Topological chaos and chaotic iterations Application to Hash functions

Christophe Guyeux
Jacques M. Bahi, Senior Member IEEE

***Abstract*— This paper introduces a new notion of chaotic algorithms. These algorithms are iterative and are based on so-called chaotic iterations. Contrary to all existing studies on chaotic iterations, we are not interested in stable states of such iterations but in their possible unpredictable behaviors. By establishing a link between chaotic iterations and the notion of Devaney's topological chaos, we give conditions ensuring that these kind of algorithms produce topological chaos. This leads to algorithms that are highly unpredictable. After presenting the theoretical foundations of our approach, we are interested in its practical aspects. We show how the theoretical algorithms give rise to computer programs that produce true topological chaos, then we propose applications in the area of information security.**

I. INTRODUCTION

The use of chaos in various fields of information security such as data hiding, hash functions, or pseudo-random number generators is almost always based on the conception of algorithms that include known chaotic maps such as the logistic map. The goal is to obtain an algorithm which preserves the chaotic properties of the included chaotic functions. For example, in [13] and [12], a watermark W is encrypted in W_e by using the bitwise exclusive or: $W_e = W \otimes X$, where X is a logistic map. Then, pixels of the carrier image designed to embed these bits are selected with the 2-D Arnold's cat map. A similar use of chaotic maps for watermarking can be found in e.g. [15], [10], [7], [3] and [4]. In the domain of hash functions, the use of chaotic maps is seen in e.g. [6], [11], [14] and [8]. However, without rigorous proof it is not indisputable that an algorithm that includes chaotic functions preserves chaos properties: for example, using the logistic function with other "obvious" parameters does not guarantee that the algorithm is chaotic. Moreover, even if the algorithm obtained by the inclusion of chaotic maps is itself chaotic, the implementation of this algorithm on a machine can cause it to lose its chaotic nature. This is due to the finite nature of the machine numbers set. These issues are discussed in this document.

In this paper we don't simply integrate chaotic maps hoping that the security algorithm remains chaotic, we conceive algorithms for computer security that we mathematically prove chaotic. We raise the question of their implementation, proving in doing so that it is possible to design a chaotic algorithm and a chaotic computer program.

Christophe Guyeux and Jacques M. Bahi are with the Computer Science Laboratory LIFC, University of Franche-Comte, 16, route de Gray - 25030 Besançon, France (phone: +33 381 666948; email: {christophe.guyeux, jacques.bahi}@univ-fcomte.fr).

The chaos theory we consider is Devaney's topological chaos. In addition to being recognized as one of the best mathematical definition of chaos, this theory offers a framework with qualitative and quantitative tools to evaluate the notion of unpredictability. As an application of our fundamental results, we are interested in the area of information security. We propose in this paper a new approach of security which is based on unpredictability as it is defined by Devaney's chaos.

The paper begins by introducing the theoretical foundation of the new approach. We recall the definition of Devaney's topological chaos as well as 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. Because chaotic iterations are very suited for computer programming, this link allows us to generate Devaney's chaos topological in the computer science field. After having studied the theoretical aspects of our approach we focus on practical aspects. The important question is how to preserve the topological chaos properties in a set of a finite number of states. This question is answered by introducing a concept we call *secure chaotic information machine*. This is a Mealy machine generating chaos as defined by Devaney (Section IV). We also give some applications of our approach of chaos, in the domain of information security. Algorithms intended for information security and based on this new approach are explained in Section V, in the hash function domain

The rest of this paper is organized as follows. In Section II, 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 III. In Section IV, the question on how to apply the theoretical result is raised and applications in the computer science field are given in Section V. The paper ends with a conclusion in which our contribution is summarized and the planned future work is discussed.

II. BASIC RECALLS

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

A. Devaney's chaotic dynamical systems

Consider a metric space $(\mathcal{X}, d)$ and a continuous function $f : \mathcal{X} \longrightarrow \mathcal{X}$.

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

Definition An element (a point) x is a *periodic element* (point) for f of period $n \in \mathbb{N}^*$, if $f^n(x) = x$. The set of periodic points of f is denoted $Per(f)$.

Definition $(\mathcal{X}, f)$ is said to be *regular* if the set of periodic points is dense in $\mathcal{X}$,

$$\forall x \in \mathcal{X}, \forall \varepsilon > 0, \exists p \in Per(f), d(x, p) \leqslant \varepsilon.$$

Definition 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 exists $y \in V$ and $n \geqslant 0$ such that $|f^n(x) - f^n(y)| > \delta$. δ is called the *constant of sensitivity* of f.

