmes, les capteurs en constituant les cellules. Certains problèmes de sécurité rencontrés dans ce genre de réseaux pourraient peut-être être résolus en tirant profit de calculs ou d'échanges imprévisibles entre les capteurs.

Conclusions et Annexes

CHAPITRE 30

Conclusions

La vie est une lutte de tous les instants avec la certitude d'être vaincu.

SCHOPENHAUER

I. SYNTHÈSE

Nous nous sommes intéressés à la possibilité de concevoir des programmes informatiques évoluant de manière imprévisible.

Pour réaliser une étude rigoureuse, il nous fallait un modèle mathématique pour décrire ce que l'on entend par « évolution imprévisible d'un programme sur ordinateur ». À cette fin, nous avons introduit une notion de *systèmes itératifs* et redéfini dans ce cadre : le synchronisme et l'asynchronisme, la convergence et et la non convergence, qu'elle soit conditionnelle ou inconditionnelle. Ces systèmes itératifs généralisent les itérations séries, parallèles et chaotiques, et lorsqu'ils sont utilisés sur des produits cartésiens d'ensembles booléens, ils modélisent assez bien l'action d'un programme sur ordinateur. Ces systèmes itératifs ont d'abord été étudiés sous l'angle des mathématiques discrètes, la recherche d'imprévisibilité faisant alors référence à la non convergence du système.

Nous avons commencé par nous intéresser aux plus simples de ces systèmes itératifs : les itérations parallèles. Plus précisément, nous avons cherché à savoir si de tels systèmes pouvaient se retrouver en situation de non convergence. Tel n'était pas le cas, le problème étant qu'une fois la condition initiale fournie, l'utilisation de la même fonction à chaque itération conduisait à un système convergent ou cyclique. Cette première étude nous a conduit à nous intéresser aux itérations dites « chaotiques » : ces systèmes itératifs restent élémentaires à calculer, et du fait de l'insertion d'une nouvelle entrée à chaque itérée, il nous a semblé que les travers des itérations parallèles pouvaient ainsi être contournés.

Grâce aux outils et résultats issus des mathématiques discrètes, nous avons principalement déduit deux critères pour que de tels systèmes itératifs soient non convergents. D'une part, il faudrait que la fonction d'itération soit non contractante, d'autre part que la stratégie d'itération ne soit pas pseudo-périodique. Il nous a semblé que la seconde contrainte se rapportait plus aux données fournies, aux conditions initiales, et que seule la première contrainte était spécifiquement liée au programme. Nous

avons déduit de cette étude un exemple canonique d'itérations chaotiques non convergentes, celles réalisées à partir de la négation vectorielle, qui nous semblait avoir l'un des potentiels les plus riches qui soient pour notre recherche d'imprévisibilité.

Il nous a ensuite fallu donner corps à la notion d'imprévisibilité : la non convergence n'était pas suffisamment riche pour les objectifs que l'on visait. La théorie mathématique du chaos nous a semblé être une bonne candidate, au moins pour initier l'étude d'imprévisibilité. C'est pourquoi nous avons fait le point sur les différentes notions de chaos mathématique.

L'étape suivante a consisté à faire en sorte de pouvoir étudier les itérations chaotiques au sein de la théorie du chaos. Au vu des applications visées, la définition d'une topologie τ pertinente était tout d'abord requise. Puis, une étape de modélisation a été nécessaire, pour nous permettre d'étudier l'espace sur lequel on itérait : cet espace est infini indénombrable, compact et complet. La continuité et la surjectivité des itérations chaotiques ont été établies, et l'étude de ses points périodiques a été menée en partie.

Une fois ce cadre topologique établi, il nous a été possible d'étudier le chaos des IC. Celui au sens de la multiplicité des périodes a pu être établi pour les itérations chaotiques généralisées de la négation vectorielle, quand le chaos selon Devaney a été vérifié pour toutes les IC de cette fonction. La sensibilité aux conditions initiales a été établie indépendamment du théorème de Banks, et la constante de sensibilité de la négation vectorielle a été évaluée. Nous avons alors cherché à savoir si d'autres fonctions d'itérations rendaient les IC chaotiques selon Devaney. Une caractérisation de ces dernières, des éléments de C, a été établie en se basant sur la forte connexité du GTPIC. Nous en avons profité pour déterminer la taille de C, et prouver que les points périodiques des IC de $f \in C$ sont dénombrables.

Après cette étude du chaos selon Devaney, nous sommes passés à la découverte d'autres formes de désordre : l'expansivité, le mélange topologique, le chaos selon Knudsen, selon Li-Yorke, *etc.* Au niveau quantitatif, la constante d'expansivité et l'entropie topologique ont été mesurées. Il nous restait à estimer l'exposant de Lyapunov, au moins pour la négation vectorielle, mais cet exposant suppose que la fonction d'itérations soit dérivable. Nous nous sommes alors rendu compte que les IC avec la négation vectorielle pouvaient se réécrire sous la forme d'une itération d'une fonction linéaire par morceaux sur un intervalle de $\mathbb{R}$, ce qui nous a permis d'en mesurer l'exposant de Lyapunov.

La réécriture des IC sous la forme d'une itération sur $\mathbb{R}$ était intéressante pour pouvoir comparer cette dernière à d'autres fonctions dans les applications que l'on visait. Seulement, la topologie sur $\mathbb{R}$ issue de la semi-conjugaison de τ n'était pas une topologie usuelle (la topologie de l'ordre). Nous nous sommes alors posés la question des conséquences de ce fait. Nous nous sommes rendus compte que τ était plus fine que la topologie de l'ordre, et que dans ce cas le chaos selon Devaney était préservé. Plus précisément, nous en sommes venus à la conclusion que le chaos selon Devaney des IC était plus riche que celui des fonctions usuellement itérées sur $\mathbb{R}$. Ces constatations nous ont conduit à réfléchir à des approches relatives et absolues au chaos selon Devaney.

