



**HAL**  
open science

# Cryptography-Based Chaos via Geometric Undersampling of Ring-Coupled Attractors

René Lozi

► **To cite this version:**

René Lozi. Cryptography-Based Chaos via Geometric Undersampling of Ring-Coupled Attractors. A. H. Siddiqi, P. Manchanda, R. Bhardwaj. *Mathematical Models, Methods and Applications*, Springer, pp.1-30, 2015, 978-981-287-971-4. hal-01333364

**HAL Id: hal-01333364**

**<https://hal.science/hal-01333364>**

Submitted on 17 Jun 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# CRYPTOGRAPHY BASED CHAOS VIA GEOMETRIC UNDERSAMPLING OF RING-COUPLED ATTRACTORS

RENÉ LOZI<sup>1</sup>

**Abstract.** We propose a new mechanism for undersampling chaotic numbers obtained by the ring coupling of one-dimensional maps. In the case of 2 coupled maps this mechanism allows the building of a PRNG which passes all NIST Test.

This new geometric undersampling is very effective for generating 2 parallel streams of pseudo-random numbers, as we show, computing carefully their properties, up to sequences of  $10^{12}$  consecutive iterates of the ring coupled mapping which provides more than  $3.35 \times 10^{10}$  random numbers in very short time. Both 3 and 4 dimension cases can be managed in the same way.

In addition we recall a novel method of noise-resisting ciphering. The originality lies in the use of a chaotic pseudo-random number generator: several co-generated sequences can be used at different steps of the ciphering process, since they present the strong property of being uncorrelated. Each letter of the initial alphabet of the plain text is encoded as a subinterval of  $[-1,1]$ . The bounds of each interval are defined in function of the known bound of the additive noise. A pseudo-random sequence is used to enhance the complexity of the ciphering. The transmission consists of a substitution technique inside a chaotic carrier, depending on another cogenerated sequence. This novel noise-resisting ciphering method can be used with geometric undersampling when 4 mappings are coupled.

## CONTENTS

Introduction	2
1. Recovering chaotic properties of numerically computed chaotic numbers	2
1.1. Numerical approximation of chaotic numbers	2
1.2. Very long periodic orbits for ultra-weakly coupled tent map	4
1.2.1. 2-coupled symmetric tent map	4
1.2.2. $p$ -coupled symmetric tent map	4
1.2.3. Computation of approximated invariant measure	6
2. The route from chaos to pseudo-randomness via chaotic undersampling	8
2.1. Chaotic under-sampling	8
2.2. Chaotic mixing	9
2.3. Enhanced chaotic under-sampling	10
2.4. A window of emergence of randomness	10
3. Geometric undersampling	10
3.1. Pseudo-random numbers generated by ring coupled mapping	11
3.2. Ring coupling of 2-dimensional symmetric tent map	12
3.2.1. Critical lines	13
3.2.2. Markov partition	14
3.2.3. Exact computation of invariant measure associated to $M_2$	16
3.3. Geometric undersampling	17
3.3.1. Algorithm of geometric undersampling	17
3.3.2. Numerical tests	18
4. Noise-resisting ciphering	20
4.1. Ciphering principle	20
4.2. Transmission principle	24
4.3. Decoding principle	24
4.4. Numerical illustration	24
Conclusion	25

---

<sup>1</sup> Université de Nice Sophia-Antipolis, Laboratoire J. A. Dieudonné, UMR CNRS 7351, Parc Valrose, 06108 NICE, Cedex 02, France, email : [rlozi@unice.fr](mailto:rlozi@unice.fr)

## INTRODUCTION

During the last decade, it has been emphasized that the undersampling of sequence of chaotic numbers is an efficient tool in order to build pseudo-random number generators (PRNG) [15]. Randomness appears to be an emergent property of complex systems of coupled chaotic maps [16]. Several kinds of coupling can be considered as ultra-weak coupling or ring coupling, ... [17]. An ultra-weak coupling recovers chaotic properties of 1-dimensional maps [12, 13] when computed with floating numbers or double precision numbers. Chaotic undersampling with thresholds based on one component of the coupled system adds random properties to the chaotic sequences. Double threshold sampled sequence (i.e., using both thresholds of different nature) improves such random properties [14]. Ring coupling deals when  $p$  1-dimensional maps are constrained on a torus [5, 26], this coupling can directly provide random numbers, without sampling or mixing, provided the number  $p$  of maps is large enough, although it is possible to combine these processes with it. However in lower dimension 2 and 3, the chaotic numbers are not equidistributed on the torus. Therefore we introduce a particular “geometric” undersampling based on the property of piecewise linearity of the invariant measure of the system of  $p$  1-dimensional ring coupled maps. This new geometric undersampling is very effective for generating parallel streams of pseudo-random numbers with a very compact mapping.

Several applications in various fields (chaotic optimization, evolutionary algorithms, secure information transmission, chaotic cryptography, ...) of such under-sampling process can be found. In this article we focus on the last ones.

- As a first example we propose a novel noise-resisting ciphering based on a large number of uncorrelated chaotic sequences. These cogenerated sequences are actually used in several steps of the ciphering process. Noisy transmission conditions are considered, with realistic assumptions. The efficiency of the proposed method for ciphering and deciphering is illustrated through numerical simulations based on ten coupled chaotic sequences [4].

- A second one is the use of such sequences in a chaotic encryption algorithm [27].

In Section 1 we briefly recall properties of chaotic mappings, when used alone or ultra-weakly coupled. Section 2 describes the route from chaos to randomness via chaotic undersampling, discovered during the last decade. In Section 3, we introduce geometric undersampling in the scope of ring coupled mapping. In Section 4, we propose in addition, a novel method of noise-resisting ciphering. The originality lies in the use of a chaotic pseudo-random number generator: several co-generated sequences can be used at different steps of the ciphering process, since they present the strong property of being uncorrelated. This novel noise-resisting ciphering method can be used with geometric undersampling when 4 mappings are coupled.

### 1. RECOVERING CHAOTIC PROPERTIES OF NUMERICALLY COMPUTED CHAOTIC NUMBERS

#### 1.1. Numerical approximation of chaotic numbers

Chaos theory studies the behavior of dynamical systems that are highly sensitive to initial conditions, an effect which is popularly referred to as the butterfly effect. Small differences in initial conditions (such as those due to rounding errors in numerical computation) yield widely diverging outcomes for chaotic systems, rendering long-term prediction impossible in general. This happens even though these systems are deterministic, meaning that their future

behavior is fully determined by their initial conditions, with no random elements involved. In other words, the deterministic nature of these systems does not make them predictable. The first example of such chaotic continuous system in the dissipative case was pointed out by the meteorologist E. Lorenz in 1963 [11].

In order to study numerically the properties of the Lorenz attractor, M. Hénon an astronomer of the observatory of Nice, France, introduced in 1976 a simplified model of the Poincaré map of this attractor [9]. The Lorenz attractor being imbedded in dimension 3, the corresponding Poincaré map is a mapping from the plane  $\mathbb{R}^2$  into  $\mathbb{R}^2$ . Hence the Hénon mapping is also defined in dimension 2 and is associated to the dynamical system

$$\begin{cases} x_{n+1} = y_n + 1 - ax_n^2, \\ y_{n+1} = bx_n \end{cases}, \quad (1)$$

which has been extensively studied since thirty six years.

More simple dynamical systems in dimension one, on the interval  $J = [-1,1] \subset \mathbb{R}$  into itself

$$x_{n+1} = f_a(x_n), \quad (2)$$

corresponding to the logistic map

$$f_a \equiv L_a(x) = 1 - ax^2, \quad (3)$$

or the symmetric tent map

$$f_a \equiv T_a(x) = 1 - a|x|, \quad (4)$$

have also been fully explored in the hope of generating random numbers easily [24]. However when a dynamical system is realized on a computer using floating point or double precision numbers, the computation is of a discretization, where finite machine arithmetic replaces continuum state space. For chaotic dynamical systems in small dimension, the discretization often has collapsing effects to a fixed point or to short cycles [6].

