On the Collision Property of Chaotic Iterations Based Post-Treatments over Cryptographic Pseudorandom Number Generators

Luigi Marangio, Christophe Guyeux and Jacques M. Bahi
Femto-ST Institute, UMR 6174 CNRS
Université de Bourgogne Franche-Comté
France

***Abstract*—There is not a proper mathematical definition of *chaos*, we have instead a quite big amount of definitions, each of one describes chaos in a more or less general context. Taking in account this, it is clear why it is hard to design an algorithm that produce *random* numbers, a kind of algorithm that could have plenty of concrete appliceautifat (anul)d bions. However we must use a finite state machine (e.g. a laptop) to produce such a sequence of *random* numbers, thus it is convenient, for obvious reasons, to redefine those aimed sequences as *pseudorandom*; also problems arise with floating point arithmetic if one wants to recover some *real* chaotic property (i.e. properties from functions defined on the real numbers). All this considerations are synthesized in the problem of the Pseudorandom number generators (PRNGs). A solution to these obstacles may be to post-operate on existing PRNGs to improve their performances, using the so-called *chaotic iterations*, i.e., specific iterations of a boolean function and a shift operator that use the inputted generator. This approach leads to a mathematical description of such PRNGs as discrete dynamical systems, on which chaos properties can be investigated using mathematical topology and measure theory. Such properties are well-formulated, and they allow us to characterize which functions improves the sensitivity to the seed, the expansivity, the ergodicity, or the topological mixing of the generator resulting from such a post-processing. Experience shows that choosing relevant boolean functions in these chaotic iterations improves the randomness of the inputted generator, for instance when considering the number of statistical tests of randomness passed successfully. If we focus on the cryptographical application of PRNGs, there are two main classical notions to be considered, namely *collision* and *avalanche effect*. In this article, we recall the chaotic properties of the proposed post-treatment and we study the collision property in families of pseudorandom sequences produced by this process.**

***Index Terms*—Pseudorandom numbers generator, cryptography.**

I. Introduction

To simulate some well-known chaotic real functions, such as a logistic map or the Arnold's cat map, is the main idea of many algorithms developed until now which aspire to be good PRNGs, programs with a lot of applications for instance in cryptography. Indeed, it is reasonable to think that elements of chaos can improve the random-like quality produced by such algorithms. A theoretical result that establish a link between a chaotic map on $\mathbb{R}$ and its floating-point counterpart, it has not yet been stated. Conversely, there are some results in the opposite direction, indicating that the numerical truncation may change drastically the statistical properties of orbits (see, *e.g.*, [1], [2]). A first attempt to avoid the use of a chaotic real map, is to consider the so-called parallel asynchronous linear iterations (PALI). Even if the domain of definition is the floating point one, the effect of round-off errors in this situations was analyzed (for instance) in [3], with numerical analysis techniques, in particular for linear fixed point systems. A second attempt to design such kinds of PRNGs is to avoid the use of floating point arithmetic, by considering Boolean functions such as the logical negation, and then an asynchronous iteration scheme that includes this function coupled with a shift. In [4], the authors proved that such iterations, viewed as an operator on a suitable discrete dynamical system, satisfy various topological properties of chaos, like mixing.

Due to the finite structure of the problem it can be an hard task formalizing a notion of *pseudorandomness* for binary sequences, however Mauduit and Sárközy [5] made an attempt in doing so. In this new framework, it is relatively easy to redefine some of the classical notions that arise from cryptography, such as collision and avalanche effect, as pointed out by Tóth in [6].

In this article we recall how to design a PRNG based on chaotic iterations, how to describe it as a dynamical system and, for the sake of completeness, we recall some of the topological properties of chaos previously obtained. Finally, we provide new investigations of such a dynamical system using the theoretical set proposed by Mauduit and Sárközy.

This research work is organized as follows. In the first two sections we recall all the basic facts, such as how to post treat an existing PRNG by using chaotic iterations. Then, in the last section, we show that the collision free property of a family of binary sequences is preserved if we post treat this family using chaotic iterations. This research work ends by a conclusion section, in which our contributions are summarized and intended future work is outlined.

II. Basic recalls

We first explain how to properly design chaotic PRNGs by using discrete dynamical systems. To accomplish this task we must introduce some notations and terminologies.