Une fois l'étude de l'imprévisibilité des itérations chaotiques réalisée, nous nous sommes posés la question de l'exploitation de ces résultats. Il nous a semblé que les principaux problèmes empêchant de réaliser pleinement le chaos sur machine pouvaient se résumer à la liste suivante : la théorie mathématique du chaos était méconnue et non exploitée, le chaos n'était souvent qu'un ingrédient intervenant à un moment donné dans un plus vaste programme (qui n'a aucune raison d'hériter de cette propriété), l'ordinateur est une machine à états finis (et finit toujours par boucler), et ce dernier ne possède pas de réels alors que les plus célèbres suites chaotiques sont définies sur $\mathbb{R}$. Nous avons pu résoudre tous ces problèmes en considérant les itérations chaotiques : l'idée était de considérer que les IC étaient le programme implanté en machine, et que la stratégie chaotique se construisait au cours des itérations à partir de données extérieures. Ce faisant, notre machine n'est plus à états finis et nous ne manipulons que des

entiers. Enfin, il nous restait à faire en sorte que nos programmes applicatifs ne se résument qu'à des itérations chaotiques.

La première de nos applications concerna la dissimulation d'informations. Nous avons montré qu'il était possible de concevoir un algorithme de dissimulation chaotique, au sens mathématique le plus fort qui soit, et de faire en sorte que ce chaos soit préservé lors de son implémentation. Nous avons illustré cela en proposant l'algorithme dhCI, dans les domaines spatial et fréquentiel (ondelettes). La raison d'être du dhCI est purement théorique, ce dernier ne nous ayant servi qu'à prouver la faisabilité de notre approche. Quelques simulations ont été réalisées, juste pour laisser entrevoir qu'à terme, cet algorithme pourrait peut-être être amélioré de façon à obtenir une solution pratique au problème de la recherche d'une méthode sûre et robuste pour la dissimulation d'information.

Notre deuxième contribution au domaine de la dissimulation d'informations à consisté à montrer que la théorie mathématique du chaos peut avoir son intérêt dans le domaine de la sécurité. Nous avons montré que tout algorithme peut se mettre sous la forme d'un système dynamique discret, et qu'il peut alors être étudié sous l'angle de la théorie du chaos. Il devient alors possible d'étudier certaines classes d'attaques qui jusqu'alors ne pouvaient l'être. De plus, prouver qu'une méthode de dissimulation stégo-sûre est aussi chaotique permet de renforcer la confiance que l'on peut avoir dans cette dernière. Pour prouver que notre approche complémentaire de la sécurité pour la dissimulation d'informations est prête à être appliquée à des cas concrets, et ne se limite pas à notre dhCI, nous avons étudié la sécurité de la technique d'étalement de spectre sous l'angle de la théorie mathématique du chaos. Nous avons prouvé que cette dernière était chaotique selon Devaney, et possédait de plus les propriétés de transitivité forte et de mélange topologique. Nous en avons profité pour évaluer sa constante de sensibilité et démontrer son absence d'expansivité. Nous pouvons conclure de cette étude que les techniques d'étalement de spectre semblent sûres pour les attaques en configuration CMA, mais que du fait de leur manque d'expansivité, il conviendrait mieux d'utiliser le dhCI dans les configurations KOA et KMA.

Notre deuxième application concerne les fonctions de hachage. Nous avons prouvé qu'il était possible d'en concevoir qui soient réellement chaotiques, et expliqué en quoi cette propriété était intéressante dans ce domaine. Une fois encore, le but n'est pas ici de proposer une solution définitive, mais juste d'illustrer une faisabilité. Quelques simulations ont été réalisées et une première évaluation a été fournie.

Enfin, les réseaux de capteurs sont le cadre de notre dernière application : ces derniers se modélisent bien à l'aide de systèmes itératifs, et sont souvent déployés dans des situations où la sécurité joue un rôle important. Nous nous sommes principalement intéressés au problème de l'agrégation sécurisée des données au sein des réseaux de capteurs sans fil. Nous avons proposé deux solutions : la première est basée sur l'utilisation d'un cryptosystème totalement homomorphe sur courbes elliptiques, la seconde repose sur l'utilisation des techniques de dissimulation d'informations. Si d'autres solutions de ce type ont déjà été proposées par le passé, elles contenaient toutes des défauts les rendant inexploitables en pratique. Nous avons dressé un état de l'art de ces techniques, expliqué en quoi elles ne convenaient pas, et montré comment résoudre ces problèmes.

II. Bilans

À notre connaissance, nous avons été les premiers à nous intéresser à la non convergence et à l'imprévisibilité des systèmes itératifs des mathématiques discrètes (les « systèmes dynamiques discrets » de François Robert [Rob86]). Après avoir fait le point sur ce que l'on pouvait tirer des résultats de convergence des mathématiques discrètes, nous sommes passés à la topologie. Nous avons montré que les itérations chaotiques portent bien leur nom : elles engendrent du chaos dans les différents sens que ce terme a dans la théorie mathématique du même nom. Nous savions que les IC pouvaient s'avérer

intéressantes pour obtenir de meilleures convergences que les itérations séries et parallèles. Nous avons pour notre part montré que leur comportement non convergent était très riche.

Nous avons ensuite expliqué comment exploiter cette imprévisibilité dans les domaines de la dissimulation de l'information et des fonctions de hachage. Nous avons montré qu'il était possible, en s'y prenant bien, de concevoir des programmes dans ces domaines qui soient chaotiques au sens le plus rigoureux possible. Nous avons expliqué comment cela était possible, et quel pouvait en être l'intérêt. Entre autre, nous avons proposé de voir les différentes notions de chaos topologique comme autant de notions de sécurité. Cela nous a permis de montrer qu'un certain nombre de catégories d'attaques pouvaient être contrées par des programmes possédant certaines qualités chaotiques.

Enfin, nous avons apporté deux solutions pratiques effectives au problème de l'agrégation sécurisé des données dans les réseaux de capteurs sans fil.

III. PERSPECTIVES

Au niveau théorique, nous souhaiterions régler définitivement le problème concernant le choix de la meilleure topologie (voir chapitre 16), mieux comprendre l'impact de ce choix. Poursuivre la découverte des propriétés topologiques des itérations chaotiques, notamment en étendant l'étude des diverses notions de chaos à toutes les fonctions de C, serait intéressant : des fonctions pourraient avoir de meilleures propriétés qualitatives ou quantitatives que la négation vectorielle. Nous aimerions de plus considérer les itérations chaotiques généralisées et les itérations avec retard, afin de voir s'il pourrait y avoir un quelconque intérêt, en terme d'imprévisibilité, à généraliser le propos. Nous voudrions ensuite réaliser une étude approfondie des systèmes itératifs (définition 1) sous l'approche mathématiques discrètes (recherche de convergence, et de non convergence) et sous l'approche topologique. Nous sommes convaincus que ces systèmes itératifs ont leur importance, ils peuvent apporter quelque chose tant d'un point de vue théorique que d'un point de vue pratique. D'autres notions de chaos pourraient être intéressantes : il faudrait alors voir s'il pourrait y avoir un quelconque intérêt à les étudier. Nous imaginons poursuivre ensuite cette étude théorique de l'imprévisibilité en ayant recours aux branches suivantes des mathématiques, qui nous semblent toutes porteuses pour les objectifs visés : théorie des graphes, des automates, de la complexité, de la mesure et des probabilités. Toutes ces approches nous semblent intéressantes pour donner plus de corps à la notion d'imprévisibilité, et mieux étudier les systèmes itératifs sous cet aspect.