It seems that the computation of numerical approximations of the periodic orbits leads to unpredictable and somewhat enigmatic results. As says O. E. Lanford III [10] “The reason is that because of the expansivity of the mapping the growth of roundoff error normally means that the computed orbit will remain near the true orbit with the chosen initial condition only for a relatively small number of steps typically of the order of the number of bits of precision with which the calculation is done. It is true that the above mapping like many ‘chaotic’ mappings satisfies a shadowing theorem (*see* [20, 21]) which ensures that the computed orbit stays near to some true orbit over arbitrarily large numbers of steps. The flaw in this idea as an explanation of the behavior of computed orbits is that the shadowing theorem says that the computed orbit approximates some true orbit but not necessarily that it approximates a typical one.”

The collapsing of iterates of dynamical systems or at least the existence of very short periodic orbits, their non constant invariant measure, and the easily recognized shape of the function in the phase space avoid the use of one-dimensional map (logistic, baker, or tent, ...) as a Pseudo Random Number Generator (see [18] for a survey).

**Remark 1.1** However, the very simple implementation in computer program of chaotic dynamical systems led some authors to use it as a base of cryptosystem [3, 2]. In addition it seems that for some applications, chaotic numbers are more efficient than random numbers. That is the case for evolutionary algorithms [22, 25] or chaotic optimization [1].

In this paper we show how to overcome the poor quality of chaotic generators using geometric undersampling. This special undersampling we introduce in this article is one among others undersampling processes we have studied before. In order to explain the difference between these processes we give in Sec. 2 a brief survey of them. Before doing this survey, we have to show how to stabilize the chaotic properties of chaotic number when realized on a computer.

## 1.2. Very long periodic orbits for ultra-weakly coupled tent map

The first step in order to preserve the genuine chaotic properties of the continuous models in numerical experiments is reached considering ultra-weak multidimensional coupling of  $p$  one-dimensional dynamical systems instead of solely a one-dimensional map.

### 1.2.1. 2-coupled symmetric tent map

In order to simplify the presentation below, we use as an example the symmetric tent map (4) with the parameter value  $a = 2$ , later denoted simply as  $f$ , even though others chaotic map of the interval (as the logistic map, the baker transform, ...) can be used for the same purpose (as a matter of course, the invariant measure of the chaotic map considered is preserved).

When  $p = 2$ , the system is simply described by Eq. (5)

$$\begin{cases} x_{n+1} = (1-\varepsilon_1)f(x_n) + \varepsilon_1 f(y_n) \\ y_{n+1} = \varepsilon_2 f(x_n) + (1-\varepsilon_2)f(y_n) \end{cases}, \quad (5)$$

We use generally  $\varepsilon_1 = 10^{-7}$ ,  $\varepsilon_2 = 2\varepsilon_1$  when computations are done using floating points or  $\varepsilon_1 = 10^{-14}$  for double precision numbers. In both cases, with these numerical values, the collapsing effect disappears and the invariant measure of any component is the Lebesgue measure [12] as we show below. In the case of computation using floating points, starting from most initial condition, it is possible to find a Mega-Periodic orbit (i.e. with period equal to 1,320,752). When computations are done with double precision number it is not possible to find any periodic orbit, up to  $n = 5 \times 10^{11}$  iterations. In [12] the computations have been performed on a Dell computer with a Pentium IV microprocessor using a Borland C compiler computing with ordinary (IEEE-754) double precision numbers.

When  $\varepsilon_1$  converges towards 0, the iterates of each component  $x_n$  and  $y_n$  of equation (5) converge to the Lebesgue measure (Fig. 1).

### 1.2.2. $p$ -coupled symmetric tent map

More generally, the coupling of  $p$  maps takes the form

$$X_{n+1} = F(X_n) = A \cdot (\underline{f}(X_n)), \quad (6)$$

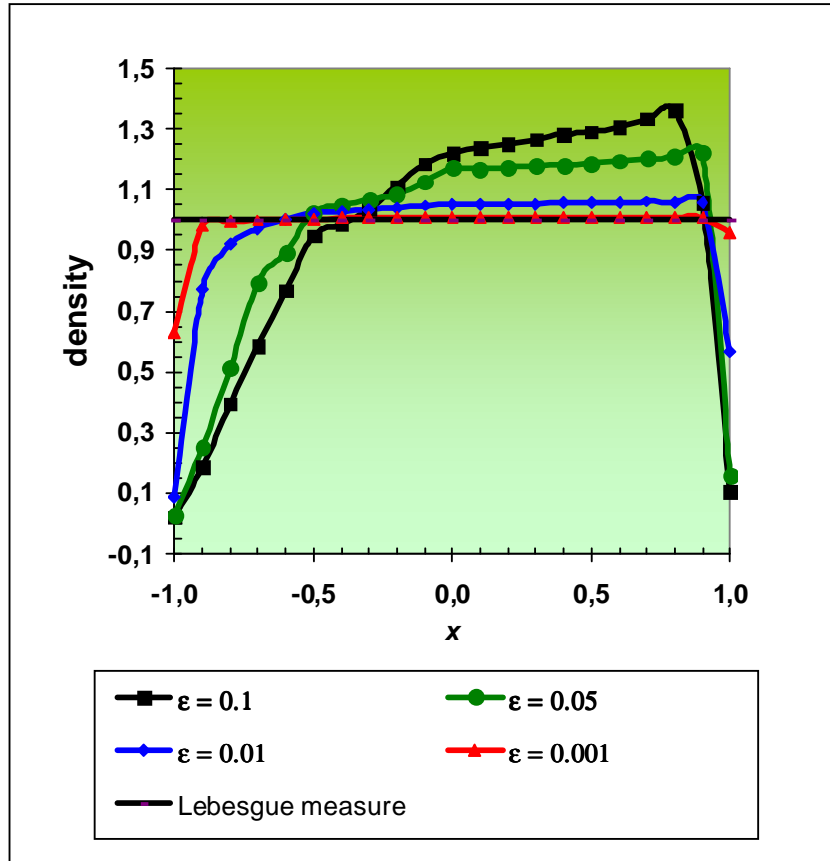
where

$$\underline{f}(X_n) = \begin{pmatrix} f(x_n^1) \\ \vdots \\ f(x_n^p) \end{pmatrix}, \quad X_n = \begin{pmatrix} x_n^1 \\ \vdots \\ x_n^p \end{pmatrix}, \quad (7)$$

and

$$A = \begin{pmatrix} \varepsilon_{1,1} = I - \sum_{j=2}^{j=p} \varepsilon_{1,j} & \varepsilon_{1,2} & \cdots & \varepsilon_{1,p-1} & \varepsilon_{1,p} \\ \varepsilon_{2,1} & \varepsilon_{2,2} = I - \sum_{j=1, j \neq 2}^{j=p} \varepsilon_{2,j} & \cdots & \varepsilon_{2,p-1} & \varepsilon_{2,p} \\ \vdots & \ddots & & \vdots & \vdots \\ \vdots & & & \vdots & \vdots \\ \varepsilon_{p,1} & \cdots & \cdots & \varepsilon_{p,p-1} & \varepsilon_{p,p} = I - \sum_{j=1}^{j=p-1} \varepsilon_{p,j} \end{pmatrix}, \quad (8)$$

with  $\varepsilon_{i,i} = I - \sum_{j=1, j \neq i}^{j=p} \varepsilon_{i,j}$  on the diagonal (the matrix  $A$  is always a stochastic matrix iff the coupling constants verify  $\varepsilon_{i,j} > 0$  for every  $i$  and  $j$ ).



**Figure 1.** Density of iterates of 2-coupled symmetric tent maps, double precision,  $N_{disc} = 10^5$ ,  $\varepsilon_2 = 2\varepsilon_1$ ,  $\varepsilon_1 = 10^{-1}$  to  $10^{-3}$ ,  $N_{iter} = 10^8$ , initial values  $x_0 = 0.330$ ,  $y_0 = 0.3387564$ .

It is noteworthy that these families of very weakly coupled maps are more powerful than the usual formulas used to generate chaotic sequences, mainly because only additions and multiplications are used in the computation process, no division being required. Moreover the computations are done using floating point or double precision numbers, allowing the use of the powerful Floating Point Unit (FPU) of the modern microprocessors. In addition, a large

part of the computations can be parallelized taking advantage of the multicore microprocessors which appear on the market of laptop computers.