In the remainder of this article, $\mathbb{N}$ is the set of natural (non-negative) numbers, while $\mathbb{N}^*$ stands for the positive integers $1, 2, 3, \ldots$ s_n denotes the n^{th} term of a sequence s while $X^{\mathbb{N}}$ is the set of all sequences whose elements belong to a given set X; also we denote by $(x^n)_{n\in\mathbb{N}}$ a sequence of those sequences. Instead of working with sequences of length N in $\{0,1\}^{\mathsf{N}}$, it will be useful considering sequences in $\{-1,+1\}^{\mathsf{N}}$. The set $\{-1,1\}$ is denoted $\mathbb{2}$.

Definition 1 Let $N \in \mathbb{N}^*$, $f : \mathbb{2}^{\mathsf{N}} \longrightarrow \mathbb{2}^{\mathsf{N}}$ be a function, and $s \in [1,\mathsf{N}]^{\mathbb{N}}$ be a sequence of integers between 1 and N. The so-called *chaotic iterations* are defined by $x^0 \in \mathbb{2}^{\mathsf{N}}$ 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 component of the vector x^n is updated; let us explain it with an example.

Example 1 For the sake of concreteness, let us consider that $\mathsf{N} = 3$. Let s be the sequence $s = (123123123123...)$ and let f be the Boolean negation (where in this case the boolean negation of -1 is 1 and viceversa). If we start with input (111), the chaotic iterations will produce the following output

$$(1,1,1) \to (-1,1,1) \to (-1,-1,1) \to (-1,-1,-1) \to$$
$$\to (1,-1,-1) \to \ldots$$

At each iteration step, we look at the correspondent element of the sequence s (at the n-th step, we look at the n-th element of s), and then we apply f to the element of the input sequence suggested by the element of s that we are considering. In the example, for instance, at step number 5 we need to change the second bit of the input, since the fifth element of s is 2.

We shall use the term "chaotic" (and similar) with various meaning throughout this paper and each of these terms has no link a priori with the other ones. In the definition of chaotic iterations, there is a sequence s of components to update as input, and a sequence of binary vectors as output. In other words, given a function f, these chaotic iterations transform a sequence $s = (s_n)_{n\in\mathbb{N}}$ in another sequence $(x_n)_{n\in\mathbb{N}}$. If s is provided by a pseudorandom number generator, we thus have defined a way to modify the produced sequence, leading to a post-treatment on this generator.

Let $\mathsf{N} \in \mathbb{N}^*$ and $f : \mathbb{2}^{\mathsf{N}} \to \mathbb{2}^{\mathsf{N}}$. Any sequence $u \in [1,\mathsf{N}]^{\mathbb{N}}$, provided by an inputted pseudorandom number generator, defines a "chaotic iterations based" PRNG, which is denoted by *CIPRNG*$^1_f(u)$. It is defined by [7], [8]:

$$x^0 \in \mathbb{2}^{\mathsf{N}}, \forall n \in \mathbb{N}, \forall i \in [1,\mathsf{N}], x_i^{n+1} = \begin{cases} f(x^n)_i & \text{if } i = u_n, \\ x_i^n & \text{else.} \end{cases}$$

The outputted sequence produced by this *CIPRNG*$^1_f(u)$ generator is $(x^n)_{n\in\mathbb{N}}$.

However for several reasons to be discuss later, it will be useful to consider another version of this generator, in which is allow to update more than one digit at each iteration step. If $\mathsf{P} \subset \mathbb{N}^*$ is a non empty and finite set of integers, any couple $(u,v) \in [1,\mathsf{N}]^{\mathbb{N}} \times \mathsf{P}^{\mathbb{N}}$ defines another chaotic iterations based PRNG, which is denoted by *CIPRNG*$^2_f(u,v)$ (defined in [9]). It is defined as follows:

$$\begin{cases} x^0 \in \mathbb{2}^{\mathsf{N}} \\ \forall n \in \mathbb{N}, \forall i \in [1,\mathsf{N}], x_i^{n+1} = \begin{cases} f(x^n)_i & \text{if } i = u^n \\ x_i^n & \text{else} \end{cases} \\ \forall n \in \mathbb{N}, y^n = x^{v^n}. \end{cases}$$