D'un point de vue pratique, nous avons proposé deux illustrations de la faisabilité de la conception d'un programme chaotique, la première dans le domaine de l'information dissimulée, la seconde dans celui des fonctions de hachage. Nous aimerions transformer ces preuves de faisabilité en réalisations concrètes satisfaisantes. Pour ce faire, il nous faudra réfléchir plus précisément à la complexité des algorithmes et à leur sécurité au sens classique du terme. Pour la dissimulation d'informations, la robustesse du dhCI devra être étudiée plus en détail, et la recherche des configurations telles que le dhCI soit stégo-sûr devra être menée. Pour notre fonction de hachage, l'étude de la résistance aux collisions devra notamment être réalisée. À ces études pratiques de sécurité devront s'ajouter des preuves théoriques faisant appel à la théorie de la complexité et aux probabilités. Ces études théoriques et pratiques nécessiteront sûrement une remise en question en profondeur de nos deux algorithmes illustratifs.

Nous souhaiterions alors étendre le nombre de connexions entre les propriétés issues de la théorie du chaos, et les différentes classes d'attaque que l'on rencontre dans nos trois domaines applicatifs. Par exemple, nous aimerions comprendre quelles sont les conséquences d'un système ayant une bonne entropie topologique, ou un exposant de Lyapunov élevé, dans le cadre de la dissimulation d'informations : les liens entre cette approche complémentaire de la sécurité et les approches existantes qui ont fait leur preuve devront être mieux comprises. Nous souhaiterions aussi poursuivre l'étude du chaos de

l'étalement de spectre, et étudier de la même manière d'autres algorithmes célèbres de dissimulation. Le même genre d'études devrait pouvoir se faire dans le domaine des fonctions de hachage et dans celui des réseaux de capteurs sans fil : considérer des méthodes existantes, les formuler, évaluer leur niveau de sécurité sous l'angle topologique, et comparer notre notion de sécurité à celles existantes dans ces domaines.

Dans le cadre des réseaux de capteurs sans fil, nous envisageons également d'approfondir notre approche dissimulation de l'information pour l'agrégation sécurisée des données, et réaliser des expérimentations grandeur nature. Nous aimerions de plus tirer profit de notre théorie des systèmes itératifs, et du chaos des IC dans ce domaine, notamment dans le cadre de la surveillance de régions et de la détection d'intrusion [MPH09, MSP09].

La possibilité de concevoir des programmes chaotiques pourrait s'étendre à de nouvelles applications. Nous avons vu dans ce document qu'un certain nombre d'auteurs utilisent les réseaux neuronaux pour concevoir des algorithmes de dissimulation d'information ou des fonctions de hachage. La plupart du temps, ces réseaux neuronaux sont couplés avec des suites chaotiques, et l'algorithme résultant est supposé chaotique. Nous souhaiterions pour notre part concevoir des réseaux neuronaux aux comportements chaotiques (selon Devaney, *etc.*) Dans le même esprit, nous pensons pouvoir utiliser nos travaux dans le cadre de la cryptographie symétrique [CCZ98] : nous retrouvons dans les chiffrements par blocs un système itératif, et l'ensemble des médias numériques en entrée [Fil00]. Ainsi, il serait possible soit de concevoir des cryptosystèmes symétriques aux comportements prouvés chaotiques suivant les diverses acceptions de ce terme en topologie, soit d'étudier les cryptosystèmes existants afin de voir s'il sont ou non prévisibles (notre approche de sécurité). De même, il nous semble possible de concevoir des virus polymorphes [Fil09, Fil10] au comportement chaotique le plus pur : l'état du système serait le code du virus, le terme courant de la stratégie serait fonction de l'identifiant et du contenu de la machine hôte, et la fonction d'itération comprendrait le polymorphisme, la modification du support hôte, et la détermination de la prochaine machine. Enfin, l'approche « hardware » du chaos nous semble riche de promesses, et nous pensons pouvoir apporter des preuves mathématiques et des programmes au comportement chaotiques en complément aux dispositifs de chaos physique de Laurent Larger [LD10].

Enfin, comme les nombres univers (qui, tels le nombre de Champernowne ou π, contiennent dans leurs décimales toutes les suites finies) contiennent tous les nombres possibles, il nous semble que nos itérations chaotiques contiennent toutes les implémentations passées et à venir. Il s'agit là d'une reformulation de la propriété de chaos selon Knudsen, dans le cadre du monde discret des programmes informatiques. Dit autrement, en partant de l'état initial correspondant au vecteur nul couplé à la stratégie dont les termes sont les chiffres de π, et en utilisant la négation vectorielle, on obtiendra au cours des itérations tous les contenus mémoire possibles avec toutes les entrées possibles, c'est-à-dire toutes les exécutions de tous les programmes possibles sur toutes les machines possibles. En ce sens, les itérations chaotiques sont un système itératif universel dont les conséquences de ce fait mériteraient d'être explorées.

ANNEXE A

Le SHA-1

Une fonction cryptographique de hachage fréquemment utilisée est le SHA-1. Elle intervient, par exemple, dans le DSS (*Digital Signature Standard*). Nous allons décrire son algorithme, puisque nous nous en sommes inspirés partiellement au chapitre 25.

I. NORMALISATION DU MESSAGE À HACHER

Soit $x \in \{0;1\}^*$, dont la longueur $||x||$ est supposée plus petite que 2^{64}. Pour calculer la valeur hachée de x, on procède ainsi...

On commence par ajouter des éléments à x pour que sa longueur soit un multiple de 512, en procédant de la manière suivante :

1. 1 est ajouté à la fin de x (*i.e.* $x \leftarrow x \circ 1$).
2. On ajoute des 0 à la fin de x pour que $||x||$ soit de la forme $512n - 64$.
3. $||x||$ est écrite en base 2 sur 64 bits, qui sont rajoutés à la fin du mot ci-dessus.

