Chapter 1
A Topological Study of Chaotic Iterations Application to Hash Functions

Christophe Guyeux and Jacques M. Bahi

Abstract Chaotic iterations, a tool formerly used in distributed computing, has recently revealed various interesting properties of disorder leading to its use in the computer science security field. In this paper, a comprehensive study of its topological behavior is proposed. It is stated that, in addition to being chaotic as defined in the Devaney's formulation, this tool possesses the property of topological mixing. Additionally, its level of sensibility, expansivity, and topological entropy are evaluated. All of these properties lead to a complete unpredictable behavior for the chaotic iterations. As it only manipulates binary digits or integers, we show that it is possible to use it to produce truly chaotic computer programs. As an application example, a truly chaotic hash function is proposed in two versions. In the second version, an artificial neural network is used, which can be stated as chaotic according to Devaney.

Key words: Topological chaos; Chaotic iterations; Hash functions; Neural Networks.

1.1 INTRODUCTION

Chaotic iterations (CIs) were formerly a way to formalize distributed algorithms through mathematical tools [9]. By using these CIs, it was thus possible to study the convergence of synchronous or asynchronous programs over parallel, distributed, P2P, grid, or GPU platforms, in a view to solve linear and non-linear systems. We have proven at the IEEE World Congress on Computational Intelligence (WCCI'10) that CIs can behave chaotically, as it is defined by Devaney [5]. These proofs have

Christophe Guyeux and Jacques M. Bahi
Computer Science Laboratory LIFC, University of Franche-Comté
16, route de Gray - 25030 Besançon, France
Phone: +33 381666948; email: {christophe.guyeux, jacques.bahi}@univ-fcomte.fr.

been improved and more detailed in [4]. In this paper, which is an extension of [5, 4], we notably enlarge the theoretical study of CIs, among other things by computing its topological entropy, to obtain a comprehensive evaluation of its topological behavior. This study leads us to the conclusion that the chaos of CIs is very intense and constitutes a useful tool to be used in the computer science security field.

Chaos in information security fields as digital watermarking [10, 11], hash functions [24, 26], or pseudo-random number generators, is often disputed. This is due to the fact that this use is almost always based on the conception of algorithms that only include "somewhere" some well-known chaotic real functions like logistic, tent, or Arnold's cat maps, to obtain a program supposed to express these chaotic properties [14, 25, 22]. However, using such functions with other "obvious" parameters does not guarantee that the whole algorithm still remains chaotic. Such an assumption should at least be discussed. Moreover, even if the algorithm obtained by the inclusion of chaotic maps is itself proven to be chaotic, its implementation on a machine can lead to the fact that this chaotic nature is lost. This is due to the finite cardinality of the machine numbers set.

In this paper, as in [5, 4], we do not simply integrate chaotic maps into algorithms hoping that the result remains chaotic. We conceive algorithms for computer security that we have mathematically proven to be chaotic, as it is defined in the Devaney's theory. We raise the question of their implementation, proving in doing so that it is possible to design both a chaotic algorithm and its associated chaotic computer program. The chaos theory we consider is taken from the mathematical topology. It encompasses the well-known Devaney's definition of chaos and the notions of expansivity, topological entropy, and topological mixing. These notions of unpredictability are the most established ones into the mathematical theory of chaos. Our fundamental study is motivated by the desire to produce chaotic programs in the area of information security.

The paper begins by introducing the theoretical foundation of this approach. On the one hand we recall the definition of Devaney's topological chaos and on the other hand the definition of discrete chaotic iterations. Although these definitions are distinct from each other, we establish a link between them by giving conditions under which chaotic discrete iterations generate a Devaney's topological chaos. This study is deepened by giving some qualitative and quantitative evaluations of the disorder generated by chaotic iterations. These evaluations are not present in [4, 5]. We will focus in this paper on the notions of expansivity, topological mixing, and topological entropy. The proofs that the considered space is separated and compact have never been published before. Furthermore, the evaluation of the topological entropy is completely new. Then, because chaotic iterations are very suited for computer programming, this link allows us to generate programs in the computer science field that behave chaotically. This link was formerly presented in [5, 4] with some errors that are corrected here.

After having studied the theoretical aspects of our approach we focus on the practical ones. The important question is how to preserve the topological chaos properties in a set of a finite number of states. This question is answered in Section 1.5, by manipulating only integers and considering the use of new data at each iteration.

The general algorithm based on our approach, formerly presented in [5], is now explained in detail in Section 1.6. It is rewritten in the next section as an artificial neural network that can compute hash values while behaving chaotically. This new application is completely new and has never been published before.

The remainder of this paper is organized in the following way. In Section 1.2, the definitions of Devaney's chaos and discrete chaotic iterations are recalled. A link between these two notions is established and sufficient conditions to obtain Devaney's topological chaos from discrete chaotic iterations are given in Section 1.3. The chaotic behavior of CIs is deepened in Section 1.4, by studying some qualitative and quantitative properties of disorder. In Section 1.5, the question on how to preserve these chaotic properties into computers is answered. Then in Section 1.6 the general hash function scheme is given and illustrated, whereas in Section 1.7 it is applied to produce an artificial neural network able to hash some values in a chaotic manner. The paper ends by a conclusion section in which our contribution is summarized and planned future work is discussed.

1.2 BASIC RECALLS

This section is devoted to basic definitions and terminologies in the field of topological chaos and in the one of chaotic iterations.

1.2.1 Devaney's chaotic dynamical systems

In the sequel S^n denotes the n^{th} term of a sequence S and V_i denotes the i^{th} component of a vector V. $f^k = f \circ ... \circ f$ denotes the k^{th} composition of a function f. Finally, the following notation is used: $[\![1;N]\!] = \{1,2,\ldots,N\}$.

Consider a topological space $(\mathscr{X},\tau)$ and a continuous function $f : \mathscr{X} \to \mathscr{X}$.

Definition 1. f is said to be *topologically transitive* if, for any pair of open sets $U,V \subset \mathscr{X}$, there exists $k > 0$ such that $f^k(U) \cap V \neq \varnothing$.

Definition 2. An element x is a *periodic point* for f of period $n \in \mathbb{N}^*$ if $f^n(x) = x$.

Definition 3. f is said to be *regular* on $(\mathscr{X},\tau)$ if the set of periodic points for f is dense in $\mathscr{X}$: for any point x in $\mathscr{X}$, any neighborhood of x contains at least one periodic point (without necessarily the same period).

Definition 4. f is said to be *chaotic* on $(\mathscr{X},\tau)$ if f is regular and topologically transitive.

The chaos property is strongly linked to the notion of "sensitivity", defined on a metric space $(\mathscr{X},d)$ by:

Definition 5. f has *sensitive dependence on initial conditions* if there exists $\delta > 0$ such that, for any $x \in \mathcal{X}$ and any neighborhood V of x, there exist $y \in V$ and $n > 0$ such that $d\left(f^n(x), f^n(y)\right) > \delta$.

δ is called the *constant of sensitivity* of f.

Indeed, Banks *et al.* have proven in [7] that when f is chaotic and $(\mathcal{X}, d)$ is a metric space, then f has the property of sensitive dependence on initial conditions (this property was formerly an element of the definition of chaos). To sum up, quoting Devaney in [13], 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". Fundamentally different behaviors are consequently possible and occur in an unpredictable way.

1.2.2 Chaotic iterations

Let us consider a *system* with a finite number $\mathsf{N} \in \mathbb{N}^*$ of elements (or *cells*), so that each cell has a Boolean *state*. Having N Boolean values for these cells leads to the definition of a particular *state of the system*. A sequence which elements belong to $[\![1;\mathsf{N}]\!]$ is called a *strategy*. The set of all strategies is denoted by $\mathbb{S}$.