The outputted sequence produced by this generator is $(y^n)_{n\in\mathbb{N}}$. In other words, the first inputted generator u is the sequence of components to update, while the second inputted generator v provides the number of iterates between two outputs of the *CIPRNG*$^2_f(u,v)$ generator. Note that *CIPRNG*$^1_f(u)$ is equal to *CIPRNG*$^2_f\left(u,(1)_{n\in\mathbb{N}}\right)$, where $(1)_{n\in\mathbb{N}}$ is the sequence that is uniformly equal to 1. Pseudorandom number sequences have plenty of applications in computer science and in particular in cryptography; in [10], it has been shown how to use *CIPRNG* to produce a cryptographycally secure PRNG. In [4] the chaotic property of *CIPRNG* has been analyzed, and some of them are listed in the next section; also simulations show the good behavior of such PRNGs. An interesting particular situation has been analyzed: if we consider the boolean negation as function involved in *CIPRNG* (a good PRNG should be fast and the boolean negation is really cheap from a computational point of view), simulations shown that *CIPRNG*$^2_f(u,v)$ really improved the chaotic behavior of the inputted generator, while *CIPRNG*$^1_f(u)$ does not work as good as its "big brother". In an article that will appear, authors provided a theoretical explanation to this phenomenon by using ergodic theory; in the next sections (after some recalls) it will be shown that actually *CIPRNG*$^2_f(u,v)$, where f is the boolean negation and under some hypothesis, satisfies a classical cryptography property, namely to be *collision free*, while there is no reason to conclude that such a property is satisfy also by *CIPRNG*$^1_f(u)$. Thus we conclude that *CIPRNG*$^2_f(u,v)$ can be successfully used to improve the cryptography properties of a family of pseudorandom binary sequences.

III. Chaotic behaviour of *CIPRNG*

In this section are listed results concerning the topological randomness of the presented class of generators. They appear, for instance, in [4]. Firstly we recall the theoretical framework that can emphasize such a behavior and how to adapt the definition of *CIPRNG* to fit this framework (i.e., *CIPRNG* can be described as a dynamical system on a suitable metric space). Even if the results listed here are about *CIPRNG*^1_f, they can be easily extended to *CIPRNG*^2_f.

A. A notion of randomness: Devaney's theory of chaos

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

Definition 2 Let U, V be any pair of opens subsets of $\mathcal{X}$. If there exists an integer $k > 0$ such that $f^k(U) \cap V \neq \varnothing$, then the function f is said to be *topologically transitive*

Definition 3 If the set of periodic points for f is dense in $\mathcal{X}$ (i.e. for any point x in $\mathcal{X}$, any neighborhood of x contains at least one periodic point), then f is *regular* on $(\mathcal{X}, \tau)$; let us recall that an element x is a *periodic point* for f of period $n \in \mathbb{N}$, $n > 1$, if $f^n(x) = x$.

Definition 4 (Devaney's formulation of chaos [11])
The function f is *chaotic* on $(\mathcal{X}, \tau)$ if f is regular and topologically transitive.

Banks *et al.* have proven in [12] that, when the topological space is a metric one $(\mathcal{X}, d)$, chaos implies sensitivity, defined below:

Definition 5 The function 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.

To show that *CIPRNG*$^1_f(u)$ has a chaotic dependence regarding modifications on u is equivalent to prove that "chaotic iterations" are indeed chaotic according to the definition of Devaney recalled above.

We first recall how to define a suitable metric space where chaotic iterations are continuous. For further explanations, see, *e.g.*, [13]. Let δ be the *discrete Boolean metric*, $\delta(x, y) = 0 \Leftrightarrow x = y$. Given a function f, define the function $F_f : [1; \mathsf{N}] \times 2^{\mathsf{N}} \longrightarrow 2^{\mathsf{N}}$ by:

$$(k, E) \quad \longmapsto \quad \left(E_j.\delta(k, j) + f(E)_k.\overline{\delta(k, j)}\right)_{j \in [1; \mathsf{N}]}$$

where + and . are the Boolean addition and product operations. Consider the phase space: $\mathcal{X} = [1; \mathsf{N}]^{\mathbb{N}} \times 2^{\mathsf{N}}$, and the map defined on $\mathcal{X}$ by:

$$G_f\left(S, E\right) = \left(\sigma(S), F_f(i(S), E)\right), \tag{1}$$

where σ is the *shift* function defined by $\sigma : (S^n)_{n \in \mathbb{N}} \in [1, \mathsf{N}]^{\mathbb{N}} \longrightarrow (S^{n+1})_{n \in \mathbb{N}} \in [1, \mathsf{N}]^{\mathbb{N}}$ and i is the *initial function* $i : (S^n)_{n \in \mathbb{N}} \in [1, \mathsf{N}]^{\mathbb{N}} \longrightarrow S^0 \in [1; \mathsf{N}]$. Then the chaotic iterations proposed in Definition 1 can be described by the following discrete dynamical system, whose topological chaos can now be studied:

$$\left\{ \begin{array}{l} X^0 \in \mathcal{X} \\ X^{k+1} = G_f(X^k). \end{array} \right. \tag{2}$$