Exemple VI.2 : Soit x le mot binaire : 01100001 01100010 01100011 01100100 01100101. Après la première étape, x devient : 01100001 01100010 01100011 01100100 01100101 1. $||x|| = 41$, nous devons donc ajouter 407 zéros à x pour que cette longueur devienne 448=512-64. En représentation hexadécimale, x est maintenant :

61626364 65800000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000

À l'origine, $||x||$ était égale à 40 ; on écrit 40 en base 2, et l'on complète avec des 0 à gauche pour atteindre 64 bits, ce qui donne sur notre exemple (en représentation hexadécimale) :

00000000 00000028

et on colle ce nombre à la fin de x, pour obtenir finalement :

61626364 65800000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000028

II. Calcul de la valeur hachée

Pour calculer la valeur hachée, on a besoin des 80 fonctions de la forme suivante :

$$f_t : \{0;1\}^{32} \times \{0;1\}^{32} \times \{0;1\}^{32} \longrightarrow \{0;1\}^{32}$$

Ces fonctions sont définies ainsi [12] :

$$f_t(B,C,D) = \begin{cases} (B \wedge C) \vee (\neg B \wedge D) & \text{pour } 0 \leqslant t \leqslant 19, \\ B \oplus C \oplus D & \text{pour } 20 \leqslant t \leqslant 39, \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & \text{pour } 40 \leqslant t \leqslant 59, \\ B \oplus C \oplus D & \text{pour } 60 \leqslant t \leqslant 79. \end{cases}$$

On utilise aussi les constantes suivantes :

$$K_t = \begin{cases} 5A827999 & \text{pour } 0 \leqslant t \leqslant 19, \\ 6ED9EBA1 & \text{pour } 20 \leqslant t \leqslant 39, \\ 8F1BBCDC & \text{pour } 40 \leqslant t \leqslant 59, \\ AC62C1D6 & \text{pour } 60 \leqslant t \leqslant 79. \end{cases}$$

Soit x un mot binaire qui a été formaté en suivant la règle précédente. Sa longueur est donc divisible par 512, et on peut l'écrire comme une suite de mots de 512 bits :

$$x = M_1 M_2 \dots M_n.$$

Pour l'initialisation, on utilise les mots de 32 bits suivants : $H_0 = 67452301$, $H_1 = EFCDAB89$, $H_2 = 98BADCFE$, $H_3 = 10325476$, et $H_4 = C3D2E1F0$.

Soit $S^k(w)$ le décalage à gauche de k bits (en permutation circulaire), d'un mot w de 32 bits, et $+$ l'addition modulo 2^{16} des entiers correspondant aux mots de 16 bits.

On exécute maintenant la procédure suivante, pour $i = 1, 2, \dots, n$:

1. Écrire M_i comme une suite $M_i = W_0 W_1 \dots W_{15}$ de 16 mots de 32 bits.
2. Pour $t = 16, 17, \dots, 79$, calculer $W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$.
3. Poser $A = H_0, B = H_1, C = H_2, D = H_3$, et $E = H_4$.
4. Pour $t = 0, 1, \dots, 79$, calculer
   - $T = S^5(A) + f_t(B,C,D) + E + W_t + K_t$,
   - $E = D$, puis $D = C$, $C = S^6(B)$, $B = A$, et enfin $A = T$.
5. Ajouter A à H_0, B à H_1, C à H_2, D à H_3 et E à H_4.

La valeur hachée est alors : $\mathbf{sha\ 1}(x) = H_0 H_1 H_2 H_3 H_4$

12. $\wedge$ représente le *et* logique exécuté bit à bit, $\vee$ est le *ou* logique, quand $\neg$ est la négation.

Index

Bibliographie

[AAF08] David Arroyo, Gonzalo Alvarez, and Veronica Fernandez. On the inadequacy of the logistic map for cryptographic applications. *X Reunión Española sobre Criptología y Seguridad de la Información (X RECSI)*, 1 :77–82, 2008.

[AcKKS07] Onur Acıiçmez, Çetin Kaya Koç, and Jean-Pierre Seifert. On the power of simple branch prediction analysis. In *2007 ACM Symposium on Information, Computer and Communications Security (ASIACCS'07)*, pages 312–320. ACM Press, 2007.

[AGW05] Mithun Acharya, Joao Girao, and Dirk Westhoff. Secure comparison of encrypted data in wireless sensor networks. In *WIOPT '05 : Proceedings of the Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 47–53, Washington, DC, USA, 2005. IEEE Computer Society.

[AKM65] R. L. Adler, A. G. Konheim, and M. H. McAndrew. Topological entropy. *Trans. Amer. Math. Soc.*, 114 :309–319, 1965.

[AKS06] André Adelsbach, Stefan Katzenbeisser, and Ahmad-Reza Sadeghi. A computational model for watermark robustness. In Jan Camenisch, Christian S. Collberg, Neil F. Johnson, and Phil Sallee, editors, *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 145–160, Alexandria, VA, USA, July 2006. Springer.

[AKSX04] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Order preserving encryption for numeric data. In *SIGMOD '04 : Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 563–574, New York, NY, USA, 2004. ACM.

[Bah91] Jacques M. Bahi. *Algorithmes asynchrones pour des systèmes différentiels-algébriques. Simulation numérique sur des exemples de circuits électriques*. PhD thesis, Université de Franche-Comté, 1991.

[Bah98] Jacques M. Bahi. *Méthodes itératives dans des espaces produits. Application au calcul parallèle*. Habilitation à diriger des recherches, Université de Franche-Comté, 1998.

[Bau78] Gérard M. Baudet. Asynchronous iterative methods for multiprocessors. *J. ACM*, 25(2) :226–244, 1978.

[BB03] David Brumley and Dan Boneh. Remote timing attacks are practical. In *Proceedings of the 12th USENIX Security Symposium*, pages 1–14, 2003.

[BBCS92] J. Banks, J. Brooks, G. Cairns, and P. Stacey. On devaney's definition of chaos. *Amer. Math. Monthly*, 99 :332–334, 1992.

[BCGG99] Sebastiano Battiato, Dario Catalano, Giovanni Gallo, and Rosario Gennaro. Robust watermarking for images based on color manipulation. In Pfitzmann [Pfi00], pages 302–317.