Definition 6. The set $\mathbb{B}$ denoting $\{0,1\}$, let $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$ be a function and $S \in \mathbb{S}$ be a strategy. The so-called *chaotic iterations* are defined by $x^0 \in \mathbb{B}^{\mathsf{N}}$ and

$$\forall n \in \mathbb{N}^*, \forall i \in [\![1;\mathsf{N}]\!], x_i^n = \begin{cases} x_i^{n-1} & \text{if } S^n \neq i \\ \left(f(x^{n-1})\right)_{S^n} & \text{if } S^n = i. \end{cases}$$

In other words, at the n^{th} iteration, only the S^n-th cell is "iterated". Note that in a more general formulation, S^n can be a subset of components and $\left(f(x^{n-1})\right)_{S^n}$ can be replaced by $\left(f(x^k)\right)_{S^n}$, where $k < n$, describing for example, delays transmission [20, 15]. Finally, let us remark that the term "chaotic", in the name of these iterations, has *a priori* no link with the mathematical theory of chaos, recalled above.

1.3 CHAOTIC ITERATIONS AS DEVANEY'S CHAOS

In this section is proven that chaotic iterations are a particular case of topological chaos, as it is defined in the Devaney's formulation'.

1.3.1 The new topological space

In this section we define a suitable metric space where chaotic iterations are continuous.

1.3.1.1 Defining the iteration function and the phase space

Let δ be the *discrete Boolean metric*, $\delta(x,y) = 0 \Leftrightarrow x = y$. Given a function f, define the function:

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

where + and . are the Boolean addition and product operations. Consider the phase space:

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

and the map defined on $\mathscr{X}$:

$$G_f(S,E) = \left(\sigma(S), F_f(i(S),E)\right), \tag{1.1}$$

where σ is the *shift* function defined by $\sigma(S^n)_{n\in\mathbb{N}} \in \mathbb{S} \longrightarrow (S^{n+1})_{n\in\mathbb{N}} \in \mathbb{S}$ and i is the *initial function* $i : (S^n)_{n\in\mathbb{N}} \in \mathbb{S} \longrightarrow S^0 \in [\![1;\mathsf{N}]\!]$. Then the chaotic iterations defined in (1.2.2) can be described by the following iterations:

$$\begin{cases} X^0 \in \mathscr{X} \\ X^{k+1} = G_f(X^k). \end{cases}$$

With this formulation, a shift function appears as a component of chaotic iterations. The shift function is a famous example of a chaotic map [13] but its presence is not sufficient enough to claim G_f as chaotic. In the rest of this section we rigorously prove that under some hypotheses, chaotic iterations generate topological chaos. Furthermore, due to the suitability of chaotic iterations for computer programming [3, 23, 2] we also have proven that this is true in the computer science field.

1.3.1.2 Cardinality of $\mathscr{X}$

By comparing $\mathbb{S}$ and $\mathbb{R}$, we have the following result.

Theorem 1. *The phase space $\mathscr{X}$ has, at least, the cardinality of the continuum.*

Proof. Let φ be the map which transforms a strategy into the binary representation of an element in $[0,1[$, as follows. If the n^{th} term of the strategy is 0, then the n^{th} associated digit is 0. If this n^{th} term is not equal to 0, then the associated digit is

1. With this construction, $\varphi : [\![1;\mathsf{N}]\!]^{\mathbb{N}} \longrightarrow [0,1]$ is onto. But $]0,1[$ is isomorphic to $\mathbb{R}$ ($x \in]0,1[\mapsto tan(\pi(x-\frac{1}{2}))$ is an isomorphism), so the cardinality of $[\![1;\mathsf{N}]\!]^{\mathbb{N}}$ is greater or equal to the cardinality of $\mathbb{R}$. As a consequence, the cardinality of the Cartesian product $\mathscr{X} = [\![1;\mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$ is greater or equal to the cardinality of $\mathbb{R}$.

Remark 1. This result is independent from the number of components of the system.

1.3.1.3 A new distance

Let us define a new distance between two points $X = (S,E), Y = (\check{S},\check{E}) \in \mathscr{X}$ by

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

where

$$\begin{cases} d_e(E,\check{E}) = \sum\limits_{k=1}^{\mathsf{N}} \delta(E_k,\check{E}_k), \\ d_s(S,\check{S}) \ = \dfrac{9}{\mathsf{N}} \sum\limits_{k=1}^{\infty} \dfrac{|S^k - \check{S}^k|}{10^k}. \end{cases}$$

This new distance has been introduced in [5] to satisfy the following requirements.

- When the number of different cells between two systems is increasing, then their distance should increase too.
- In addition, if two systems present the same cells and their respective strategies start with the same terms, then the distance between these two points must be small because the evolution of the two systems will be the same for a while. Indeed, the two dynamical systems start with the same initial condition, use the same update function, and as strategies are the same for a while, then components that are updated are the same too.

The distance presented above follows these recommendations. Indeed, if the floor value $\lfloor d(X,Y) \rfloor$ is equal to n, then the systems $E,\check{E}$ differ in n cells. In addition, $d(X,Y) - \lfloor d(X,Y) \rfloor$ is a measure of the differences between strategies S and $\check{S}$. More precisely, this floating part is less than 10^{-k} if and only if the first k terms of the two strategies are equal. Moreover, if the k^{th} digit is nonzero, then the k^{th} terms of the two strategies are different.

1.3.1.4 Continuity of the iteration function

To prove that chaotic iterations are an example of topological chaos in the sense of Devaney [13], G_f must be continuous in the metric space $(\mathscr{X},d)$.

Theorem 2. *G_f is a continuous function.*

Proof. We use the sequential continuity. Let $(S^n,E^n)_{n\in\mathbb{N}}$ be a sequence of the phase space $\mathscr{X}$, which converges to (S,E). We will prove that $\big(G_f(S^n,E^n)\big)_{n\in\mathbb{N}}$ converges to $\big(G_f(S,E)\big)$. Let us recall that for all n, S^n is a strategy, thus, we consider a sequence of strategies (*i.e.*, a sequence of sequences).
As $d((S^n,E^n);(S,E))$ converges to 0, each distance $d_e(E^n,E)$ and $d_s(S^n,S)$ converges to 0. But $d_e(E^n,E)$ is an integer, so $\exists n_0\in\mathbb{N}$, $d_e(E^n,E)=0$ for any $n\geqslant n_0$. In other words, there exists a threshold $n_0\in\mathbb{N}$ after which no cell will change its state: $\exists n_0\in\mathbb{N}, n\geqslant n_0\Rightarrow E^n=E$.

In addition, $d_s(S^n,S)\longrightarrow 0$, so $\exists n_1\in\mathbb{N}, d_s(S^n,S)<10^{-1}$ for all indexes greater than or equal to n_1. This means that for $n\geqslant n_1$, all the S^n have the same first term, which is S^0: $\forall n\geqslant n_1, S^n_0=S_0$.

Thus, after the $max(n_0,n_1)^{th}$ term, states of E^n and E are identical and strategies S^n and S start with the same first term.
Consequently, states of $G_f(S^n,E^n)$ and $G_f(S,E)$ are equal, so, after the $max(n_0,n_1)^{th}$ term, the distance d between these two points is strictly less than 1.
We now prove that the distance between $\big(G_f(S^n,E^n)\big)$ and $\big(G_f(S,E)\big)$ is convergent to 0. Let $\varepsilon>0$.