To do so, a relevant distance between two points $X = (S, E), Y = (\check{S}, \check{E}) \in \mathcal{X}$ has been introduced in [13] as follows: $d(X, Y) = d_e(E, \check{E}) + d_s(S, \check{S})$, where

$$\left\{ \begin{array}{lll} d_e(E, \check{E}) & = & \displaystyle\sum_{k=1}^{\mathsf{N}} \delta(E_k, \check{E}_k), \\ d_s(S, \check{S}) & = & \displaystyle\frac{9}{\mathsf{N}} \sum_{k=1}^{\infty} \frac{|S^k - \check{S}^k|}{10^k}. \end{array} \right. \tag{3}$$

It has been established in [13] that,

Proposition 1 *G_f is continuous in the metric space $(\mathcal{X}, d)$.*

Let f be a map from 2^{N} to itself. The *asynchronous iteration graph* associated with f is the directed graph $\Gamma(f)$ defined by: the set of vertices is 2^{N}; for all $x \in 2^{\mathsf{N}}$ and $i \in [1; \mathsf{N}]$, the graph $\Gamma(f)$ contains an arc from x to $F_f(i, x)$.

It has been proven in [14] that,

Theorem 1 *Let $f : 2^{\mathsf{N}} \to 2^{\mathsf{N}}$. G_f is chaotic according to Devaney if and only if $\Gamma(f)$ is strongly connected.*

B. Further analysis: Li-York Approach, Lyapunov Exponents, Entropy, Uniformly Distributed Output

Additionally to the Devaney's chaos, a discrete dynamical system can be intrinsically complicated for various other understanding of this wish, that are not equivalent one another, like:

- *Undecomposable*: it is not the union of two nonempty closed subsets that are positively invariant ($f(A) \subset A$).
- *Total transitivity*: $\forall n \geqslant 1$, the composition function $f^n = f \circ f \circ \ldots \circ f$ is transitive.
- *Strong transitivity*: $\forall x, y \in \mathcal{X}$, $\forall r > 0$, $\exists z \in B(x, r)$, $\exists n \in \mathbb{N}$, $f^n(z) = y$.
- *Topological mixing*: for all pairs of disjoint open nonempty sets U and V, there exists $n_0 \in \mathbb{N}$ such that $\forall n \geqslant n_0, f^n(U) \cap V \neq \varnothing$.

These varieties of definitions lead to various notions of chaos. For instance, a dynamical system is chaotic according to Wiggins if it is transitive and sensible to the initial conditions. According to Knudsen if a dynamical system has a dense orbit while being sensible then we are able to use again the adjective chaotic for such a system. Also, when the properties of transitivity, regularity, and expansiveness are satisfied we fit the so called *expansive chaos*.

Let us denote by $\mathcal{C}$ the set of $f : 2^{\mathsf{N}} \to 2^{\mathsf{N}}$ such that $\Gamma(f)$ is strongly connected. We have previously established that ([15], [16]) $\forall f \in \mathcal{C}$, G_f is strongly transitive undecomposable, unstable, and chaotic as defined by Wiggins.

Further results have been obtained when considering the boolean negation f_0, which are summarized below [17].

Theorem 2 *$(\mathcal{X}, G_{f_0})$ is topologically mixing and expansive with a constant equal to 1.*

Definition 6 Let $(\mathcal{X}, d)$ a metric space and $f : \mathcal{X} \longrightarrow \mathcal{X}$ a continuous map. A scrambled couple of points is a pair $(x, y) \in \mathcal{X}^2$ such that $\liminf_{n \to \infty} d(f^n(x), f^n(y)) = 0$ and $\limsup_{n \to \infty} d(f^n(x), f^n(y)) > 0$, or in other words, the two orbits oscillate.

A scrambled set is a set in which any couple of points are a scrambled couple, whereas a Li-Yorke chaotic system is a system possessing an uncountable scrambled set.