[BCKK05] Mauro Barni, Ingemar J. Cox, Ton Kalker, and Hyoung Joong Kim, editors. *IWDW'05 : 4th International Workshop on Digital Watermarking*, volume 3710 of *Lecture Notes in Computer Science*, Siena, Italy, September 15-17 2005. Springer.

[BF03] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3) :586–615, 2003. Extended abstract in Crypto'01.

[BG10a] Jacques M. Bahi and Christophe Guyeux. An improved watermarking algorithm for internet applications. In *INTERNET'2010. The 2nd Int. Conf. on Evolving Internet*, pages ***–***, Valencia, Spain, September 2010. IEEE seccion ESPANIA.

[BG10b] Jacques M. Bahi and Christophe Guyeux. A new chaos-based watermarking algorithm. In *SECRYPT 2010, International conference on security and cryptography*, pages ***–***, Athens, Greece, 2010.

[BG10c] Jacques M. Bahi and Christophe Guyeux. Topological chaos and chaotic iterations, application to hash functions. In *WCCI'10, IEEE World Congress on Computational Intelligence*, pages ***–***, Barcelona, Spain, July 2010. IEEE.

[BGM10a] Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul. Efficient and robust secure aggregation of encrypted data in sensor networks. In *SENSORCOMM '10 : Proceedings of the 2010 Fourth International Conference on Sensor Technologies and Applications*, pages 472–477, Washington, DC, USA, 2010. IEEE Computer Society.

[BGM10b] Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul. Secure data aggregation in wireless sensor networks. homomorphism versus watermarking approach. In *AHDOC-NETS 2010, 2nd Int. Conf. on Ad Hoc Networks*, volume * of *Lecture Notes in ICST*, pages ***–***, Victoria, Canada, August 2010. To appear in the Springer LNISCT series.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. pages 325–341. 2005.

[BGW09] Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. A novel pseudo-random generator based on discrete chaotic iterations. In *Internet 2009*, pages 71–76, Cannes, France, August 2009. IEEE.

[BGW10a] Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. Improving random number generators by chaotic iterations. application in data hiding. In *ICCASM 2010, Int. Conf. on Computer Application and System Modeling*, pages ***–***, Taiyuan, China, October 2010. IEEE. to appear.

[BGW10b] Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. A pseudo random numbers generator based on chaotic iterations. application to watermarking. In *WISM 2010, Int. Conf. on Web Information Systems and Mining*, pages ***–***, Sanya, China, October 2010. To appear in the LNCS series.

[Bow71a] R. Bowen. Entropy for group endomorphisms and homogeneous spaces. *Trans. Amer. Math. Soc.*, 153 :401–414, 1971.

[Bow71b] R. Bowen. Periodic points and measures for axiom a diffeomorphisms. *Trans. Amer. Math. Soc.*, 154 :377–397, 1971.

[BR10] E. Barker and A. Roginsky. Draft nist special publication 800-131 recommendation for the transitioning of cryptographic algorithms and key sizes, 2010.

[BT88] Dimitri P. Bertsekas and John N. Tsitsiklis. Parallel and distributed iterative algorithms : a selective survey, 1988.

[BT89] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation : numerical methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[Cac98] Christian Cachin. An information-theoretic model for steganography. In *Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318. Springer Berlin / Heidelberg, 1998.

[Cas05] Claude Castelluccia. Efficient aggregation of encrypted data in wireless sensor networks. In *MobiQuitous*, pages 109–117. IEEE Computer Society, 2005.

[CB08] F. Cayre and P. Bas. Kerckhoffs-based embedding security classes for woa data hiding. *IEEE Transactions on Information Forensics and Security*, 3(1) :1–15, 2008.

[CBS06] R. Chandramouli, S. Bapatla, and K.P. Subbalakshmi. Battery power-aware encryption. *ACM transactions on information and system security*, pages 162–180, 2006.

[CCZ98] Claude Carlet, Pascale Charpin, and Victor Zinoviev. Codes, bent functions and permutations suitable for des-like cryptosystems. *Des. Codes Cryptography*, 15(2) :125–156, 1998.

[CFF05] F. Cayre, C. Fontaine, and T. Furon. Watermarking security : theory and practice. *IEEE Transactions on Signal Processing*, 53(10) :3976–3987, 2005.

[Cha85] Pascale Charpin. A description of some extended cyclic codes with application to reed-solomon codes. *Discrete Mathematics*, 56(2-3) :117–124, 1985.

[CJQZ06] Jin Cong, Yan Jiang, Zhiguo Qu, and Zhongmei Zhang. A wavelet packets watermarking algorithm based on chaos encryption. In Marina L. Gavrilova, Osvaldo Gervasi, Vipin Kumar, Chih Jeng Kenneth Tan, David Taniar, Antonio Laganà, Youngsong Mun, and Hyunseung Choo, editors, *ICCSA (1)*, volume 3980 of *Lecture Notes in Computer Science*, pages 921–928. Springer, 2006.

[CM69] D. Chazan and W. Miranker. Chaotic relaxation. *Linear algebra and its applications*, pages 199–222, 1969.

[CMK+97] Ingemar J. Cox, Senior Member, Joe Kilian, F. Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6 :1673–1687, 1997.

[CMM99] Ingemar Cox, Matt L. Miller, and Andrew L. Mckellips. Watermarking as communications with side information. In *Proceedings of the IEEE*, pages 1127–1141, 1999.

[CN03] Jung Hee Cheon and Hyun Soo Nam. A cryptanalysis of the original domingo-ferrer's algebraic privacy homomorphism, 2003.

[CPFPG05a] Pedro Comesaña, Luis Pérez-Freire, and Fernando Pérez-González. Fundamentals of data hiding security and their application to spread-spectrum analysis. In *IH'05 : Information Hiding Workshop*, pages 146–160. Lectures Notes in Computer Science, Springer-Verlag, 2005.

[CPFPG05b] Pedro Comesaña, Luis Pérez-Freire, and Fernando Pérez-González. The return of the sensitivity attack. In Barni et al. [BCKK05], pages 260–274.

[CTLC05] R.C.C. Cheung, N.J. Telle, W. Luk, and P.Y.K. Cheung. Secure encrypted-data aggregation for wireless sensor networks. *IEEE Trans. on Very Large Scale Integration Systems*, 13(9) :1048–1059, 2005.