- If $\varepsilon\geqslant 1$, we see that distance between $\big(G_f(S^n,E^n)\big)$ and $\big(G_f(S,E)\big)$ is strictly less than 1 after the $max(n_0,n_1)^{th}$ term (same state).
- If $\varepsilon<1$, then $\exists k\in\mathbb{N}, 10^{-k}\geqslant\varepsilon>10^{-(k+1)}$. But $d_s(S^n,S)$ converges to 0, so

$$\exists n_2\in\mathbb{N}, \forall n\geqslant n_2, d_s(S^n,S)<10^{-(k+2)},$$

  thus after n_2, the $k+2$ first terms of S^n and S are equal.

As a consequence, the $k+1$ first entries of the strategies of $G_f(S^n,E^n)$ and $G_f(S,E)$ are the same (G_f is a shift of strategies) and due to the definition of d_s, the floating part of the distance between (S^n,E^n) and (S,E) is strictly less than $10^{-(k+1)}\leqslant\varepsilon$.

In conclusion,

$$\forall\varepsilon>0, \exists N_0=max(n_0,n_1,n_2)\in\mathbb{N}, \forall n\geqslant N_0, d\left(G_f(S^n,E^n);G_f(S,E)\right)\leqslant\varepsilon.$$

G_f is consequently continuous.

In this section, we proved that chaotic iterations can be modeled as a dynamical system in a topological space. In the next section, we show that some chaotic iterations behave chaotically, as defined by Devaney's theory.

1.3.2 Discrete chaotic iterations as topological chaos

To prove that we are in the framework of Devaney's topological chaos, we have to find a Boolean function f such that G_f satisfies the regularity, transitivity, and sensitivity conditions. We will prove that the vectorial logical negation

$$f_0(x_1,\ldots,x_{\mathsf{N}}) = (\overline{x_1},\ldots,\overline{x_{\mathsf{N}}}) \tag{1.2}$$

is a suitable function.

1.3.2.1 Regularity

Firstly, let us prove that,

Theorem 3. *Periodic points of G_{f_0} are dense in $\mathscr{X}$.*

Proof. Let $(\check{S},\check{E}) \in \mathscr{X}$ and $\varepsilon > 0$. We are looking for a periodic point $(\widetilde{S},\widetilde{E})$ satisfying $d((\check{S},\check{E});(\widetilde{S},\widetilde{E})) < \varepsilon$. As ε can be strictly lesser than 1, we must choose $\widetilde{E} = \check{E}$. Let us define $k_0(\varepsilon) = \lfloor log_{10}(\varepsilon) \rfloor + 1$ and consider the set

$$\mathscr{S}_{\check{S},k_0(\varepsilon)} = \left\{ S \in \mathbb{S} / S^k = \check{S}^k, \forall k \leqslant k_0(\varepsilon) \right\}.$$

Then, $\forall S \in \mathscr{S}_{\check{S},k_0(\varepsilon)}, d((S,\check{E});(\check{S},\check{E})) < \varepsilon$. It remains to choose $\widetilde{S} \in \mathscr{S}_{\check{S},k_0(\varepsilon)}$ such that $(\widetilde{S},\widetilde{E}) = (\widetilde{S},\check{E})$ is a periodic point for G_{f_0}. Let

$$\mathscr{J} = \left\{ i \in \{1,...,\mathsf{N}\} / E_i \neq \check{E}_i, \text{ where } (S,E) = G_{f_0}^{k_0}(\check{S},\check{E}) \right\},$$

$i_0 = card(\mathscr{J})$, and $j_1 < j_2 < ... < j_{i_0}$ the elements of $\mathscr{J}$. Then, $\widetilde{S} \in \mathscr{S}_{\check{S},k_0(\varepsilon)}$ defined by

- $\widetilde{S}^k = \check{S}^k$, if $k \leqslant k_0(\varepsilon)$,
- $\widetilde{S}^k = j_{k-k_0(\varepsilon)}$, if $k \in \{k_0(\varepsilon)+1, k_0(\varepsilon)+2, ..., k_0(\varepsilon)+i_0\}$,
- and $\widetilde{S}^k = \widetilde{S}^j$, where $j \leqslant k_0(\varepsilon)+i_0$ is satisfying $j \equiv k \pmod{k_0(\varepsilon)+i_0}$, if $k > k_0(\varepsilon)+i_0$,

is such that $(\widetilde{S},\widetilde{E})$ is a periodic point (of period $k_0(\varepsilon)+i_0$), which is $\varepsilon-$close to $(\check{S},\check{E})$.
As a conclusion, $(\mathscr{X}, G_{f_0})$ is regular.

1.3.2.2 Transitivity

Regarding the transitivity property of G_{f_0}, we can show that,

Theorem 4. *$(\mathscr{X}, G_{f_0})$ is topologically transitive.*

Proof. Let us define $\mathscr{E} : \mathscr{X} \to \mathbb{B}^{\mathsf{N}}$, such that $\mathscr{E}(S,E) = E$. Let $\mathscr{B}_A = \mathscr{B}(X_A, r_A)$ and $\mathscr{B}_B = \mathscr{B}(X_B, r_B)$ be two open balls of $\mathscr{X}$, with $X_A = (S_A, E_A)$ and $X_B = (S_B, E_B)$. We are looking for $\widetilde{X} = (\widetilde{S},\widetilde{E})$ in $\mathscr{B}_A$ such that $\exists n_0 \in \mathbb{N}, G_{f_0}^{n_0}(\widetilde{X}) \in \mathscr{B}_B$.
$\widetilde{X}$ must be in $\mathscr{B}_A$ and r_A can be strictly lesser than 1, so $\widetilde{E} = E_A$. Let $k_0 =$

$\lfloor \log_{10}(r_A)+1 \rfloor$. Then $\forall S \in \mathbb{S}$, if $S^k = S_A^k, \forall k \leqslant k_0$, then $(S,\widetilde{E}) \in \mathscr{B}_A$. Let $(\check{S},\check{E})$ be equal to $G_{f_0}^{k_0}(S_A,E_A)$ and $c_1,...,c_{k_1}$ denote the elements of the set $\{i \in [\![1,\mathsf{N}]\!]/\check{E}_i \neq \mathscr{E}(X_B)_i\}$. So any point X of the set

$$\{(S,E_A) \in \mathscr{X} / \forall k \leqslant k_0, S^k = S_A^k \text{ and } \forall k \in [\![1,k_1]\!], S^{k_0+k} = c_k\}$$

is satisfying $X \in \mathscr{B}_A$ and $\mathscr{E}\left(G_{f_0}^{k_0+k_1}(X)\right) = E_B$. Lastly, let k_2 be $\lfloor \log_{10}(r_B) \rfloor + 1$. Then $\widetilde{X} = (\widetilde{S},\widetilde{E}) \in \mathscr{X}$ defined by:

1. $\widetilde{X} = E_A$,
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. $\forall k \in \mathbb{N}^*, \widetilde{S}^{k_0+k_1+k} = S_B^k$,

is such that $\widetilde{X} \in \mathscr{B}_A$ and $G_{f_0}^{k_0+k_1}(\widetilde{X}) \in \mathscr{B}_B$. This fact concludes the proof of the theorem.

1.3.2.3 Devaney's Chaos

In conclusion, $(\mathscr{X},G_{f_0})$ is topologically transitive and regular. Then we have the following result:

Theorem 5. *G_{f_0} is a chaotic map on $(\mathscr{X},d)$ in the sense of Devaney.*

We have proven that the set $\mathscr{C}$ of the iterate functions f so that $(\mathscr{X},G_f)$ is chaotic (according to the definition of Devaney), is a nonempty set. In future work, we will deepen the study of $\mathscr{C}$, among other things, by computing its cardinality and characterizing this set.