Let us now recall the definition of a chaotic topological system, in the sense of Devaney [5]:

Definition $f : \mathcal{X} \longrightarrow \mathcal{X}$ is said to be *chaotic* on $\mathcal{X}$ if,

1) f has sensitive dependence on initial conditions,
2) f is topologically transitive,
3) $(\mathcal{X}, f)$ is regular.

Therefore, quoting Robert Devaney: "A chaotic map possesses three ingredients: unpredictability, indecomposability and an element of regularity. A chaotic system is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or decomposed into two subsystems, because of topological transitivity. And, in the midst of this random behavior, we nevertheless have an element of regularity, namely the periodic points which are dense." Fundamentally different behaviors are thus possible and occur in an unpredictable way.

B. Chaotic iterations

In the sequel S^n denotes the n^{th} term of a sequence S, V_i denotes the i^{th} component of a vector V and $f^k = f \circ ... \circ f$ denotes the k^{th} composition of a function f. Finally, the following notation is used: $[\![1; N]\!] = \{1, 2, \ldots, N\}$. Let us consider a *system* of a finite number N of elements (or *cells*), so that each cell has a boolean *state*. Then a sequence of length N of boolean states of the cells corresponds to 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 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. Then, the so-called *chaotic iterations* are defined by $x^0 \in \mathbb{B}^{\mathsf{N}}$ and $\forall n \in \mathbb{N}^*$,

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

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 $f(x^{n-1})_{S^n}$ can be replaced by $f(x^k)_{S^n}$ (where $k \leqslant n - 1$), describing for example delays transmission (see e.g. [1]). For the general definition of such chaotic iterations, see e.g. [9].

III. CHAOTIC ITERATIONS AS DEVANEY'S CHAOS

A. The new topological space

In this section we will put our study in a topological context by defining a suitable metric space where chaotic iterations are continuous.

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:

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

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

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

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

$$G_f(S, E) = \left(\sigma(S), F_f(i(S), E)\right), \quad (2)$$

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 (II-B) can be described by the following iterations:

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

With this formulation, a shift function appears as a component of chaotic iterations. The shift function is a famous example of a chaotic map [5] but its presence is not sufficient enough to claim G_f as chaotic. In the rest of this section we prove rigorously that under some hypotheses, chaotic iterations generate topological chaos. Furthermore, due to the suitability of chaotic iterations for computer programming we also prove that this is true in the computer science field.

2) Cardinality of $\mathcal{X}$: By comparing $\mathbb{S}$ and $\mathbb{R}$, we have the result.

theorem 1 *The phase space $\mathcal{X}$ 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, or else it is equal to 1. With this construction, $\varphi : [\![1; \mathsf{N}]\!]^{\mathbb{N}} \longrightarrow [0, 1]$ is surjective. 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 $\mathcal{X} = [\![1; \mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$ is greater or equal to the cardinality of $\mathbb{R}$. ■

remark 1 *This result is independent from the number of cells of the system.*

3) A new distance: We define a new distance between two points $X = (S, E), Y = (\check{S}, \check{E}) \in \mathcal{X}$ by

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

where

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

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.

4) Continuity of the iteration function: To prove that chaotic iterations are an example of topological chaos in the sense of Devaney [5], G_f must be continuous in the metric space $(\mathcal{X}, d)$.

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 $\mathcal{X}$, which converges to (S, E). We will prove that $\left(G_f(S^n, E^n)\right)_{n\in\mathbb{N}}$ converges to $\left(G_f(S, E)\right)$. 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:

$$\exists n_0 \in \mathbb{N}, n \geqslant n_0 \Longrightarrow E^n = E.$$

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:

$$\forall n \geqslant n_1, S_0^n = S_0.$$

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 $\left(G_f(S^n, E^n)\right)$ and $\left(G_f(S, E)\right)$ is convergent to 0. Let $\varepsilon > 0$.

- If $\varepsilon \geqslant 1$, we see that distance between $\left(G_f(S^n, E^n)\right)$ and $\left(G_f(S, E)\right)$ is strictly less than 1 after the $max(n_0, n_1)^{th}$ term (same state).
- If $\varepsilon < 1$, then $\exists k \in \mathbb{N}, 10^{-k} \geqslant \varepsilon \geqslant 10^{-(k+1)}$. 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 chaotic iterations are a case of topological chaos, according to Devaney.

B. 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

$$f_0(x_1, \ldots, x_{\mathsf{N}}) = (\overline{x_1}, \ldots, \overline{x_{\mathsf{N}}}) \tag{3}$$

is a suitable function.