Moreover, a determining property of such coupled map is the high number of parameters used ( $p \times (p-1)$  for  $p$  coupled equations) which allows to choose it as cipher-keys, when used in chaos based cryptographic algorithms, due to the high sensitivity to the parameters values [16]. It at be shown also, that using control theory techniques, synchronization of two systems (6), with  $p = 2$  or  $3$ , can be reached via exact (dead-beat) or asymptotic observers [8].

### 1.2.3. Computation of approximated invariant measure

In order to assess numerical computations more accurately, we define an approximation  $P_{M,N}(x)$  of the invariant measure also called the probability distribution function linked to the  $I$ -dimensional map  $f$ , when computed with floating numbers (or numbers in double precision). For this aim we consider a regular partition of  $M$  small intervals (boxes)  $r_i$  of  $J$  defined by

$$s_i = -I + \frac{2i}{M}, \quad i = 0, M \quad (9)$$

$$r_i = [s_i, s_{i+1}[ , \quad i = 0, M-2 \text{ and } \quad r_{M-1} = [s_{M-1}, I] \quad (10)$$

the length of each box is equal to  $\frac{2}{M}$  and the  $r_i$  intervals form a partition of the interval  $J$

$$J = \bigcup_0^{M-1} r_i \quad (11)$$

All iterates  $f^{(n)}(x)$  belonging to these boxes are collected, after a transient regime of  $Q$  iterations decided *a priori*, (i.e. the first  $Q$  iterates are neglected). Once the computation of  $N + Q$  iterates is completed, the relative number of iterates with respect to  $N/M$  in each box  $r_i$  represents the value  $P_N(s_i)$ . The approximated  $P_N(x)$  defined in this article is then a step function, with  $M$  steps. As  $M$  may vary, we define

$$P_{M,N}(s_i) = \frac{M}{N} (\#r_i) \quad (12)$$

where  $\#r_i$  is the number of iterates belonging to the interval  $r_i$ .  $P_{M,N}(x)$  is normalized to 2 on the interval  $J$ .

$$P_{M,N}(x) = P_{M,N}(s_i), \quad \forall x \in r_i \quad (13)$$

In the case of  $p$ -coupled maps, we are more interested by the distribution of each component  $(x^1, x^2, x_2^1, \dots, x^p)$  of  $X$  rather than the distribution of the variable  $X$  itself in  $J^p$ . We then consider the approximated probability distribution function  $P_{M,N}(x^j)$  associated to one among several components of  $F(X)$  defined by (6), which are one-dimensional maps. In this paper we use equally  $N_{disc}$  for  $M$  and  $N_{iter}$  for  $N$ , when they are more explicit.

The discrepancies  $E_1$  (in norm  $L_1$ ),  $E_2$  (in norm  $L_2$ ) and  $E_\infty$  (in norm  $L_\infty$ ) between  $P_{N_{disc}, N_{iter}}(x^j)$  and the Lebesgue measure, which is the invariant measure associated to the symmetric tent map, are defined by

$$E_{1, N_{disc}, N_{iter}}(x^j) = \left\| P_{N_{disc}, N_{iter}}(x^j) - 1 \right\|_{L_1} \quad (14)$$

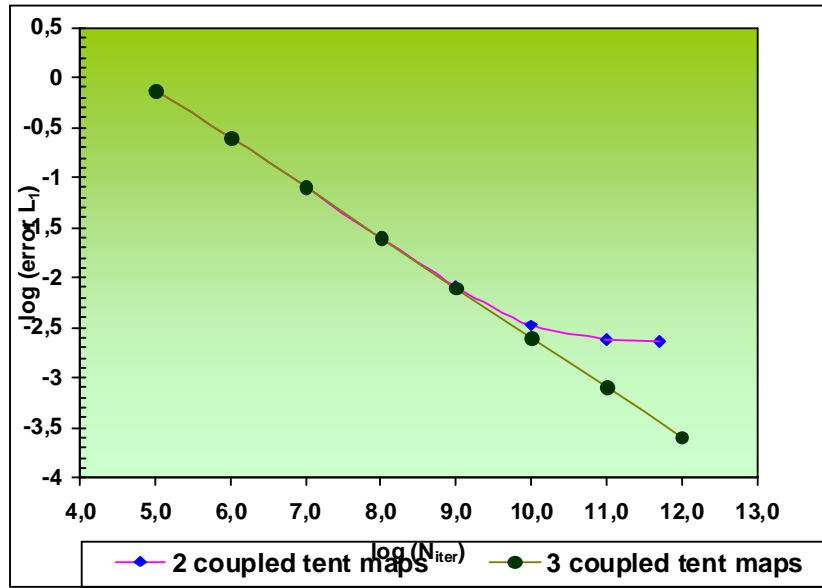
$$E_{2,N_{disc},N_{iter}}(x^j) = \left\| P_{N_{disc},N_{iter}}(x^j) - 1 \right\|_{L_2} \quad (15)$$

$$E_{\infty,N_{disc},N_{iter}}(x^j) = \left\| P_{N_{disc},N_{iter}}(x^j) - 1 \right\|_{L_\infty} \quad (16)$$

As previously said, Fig. 1 shows the convergence of the density of iterates of the components of 2-coupled symmetric tent maps to the Lebesgue measure when  $\varepsilon_1$  converges towards 0. Moreover, for a fixed value of  $N_{disc}$  when the number  $N_{iter}$  increases, the discrepancy between  $P_{N_{disc},N_{iter}}(x^j)$  and the Lebesgue measure is expected to converge towards 0, except if there exist periodic orbits of finite length lower than  $N_{iter}$  which capture the iterates. In this case whatsoever the value of  $N_{iter}$  is, the approximated distribution function converges to the distribution function of the periodic orbit, if it is unique, or to the average of the distribution functions of the periodic orbits observed, if not.

Figure 2 shows the errors  $E_{1,N_{disc},N_{iter}}(x^1)$  versus the number of iterates of the approximated distribution functions, with respect to the first variable  $x^1$ , for 2 and 3-coupled symmetric tent map. Same results are obtained for the other variables  $x^2$  or  $x^3$ .

The 3-coupled symmetric tent maps model considered here with very very small value of  $\varepsilon_1$ , seems a sterling model of generator of chaotic numbers with a uniform distribution of these numbers over the interval  $J$ . It produces very long periodic orbits: Gigaperiodic orbits (i.e. with length of period between  $10^9$  and  $10^{12}$ ) when computed with simple precision numbers, and orbits of unknown length when computed with double precision numbers. However these chaotic sequences are not at all random sequences.



**Figure 2.** Error  $E_{1,N_{disc},N_{iter}}(x^1)$  for 2 and 3-coupled symmetric tent map, double precision,  $N_{disc} = 10^5$ ,  $\varepsilon_1 = 10^{-14}$ ,  $\varepsilon_2 = 2\varepsilon_1$ ,  $N_{iter} = 10^5$  to  $10^{12}$ . Initial values  $x_0^1 = 0.330$ ,  $x_0^2 = 0.3387564$ ,  $x_0^3 = 0.331353429$ .



## 2. THE ROUTE FROM CHAOS TO PSEUDO-RANDOMNESS VIA CHAOTIC UNDERSAMPLING

Chaotic numbers are not pseudo-random numbers because the plot of the couples of any component  $(x_n^l, x_{n+1}^l)$  of iterated points  $(X_n, X_{n+1})$  in the corresponding phase plane reveals the map  $f$  used as one-dimensional dynamical systems to generate them via Eq. (6). Nevertheless, we have recently introduced a family of enhanced Chaotic Pseudo Random Number Generators (CPRNG) in order to compute faster long series of pseudorandom numbers with desktop computer [14, 15]. This family is based on the previous ultra weak coupling which is improved in order to conceal the chaotic genuine function.

In this Section, we describe briefly how works this first process of undersampling, the chaotic one.