1.4 TOPOLOGICAL PROPERTIES OF CHAOTIC ITERATIONS

In this section, some qualitative and quantitative topological properties for chaotic iterations with G_{f_0} will be studied in detail. These properties reinforce the chaotic behavior of the system.

1.4.1 Topological mixing

The topological mixing is a strong version of transitivity:

Definition 7. A discrete dynamical system is said to be *topologically mixing* if and only if, for any couple of disjoint open set $U, V \neq \emptyset$, $n_0 \in \mathbb{N}$ can be found so that $\forall n \geqslant n_0, f^n(U) \cap V \neq \emptyset$.

We have the result [16],

Theorem 6. *$(\mathscr{X}, G_{f_0})$ is topologically mixing.*

This result is an immediate consequence of the lemma below.

Lemma 1. *For any open ball B of $\mathscr{X}$, an index n can be found such that $G^n_{f_0}(B) = \mathscr{X}$.*

Proof. Let $B = B((E,S),\varepsilon)$ be an open ball, which the radius can be considered as strictly less than 1. The elements of B all have the same state E and are such that an integer $k\,(= -\log_{10}(\varepsilon))$ satisfies:

- all the strategies of B have the same k first terms,
- after the index k, all values are possible.

Then, after k iterations, the new state of the system is $G^k_{f_0}(E,S)_1$ and all the strategies are possibles (any point of the form $(G^k_{f_0}(E,S)_1, \hat{\mathrm{S}})$, with any $\hat{\mathrm{S}} \in \mathbb{S}$, is reachable from B).

Let $(E',S') \in \mathscr{X}$. We will prove that any point of $\mathscr{X}$ is reachable from B.

Indeed, let s_i be the list of the different cells between $G^k_{f_0}(E,S)_1$ and E', and $|s|$ its size. The point $(\check{E},\check{S})$ of B defined by:

- $\check{E} = E$,
- $\check{S}^i = S^i, \forall i \leqslant k$,
- $\check{S}^{k+i} = s_i, \forall i \leqslant |s|$,
- $\forall i \in \mathbb{N}, S^{k+|s|+i} = S'^i$.

is such that $G^{k+|s|}_{f_0}(\check{E},\check{S}) = (E',S')$. This conclude the proof of the lemma.

1.4.2 Quantitative measures

1.4.2.1 General definitions

In Section 1.3.2.3 we have proven that discrete chaotic iterations produce a topological chaos by checking two qualitative properties, namely transitivity and regularity. This mathematical framework offers tools to measure this chaos quantitatively.
The first of these measures is the constant of sensitivity defined in Definition 5. Intuitively, a function f having a constant sensitivity equal to δ implies that there exists points arbitrarily close to any point x, which eventually separate from x by at least δ under some iterations of f. This induces that an arbitrarily small error on an initial condition *might be* magnified upon iterations of f.

Another important tool is defined below.

Definition 8. A function f is said to have the property of *expansivity* if

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

Then, ε is the *constant of expansivity* of f. We also say that f is ε-expansive.

A function f has a constant of expansivity equal to ε if an arbitrarily small error on any initial condition is *always* magnified until ε.

1.4.2.2 Sensitivity

The sensitive dependence on the initial conditions has been shown as a consequence of the regularity and the transitivity of chaotic iterations. However, in the set of machine numbers, we have shown in [5] that the notion of regularity must be redefined. This is the reason why this sensitivity should be proven without using the result of Banks [7], to be sure that this dependence is preserved in practical use of chaotic iterations.

In addition, the constant of sensitivity will be obtained during this proof.

Theorem 7. *$(\mathscr{X}, G_{f_0})$ has sensitive dependence on initial conditions and its constant of sensitivity is equal to* $\mathsf{N}-1$.

Proof. Let $\check{X} = (\check{S}, \check{E}) \in \mathscr{X}$. We are looking for $\widetilde{X} = (\widetilde{S}, \widetilde{E}) \in \mathscr{X}$ such that $d(\check{X}, \widetilde{X}) \leqslant \delta$ and $\exists n_0 \in \mathbb{N}$, $d\left(G_{f_0}^{n_0}(\check{X}); G_{f_0}^{n_0}(\widetilde{X})\right) \geqslant \mathsf{N}-1$. Let k_0 be $\lfloor \log_{10}(\delta) \rfloor + 1$. So, if $S \in \{S \in \mathbb{S} / \forall k \leqslant k_0, S^k = \check{S}^k\}$, then $d\left((S, \check{E}), (\check{S}, \check{E})\right) \leqslant \delta$.

Let $\mathscr{J} = \left\{ i \in [\![1, \mathsf{N}]\!] / \ \mathscr{E}\left(G_{f_0}^{k_0}(\check{S}, \check{E})\right)_i = \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{S}, \check{E})\right)_i \right\}$ and $p = card\,(\mathscr{J})$.

If $p = \mathsf{N}$, then $(\widetilde{S}, \widetilde{E}) \in \mathscr{X}$ defined by:

1. $\widetilde{E} = \check{E}$,
2. $\forall k \leqslant k_0, \widetilde{S}^k = \check{S}^k$,
3. $\forall k \in [\![1, \mathsf{N}]\!], \widetilde{S}^{k_0+k} = k$,
4. $\forall k > k_0 + \mathsf{N}, \widetilde{S}^k = 1$.

satisfies $d((\widetilde{S}, \widetilde{E}); (\check{S}, \check{E})) < \delta$ and $\forall i \in [\![1, \mathsf{N}]\!], \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\widetilde{S}; \widetilde{E})\right)_i \neq \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{S}; \check{E})\right)_i$, so the result is obtained.

Else, let $j_1 < j_2 < ... < j_p$ be the elements of $\mathscr{J}$ and $j_0 \notin \mathscr{J}$. Then $\widetilde{X} = (\widetilde{E}, \widetilde{S}) \in \mathscr{X}$ defined by

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. $\forall k \in \mathbb{N}^*, \widetilde{S}^{k_0+p+k} = j_0$.

is such that $d(\check{X}, \widetilde{X}) < \delta$. In addition, $\forall i \in [\![1, p]\!], \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{X})\right)_{j_i} \neq \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\widetilde{X})\right)_{j_i}$, because:

- $\forall i \in [\![1,\mathsf{N}]\!], \mathscr{E}\left(G_{f_0}^{k_0}(\check{X})\right)_i = \mathscr{E}\left(G_{f_0}^{k_0}(\widetilde{X})\right)_i$, due to the definition of k_0.
- $\forall i \in [\![1,p]\!], j_i \in \mathscr{J} \Rightarrow \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{X})\right)_{j_i} = \mathscr{E}\left(G_{f_0}^{k_0}(\check{X})\right)_{j_i}$, according to the definition of $\mathscr{J}$.
- $\forall i \in [\![1,p]\!], j_i$ appears exactly one time in $\widetilde{S}^{k_0}, \widetilde{S}^{k_0+1}, ..., \widetilde{S}^{k_0+\mathsf{N}}$, so

$$\mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\widetilde{X})\right)_{j_i} \neq \mathscr{E}\left(G_{f_0}^{k_0}(\widetilde{X})\right)_{j_i}.$$

Lastly, $\forall i \in [\![1,\mathsf{N}]\!] \setminus \{j_0, j_1, ..., j_p\}, \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\widetilde{X})\right)_i \neq \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{X})\right)_i$, because:

- $\forall i \in [\![1,\mathsf{N}]\!], \mathscr{E}\left(G_{f_0}^{k_0}(\check{X})\right)_i = \mathscr{E}\left(G_{f_0}^{k_0}(\widetilde{X})\right)_i$,
- $i \notin \mathscr{J} \Rightarrow \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{X})\right)_i \neq \mathscr{E}\left(G_{f_0}^{k_0}(\check{X})\right)_i$,
- $i \notin \{\widetilde{S}^{k_0}, \widetilde{S}^{k_0+1}, ..., \widetilde{S}^{k_0+\mathsf{N}}\} \Rightarrow \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\widetilde{X})\right)_i = \mathscr{E}\left(G_{f_0}^{k_0}(\widetilde{X})\right)_i$.