1) Regularity:

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

Proof: Let $(\check{S}, \check{E}) \in \mathcal{X}$ 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

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

Then, $\forall S \in \mathcal{S}_{\check{S}, k_0(\varepsilon)}, d((S, \check{E}); (\check{S}, \check{E})) < \varepsilon$. It remains to choose $\widetilde{S} \in \mathcal{S}_{\check{S}, k_0(\varepsilon)}$ such that $(\widetilde{S}, \widetilde{E}) = (\widetilde{S}, \check{E})$ is a periodic point for G_{f_0}. Let $\mathcal{J} = \left\{i \in \{1, ..., \mathsf{N}\} / E_i \neq \check{E}_i, \text{ where } (S, E) = G_{f_0}^{k_0}(\check{S}, \check{E})\right\}$, $i_0 = card(\mathcal{J})$ and $j_1 < j_2 < ... < j_{i_0}$ the elements of $\mathcal{J}$. Then, $\widetilde{S} \in \mathcal{S}_{\check{S}, k_0(\varepsilon)}$ 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 $\varepsilon-$closed to $(\check{S}, \check{E})$.

As a conclusion, $(\mathcal{X}, G_{f_0})$ is regular. ■

2) Transitivity:

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

Proof: Let us define $\mathcal{E} : \mathcal{X} \to \mathbb{B}^{\mathbb{N}}$, such that $\mathcal{E}(S, E) = E$. Let $\mathcal{B}_A = \mathcal{B}(X_A, r_A)$ and $\mathcal{B}_B = \mathcal{B}(X_B, r_B)$ be two open balls of $\mathcal{X}$, 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 $\mathcal{B}_A$ such that $\exists n_0 \in \mathbb{N}, G^{n_0}_{f_0}(\widetilde{X}) \in \mathcal{B}_B$.
$\widetilde{X}$ must be in $\mathcal{B}_A$ 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 $S^k = S^k_A, \forall k \leqslant k_0$, then $(S, \widetilde{E}) \in \mathcal{B}_A$. Let us notice $(\check{S}, \check{E}) = G^{k_0}_{f_0}(S_A, E_A)$ and $c_1, ..., c_{k_1}$ the elements of the set $\{i \in [\![1, \mathsf{N}]\!] / \check{E}_i \neq \mathcal{E}(X_B)_i\}$. So any point X of the set

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

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

1) $\widetilde{X} = E_A$,
2) $\forall k \leqslant k_0, \widetilde{S}^k = S^k_A$,
3) $\forall k \in [\![1, k_1]\!]$, $\widetilde{S}^{k_0+k} = c_k$,
4) $\forall k \in \mathbb{N}^*, \widetilde{S}^{k_0+k_1+k} = S^k_B$,

is such that $\widetilde{X} \in \mathcal{B}_A$ and $G^{k_0+k_1}_{f_0}(\widetilde{X}) \in \mathcal{B}_B$. ■

3) Sensitive dependence on initial conditions:

theorem 5 $(\mathcal{X}, G_{f_0})$ *has sensitive dependence on initial conditions.*

Proof: Banks *et al.* proved in [2] that having sensitive dependence is a consequence of being regular and topologically transitive. ■

4) Devaney's Chaos: In conclusion, $(\mathcal{X}, G_{f_0})$ is topologically transitive, regular and has sensitive dependence on initial conditions. Then we have the following result:

theorem 6 G_{f_0} *is a chaotic map on* $(\mathcal{X}, d)$ *in the sense of Devaney.*

remark 2 *We have proven that the set C of the iterate functions f so that* $(\mathcal{X}, G_f)$ *is chaotic (according to the definition of Devaney), is a nonempty set. In future work, we will deepen the study of C, among other things, by computing its cardinality and characterizing this set.*

IV. CHAOS IN A FINITE STATE MACHINE

A. The approach presented in this paper

In the section above, it has been proven that discrete chaotic iterations can be put in the field of discrete dynamical systems:

$$\begin{cases} x^0 \in \mathcal{X} \\ x^{n+1} = G_f(x^n), \end{cases} \tag{4}$$

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.