We have previously stated that [18],

Theorem 3 *Chaotic iterations are chaotic as defined by Li and Yorke.*

Theorem 4 *Chaotic iterations using the boolean negation have a topological entropy equal to $ln(\mathsf{N})$.*

Theorem 5 *Chaotic iterations using the boolean negation have an exponent of Lyapunov equal to* $ln(\mathsf{N})$.

Finally, it has been proven in [14] that,

Theorem 6 *Let* $f : 2^n \to 2^n$, $\Gamma(f)$ *its iteration graph,* $\check{M}$ *its adjacency matrix and* M *a* $n \times n$ *matrix defined by* $M_{ij} = \frac{1}{n}\check{M}_{ij}$ *if* $i \neq j$ *and* $M_{ii} = 1 - \frac{1}{n}\sum_{j=1,j\neq i}^{n} \check{M}_{ij}$ *otherwise.*

If $\Gamma(f)$ *is strongly connected, then the output of CIPRNG*$_f^1(u)$ *follows a law that tends to the uniform distribution if and only if* M *is a double stochastic matrix.*

These results of topological chaos and uniform distribution have initially led us to study the possibility of building a pseudorandom number generator (PRNG) based on chaotic iterations. As G_f, defined on the domain $[1;\mathsf{N}]^{\mathbb{N}} \times 2^{\mathsf{N}}$, is built from Boolean networks $f : 2^{\mathsf{N}} \to 2^{\mathsf{N}}$, we can preserve the theoretical properties on G_f during implementations (due to the discrete nature of f).

IV. Collision property

A. Basic recalls

It should be clear from the last section that there are several ways to define the notion of pseudorandomness. Recently a constructive approach to this notion was initiated by Mauduit and Sárközy [5], and during these years it has been extended to more general situations. V.Toth in [6] adapted the classical cryptographical notions of collision and avalanche effect to this new framework. A good survey of this new approach can be found in [19]; in this section we will recall the basic definition and in the next section we will discuss the behaviour of *CIPRNG* in this new framework, with particular focus on the collision property. Let $N \in \mathbb{N}$, let $\mathcal{S}$ be a given set (e.g., a set of certain polynomials or the set of all binary sequences of a given length much less than N), to each $s \in \mathcal{S}$ we assign a unique binary sequence

$$E_N(s) = (e_1, \dots, e_N) \in \{-1, +1\}^N,$$

and let $\mathcal{F} = \mathcal{F}(\mathcal{S})$ denote the family of the binary sequences obtained in this way:

$$\mathcal{F} = \mathcal{F}(\mathcal{S}) = \{E_N(s) : s \in \mathcal{S}\}.$$

Definition 7 We have a *collision* in $\mathcal{F}$ if there are two different elements $s \neq s' \in \mathcal{S}$ such that

$$E_N(s) = E_N(s');$$

if there is no collision in $\mathcal{F}$ then $\mathcal{F}$ is said to be *collision free*.

An ideally good family of pseudorandom binary sequence should be collision free or, at least, the number of collision should be limited. We can reformulate this notion.

Definition 8 Let $N \in \mathbb{N}$ and let $E_N = (e_1, \dots, e_N)$, $E'_N = (e'_1, \dots, e'_N)$ be two sequences of $\{-1, +1\}^N$, then the *distance* between E_N and E'_N is defined by

$$d(E_N, E'_N) = |\{n : 1 \leq n \leq N,\ e_n \neq e'_n\}|.$$

Moreover the *distance minimum* $m(\mathcal{F})$ of $\mathcal{F}$ can be defined as

$$m(\mathcal{F}) = \min_{s \neq s',\, s,s' \in S} d(E_N(s), E_N(s')).$$

With the definitions above easily follows that a family $\mathcal{F}$ is collision free if and only if $m(\mathcal{F}) > 0$.

B. Collisions in CIPRNG^1

Let $\mathcal{F}(\mathcal{S})$ be a family of binary sequences generated, for instance, by a PRNG (or several PRNGs); we would like to post process this family with $CIPRNG$ and see when we are able to improve the distance minimum of the inputted family.

Firstly we investigate a very simple case, i.e. the effects of $CIPRNG_f^1$ where f is the boolean negation; with abuse of notation (not really) let $S = \mathcal{F}(\mathcal{S})$ be a set of strategies of length k, let $u = 1^N$ be the input of $CIPRNG_f^1$ and let $F = \{CIPRNG_f^1(s) : s \in S\}$ the outputted family of binary sequences. We want to study collisions in F, i.e. we want to see what kind of conditions we need in order to get $m(F) > 0$.