### 2.1. Chaotic undersampling

In order to hide  $f$  in the phase space  $(x_n^l, x_{n+1}^l)$  two mechanisms are used. The pivotal idea of the first one mechanism is to sample chaotically the sequence  $(x_0^l, x_1^l, x_2^l, \dots, x_n^l, x_{n+1}^l, \dots)$  generated by the  $l$ -th component  $x^l$ , selecting  $x_n^l$  every time the value  $x_n^m$  of the  $m$ -th component  $x^m$ , is strictly greater (or smaller) than a threshold  $T \in J$ , with  $l \neq m$ , for  $1 \leq l, m \leq p$ .

That is to say to extract the subsequence  $(x_{n_{(0)}}^l, x_{n_{(1)}}^l, x_{n_{(2)}}^l, \dots, x_{n_{(q)}}^l, x_{n_{(q+1)}}^l, \dots)$  denoted here  $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \dots, \overline{x_q}, \overline{x_{q+1}}, \dots)$  of the original one, in the following way given  $1 \leq l, m \leq p, l \neq m$

$$\begin{cases} n_{(-1)} = -1 \\ \overline{x_q} = x_{n_{(q)}}^l, \text{ with } n_{(q)} = \underset{r \in \mathbb{N}}{\text{Min}} \{ r > n_{(q-1)} \mid x_r^m > T \} \end{cases} \quad (17)$$

The sequence  $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \dots, \overline{x_q}, \overline{x_{q+1}}, \dots)$  is then the sequence of chaotic pseudo-random numbers.

The above mathematical formula can be best understood in algorithmic way. The pseudo-code, for computing iterates of (17) corresponding to  $N$  iterates of (6) is:

```

 $X_0 = (x_0^1, x_0^2, \dots, x_0^{p-1}, x_0^p) = \text{seed}$ 
 $n = 0; q = 0$ 
do { while  $n < N$ 
  do { while  $(x_n^m \leq T)$  compute  $(x_n^1, x_n^2, \dots, x_n^{p-1}, x_n^p); n ++$  }
  compute  $(x_n^1, x_n^2, \dots, x_n^{p-1}, x_n^p);$  then  $n(q) = n; \overline{x_q} = x_{n(q)}^l; n ++; q ++$  }
```

This chaotic sampling is possible due to the independence of each component of the iterated points  $X_n$  versus the others [13].

**Remark 2.1.** Albeit the number  $NSampl_{iter}^l$  of pseudo-random numbers  $\overline{x_q}$  corresponding to the computation of  $N$  iterates is not known *a priori*, considering that the selecting process is

again linked to the uniform distribution of the iterates of the tent map on  $J$ , this number is equivalent to  $\frac{2N}{1-T}$ .

## 2.2. Chaotic mixing

A second mechanism can improve the unpredictability of the pseudo-random sequence generated as above, using synergistically all the components of the vector  $X_n$ , instead of two. Given  $p - 1$  thresholds

$$T_1 < T_2 < \dots < T_{p-1} \in J \quad (18)$$

and the corresponding partition  $J_0 = [-1, T_1]$ ,  $J_1 = ]T_1, T_2[$ ,  $J_k = ]T_k, T_{k+1}[$  for  $1 < k < p-1$  and  $J_{p-1} = ]T_{p-1}, 1[$ , with

$$J = \bigcup_{k=0}^{p-1} J_k \quad (19)$$

(note that this partition of  $J$  is different from the regular previous one (11) used for the approximated distribution function).

The second simple mechanism is based on the chaotic undersampling combined with a chaotic mixing of the  $p-1$  sequences

$$(x_0^1, x_1^1, x_2^1, \dots, x_n^1, x_{n+1}^1, \dots), (x_0^2, x_1^2, x_2^2, \dots, x_n^2, x_{n+1}^2, \dots), \dots, (x_0^{p-1}, x_1^{p-1}, x_2^{p-1}, \dots, x_n^{p-1}, x_{n+1}^{p-1}, \dots)$$

, . . . .

using the last one  $(x_0^p, x_1^p, x_2^p, \dots, x_n^p, x_{n+1}^p, \dots)$  in order to distribute the iterated points with respect to this given partition, defining the subsequence  $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \dots, \overline{x_q}, \overline{x_{q+1}}, \dots)$  (in pseudo-code) by

$$\begin{aligned} X_0 &= (x_0^1, x_0^2, \dots, x_0^{p-1}, x_0^p) = \text{seed} \\ n &= 0; q = 0 \\ \text{do } \{ &\text{while } n < N \\ &\text{do } \{ \text{while } (x_n^p \in J_0) \text{ compute } (x_n^1, x_n^2, \dots, x_n^{p-1}, x_n^p); n++ \} \\ &\text{compute } (x_n^1, x_n^2, \dots, x_n^{p-1}, x_n^p) \\ &\text{let } k \text{ be such that } x_n^p \in J_k; \text{ then } n(q) = n; \overline{x_q} = x_{n(q)}^k; n++; q++ \} \end{aligned}$$

**Remark 2.2.** In this case also,  $NSampl_{iter}$  is not known *a priori*, however, considering that the selecting process is linked to the uniform distribution of the iterates of the tent map on  $J$ ,

$$\text{one has } NSampl_{iter} \approx \frac{2N}{1-T_1}.$$

**Remark 2.3.** This second mechanism is more or less linked to the whitening process [28, 29].

**Remark 2.4.** Actually, one can choose any of the components in order to sample and mix the sequence, not only the last one.

### 2.3. Enhanced chaotic undersampling

One can eventually improve the CPRG previously introduced with respect to the infinity norm instead of the  $L_1$  or  $L_2$  norms because the  $L_\infty$  norm is more sensitive than the others to reveal the concealed  $f$  [14]. For this purpose we introduce a second kind of threshold  $T' \in \mathbb{N}$ , together with  $T_1, \dots, T_{p-1} \in J$  such that the subsequence  $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \dots, \overline{x_q}, \overline{x_{q+1}}, \dots)$  is defined (in pseudo-code) by

```

 $X_0 = (x_0^1, x_0^2, \dots, x_0^{p-1}, x_0^p) = seed$ 
 $n = 0, q = 0$ 
do { while  $n < N$ 
  do { while ( $n \leq n_{(q-1)} + T'$  and  $x_n^p \in J_0$ )
    compute  $(x_n^1, x_n^2, \dots, x_n^{p-1}, x_n^p); n++$  }
  compute  $(x_n^1, x_n^2, \dots, x_n^{p-1}, x_n^p)$ 
  let  $k$  be such that  $x_n^p \in J_k$ 
  then  $n(q) = n; \overline{x_q} = x_{n(q)}^k; n++; q++$  }
```

**Remark 2.5.** In this case also,  $NSampl_{iter}$  is not known *a priori*, it is more complicated to give an equivalent to it. However, considering that the selecting process is linked to the uniform distribution of the iterates of the tent map on  $J$ , and to the second threshold  $T'$ , it comes that

$$NSampl_{iter} \leq \text{Min} \left\{ \frac{2N}{1-T_1}, \frac{N}{T'} \right\}.$$

**Remark 2.6.** The second kind of threshold  $T'$  can also be used with only the chaotic sampling, without the chaotic mixing.

### 2.4. A window of emergence of randomness

In [15, 16] we show that if one considers the errors  $E_{1, N_{disc}, N_{iter}}(x) = \|P_{N_{disc}, N_{iter}}(x) - 1\|_{L_1}$ ,  $E_{2, N_{disc}, N_{iter}}(x) = \|P_{N_{disc}, N_{iter}}(x) - 1\|_{L_2}$  and  $E_{\infty, N_{disc}, N_{iter}}(x) = \|P_{N_{disc}, N_{iter}}(x) - 1\|_{L_\infty}$  together with the correlated distribution functions which assess the independence of each component of the iterated points  $X_n$  versus the others, a window of emergence comes clearly into sight for the values  $\varepsilon_i \in [10^{-15}, 10^{-7}]$ , in the case  $p = 4$  and  $\varepsilon_{i,j} = \varepsilon_i = i\varepsilon_1$ . We have also performed NIST test developed by the National Institute of Standards and Technology [23], in order to check carefully the random nature of such numbers [7].

Then there is a route from chaos to randomness using the process of chaotic undersampling.