So, in this case, $\forall i \in [\![1,\mathsf{N}]\!] \setminus \{j_0\}, \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\widetilde{S};\widetilde{E})\right)_i \neq \mathscr{E}\left(G_{f_0}^{k_0+\mathsf{N}}(\check{S};\check{E})\right)_i$ and the result of sensitivity is still obtained.

1.4.2.3 Expansivity

In this section we offer the proof that chaotic iterations are expansive [16] when f_0 is the update function:

Theorem 8. *$(\mathscr{X}, G_{f_0})$ is an expansive chaotic system. Its constant of expansivity is equal to 1.*

Proof. If $(S,E) \neq (\check{S};\check{E})$, then:

- Either $E \neq \check{E}$ and so at least one cell is not in the same state in E and $\check{E}$. Consequently the distance between (S,E) and $(\check{S};\check{E})$ is greater or equal to 1.
- Or $E = \check{E}$. So the strategies S and $\check{S}$ are not equal. Let n_0 be the first index such that the terms S and $\check{S}$ differ. Then $\forall k < n_0, G_{f_0}^{n_0}(S,E) = G_{f_0}^{k}(\check{S},\check{E})$, and $G_{f_0}^{n_0}(S,E) \neq G_{f_0}^{n_0}(\check{S},\check{E})$.
  As $E = \check{E}$, the cell that has changed in E at the n_0-th iterate is not the same than the cell that has changed in $\check{E}$, so the distance between $G_{f_0}^{n_0}(S,E)$ and $G_{f_0}^{n_0}(\check{S},\check{E})$ is greater or equal to 2.

So the expansivity property is established.

Remark 2. Expansivity is a kind of avalanche effect: any initial error is always magnified when iterating the system.

Remark 3. $(\mathscr{X}, G_{f_0})$ is not A-expansive, for any $A > 1$: let us consider two points $X = (E,S)$ and $X' = (E',S')$ with the same strategy ($S = S'$) and only one different cell ($d_e(E,E') = 1$). So, $\forall n \in \mathbb{N}, d_e\left(\mathscr{E}\left(G_{f_0}^{n}(X)\right), \mathscr{E}\left(G_{f_0}^{n}(X')\right)\right) = 1$.

1.4.3 Topological Entropy

Another important tool to measure the chaotic behavior of a dynamical system is the topological entropy, which is defined for compact topological spaces. Before studying the entropy of CIs, we must then check that $(\mathscr{X},d)$ is compact.

1.4.3.1 Compacity Study

In this section, we will prove that $(\mathscr{X},d)$ is a compact topological space, in order to study its topological entropy later. Firstly, as $(\mathscr{X},d)$ is a metric space, it is separated. It is however possible to give a direct proof of this result:

Theorem 9. *$(\mathscr{X},d)$ is a separated space.*

Proof. Let $(E,S) \neq (\hat{\mathrm{E}},\hat{\mathrm{S}})$ two points of $\mathscr{X}$.

1. If $E \neq \hat{\mathrm{E}}$, then the intersection between the two balls $\mathscr{B}\left((E,S),\frac{1}{2}\right)$ and $\mathscr{B}\left((\hat{\mathrm{E}},\hat{\mathrm{S}}),\frac{1}{2}\right)$ in empty.
2. Else, it exists $k \in \mathbb{N}$ such that $S^k \neq \hat{\mathrm{S}}^k$, then the balls $\mathscr{B}\left((E,S),10^{-(k+1)}\right)$ and $\mathscr{B}\left((\hat{\mathrm{E}},\hat{\mathrm{S}}),10^{-(k+1)}\right)$ can be chosen.

The sequential characterization of the compacity for metric spaces can now be used to obtain the following result.

Theorem 10. *$(\mathscr{X},d)$ is a compact metric space.*

Proof. Let $(E^n,S^n)_{n\in\mathbb{N}}$ be a sequence of $\mathscr{X}$.

1. A state $E^{\tilde{\mathrm{n}}}$ which appears an infinite number of time in this sequence can be found. Let
$$I = \{(E^n,S^n)\big/E^n = E^{\tilde{\mathrm{n}}}\}.$$
For all $(E,S) \in I$, $S_0^n \in [\![1,\mathsf{N}]\!]$, and I is an infinite set. Then $\tilde{\mathrm{k}} \in [\![1,\mathsf{N}]\!]$ can be found such that an infinite number of strategies of I starts with $\tilde{\mathrm{k}}$.
Let n_0 be the smallest integer such that $E^n = E^{\tilde{\mathrm{n}}}$ and $S_0^n = \tilde{\mathrm{k}}$.
2. The set
$$I' = \{(E^n,S^n)\big/E^n = E^{n_0} \text{ and } S_0^n = S_0^{n_0}\}$$
is infinite, then one of the elements of $[\![1,\mathsf{N}]\!]$ will appear an infinite number of times in the S_1^n of I': let us call it $\tilde{\mathrm{I}}$.
Let n_1 be the smallest n such that $(E^n,S^n) \in I'$ and $S_1^n = \tilde{\mathrm{I}}$.
3. The set
$$I'' = \{(E^n,S^n) | E^n = E^{n_0} \text{ and } S_0^n = S_0^{n_0} \text{ and } S_1^n = S_1^{n_1}\}$$
is infinite, *etc.*

Let $l = \left(E^{n_0},\left(S_k^{n_k}\right)_{k\in\mathbb{N}}\right)$, then the subsequence (E^{n_k},S^{n_k}) converges to l.

1.4.3.2 Topological entropy

Let (X,d) be a compact metric space and $f : X \to X$ be a continuous map. For each natural number n, a new metric d_n is defined on X by

$$d_n(x,y) = \max\{d(f^i(x), f^i(y)) : 0 \leq i < n\}.$$

Given any $\varepsilon > 0$ and $n \geqslant 1$, two points of X are ε-close with respect to this metric if their first n iterates are ε-close.

This metric allows one to distinguish in a neighborhood of an orbit the points that move away from each other during the iteration from the points that travel together. A subset E of X is said to be (n,ε)-separated if each pair of distinct points of E is at least ε apart in the metric d_n. Denote by $H(n,\varepsilon)$ the maximum cardinality of an (n,ε)-separated set,

Definition 9. The *topological entropy* of the map f is defined by (see e.g. [1] or [8])

$$h(f) = \lim_{\varepsilon \to 0} \left(\limsup_{n \to \infty} \frac{1}{n} \log H(n,\varepsilon) \right).$$

We have the result,

Theorem 11. *Entropy of $(\mathscr{X}, G_f)$ is infinite.*

Proof. Let $\mathrm{E}, \check{\mathrm{E}} \in \mathbb{B}^{\mathsf{N}}$ such that $\exists i_0 \in [\![1,N]\!], \mathrm{E}_{i_0} \neq \check{\mathrm{E}}_{i_0}$. Then, $\forall \mathrm{S}, \check{\mathrm{S}} \in \mathscr{S}$,

$$d((\mathrm{E},\mathrm{S});(\check{\mathrm{E}},\check{\mathrm{S}})) \geqslant 1$$

But the cardinal c of $\mathscr{S}$ is infinite, then $\forall n \in \mathbb{N}, c > e^{n^2}$.