For better notations we shall write $C(s)$ instead of $CIPRNG_{f,N}^1(u, s)$, since the input string u and the length N are fixed. Now observe that if $C(s) = (c_1(s), \dots, c_N(s))$, then for $n = 1, 2, \dots, N$ we have

$$c_n(s)c_n(t) = \begin{cases} +1 & \text{if } c_n(s) = c_n(t), \\ -1 & \text{if } c_n(s) \neq c_n(t) \end{cases}$$

thus

$$\frac{1}{2}(1 - c_n(s)c_n(t)) = \begin{cases} 0 & \text{if } c_n(s) = c_n(t), \\ +1 & \text{if } c_n(s) \neq c_n(t) \end{cases}$$

It follows that

$$\begin{aligned} d(C(s), C(t)) &= \sum_{n=1}^{N} \frac{1}{2}(1 - c_n(s)c_n(t)) \\ &= \frac{1}{2}(N - \sum_{n=1}^{N} c_n(s)c_n(t)) \\ &= \frac{1}{2}(N - \sum_{n=1}^{N} (-1)^{|n \in s| + |n \in t| (2)}), \end{aligned}$$

where $|n \in s|$ is the number of occurrences of the symbols n in the string s. To understand the last equality in the expression above observe that the n-th digit of $c(s)$ (i.e. $c_n(s)$) is equal to 1 if n appear an even numbers of times in the strategy (since the boolean negation is such that $f \circ f = id$ and we started from input $u = 1^N$), otherwise is equal to -1.

So far from now we obtained that if we want to estimate

$$m(F) = \min_{s \neq t} d(C(s), C(t)),$$

we should be able to estimate the following quantity

$$\max_{s \neq t} \sum_{n=1}^{N} (-1)^{|n \in s| + |n \in t| (2)}.$$

However, since the strategies are computed in some way by some PRNG (or several PRNGs), is not clear why assuming the collision free property of the set of the strategies should led to the collision free property for the outputted family, that now can be expressed by the following formula

$$\max_{s \neq t} \sum_{n=1}^{N} (-1)^{|n \in s| + |n \in t| (2)} < N.$$

C. Collisions in *CIPRNG*^{2}

In the last section it has been shown that there is no reasons to conclude that *CIPRNG*^{1} preserves the collision free property; in this section it will be show that *CIPRNG*^{2} is more suitable for this task. As in the previous section, let $S = \mathcal{F}(\mathcal{S})$ be a set of strategies generated in some way and let $P = \mathcal{F}'(\mathcal{P})$ be a set of *iterations strings* (pseudo) randomly generated (by some PRNGs). Recall that the strategies are sequences in $\{1, \ldots, N\}^k$ while the iteration strings are sequences in $\{2, 3\}^{k_1}$; the set $\{2, 3\}$ can be actually any finite set of integers with $k_1 < k$.

We fix the input $u = 1^N$ and we want to post process the family $S \times P$ with $CIPRNG_f^2(s, p) = C(s, p)$, where f is the boolean negation and $(s, p) \in S \times P$.

As before we want an estimate of this quantity

$$d(C(s,p), C(t,q)) = \frac{1}{2}(N - \sum_{n=1}^{N} c_n(s,p) c_n(t,q));$$

recalling the structure of $CIPRNG^2$, we can observe that an iteration string p induces a partition of the strategy s (in block of size 2 or 3 in this case), thus we can write

$$s = s_1 s_2 \ldots s_r,$$

for some r, where $|s_i| = p_i$ for $i = 1, \ldots, r-1$ and $p_r \leq |s_r| \leq p_r + \max\{2, 3\} - 1$. To avoid the fact that the last block of a strategy can have size different from 2 or 3, we will just consider $r - 1$ iterations instead of r. Due to the fact that we are using the boolean negation and we started with input 1^N, we have

$$\begin{aligned} d(C(s,p), C(t,q)) &= \sum_{n=1}^{N} \frac{1}{2}(1 - c_n(s,p) c_n(t,q)) \\ &= \frac{1}{2}(N - \sum_{n=1}^{N} c_n(s,p) c_n(t,q)) \\ &= \frac{1}{2}(N - \sum_{n=1}^{N} (-1)^{|n \in s| + |n \in t| (2)}), \\ &= \frac{1}{2}(N - \sum_{n=1}^{N} (-1)^{\sum_{j=1}^{r-1} |n \in s_j| + |n \in t_j| (2)}) \end{aligned}$$