## 3 GEOMETRIC UNDERSAMPLING

The previous route from chaos to randomness uses chaotic undersampling. It is possible to couple in another way  $p$  tent maps on the torus  $J^p = [-1, 1]^p \subset \mathbb{R}^p$ , which can directly provide random numbers, without sampling or mixing, provided  $p$  is large enough, although it is possible to combine these processes with it. After reviewing this ring coupling in high

dimension, we introduce the new geometric undersampling in order to obtain randomness with small values of  $p$  (for example  $p = 2$ ).

### 3.1. Pseudo-random numbers generated by ring coupled mapping

Consider the mapping defined on the  $p$ -dimensional torus  $M_p : J^p \rightarrow J^p$

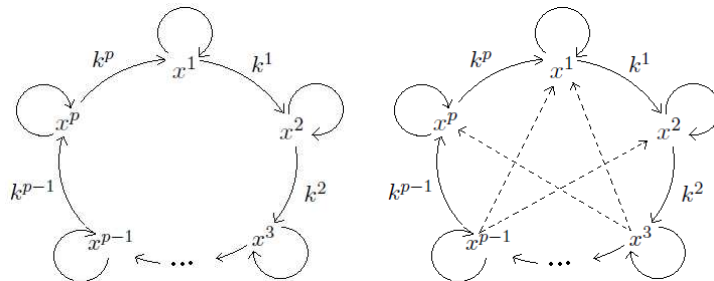
$$M_p \begin{pmatrix} x_n^1 \\ x_n^2 \\ \vdots \\ x_n^p \end{pmatrix} = \begin{pmatrix} x_{n+1}^1 \\ x_{n+1}^2 \\ \vdots \\ x_{n+1}^p \end{pmatrix} = \begin{pmatrix} 1 - 2|x_n^1| + k_1 \times x_n^2 \\ 1 - 2|x_n^2| + k_2 \times x_n^3 \\ \vdots \\ 1 - 2|x_n^m| + k_m \times x_n^1 \end{pmatrix} \quad (20)$$

with the parameters  $k_i \in \{-1, 1\}$ . In order to confine every variable  $x_n^j$  on  $J^p$  we do, for every iteration, the transform

$$\begin{cases} \text{if } (x_{n+1}^j < -1) & \text{add } 2 \\ \text{if } (x_{n+1}^j > 1) & \text{subtract } 2 \end{cases} \quad (21)$$

The particularity of this coupling is that each variable  $x^j$  is coupled only with itself and  $x^{j+1}$ , as displayed on Fig. 3a. At first glance, in order to enrich the random properties of the map, it could seem interesting to add supplementary cross couplings between these variables, as shown on Fig. 3b. However in this case a cross-coupling is inappropriate because it would increase the determinism and therefore deteriorate the statistical properties which we are looking for.

To evaluate the random properties of these generators, the set of NIST tests have been used again.



**Figure 3a** (left). Ring coupling between the variables  $x^j$ .

**Figure 3b** (right). Cross coupling between the variables  $x^j$ .

The random properties validations of both a 4-dimensional system and a 10-dimensional one have been carried out [5]. For this purpose, the chaotic carrier output needs to be quantised and binarized (0 and 1) in order to be validated as being random using NIST tests. Therefore, different methods of binarization (converting real signals to binary ones) have been implemented and compared.

A first 1-bit binarization has been applied to the system (21) output, defined as  $y_n = x_n^j$  with  $j \in \llbracket 1, p \rrbracket$

$$\begin{cases} \text{if } (y_n \geq 0) & b = 1 \\ \text{else} & b = 0 \end{cases} \quad (22)$$

The results showed to be highly sensitive to the type of binarization. Eventually, after testing several different methods, a 32-bit binarization has been chosen as being the most suitable solution. Because the system is confined to the  $p$ -dimensional torus  $J^p$ , 31 bits are assigned to represent the decimal part, and 1 bit to the sign. To illustrate the results, the NIST tests for the 4-dimensional system with parameters  $k_i \in (-1)^{i+1}$  are shown in Fig. 4. The chosen conditions are: length of the original sequence =  $10^8$  bits, length of bit string =  $10^6$  bits, quantity of bit strings = 100. The output of the system has been arbitrary chosen as being:  $y = x_n^4$ .

Furthermore, as the results show their independence from the initial conditions, every bit string in this test is the resulting sequence of a different randomly chosen initial condition. The criterion for a successful test is that the  $p$ -value has to be superior to the significance level (0.01 for this case). For the present model, all tests were successful thus the sequences can be accepted as being random.

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES												
generator is <data/lozi_10_positif.txt>												
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	5	13	9	9	12	6	19	8	11	0.102526	96/100	Frequency
11	16	9	10	10	10	14	6	8	6	0.437274	99/100	BlockFrequency
11	5	8	11	10	5	11	11	13	15	0.419021	97/100	CumulativeSums
8	6	17	10	10	6	7	11	15	10	0.213309	97/100	CumulativeSums
5	8	17	15	6	8	6	14	10	11	0.075719	99/100	Runs
11	11	10	13	9	5	8	8	15	10	0.637119	99/100	LongestRun
6	8	17	14	10	8	9	15	7	6	0.122325	99/100	Rank
9	10	9	13	10	10	9	8	12	10	0.991468	99/100	FFT
14	15	8	10	14	10	11	9	4	5	0.191687	98/100	NonOverlappingTemplate
10	8	11	9	9	13	7	12	10	11	0.964295	99/100	OverlappingTemplate
13	16	6	8	7	10	13	10	8	9	0.455937	100/100	Universal
9	10	12	8	10	11	5	14	11	10	0.816537	97/100	ApproximateEntropy
6	5	6	5	9	11	5	6	8	5	0.637119	65/66	RandomExcursions
3	5	6	7	10	10	9	6	4	6	0.407091	65/66	RandomExcursionsvariant
3	8	8	12	12	9	13	8	13	14	0.319084	100/100	Serial
4	3	12	18	12	8	8	14	9	12	0.028817	100/100	LinearComplexity

**Figure 4.** Example of NIST Test for  $k^i = (-1)^{i+1}$ ,  $i = 1, 4$ , each sequence of components satisfies the NIST test for randomness.

### 3.2. Ring coupling of 2-dimensional symmetric tent map

Although the system (20) is a good PRNG when  $p \geq 4$ , in lower dimension 2 and 3, the chaotic numbers are not equidistributed on the torus (see Fig. 5).

In order to improve the ring coupling mechanism in low dimension, we introduce now a very new type of undersampling based on geometric nature of the invariant measure. We present this very new mechanism which allows the emergence of randomness from chaos, in the simplest case, the 2-dimensional ring mapping  $M_2$  on the square  $J^2$ , with  $k^1 = k^2 = 1$ .

Let  $M_2$  be defined by

$$\begin{cases} x_{n+1}^1 = I - 2|x_n^1| + x_n^2 \\ x_{n+1}^2 = I - 2|x_n^2| + x_n^1 \end{cases} \quad (23)$$

$$\text{with } \begin{cases} \text{if } (x_{n+1}^j < -1) \text{ add } 2 \\ \text{if } (x_{n+1}^j > 1) \text{ subtract } 2 \end{cases} \quad (24)$$

### 3.2.1. Critical lines

Figure 5 shows the distribution of the iterates of system (23) (the transient of the first  $10^6$  iterations has been cut off). It can be observed that the attractor contains regions where the point density is lower, and two lozenge-like holes. It is possible to define critical lines which form a partition of the square  $J^2$ . The critical lines CL [19] are singularities of dimension 1 and represent an important tool for the analysis of noninvertible maps. The holes on Fig. 5 are completely delimited by segments of the critical lines  $CL_1^{A1}, CL_1^{B4}, CL_1^{C2}, CL_1^{D4}$ , and  $CL_1^{A2}, CL_1^{B3}, CL_1^{C1}, CL_1^{D3}$ , defined below.

The critical lines separate regions of the phase space with different number of preimages (backward iterates). In the case of piecewise linear maps, they are the first iterates of the lines of discontinuity  $CL_{-1}$  of the system.