[CXZ06] Zhu Congxu, Liao Xuefeng, and Li Zhihua. Chaos-based multipurpose image watermarking algorithm. *Wuhan University Journal of Natural Sciences*, 11 :1675–1678, 2006. 10.1007/BF02831848.

[DEF10] Anthony Desnos, Robert Erra, and Eric Filiol. Processor-dependent malware... and codes, 2010.

[Del] Delicious social bookmarking, http ://delicious.com/.

[Dev03] Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems, 2nd Edition*. Westview Pr., March 2003.

[DF02] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In *ISC '02 : Proceedings of the 5th International Conference on Information Security*, pages 471–483, London, UK, 2002. Springer-Verlag.

[DGW04] Zhao Dawei, Chen Guanrong, and Liu Wenbo. A chaos-based robust wavelet-domain watermarking algorithm. *Chaos, Solitons and Fractals*, 22 :47–54, 2004.

[Dob96] Hans Dobbertin. Cryptanalysis of md4. In *Proceedings of the Third International Workshop on Fast Software Encryption*, pages 53–69, London, UK, 1996. Springer-Verlag.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.

[FB08] Teddy Furon and Patrick Bas. Broken arrows. *EURASIP J. Inf. Secur.*, 2008 :1–13, 2008.

[Fil00] Eric Filiol. Decimation attack of stream ciphers. In Bimal K. Roy and Eiji Okamoto, editors, *INDOCRYPT*, volume 1977 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2000.

[Fil08] Eric Filiol. Passive and active leakage of secret data from non networked computer. In *Black Hat*, 2008.

[Fil09] Eric Filiol. *Les virus informatiques : techniques virales et antivirales avancées*. 2009.

[Fil10] Eric Filiol. Viruses and malware. In Peter P. Stavroulakis and Mark Stamp, editors, *Handbook of Information and Communication Security*, pages 747–769. Springer, 2010.

[For98] Enrico Formenti. *Automates cellulaires et chaos : de la vision topologique à la vision algorithmique*. PhD thesis, École Normale Supérieure de Lyon, 1998.

[For03] Enrico Formenti. *De l'algorithmique du chaos dans les systèmes dynamiques discrets*. PhD thesis, Université de Provence, 2003.

[Fri] The frick collection, http ://www.frick.org/.

[Fri98a] Alain Frisch. Entropie topologique et définition du chaos, 1998. [En ligne ; Page disponible le 12-août-2010].

[Fri98b] Alain Frisch. Entropie topologique et définition du chaos, 1998. Rapport de tipe.

[FSSM05] Peng Fei, Qiu Shui-Sheng, and Long Min. A secure digital signature algorithm based on elliptic curve and chaotic mappings. *Circuits Systems Signal Processing*, 24, No. 5 :585–597, 2005.

[Fur02] T. Furon. Security analysis, 2002. European Project IST-1999-10987 CERTIMARK, Deliverable D.5.5.

[Fur05] Teddy Furon. A survey of watermarking security. In Barni et al. [BCKK05], pages 201–215.

[GB10] Christophe Guyeux and Jacques M. Bahi. Hash functions using chaotic iterations. *Journal of Algorithms & Computational Technology*, 4(2) :167–182, 2010.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09 : Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178, New York, NY, USA, 2009. ACM.

[GFB10] Christophe Guyeux, Nicolas Friot, and Jacques M. Bahi. Chaotic iterations versus spread-spectrum : chaos and stego security. In *IIH-MSP'10, 6-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages ***–***, Darmstadt, Germany, October 2010. To appear.

[GLS88] C. Gotsman, D. Lehmann, and E. Shamir. Asynchronous dynamics of random boolean networks. In *San Diego '88 Neural Networks Conference*, 1988.

[GSW04] J. Girao, M. Schneider, and D. Westhoff. Cda : Concealed data aggregation in wireless sensor networks. In *Proceedings of the ACM Workshop on Wireless Security*, 2004.

[GTOMD05] G.S.El-Taweel, H.M. Onsi, M.Samy, and M.G. Darwish. Secure and non-blind watermarking scheme for color images based on dwt. *ICGST International Journal on Graphics, Vision and Image Processing*, 05 :1–5, April 2005.

[HKB09] A. Houmansadr, N. Kiyavash, and N. Borisov. Rainbow : A robust and invisible non-blind watermark for network flows. In *NDSS'09 : 16th Annual Network and Distributed System Security Symposium*, 2009.

[HMV04] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer Professional Computing, 2004.

[HST] Shih-I Huang, Shiuhpyng Shieh, and J. Tygar. Secure encrypted-data aggregation for wireless sensor networks. *Wireless Networks*.

[HST10] Shih-I Huang, Shiuhpyng Shieh, and J. D. Tygar. Secure encrypted-data aggregation for wireless sensor networks. *Wirel. Netw.*, 16(4) :915–927, 2010.

[Kal01] T. Kalker. Considerations on watermarking security. pages 201–206, 2001.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX :5–83, January 1883.

[Knu94a] Knudsen. Chaos without nonperiodicity. *Amer. Math. Monthly*, 101, 1994.

[Knu94b] C. Knudsen. *Aspects of noninvertible dynamics and chaos*. PhD thesis, Technical University of Denmark, 1994.

[Koc95] Paul C. Kocher. Cryptanalysis of diffie-hellman, rsa, dss, and other systems using timing attacks (extended abstract). In *Advances in Cryptology, CRYPTO '95 : 15th Annual International Cryptology Conference*, pages 27–31. Springer-Verlag, 1995.

[KP00] Stefan Katzenbeisser and Fabien A. Petitcolas, editors. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, Inc., Norwood, MA, USA, 2000.

[KSAT06] L. Kocarev, J. Szczepanski, J.M. Amigo, and I. Tomovski. Discrete chaos - i : Theory. *IEEE Trans. on Circuits Systems*, 53 :1300–1309, 2006.

[LC09a] H.-Y. Lin and T.-C. Chiang. Cooperative secure data aggregation in sensor networks using elliptic curve based cryptosystems. In *CDVE*, pages 384–387, 2009.

[LC09b] Hua-Yi Lin and Tzu-Chiang Chiang. Cooperative secure data aggregation in sensor networks using elliptic curve based cryptosystems. In *CDVE'09 : Proceedings of the 6th international conference on Cooperative design, visualization, and engineering*, pages 384–387, Berlin, Heidelberg, 2009. Springer-Verlag.