So now our aim is to prove the following inequality

$$\max_{(s,p) \neq (t,q)} \sum_{n=1}^{N} (-1)^{\sum_{j=1}^{r-1} |n \in s_j| + |n \in t_j| (2)} < N.$$

Suppose N prime (in this way things works much better, but probably this is not necessary) and suppose for an absurdum that there are $(s, p) \neq (t, q)$ such that

$$\sum_{n=1}^{N} (-1)^{\sum_{j=1}^{r-1} |n \in s_j| + |n \in t_j| (2)} = N;$$

thus

$$\begin{aligned} N &= \sum_{n=1}^{N} (-1)^{\sum_{j=1}^{r-1} |n \in s_j| + |n \in t_j| (2)} \\ &= \sum_{n=1}^{N} \prod_{j=1}^{r-1} (-1)^{|n \in s_j| + |n \in t_j| (2)} \\ &= \prod_{j=1}^{r-1} \sum_{n=1}^{N} (-1)^{|n \in s_j| + |n \in t_j| (2)} \end{aligned}$$

We will show now that, under the assumption that the family of the iteration strings P is collision free, then there is some j^* such that $\sum_{n=1}^{N} (-1)^{|n \in s_{j^*}| + |n \in t_{j^*}| (2)}$ is neither 1 nor N, thus we reach an absurdum since N is prime.

In fact, if the family P is collision free, then for any two elements, there is at least one index with different digits in those two elements. Thus, they induced a different partition of the strategies, i.e., there is a j^* such that (without loss of generality) $|s_{j^*}| = 2$ and $|t_{j^*}| = 3$. There are two cases, either $s_{j^*} = AA$ or $s_{j^*} = AB$, where $A, B \in \{1, \ldots, N\}$; in each of these cases, we can compute all the possible configurations of t_{j^*} and the correspondent values.

case 1 In this case the possible configurations of t_{j^*} are

$t_{j^*} = BCD$ in this case A appears with the same parity in both s and t, while B, C, D appear with different parity, and then there are other $N - 4$ letters that appear with the same parity, thus the total is $1 + (-3) + N - 4 = N - 6$;

$t_{j^*} = ABC$ $(-3) + (N - 3) = N - 6$;

$t_{j^*} = ABB$ $(-1) + (1) + N - 2 = N - 2$;

$t_{j^*} = AAB$ $(1) + (-1) + N - 2 = N - 2$;

$t_{j^*} = AAA$ $(-1) + (N - 1) = N - 2$.

and this prove the first case.

case 2 For the second case, we will first consider all the possible configuration of t_{j^*} without any letter in common with s_{j^*}. Then, all the possible configuration with one letter in common, and finally the case with two letters in common.

$t_{j^*} = CDE$ $(-5) + (N - 5) = N - 10$;

$t_{j^*} = ACD$ observe that we can assume that the letter in common is A; here we have $(1) + (-1) + (-2) + N - 4 = N - 6$;

$t_{j^*} = ACC$ $(1) + (-1) + (N - 2) = N - 2$;

$t_{j^*} = AAC$ $(-1) + (-1) + (-1) + N - 3 = N - 6$;

$t_{j^*} = ABC$ $(+2) + (-1) + N - 3 = N - 2$;

$t_{j^*} = ABA$ $(1) + (-1) + N - 2 = N - 2$.

V. Conclusion and future work

Designing a pseudorandom number generator is a complicated task. In this article we presented a way to post treat a family of binary sequences in order to improve their randomness. After having recalled the structure of such a post treatment and its properties, we proved in the new framework introduced by Mauduit and Sárközy, that the post treatment preserves the collision free property of the family, a classical notion in cryptography.

In future work we intend to extend our investigation to the avalanche effect property, and more in general we would like to proof that such a post treatment actually increase the distance minimum of a family of binary sequences. Seems reasonable to think that the structure of the chaotic iterations strictly depend from the boolean function involved in the process, we would like to explore a possible characterization of such a property.

Acknowledgement

I would like to say thank you to professor Stefano Galatolo for his guide through this new world (and for saying me where to put his name in this article).

References

[1] S. Galatolo, I. Nisoli, and C. Rojas, "Probability, statistics and computation in dynamical systems," *Mathematical Structures in Computer Science*, vol. 24, no. 3, 2014.