For the two dimensional system (23), there are four groups of critical lines  $CL$  with preimages  $CL_{-1}$  given by

**Critical lines A:**  $CL_{-1}^A : x^1 = 0$

$$\text{and } \begin{cases} CL_1^{A1} : x^2 = -2x^1 - 1 & \text{if } x^2 > 0 \\ CL_1^{A2} : x^2 = 2x^1 - 1 & \text{if } x^2 < 0 \end{cases} \quad (25)$$

**Critical lines B:**  $CL_{-1}^B : x^1 = -1$

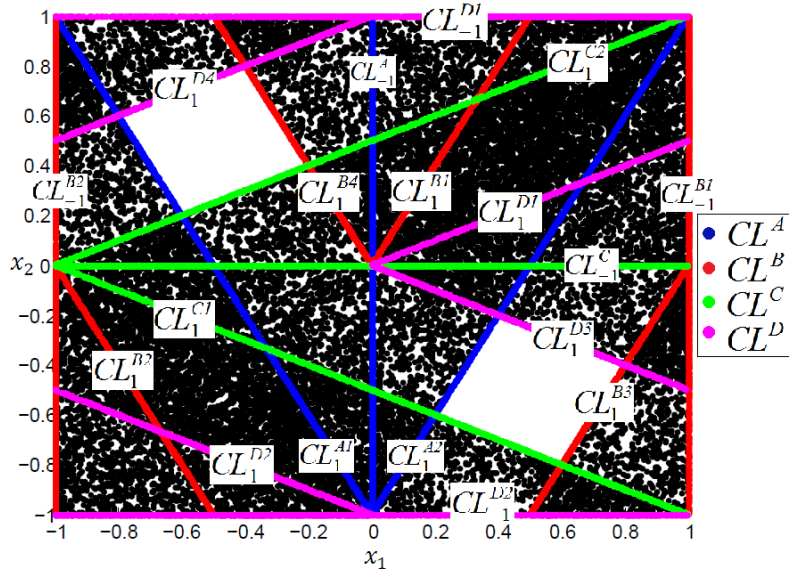
$$\text{and } \begin{cases} CL_1^{B1} : x^2 = 2x^1 & \text{if } x^2 < 0, x^1 \in [0, 0.5] \\ CL_1^{B2} : x^2 = -2x^1 - 2 & \text{if } x^2 > 0, x^1 \in [-1, -0.5] \\ CL_1^{B3} : x^2 = 2x^1 - 2 & \text{if } x^2 < 0, x^1 \in [0.5, 1] \\ CL_1^{B4} : x^2 = -2x^1 & \text{if } x^2 > 0, x^1 \in [-0.5, 0] \end{cases} \quad (26)$$

**Critical lines C:**  $CL_{-1}^C : x^2 = 0$

$$\text{and } \begin{cases} CL_1^{C1} : x^2 = -\frac{1}{2}(x^1 + 1) & \text{if } x^1 > 0 \\ CL_1^{C2} : x^2 = \frac{1}{2}(x^1 + 1) & \text{if } x^1 < 0 \end{cases} \quad (27)$$

**Critical lines D:**  $CL_{-1}^D : x^2 = -1$

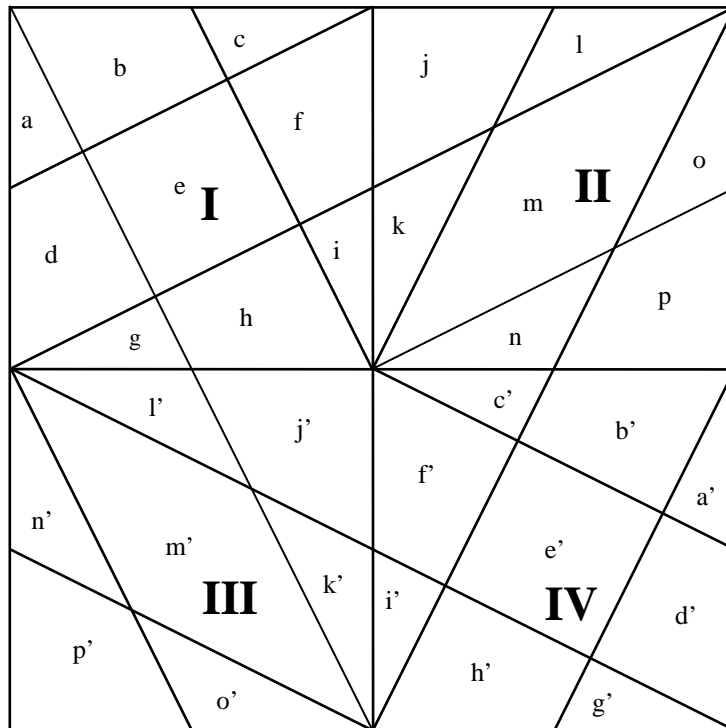
$$\text{and } \begin{cases} CL_1^{D1} : x^2 = \frac{x^1}{2} & \text{if } x^1 < 0, x^2 \in [0, 0.5] \\ CL_1^{D2} : x^2 = -\frac{x^1}{2} - 1 & \text{if } x^1 > 0, x^2 \in [-1, -0.5] \\ CL_1^{D3} : x^2 = -\frac{x^1}{2} & \text{if } x^1 > 0, x^2 \in [-0.5, 0] \\ CL_1^{D4} : x^2 = \frac{x^1}{2} + 1 & \text{if } x^1 < 0, x^2 \in [0.5, 1] \end{cases} \quad (28)$$



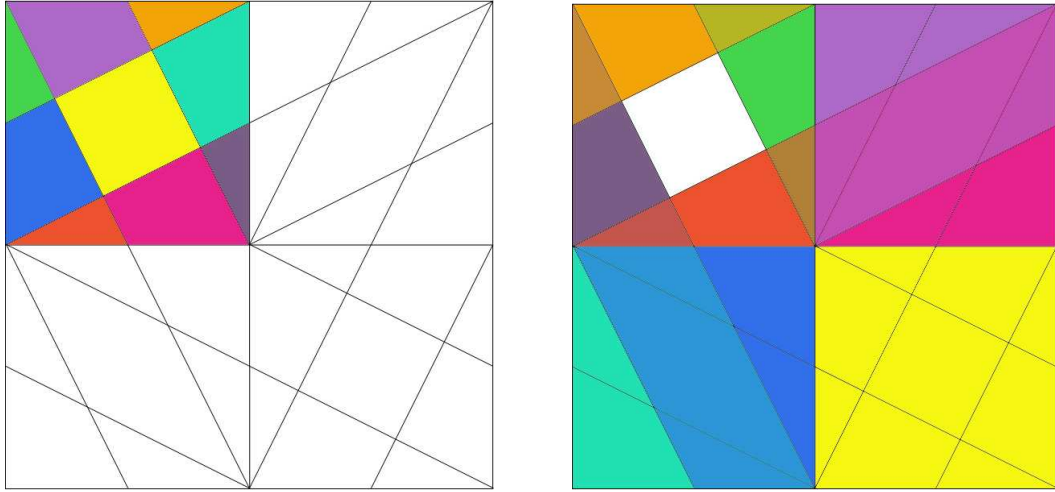
**Figure 5.** Critical lines of the map  $M_2$  on the torus  $J^2$  (a square) [26].

### 3.2.2. Markov partition of the square

Our aim is first to use the partition defined by these critical lines in order to do a cell-to-cell analysis and, by the means of a Markov process, to compute explicitly the invariant measure of iterates associated to system (23). Figure 6 displays the 32 sub-regions of the square  $J^2$ , labelled from a to p and a' to p'. For clarity of the presentation, we have labelled from (I) to (IV), the four quadrants of  $J^2$ .