[LD10] L. Larger and J.M. Dudley. Nonlinear dynamics Optoelectronic chaos. *Nature*, 465(7294) :41–42, 05 2010.

[Led02] Eshter Ledoux. Introduction à la théorie du chaos, 2002. [En ligne ; Page disponible le 12-août-2010].

[LN08] An Liu and Peng Ning. Tinyecc : A configurable library for elliptic curve cryptography in wireless sensor networks. In *7th International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pages 245–256, April 2008.

[LV01] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Jour. of the International Association for Cryptologic Research*, 14(4) :255–293, 2001.

[LX07] Zhen Liu and Lifeng Xi. Image information hiding encryption using chaotic sequence. In *KES '07 : Knowledge-Based Intelligent Information and Engineering Systems and the XVII Italian Workshop on Neural Networks on Proceedings of the 11th International Conference*, pages 202–208, Berlin, Heidelberg, 2007. Springer-Verlag.

[LY75] T. Y. Li and J. A. Yorke. Period three implies chaos. *Amer. Math. Monthly*, 82(10) :985–992, 1975.

[LYGL07] Shao-Hui Liu, Hong-Xun Yao, Wen Gao, and Yong-Liang Liu. An image fragile watermark scheme based on chaotic image pattern and pixel-pairs. *Applied Mathematics and Computation*, 185 :869–882, 2007.

[MDS98] M. Martelli, M. Dang, and T. Seph. Defining chaos. *Mathematics Magazine*, 71 :112–122, 1998.

[MF03] H.S. Malvar and D. Florêncio. Improved spread spectrum : A new modulation technique for robust watermarking. *IEEE Trans. Signal Proceeding*, 53 :898–905, 2003.

[Mie75a] J.-C. Miellou. Algorithmes de relaxation chaotique à retards. *Rairo*, R1 :148–162, 1975.

[Mie75b] J.-C. Miellou. Itérations chaotiques à retards, étude de la convergence dans le cas d'espaces partiellement ordonnés. *C.R.A.S. Paris*, 280 :233–236, 1975.

[Mit99] Thomas Mittelholzer. An information-theoretic approach to steganography and watermarking. In Pfitzmann [Pfi00], pages 1–16.

[MKP08] Aidan Mooney, John G. Keating, and Ioannis Pitas. A comparative study of chaotic and white noise signals in digital watermarking. *Chaos, Solitons and Fractals*, 35 :913–921, 2008.

[MPH09] Moufida Maimour, CongDuc Pham, and Doan B. Hoang. A congestion control framework for handling video surveillance traffics on wsn. In *CSE (2)*, pages 943–948. IEEE Computer Society, 2009.

[MS85] Jean-Claude Miellou and Pierre Spitéri. Un critère de convergence pour des méthodes générales de point fixe. *Rairo – Modélisation mathématique et analyse numérique*, 19(4) :645–669, 1985.

[MSP09] Abdallah Makhoul, Rachid Saadi, and CongDuc Pham. Coverage and adaptive scheduling algorithms for criticality management on video wireless sensor networks. In *ICUMT*, pages 1–8. IEEE, 2009.

[PBA10] Andrea Pellegrini, Valeria Bertacco, and Todd M. Austin. Fault-based attack of rsa authentication. In *DATE*, pages 855–860. IEEE, 2010.

[Pet07] *On Concealed Data Aggregation for WSNs*, 2007.

[PFCTPPG06] Luis Perez-Freire, Pedro Comesana, Juan Ramon Troncoso-Pastoriza, and Fernando Perez-Gonzalez. Watermarking security : a survey. In *LNCS Transactions on Data Hiding and Multimedia Security*, 2006.

[Pfi00] Andreas Pfitzmann, editor. *IH'99 : 3rd International Workshop on Information Hiding*, volume 1768 of *Lecture Notes in Computer Science*, Dresden, Germany, September 29 - October 1. 2000. Springer.

[PFPgC06] Luis Perez-Freire, F. Pérez-gonzalez, and Pedro Comesaña. Secret dither estimation in lattice-quantization data hiding : A set-membership approach. In Edward J. Delp and Ping W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, California, USA, January 2006. SPIE.

[PFPGFC06] L. Pérez-Freire, F. Pérez-González, T. Furon, and P. Comesaña. Security of lattice-based data hiding against the known message attack. *IEEE Trans. on Information Forensics and Security*, 1(4) :421–439, dec 2006.

[Poe39] Edgar Alan Poe. The raven, April 1839. American Museum (Baltimore).

[PQL05] F. Peng, S.-S. Qiu, and M. Long. One way hash function construction based on two-dimensional hyperchaotic mappings. *Acta Phys. Sinici.*, 54 :98–104, 2005.

[QSLC02] J-J. Quisquater, D. Samyde, Université Catholique De Louvain, and Groupe Crypto. Side channel cryptanalysis, 2002.

[Rob86] François Robert. *Discrete Iterations, a Metric Study*, volume 6 of *Series in Computational Mathematics*. Springer-Verlag, 1986.

[Rue01] Sylvie Ruette. *Chaos en dynamique topologique, en particulier sur l'intervalle, mesures d'entropie maximale*. PhD thesis, Université d'Aix-Marseille II, 2001.

[Sau02] Boris Saulnier. Entropie topologique. Master's thesis, DEA Sémantique, Preuves et Langages, Paris 7, 2002.

[Sch80] Laurent Schwartz. *Analyse : topologie générale et analyse fonctionnelle*. Hermann, 1980.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28 :656–715, 1949.

[Sim84] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *Advances in Cryptology, Proc. CRYPTO'83*, pages 51–67, 1984.

[Spi74] Pierre Spitéri. *Contribution à l'étude de la stabilite au sens de liapounov de certains systemes differentiels non lineaires*. PhD thesis, Université de Franche-Comté, 1974.

[SQW+01] Li Shujun, Li Qi, Li Wenmin, Mou Xuanqin, and Cai Yuanlong. Statistical properties of digital piecewise linear chaotic maps and their roles in cryptography and pseudo-random coding. *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, 1 :205–221, 2001.

[Sti02] Douglas R. Stinson. *Cryptography : Theory and Practice, Second Edition*. Chapman & Hall/CRC, February 2002.

[TNM90] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding secret information into a dithered multi-level image. In *IEEE Military Communications Conference*, 1990.