Two major problems typically occur when trying to develop a computer program with chaotic behavior. First, computers have a finite number of states, so the computations always enter into cycles. Second, the properties of chaotic algorithms are inherited from a real chaotic sequence (like a logistic map) and this behavior is lost when computing floating-point numbers (unlike real numbers, floating-point numbers have a finite decimal part). These two problems are solved in this paper due to the two following ideas:

1) Chaotic iterations are Mealy machines. At each iteration, data corresponding to the current strategy II-B are taken from the outside world, then computations are realized into the memory (the updates of the finite state of the system). The last state is returned after a desired number of iterations. Contrary to the existing points of view, based on a Moore machine, this machine can pass two times in a same state, without continuing the same evolution. Section IV-B explains in detail this original contribution, which allows the realization of true chaos in computers.
2) As mentioned above, the strategy S defined in II-B will not depend on real numbers, but on integers taken from the outside world. We work with the set $\mathcal{X}$ defined in Subsection III-A.1 which has the cardinality of the continuum. Section IV-C discusses the consequences of dealing with finite strategies in practice.

B. A chaotic Mealy machine

The algorithms considered chaotic usually follow the principle of a Moore machine. After having received its initial states, 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 is that when a machine with a finite number of states reaches a same state twice, the two following evolutions are identical. Such an algorithm always enters into a cycle. This behavior is highly predictable and cannot be set as chaotic. Attempts to define a discrete notion of chaos have been proposed, but they are not completely satisfactory and are less recognized than the notion of Devaney's topological chaos. This problem does not occur in a Mealy machine. This finite state transducer generates an output O computed from its current state E *and* the current value of an input S (Fig. 1). By this accord, even if the machine reaches the same state twice, the corresponding following evolutions may be completely different depending on the values of the inputs. The method presented here is based on such a machine. Indeed, chaotic iterations are a Mealy machine: at each iteration, the computations take into account new inputs (strategies) which are obtained, for example, from the media on which our algorithm applies. Roughly speaking, as the set of all media is infinite, we obtain a finite state machine which can evolve in infinite

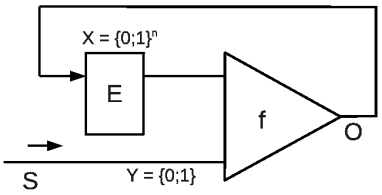


Fig. 1. Mealy machine for chaotic algorithms.

ways, thus making it possible to obtain a true chaos in computers.

Definition A Mealy machine is said to be chaotic if this machine has a chaotic behavior, as expressed by Devaney.

The Mealy machine we used in this document will be the chaotic iterations with G_{f_0} as iterate function. Because these chaotic iterations satisfy the Devaney's definition of chaos, as stated in Section III-B.4, we can conclude that our Mealy machine is a chaotic machine.

C. 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, as suggested before, it is not necessary to store all the terms of the strategy 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. Of course, the previous example is a simplistic one. A chaotic procedure should to be introduced to generate the terms of the strategy from the stream of characters.

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.

The proofs of continuity and transitivity are independent of the finiteness of the length of strategies (sequences of $\mathbb{S}$). Sensitivity can be proved too in this situation. 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 [5], 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 define below the notion of *periodic but finite* sequences.

Definition 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 $(E,S) \in \mathcal{X}$ is said to be *periodic but finite*, if its strategy S is periodic but finite.

For example, $(1,2,1,2,1,2,1,2)$ (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 III-B.1), it can be proven that the set of periodic but finite points is dense on $\mathcal{X}$, hence obtaining a desired element of regularity in finite sets, as quoted by Devaney ([5], 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 sensitivity and transitivity: two points close to each other can have fundamentally different behaviors.

V. HASH FUNCTIONS BASED ON TOPOLOGICAL CHAOS

A. Introduction

The use of chaotic maps to generate hash algorithms has seen several developments in recent years. In [6] 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.* [11], [14] and [8]. However, as for digital watermarking, the use of any chaotic map does not guarantee that the resulting hash function would behave chaotically too. To 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 the topological chaos property. Thus, various desired properties in this domain are guaranted by our approach. For example, the avalanche criterion is closely linked to the sensitivity property.

B. A chaotic machine for hash functions

In this section, we explain a new way to obtain a hash value of a digital medium described by a binary sequence. It is based on chaotic iterations and satisfies the topological chaos property. 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 sequence, we describe in detail how to obtain this initial condition from the original medium.

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 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 to 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
```

The whole string is copied, but in the opposite direction. 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 our obtained string into blocks of 256 bits and apply the exclusive-or function, obtaining a 256 bits sequence.