Then for all $n \in \mathbb{N}$, the maximal number $H(n,1)$ of $(n,1)-$separated points is greater than or equal to e^{n^2}, so

$$h_{top}(G_f,1) = \overline{lim} \frac{1}{n} log\,(H(n,1)) > \overline{lim} \frac{1}{n} log\left(e^{n^2}\right) = \overline{lim}\,(n) = +\infty.$$

But $h_{top}(G_f,\varepsilon)$ is an increasing function when ε is decreasing, then

$$h_{top}\left(G_f\right) = \lim_{\varepsilon \to 0} h_{top}(G_f,\varepsilon) > h_{top}(G_f,1) = +\infty,$$

which concludes the evaluation of the topological entropy of G_f.

We have proven that it is possible to find f, such that chaotic iterations generated by f can be described by a chaotic and entropic map on a topological space in the sense of Devaney. We have considered a finite set of states $\mathbb{B}^{\mathsf{N}}$ and a set $\mathbb{S}$ of strategies composed by an infinite number of infinite sequences. In the following section we will discuss the impact of these assumptions in the context of the finite set of machine numbers.

1.5 CHAOS IN A FINITE STATE MACHINE

Let us now explain how it is possible to have a true chaos in a finite state machine.

1.5.1 A program with a chaotic behavior

In the Section 1.3 we have proven that discrete chaotic iterations can be put in the field of discrete dynamical systems:

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

where $(\mathcal{X}, d)$ is a metric space and G_f is a continuous function. Thus, it becomes possible to study the topological behavior of those chaotic iterations. Precisely, it has been proven that if the iterate function is based on the vectorial logical negation f_0, then chaotic iterations generate chaos according to Devaney. Therefore chaotic iterations, as Devaney's topological chaos, satisfy: sensitive dependence on the initial conditions, unpredictability, indecomposability, and uniform repartition. Additionally, G_{f_0} has been proven to be expansive and topologically mixing, and its topological entropy has been computed. Our intention is now to use these chaotic iterations that are highly unpredictable, to build programs in the computer science security field. Furthermore, we will give in Section 1.7 a link between CIs and artificial neural networks, thus it is possible to make them behave chaotically.

Up to now, most of computer programs presented as chaotic lose their chaotic properties while computing in the finite set of machine numbers. The algorithms that have been presented as chaotic usually act as follows. After having received its initial state, the machine works alone with no interaction with the outside world. Its outputs only depend on the different states of the machine. The main problem which prevents speaking about chaos in this particular situation is that when a finite state machine reaches a same internal state twice, the two future evolutions are identical. Such a machine always finishes by entering into a cycle while iterating. This highly predictable behavior cannot be set as chaotic, at least as expressed by Devaney. Some attempts to define a discrete notion of chaos have been proposed, but they are not completely satisfactory and are less recognized than the notions exposed in this paper.

The stated problem can be solved in the following way. The computer must generate an output *O* computed from its current state *E and* the current value of an input *S*, which changes at each iteration (Fig. 1.1). Therefore, it is possible that the machine presents the same state twice, but with two future evolutions completely different, depending on the values of the input. By doing so, we thus obtain a machine with a finite number of states, which can evolve in infinitely different ways,

due to the new values provided by the input at each iteration. Thus such a machine can behave chaotically, as defined in the Devaney's formulation.

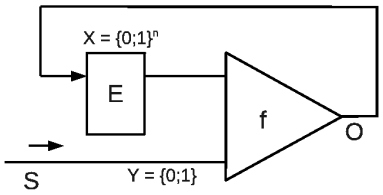


Fig. 1.1 A chaotic finite-state machine. At each iteration, a new value is taken from the outside world (S). It is used by f as input together with the current state (E).

1.5.2 The practical case of finite strategies

It is worthwhile to notice that even if the set of machine numbers is finite, we deal in practice with the *infinite* set of strategies that have *finite but unbounded lengths*. Indeed, it is not necessary to store all of the terms of these strategies in the memory. Only its n^{th} term (an integer less than or equal to N) has to be stored at the n^{th} step, as it is illustrated in the following example. Let us suppose that a given text is input from the outside world into the computer character by character and that the current term of the strategy is computed from the ASCII code of the current stored character. Since the set of all possible texts of the outside world is infinite and the number of their characters is unbounded, we work with an infinite set of finite but unbounded strategies.

In the computer science framework, we also have to deal with a finite set of states of the form $\mathbb{B}^{\mathsf{N}}$ and as stated before an infinite set $\mathbb{S}$ of strategies. The sole difference with the theoretical study is that instead of being infinite the sequences of S are finite with unbounded length, as any reasonable program must obviously finish one day.

The proofs of continuity and transitivity stated previously are independent of the finiteness of the length of strategies (sequences of $\mathbb{S}$). Sensitivity can be proven too in this situation (see Section 1.4.2.2). So even in the case of finite machine numbers, we have the two fundamental properties of chaos: sensitivity and transitivity, which respectively implies unpredictability and indecomposability (see [13], p.50). The regularity supposes that the sequences are of infinite lengths. To obtain the analogous of regularity in the context of finite sets, we can for example define a notion of *periodic but finite* sequences.

Definition 10. A strategy $S \in \mathbb{S}$ is said to be *periodic but finite* if S is a finite sequence of length n and if there exists a divisor p of n, $p \neq n$, such that $\forall i \leqslant n-p, S^i = S^{i+p}$. A point $(E,S) \in \mathscr{X}$ is said to be *periodic but finite*, if its strategy S is periodic but finite.

In this situation, $(1,2,1,2,1,2,1,2)$ (p=2) and $(2,2,2)$ (p=1), are periodic but finite. This definition can be interpreted as the analogous of periodicity definition on finite strategies. Following the proof of regularity (Section 1.3.2.1), it can be proven that the set of periodic but finite points is dense on $\mathscr{X}$, hence obtaining a desired element of regularity in finite sets, as quoted by Devaney ([13], p.50): "two points arbitrary close to each other could have completely different behaviors, the one could have a cyclic behavior as long as the system iterates while the trajectory of the second could 'visit' the whole phase space". It should be recalled that the regularity was introduced by Devaney in order to counteract the effects of transitivity: two points close to each other can have fundamentally different behaviors.

In the following we explain how to use chaotic iterations in the computer science security field, by using an illustrative example. In this sense, we show two different ways to compute chaotic hash functions, the second one using neural networks.

1.6 HASH FUNCTIONS WITH TOPOLOGICAL CHAOS PROPERTIES

In this section, a concrete example of a chaotic program is given in the computer science security field.

1.6.1 Introduction

The use of chaotic maps to generate hash algorithms has seen several developments in recent years. In [14] for example, a digital signature algorithm based on an elliptic curve and chaotic mapping is proposed to strengthen the security of an elliptic curve digital signature algorithm. Other examples of the generation of a hash function using chaotic maps can be found in, *e.g.*, [24, 26, 19]. Neural networks that have learned a continuous chaotic map have been proposed too in recent years [17], to achieve hash functions requirements.