[Van93] Tirkel Rankin Van. Electronic water mark, 1993.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology –*

EUROCRYPT 2010, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer Berlin / Heidelberg, 2010.

[Wag03] David Wagner. Cryptanalysis of an algebraic privacy homomorphism. In *Information Security*, volume 2851 of *Lecture Notes in Computer Science*, pages 234–239. Springer Berlin, Heidelberg, 2003.

[WBGF10] Qianxue Wang, Jacques M. Bahi, Christophe Guyeux, and Xaole Fang. Randomness quality of CI chaotic generators. application to internet security. In *INTERNET'2010. The 2nd Int. Conf. on Evolving Internet*, pages ***–***, Valencia, Spain, September 2010. IEEE seccion ESPANIA.

[WG07] Xianyong Wu and Zhi-Hong Guan. A novel digital watermark algorithm based on chaotic maps. *Physics Letters A*, 365(5-6) :403 – 406, 2007.

[WGA06] Dirk Westhoff, Joao Girao, and Mithun Acharya. Concealed data aggregation for reverse multicast traffic in sensor networks : Encryption, key distribution, and routing adaptation. *IEEE Transactions on Mobile Computing*, 5(10) :1417–1431, 2006.

[WGW07] Xianyong Wu, Zhi-Hong Guan, and Zhengping Wu. A chaos based robust spatial domain watermarking algorithm. In *ISNN '07 : Proceedings of the 4th international symposium on Neural Networks*, pages 113–119, Berlin, Heidelberg, 2007. Springer-Verlag.

[WSL] Haodong Wang, Bo Sheng, and Qun Li. Elliptic curve cryptography-based access control in sensor networks. *Int. J. Security and Networks*.

[WZZ03] X. M. Wang, J. S. Zhang, and W. F. Zhang. One-way hash function construction based on the extended chaotic maps switch. *Acta Phys. Sinici.*, 52, No. 11 :2737–2742, 2003.

[XLW09] Di Xiao, Xiaofeng Liao, and Yong Wang. Improving the security of a parallel keyed hash function based on chaotic maps. *Physics Letters A*, 373(47) :4346 – 4353, 2009.

[YLP06] Yu Yu, Jussipekka Leiwo, and Benjamin Premkumar. A study on the security of privacy homomorphism. In *ITNG '06 : Proceedings of the Third International Conference on Information Technology : New Generations*, pages 470–475, Washington, DC, USA, 2006. IEEE Computer Society.

[ZLDD08] Wei Zhang, Yonghe Liu, Sajal K. Das, and Pradip De. Secure data aggregation in wireless sensor networks : A watermark based authentication supportive approach. *Pervasive and Mobile Computing*, 4(5) :658 – 680, 2008.

[ZZX+04] Jian Zhao, Mingquan Zhou, Hongmei Xie, Jinye Peng, and Xin Zhou. A novel wavelet image watermarking scheme combined with chaos sequence and neural network. In Fuliang Yin, Jun Wang, and Chengan Guo, editors, *ISNN (2)*, volume 3174 of *Lecture Notes in Computer Science*, pages 663–668. Springer, 2004.

RÉSUMÉ

Les itérations chaotiques, un outil issu des mathématiques discrètes, sont pour la première fois étudiées pour obtenir de la divergence et du désordre. Après avoir utilisé les mathématiques discrètes pour en déduire des situations de non convergence, ces itérations sont modélisées sous la forme d'un système dynamique et sont étudiées topologiquement dans le cadre de la théorie mathématique du chaos. Nous prouvons que leur adjectif « chaotique » a été bien choisi : ces itérations sont du chaos aux sens de Devaney, Li-Yorke, l'expansivité, l'entropie topologique et l'exposant de Lyapunov, *etc.* Ces propriétés ayant été établies pour une topologie autre que la topologie de l'ordre, les conséquences de ce choix sont discutées. Nous montrons alors que ces itérations chaotiques peuvent être portées telles quelles sur ordinateur, sans perte de propriétés, et qu'il est possible de contourner le problème de la finitude des ordinateurs pour obtenir des programmes aux comportements *prouvés* chaotiques selon Devaney, *etc.* Cette manière de faire est respectée pour générer un algorithme de tatouage numérique et une fonction de hachage chaotiques au sens le plus fort qui soit. À chaque fois, l'intérêt d'être dans le cadre de la théorie mathématique du chaos est justifié, les propriétés à respecter sont choisies suivant les objectifs visés, et l'objet ainsi construit est évalué. Une notion de sécurité pour la stéganographie est introduite, pour combler l'absence d'outil permettant d'estimer la résistance d'un schéma de dissimulation d'information face à certaines catégories d'attaques. Enfin, deux solutions au problème de l'agrégation sécurisée des données dans les réseaux de capteurs sans fil sont proposées.

Mots-clés: Théorie du Chaos ; Systèmes Dynamiques Discrets ; Itérations Chaotiques ; Sécurité ; Fonctions de Hachage ; Stéganalyse ; Réseaux de Capteurs.

ABSTRACT

For the first time, the divergence and disorder properties of "chaotic iterations", a tool taken from the discrete mathematics domain, are studied. After having used discrete mathematics to deduce situations of non-convergence, these iterations are modeled as a dynamical system and are topologically studied into the framework of the mathematical theory of chaos. We prove that their adjective "chaotic" is well chosen : these iterations are chaotic, according to the definitions of Devaney, Li-Yorke, expansivity, topological entropy, Lyapunov exponent, and so on. These properties have been established for a topology different from the order topology, thus the consequences of this choice are discussed. We show that these chaotic iterations can be computed without any loss of properties, and that it is possible to circumvent the problem of the finiteness of computers to obtain programs that are proven to be chaotic according to Devaney, *etc.* The procedure proposed in this document is followed to generate a digital watermarking algorithm and a hash function, which are chaotic according to the strongest possible sense. At each time, the advantages of being chaotic as defined in the mathematical theory of chaos is justified, the properties to check are chosen depending on the objectives to reach, and the programs are evaluated. A novel notion of security for steganography is introduced, to address the lack of tool for estimating the strength of an information hiding scheme against certain types of attacks. Finally, two solutions to the problem of secure data aggregation in wireless sensor networks are proposed.

Keywords: Chaos Theory ; Discrete Dynamical Systems ; Chaotic Iterations ; Security ; Hash Functions ; Steganalysis ; Sensor Networks.