```
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
```

So, in the context of Subsection V-B, N = 256 and E is the above obtained sequence of 256 bits.

We now have the definitive length of our digest. Note that a lot of texts have the same string. This is not a problem because the strategy we will build depends on the whole text. Let us now build the strategy S.

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 [5].

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 per 4 bits and translated into hexadecimal numbers, to obtain the hash value:

```
63A88CB6AF0B18E3BE828F9BDA4596A6
A13DFE38440AB9557DA1C0C6B1EDBDBD
```

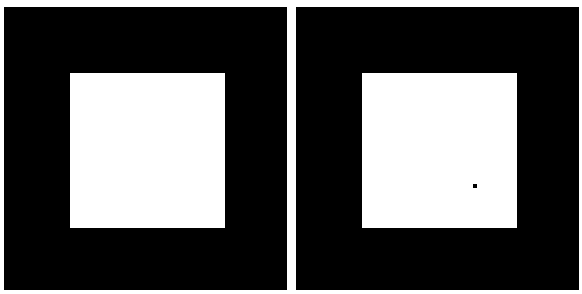
To compare, if instead of using the text "*The original text*" we took "*the original text*", the hash function returns:

```
33E0DFB5BB1D88C924D2AF80B14FF5A7
B1A3DEF9D0E831194BD814C8A3B948B3
```

In this paper, the generation of hash value is done with the vectorial boolean negation f_0 defined in eq. (3). 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.

C. Application example

Let us consider the two black and white images of size 64×64 in Fig. 2, in which the pixel in position (40,40) has been changed. In this case, our hash function returns:



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

Fig. 2. Hash of some black and white images.

```
34A5C1B3DFFCC8902F7B248C3ABEFE2C
9C9538E5104D117B399C999F74CF1CAD
```

for the Fig. 2(a) and

```
5E67725CAA6B7B7434BE57F5F30F2D3D
57056FA960B69052453CBC62D9267896
```

for the Fig. 2(b).

Let us consider now the two 256 graylevel images of Lena (256×256 pixels) in figure 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, our hash function

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

Fig. 3. Hash of some grayscale level images.

returns:

```
FA9F51EFA97808CE6BFF5F9F662DCD73
8C25101FE9F7F427CD4E2B8D40331B89
```

for the left Lena and

```
BABF2CE1455CA28F7BA20F52DFBD24B7
6042DC572FCCA4351D264ACF4C2E108B
```

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 our hash functions and resistance under collisions will be studied in future work.

VI. CONCLUSION

In this paper, a new approach to generate algorithms with chaotic behaviors 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. This is the first time that such an approach is considered for 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. A simple application is proposed in the area of hash functions. The security in this case is defined by the unpredictability of the behavior of the proposed algorithm. The algorithm derived from our approach satisfies important properties of topological chaos such as sensitivity to initial conditions, uniform repartition (as a result of the transitivity), and unpredictability. The results expected in our study have been experimentally checked. The choices made in this first study are simple: 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. 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, such as entropy, will be explored and their interest in the information security framework will be shown.

REFERENCES

[1] J. M. Bahi and S. Contassot-Vivier. Stability of fully asynchronous discrete-time discrete state dynamic networks. *IEEE Transactions on Neural Networks*, 13(6):1353–1363, 2002.

[2] J. Banks, J. Brooks, G. Cairns, and P. Stacey. On devaney's definition of chaos. *Amer. Math. Monthly*, 99:332–334, 1992.

[3] Jin Cong, Yan Jiang, Zhiguo Qu, and Zhongmei Zhang. A wavelet packets watermarking algorithm based on chaos encryption. *Lecture Notes in Computer Science*, 3980:921–928, 2006.

[4] Zhao Dawei, Chen Guanrong, and Liu Wenbo. A chaos-based robust wavelet-domain watermarking algorithm. *Chaos, Solitons and Fractals*, 22:47–54, 2004.

[5] Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems, 2nd Edition*. Westview Pr (Short Disc), March 2003.

[6] 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.

[7] 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.

[8] 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.

[9] F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. 1986.

[10] 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.

[11] 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.

[12] Xianyong Wu and Zhi-Hong Guan. A novel digital watermark algorithm based on chaotic maps. *Physics Letters A*, 365(5-6):403 – 406, 2007.

[13] Xianyong Wu, Zhi-Hong Guan, and Zhengping Wu. A chaos based robust spatial domain watermarking algorithm. *Lecture Notes in Computer Science*, 4492:113–119, 2007.

[14] 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.

[15] Jian Zhao, Mingquan Zhou, Hongmei Xie, Jinye Peng, and Xin Zhou. A novel wavelet image watermarking scheme combined with chaos sequence and neural network. *Lecture Notes in Computer Science*, 3174:663–668, 2004.