Note that using any chaotic map does not guarantee that the resulting hash function would behave chaotically too. To the best of our knowledge, this point is not discussed in these referenced papers, however it should be considered as important. We define in this section a new way to construct hash functions based on chaotic iterations. As a consequence of the theory presented before, the generated hash functions satisfy various topological chaos properties. Thus, properties required for hash functions are guaranteed by our approach. For example, the avalanche criterion is deduced from the expansivity property.

1.6.2 A chaotic hash function

In this section, we explain a new way to obtain a digest of a digital medium described by a binary sequence. It is based on chaotic iterations and satisfies various topological chaos properties. The hash value will be the last state of some chaotic iterations: the initial state X_0, finite strategy S, and iterate function must then be defined.

The initial condition $X_0 = (S,E)$ is composed by a $\mathsf{N} = 256$ bits sequence E and a chaotic strategy S. In the following section, we describe in detail how to obtain this initial condition from the original medium.

1.6.2.1 How to obtain E

The first step of our algorithm is to transform the message in a normalized 256 bits sequence E. To illustrate this step inspired by SHA-1, we state that our original text is: "*The original text*". Each character of this string is replaced by its ASCII code (on 7 bits). Then, we add a 1 at the end of this string.

```
10101001 10100011 00101010 00001101 11111100 10110100
11100111 11010011 10111011 00001110 11000100 00011101
00110010 11111000 11101001
```

So, the binary value (1111000) of the length of this string (120) is added, with another 1:

```
10101001 10100011 00101010 00001101 11111100 10110100
11100111 11010011 10111011 00001110 11000100 00011101
00110010 11111000 11101001 11110001
```

This string is inverted (the last bit is now the first one) and the two new substrings are concatenated. This gives:

```
10101001 10100011 00101010 00001101 11111100 10110100
11100111 11010011 10111011 00001110 11000100 00011101
00110010 11111000 11101001 11110001 00011111 00101110
00111110 10011001 01110000 01000110 11100001 10111011
10010111 11001110 01011010 01111111 01100000 10101001
10001011 0010101
```

So, we obtain a multiple of 512, by duplicating this string enough and truncating at the next multiple of 512. This string in which the whole original text is contained, is denoted by D.

Finally, we split the new string into blocks of 256 bits and apply the exclusive-or function, obtaining a 256 bits sequence in a manner inspired by the SHA-X algorithms.

```
11111010 11100101 01111110 00010110 00000101 11011101
00101000 01110100 11001101 00010011 01001100 00100111
01010111 00001001 00111010 00010011 00100001 01110010
01000011 10101011 10010000 11001011 00100010 11001100
10111000 01010010 11101110 10000001 10100001 11111010
10011101 01111101
```

In the context of Subsection 1.6.2, $\mathsf{N} = 256$, and E is the above obtained sequence of 256 bits: the given message has been compressed into a 256 binary string.

We now have the definitive length of our digest. Note that a lot of texts have the same normalized string. This is not a problem because the strategy we will build depends on the whole text too, in such a way that two different texts lead to two different strategies. Let us now build the strategy S.

1.6.2.2 How to choose S

To obtain the strategy S, an intermediate sequence (u^n) is constructed from D as follows:

- D is split into blocks of 8 bits. Then u^n is the decimal value of the n^{th} block.
- A circular rotation of one bit to the left is applied to D (the first bit of D is put on the end of D). Then the new string is split into blocks of 8 bits another time. The decimal values of those blocks are added to (u^n).
- This operation is repeated again 6 times.

It is now possible to build the strategy S:

$$S^0 = u^0, \quad S^n = (u^n + 2 \times S^{n-1} + n) \ (mod\ 256).$$

S will be highly dependent to the changes of the original text, because $\theta \longmapsto 2\theta \ (mod\ 1)$ is known to be chaotic as defined by Devaney's theory [13].

1.6.2.3 How to construct the digest

To construct the digest, chaotic iterations are done with initial state X^0,

$$\begin{array}{rccc} f: & [\![1,256]\!] & \longrightarrow & [\![1,256]\!] \\ & (E_1,\ldots,E_{256}) & \longmapsto & (\overline{E_1},\ldots,\overline{E_{256}}), \end{array}$$

as iterate function, and S for the chaotic strategy.

The result of those iterations is a 256 bits vector. Its components are taken 4 bits at a time and translated into hexadecimal numbers, to obtain the hash value:

```
63A88CB6AF0B18E3BE828F9BDA4596A6A13DFE38440AB9557DA1C0C6B1EDBDBD
```

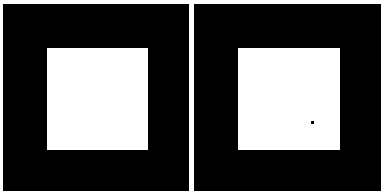
To compare, if instead of using the text "*The original text*" we took "*the original text*", the hash function returns:

```
33E0DFB5BB1D88C924D2AF80B14FF5A7B1A3DEF9D0E831194BD814C8A3B948B3
```

In this paper, the generation of hash value is done with the vectorial Boolean negation f_0 defined in eq. (1.2). Nevertheless, the procedure remains general and can be applied with any function f such that G_f is chaotic. In the following subsection, a complete example of the procedure is given.

1.6.3 Application example

Consider two black and white images of size 64×64 in Fig. 1.2, in which the pixel in position (40,40) has been changed. In this case, the hash function returns:



(a) Original image. (b) Modified image.

Fig. 1.2 Hash of some black and white images.

```
34A5C1B3DFFCC8902F7B248C3ABEFE2C9C9538E5104D117B399C999F74CF1CAD
```

for the Fig. 2(a) and

```
5E67725CAA6B7B7434BE57F5F30F2D3D57056FA960B69052453CBC62D9267896
```

for the Fig. 2(b).

Consider two 256 graylevel images of Lena (256×256 pixels) in figure 1.3, in which the grayscale level of the pixel in position (50,50) has been transformed from 93 (fig. 3(a)) to 94 (fig. 3(b)). In this case, the hash function returns:

```
FA9F51EFA97808CE6BFF5F9F662DCD738C25101FE9F7F427CD4E2B8D40331B89
```

for the left Lena and

(a) Original lena. (b) Modified lena.

Fig. 1.3 Hash of some grayscale level images.

```
BABF2CE1455CA28F7BA20F52DFBD24B76042DC572FCCA4351D264ACF4C2E108B
```

for the right Lena.

These examples give an illustration of the avalanche effect obtained by this algorithm. A more complete study of the properties possessed by the hash functions and resistance under collisions will be studied in a future work.

1.7 A CHAOTIC NEURAL NETWORK AS HASH FUNCTION

A hash function can be achieved in two stages: the compression of the message (mapping a binary sequence of any length $n \in \mathbb{N}$ into a message of a fixed length belonging into $\mathbb{B}^{\mathsf{N}}$, for a given fixed length $\mathsf{N} \in \mathbb{N}$) and the hash of the compressed message [17]. As several compression functions have yet been proposed to achieve the first stage, we will only focus on the second stage and we will explain how to build a neural network that realize it. This neural network that hashes compressed messages will behave chaotically, as it is defined by the Devaney's theory.

Let us firstly explain how it is possible to build a neural network that behaves chaotically. Consider $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{N}^{\mathsf{N}}$ and a MLP which recognize F_f. That means, for all $(k,x) \in [\![1;\mathsf{N}]\!] \times \mathbb{B}^{\mathsf{N}}$, the response of the output layer to the input (k,x) is $F_f(k,x)$. We thus connect the output layer to the input one as it is depicted in Figure 1.4, leading to a global recurrent artificial neural network (ANN) working as follows [6]:

- At the initialization stage, the ANN receives a Boolean vector $x^0 \in \mathbb{B}^{\mathsf{N}}$ as input state, and $S^0 \in [\![1;\mathsf{N}]\!]$ in its input integer channel $i()$. Thus, $x^1 = F_f(S^0,x^0) \in \mathbb{B}^{\mathsf{N}}$ is computed by the neural network.
- This state x^1 is published as an output. Additionally, x^1 is sent back to the input layer, to act as Boolean state in the next iteration.