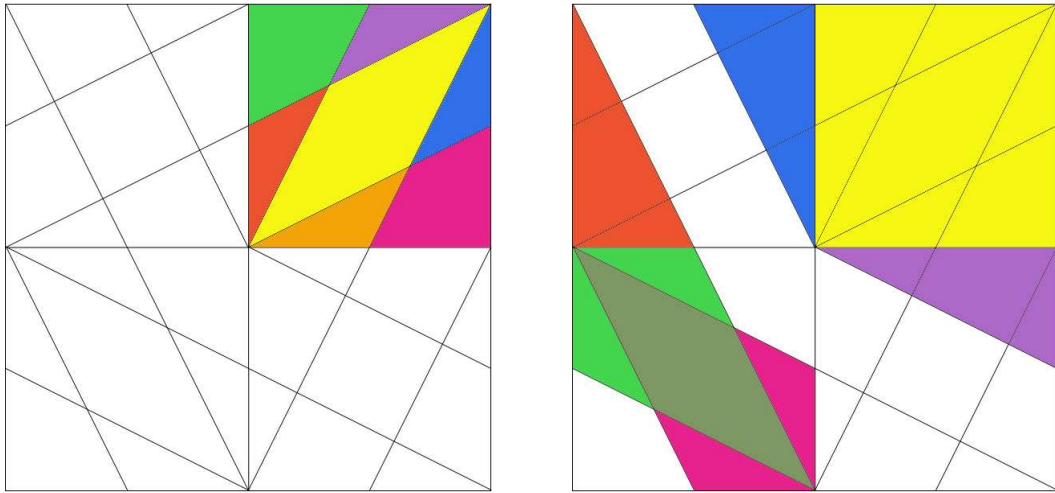
**Figure 6.** The 32 sub-regions for a partition of the square  $J^2$ .



**Figure 7 a (left).** The nine regions  $a$  to  $i$  of quadrant (I)

**Figure 7 b (right).** The images  $M_2(I)$  of the nine regions of quadrant (I).

Straightforward computation shows that the images of each region, by the mapping  $M_2$ , is either one, two or three regions of the same partition of the square  $J^2$ . Figures 7a and 7b display the images of the regions embedded in the first quadrant (I). Figures 8a and 8b display the images of the regions embedded in the second quadrant (II). The colour is the same for every region and its corresponding image, except when two regions are mapped on the same region, in this case there is a mix of colours on the common part of the image.



**Figure 8 a (left).** The seven regions  $j$  to  $p$  of quadrant (II)

**Figure 8 b (right).** The images  $M_2(II)$  of the seven regions of quadrant (II).

The overall correspondence between regions of the partition and their image is given by the Markov matrix  $M_a$  which is displayed on Table 1. The computation of the coefficients of this matrix, which are rational numbers, is based on the ratios of surfaces of bounded regions.



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	a'	b'	c'	d'	e'	f'	g'	h'	i'	j'	k'	l'	m'	n'	o'	p'
a				x		y			x																							
b										t	s	s	u		s																	
c	x	y	x																													
d																																
e																	p	q	p	q	r	q	p	q	p							
f																																
g																																
h																																
i	x			y					x																							
j																																
k	x			y					x																							
l																	x	y	x													
m																																
n																																
o			x			y			x																							
p																																
a'																	x			y												
b'																																
c'																																
d'																																
e'	p	q	p	q	r	q	p	q	p																							
f'																																
g'																																
h'																																
i'																																
j'																																
k'																																
l'																																
m'																																
n'																																
o'	x	y	x																													
p'																																

**Table 1.** Markov Matrix  $M_a$ .

In order to display the  $32 \times 32$  matrix  $M_a$  on one page, we have labelled the coefficients using letters which are not related to the names of the regions.

$$\left\{ \begin{array}{l} o = \frac{1}{12}; p = \frac{1}{20}; q = \frac{3}{20}; r = \frac{4}{20}; \\ s = \frac{1}{9}; t = \frac{2}{9}; u = \frac{4}{9}; v = \frac{1}{6}; \\ w = \frac{1}{3}; x = \frac{1}{5}; y = \frac{3}{5}; z = \frac{2}{3}; \end{array} \right. \quad (29)$$

### 3.2.3. Exact computation of invariant measure associated to $M_2$

With the help of Markov matrix  $M_a$ , it is straightforward to compute explicitly the invariant measure associated to  $M_2$ . For every region on Figure 6, we define a quantity of initial points called  $Q^i$ ,  $i = 1, 32$  uniformly scattered on it, and we compute its surface  $S_i$ . We normalise both quantities to  $\sum_i Q^i = |Q| = 4$ , and  $\sum_i S_i = |S| = 4$ . Hence it is possible to define the density of iterates on each region.

$$d^i = \frac{Q^i}{S_i} \quad (30)$$

Let  $Q = \begin{pmatrix} Q^1 \\ \vdots \\ Q^{32} \end{pmatrix}$  and  $D = \begin{pmatrix} d^1 \\ \vdots \\ d^{32} \end{pmatrix}$  the vectors of quantities and densities obtained applying

(30) to every region. Then starting from an arbitrary initial repartition of points on  $J^2$ , say

$$Q_0 = \begin{pmatrix} Q_0^1 \\ \vdots \\ Q_0^{32} \end{pmatrix}, \text{ and applying repeatedly the equation}$$

$$Q_{m+1} = M_a^t Q_m \quad (31)$$

The sequence of vectors  $\{Q_m\}_{m \in \mathbb{N}}$  converges to a limit vector  $\bar{Q}$  which satisfies

$$\bar{Q} = M_a^t \bar{Q} \quad (32)$$

and gives the invariant measure, the density of which is the vector  $\bar{D}$ , using (30).

### Numerical results:

Starting from  $Q_0 = \begin{pmatrix} Q_0^1 \\ \vdots \\ Q_0^{32} \end{pmatrix} = \begin{pmatrix} 1/8 \\ \vdots \\ 1/8 \end{pmatrix}$ ,  $\bar{Q}$  is obtained rapidly, as

$$Q_{500} = \begin{pmatrix} Q_{500}^1 \\ Q_{500}^2 \\ Q_{500}^3 \\ Q_{500}^4 \\ \vdots \\ Q_{500}^{29} \\ Q_{500}^{30} \\ Q_{500}^{31} \\ Q_{500}^{32} \end{pmatrix} = \begin{pmatrix} 1/14 \\ 3/28 \\ 1/14 \\ 3/28 \\ \vdots \\ 4/7 \\ 3/28 \\ 3/28 \\ 1/7 \end{pmatrix} = \bar{Q}, \text{ which gives using (30), } D_{500} = \begin{pmatrix} d_{500}^1 \\ d_{500}^2 \\ d_{500}^3 \\ d_{500}^4 \\ \vdots \\ d_{500}^{29} \\ d_{500}^{30} \\ d_{500}^{31} \\ d_{500}^{32} \end{pmatrix} = \begin{pmatrix} 10/7 \\ 5/7 \\ 10/7 \\ 5/7 \\ \vdots \\ 12/7 \\ 9/7 \\ 9/7 \\ 6/7 \end{pmatrix} = \bar{D}.$$

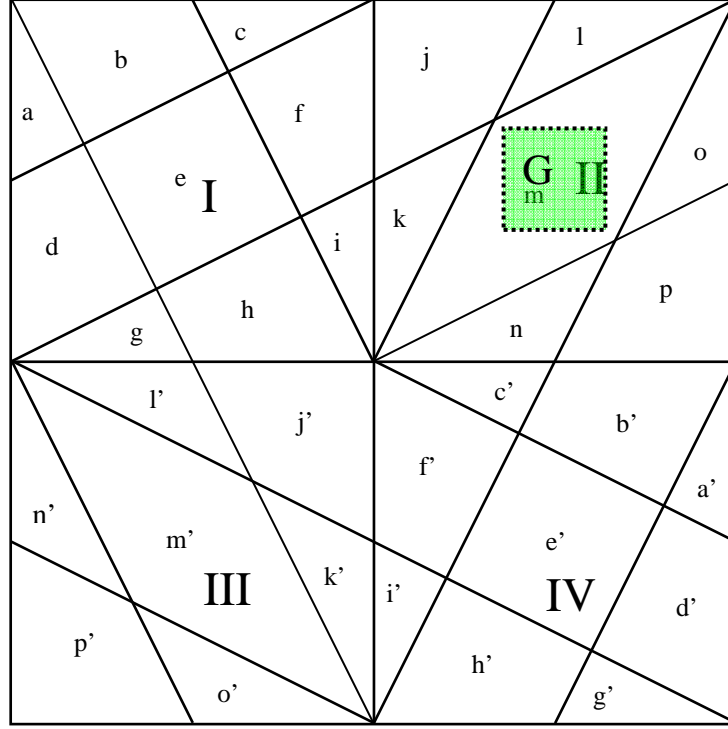
**Remark 3.1.** Computing directly this density, iterating (23) up to  $10^{11}$  iterates leads to the same result.