[2] P.-A. Guiheneuf, "Dynamical properties of spatial discretizations of a generic homeomorphism," *Ergodic Theory and Dynamical Systems*, vol. 35, no. 5, pp. 1474–1523, 2015.

[3] J. M. Bahi, "Asynchronous iterative algorithms for nonexpansive linear systems," *Journal of Parallel and Distributed Computing*, vol. 60, pp. 92–112, 2000.

[4] C. Guyeux and J. Bahi, "A topological study of chaotic iterations. application to hash functions," in *CIPS, Computational Intelligence for Privacy and Security*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 394, pp. 51–73, revised and extended journal version of an IJCNN best paper.

[5] C. a. Maudit, "On finite pseudoradom binary sequences i: The measure of pseudorandomness, the legendre symbol."

[6] V. Tóth, "Collision and avalanche effect in families of pseudorandom binary sequences." *Period. Math. Hungar.*, vol. 55, pp. 185–196, 2007.

[7] J. Bahi, C. Guyeux, and Q. Wang, "A novel pseudo-random generator based on discrete chaotic iterations," in *INTERNET'09, 1-st Int. Conf. on Evolving Internet*, Cannes, France, Aug. 2009, pp. 71–76. [Online]. Available: http://dx.doi.org/10.1109/INTERNET.2009.18

[8] J. M. Bahi, C. Guyeux, and Q. Wang, "Improving random number generators by chaotic iterations. Application in data hiding," in *ICCASM 2010, Int. Conf. on Computer Application and System Modeling*, Taiyuan, China, Oct. 2010, pp. V13–643–V13–647. [Online]. Available: http://dx.doi.org/10.1109/ICCASM.2010.5622199

[9] Q. Wang, J. Bahi, C. Guyeux, and X. Fang, "Randomness quality of CI chaotic generators. application to internet security," in *INTERNET'2010. The 2nd Int. Conf. on Evolving Internet*. Valencia, Spain: IEEE Computer Society Press, Sep. 2010, pp. 125–130, best Paper award. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/INTERNET.2010.30

[10] J. M. Bahi, R. Couturier, C. Guyeux, and P.-C. Heam, "Efficient and cryptographically secure generation of chaotic pseudorandom numbers on gpu," *The journal of Supercomputing*, vol. 71(10), pp. 3877–3903, 2015.

[11] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Redwood City, CA: Addison-Wesley, 1989.

[12] J. Banks, J. Brooks, G. Cairns, and P. Stacey, "On Devaney's definition of chaos," *Amer. Math. Monthly*, vol. 99, pp. 332–334, 1992.

[13] J. M. Bahi and C. Guyeux, "Topological chaos and chaotic iterations, application to hash functions," in *WCCI'10, IEEE World Congress on Computational Intelligence*, Barcelona, Spain, Jul. 2010, pp. 1–7, best paper award.

[14] J. Bahi, J.-F. Couchot, C. Guyeux, and A. Richard, "On the link between strongly connected iteration graphs and chaotic boolean discrete-time dynamical systems," in *FCT'11, 18th Int. Symp. on Fundamentals of Computation Theory*, ser. LNCS, vol. 6914, Oslo, Norway, Aug. 2011, pp. 126–137.

[15] J. Bahi, X. Fang, C. Guyeux, and Q. Wang, "Evaluating quality of chaotic pseudo-random generators. application to information hiding," *IJAS, International Journal On Advances in Security*, vol. 4, no. 1-2, pp. 118–130, 2011.

[16] C. Guyeux and J. M. Bahi, "A topological study of chaotic iterations. application to hash functions," *CIPS, Computational Intelligence for Privacy and Security*, vol. 394, no. Studies in Computational Intelligence, pp. 51–73, 2012, revised and extended journal version of an IJCNN best paper.

[17] C. Guyeux, N. Friot, and J. Bahi, "Chaotic iterations versus spread-spectrum: chaos and stego security," in *IIH-MSP'10, 6-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Darmstadt, Germany, Oct. 2010, pp. 208–211. [Online]. Available: http://dx.doi.org/10.1109/IIHMSP.2010.59

[18] C. Guyeux, "Le désordre des itérations chaotiques et leur utilité en sécurité informatique," Ph.D. dissertation, Université de Franche-Comté, 2010.

[19] Sárk "On pseudorandomness of families of binary sequences."