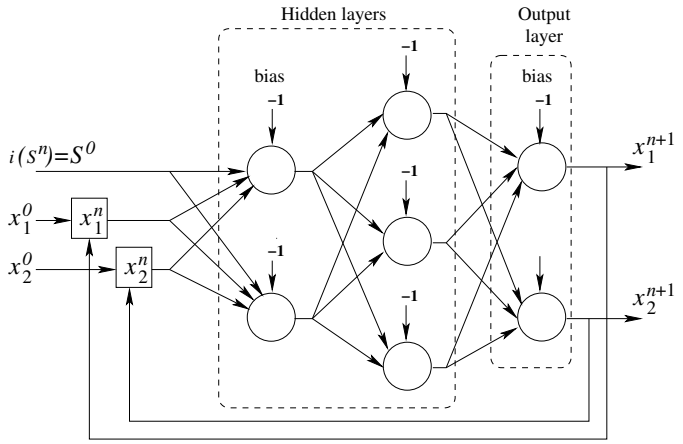


Fig. 1.4 Example of global recurrent neural network modeling function F_f such that $x^{n+1} = \left(x_1^{n+1}, x_2^{n+1}\right) = F_f\left(i(S^n), (x_1^n, x_2^n)\right)$

- At iteration number n, the recurrent neural network receives the state $x^n \in \mathbb{B}^{\mathsf{N}}$ from its output layer and $i\left(S^n\right) \in [\![1;\mathsf{N}]\!]$ from its input integer channel $i()$. It can thus calculate $x^{n+1} = F_f\left(i\left(S^n\right), x^n\right) \in \mathbb{B}^{\mathsf{N}}$, which will be the new output of the network.

Obviously, this particular MLP produce exactly the same values than CIs with update function f. That is, such MLPs are equivalent to CIs with f as update function. However, the compression stage of the hash function presented in the previous section can be resumed to making chaotic iterations over the compressed message. As chaotic iterations can be obtained with a neural network, we can thus realize this stage with a (chaotic) neural network. Finally, it is important to remark that the proposed hash function can be implemented into a global neural network, as various compression neural networks can be found in the literature [18, 21, 12]: we just have to replace our compression stage, inspired by SHA-X, with one of these compression ANN.

1.8 CONCLUSION

In this paper, a new approach to compute programs with a chaotic behavior is proposed. This approach is based on the well-known Devaney's topological chaos. The algorithms which are of iterative nature are based on the so-called chaotic iterations. This is achieved by establishing a link between the notions of topological chaos and chaotic iterations. Indeed, we are not interested in stable states of such iterations as it has always been the case in the literature, but in their unpredictable behavior. After a solid theoretical study, we consider the practical implementation of the proposed

algorithms by evaluating the case of finite sets. We study the behavior of the induced computer programs proving that it is possible to design true chaotic computer programs.

An application is proposed in the area of information security: a new hash function is presented, the security in this case is guaranteed by the unpredictability of the behavior of the proposed algorithms. The algorithms derived from our approach satisfy important properties of topological chaos such as sensitivity to initial conditions, uniform repartition (as a result of the transitivity), unpredictability, expansivity, and topological mixing. Moreover, its topological entropy is infinite. The results expected in our study have been experimentally checked. The choices made in this first study are simple: compression function inspired by SHA-1, negation function for the iteration function, *etc.* The aim was not to find the best hash function, but to give simple illustrated examples to prove the feasibility in using the new kind of chaotic algorithms in computer science. Finally, we have shown how the mathematical framework of topological chaos offers interesting qualitative and qualitative tools to study the algorithms based on our approach.

In future work, we will investigate other choices of iteration functions and chaotic strategies. We will try to characterize transitive functions. Other properties induced by topological chaos will be explored and their interest in the information security framework will be deepened.

References

1. R. L. Adler, A. G. Konheim, and M. H. McAndrew. Topological entropy. *Trans. Amer. Math. Soc.*, 114:309–319, 1965.
2. Jacques Bahi and Christophe Guyeux. A new chaos-based watermarking algorithm. In *SECRYPT'10, Int. conf. on security and cryptography*, pages 455–458, Athens, Greece, July 2010. SciTePress.
3. Jacques Bahi, Christophe Guyeux, and Qianxue Wang. A novel pseudo-random generator based on discrete chaotic iterations. In *INTERNET'09, 1-st Int. Conf. on Evolving Internet*, pages 71–76, Cannes, France, August 2009.
4. Jacques M. Bahi and Christophe Guyeux. Hash functions using chaotic iterations. *Journal of Algorithms & Computational Technology*, 4(2):167–181, 2010.
5. 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 1–7, Barcelona, Spain, July 2010. Best paper award.
6. Jacques M. Bahi, Christophe Guyeux, and Michel Salomon. Building a chaotic proved neural network. *CoRR*, abs/1101.4351, 2011.
7. J. Banks, J. Brooks, G. Cairns, and P. Stacey. On devaney's definition of chaos. *Amer. Math. Monthly*, 99:332–334, 1992.
8. R. Bowen. Entropy for group endomorphisms and homogeneous spaces. *Trans. Amer. Math. Soc.*, 153:401–414, 1971.
9. D. Chazan and W. Miranker. Chaotic relaxation. *Linear algebra and its applications*, pages 199–222, 1969.
10. 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.
11. 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.
12. Christopher Cramer, Erol Gelenbe, and Hakan Bakircioglu. Video compression with random neural networks. *Neural Networks for Identification, Control, and Robotics, International Workshop*, 0:0476, 1996.
13. Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, Redwood City, CA, 2nd edition, 1989.
14. 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.
15. Christophe Guyeux. *Le désordre des itérations chaotiques et leur utilité en sécurité informatique*. PhD thesis, Université de Franche-Comté, 2010.
16. Christophe Guyeux, Nicolas Friot, and Jacques 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 208–211, Darmstadt, Germany, October 2010.
17. Yantao Li, Shaojiang Deng, and Di Xiao. A novel hash algorithm construction based on chaotic neural network. *Neural Computing and Applications*, pages 1–9, 2010.
18. Matthew V. Mahoney. Fast text compression with neural networks. In *Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference*, pages 230–234. AAAI Press, 2000.
19. 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.
20. F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. 1986.
21. O. Rudenko and M. Snytkin. Image compression based on the neural network art. *Cybernetics and Systems Analysis*, 44:797–802, 2008.
22. Chang song Zhou and Tian lun Chen. Extracting information masked by chaos and contaminated with noise: Some considerations on the security of communication approaches using chaos. *Physics Letters A*, 234(6):429 – 435, 1997.
23. Qianxue Wang, Jacques Bahi, Christophe Guyeux, and Xiaole Fang. Randomness quality of CI chaotic generators. application to internet security. In *INTERNET'2010. The 2nd Int. Conf. on Evolving Internet*, pages 125–130, Valencia, Spain, September 2010. IEEE Computer Society Press. Best Paper award.
24. 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.
25. Xianyong Wu and Zhi-Hong Guan. A novel digital watermark algorithm based on chaotic maps. *Physics Letters A*, 365(5-6):403 – 406, 2007.
26. 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.