## 3.3 Geometric undersampling

The exact computation of the density  $\bar{D}$  of the invariant measure shows that this density is constant on each region. The geometric undersampling process consists in magnifying a square  $G$  included in one region (as for example the square  $G = [0.36, 0.64] \times [0.36, 0.64]$  included in region  $m$  on Fig. 9), up to the size of the square  $J^2$ .

### 3.3.1. Algorithm of geometric undersampling

Let  $G = [x_l^1, x_r^1] \times [x_l^2, x_r^2]$  the square in which we will undersample the iterate of (23) and,  $x_{mean}^1 = \frac{x_l^1 + x_r^1}{2}$ ,  $x_{mean}^2 = \frac{x_l^2 + x_r^2}{2}$ . In algorithmic form, the pseudo-code, to geometric undersample  $N$  iterates of (23) is



**Figure 9.** The square  $G = [0.36, 0.64] \times [0.36, 0.64]$  in which iterates of (23) are geometrically undersampled.

$$X_0 = (x_0^1, x_0^2) = \text{seed}$$

$$n = 0;$$

do { while  $n < N$  compute  $(x_n^1, x_n^2)$ ; if  $(x_n^1, x_n^2) \in G$  then

$$\bar{x}_q^1 = 2 \left[ \frac{x_n^1 - x_{\text{mean}}^1}{x_r^1 - x_l^1} \right], \quad \bar{x}_q^2 = 2 \left[ \frac{x_n^2 - x_{\text{mean}}^2}{x_r^2 - x_l^2} \right]; q = q + 1; n ++ \}$$

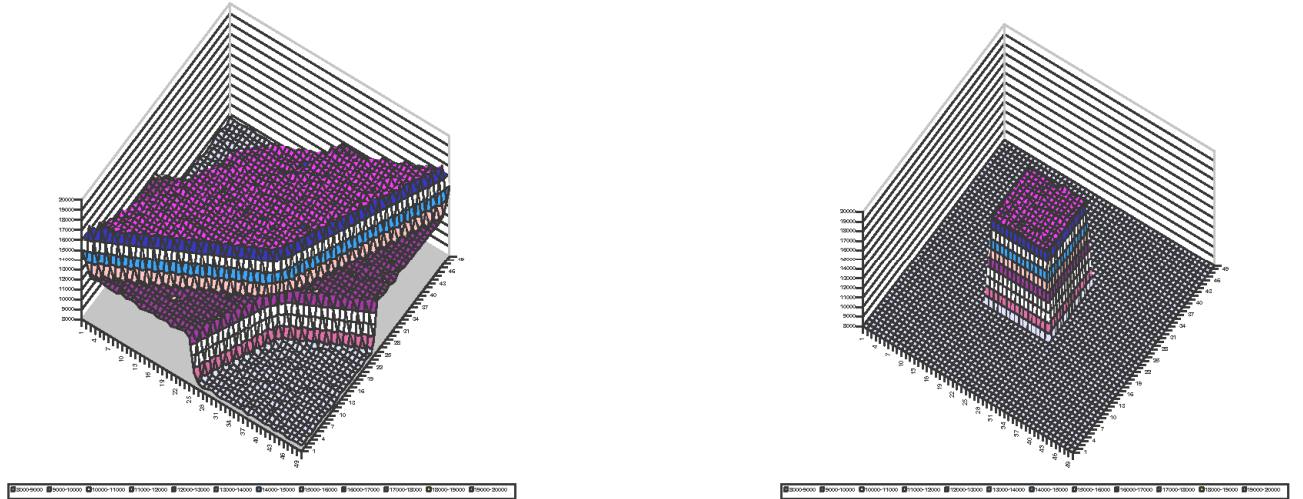
**Remark 3.2.** In this case, the undersampling process provides two streams of pseudo-random numbers.

**Remark 3.3.** In this case,  $NSampl_{iter}$  the number of geometrically undersampled iterates is not known *a priori*, however, considering that the selecting process is linked to the uniform distribution of the iterates of the tent map on  $J^2$ , one has  $NSampl_{iter} \approx \frac{(x_r^1 - x_l^1)^2}{4} \times d^m$ , where  $d^m$  is the density of the measure in region m.

### 3.3.2. Numerical tests

We have applied this process in the case of the square G of Fig. 9, with  $N = 10^{12}$  which gives  $NSampl_{iter} \approx 3.35 \times 10^{10}$ . Figure 10a displays the densities of the seven regions j, k, l, m, n, o, p of quadrant (II) which are equal to

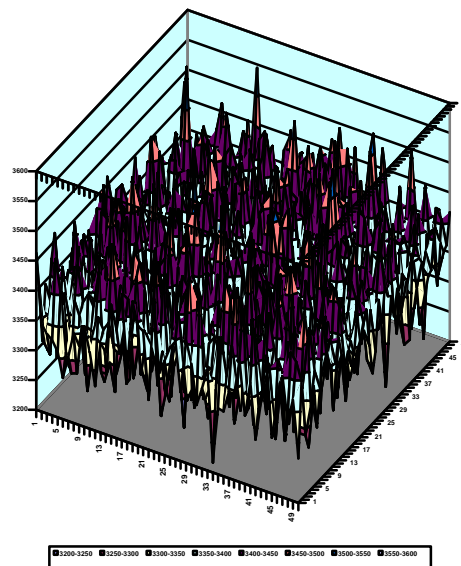
$$\begin{cases} \bar{d}^j = \frac{6}{7}; \bar{d}^k = \frac{9}{7}; \bar{d}^l = \frac{9}{7}; \bar{d}^m = \frac{12}{7}; \\ \bar{d}^n = \frac{9}{7}; \bar{d}^o = \frac{9}{7}; \bar{d}^p = \frac{6}{7}; \end{cases}$$



**Figure 10 a (left).** densities of the seven regions  $j, k, l, m, n, o, p$  of quadrant (II)

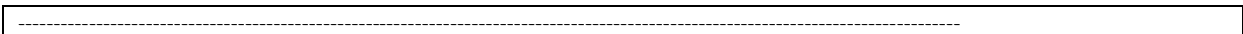
**Figure 10 b (right).** Uniform density of iterates in the square  $G = [0.36, 0.64] \times [0.36, 0.64]$  of quadrant (II).

Figure 10b shows the uniform density of iterates in the square  $G = [0.36, 0.64] \times [0.36, 0.64]$  of quadrant (II). On Fig. 11 the square is magnified up to the size of the square  $J^2$ . The vertical scale is fitted near the invariant Lebesgue measure.



**Figure 11.** Uniform density of iterates of the square  $G = [0.36, 0.64] \times [0.36, 0.64]$  magnified to the square  $J^2$ .

We have also used NIST test to confirm the random property of the geometrical undersampling process. They are all successful (Fig. 12).



RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES												
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
12	9	7	9	12	11	8	11	13	8	0.924076	99/100	Frequency
1	4	3	4	7	5	9	16	16	35	0.000000 *	100/100	BlockFrequency
9	9	10	12	11	8	9	10	10	12	0.996335	99/100	CumulativeSums
10	9	12	12	9	7	10	10	9	12	0.983453	99/100	CumulativeSums
11	12	11	8	12	7	12	6	10	11	0.883171	99/100	Runs
9	9	13	8	9	8	17	8	10	9	0.595549	100/100	LongestRun
6	11	11	11	9	8	8	14	9	13	0.798139	100/100	Rank
15	10	7	8	8	8	15	16	7	6	0.153763	97/100	FFT
12	9	10	13	9	11	7	15	4	10	0.474986	98/100	NonOverlappingTemplate
12	6	10	6	13	6	8	8	17	14	0.145326	99/100	OverlappingTemplate
18	12	13	11	9	10	5	8	9	5	0.145326	99/100	Universal
11	8	12	11	11	14	8	10	7	8	0.883171	99/100	ApproximateEntropy
3	5	6	9	4	3	7	5	6	11	0.145326	59/59	RandomExcursions
7	6	6	2	6	7	6	7	4	8	0.637119	59/59	RandomExcursions
2	6	4	5	5	6	10	6	7	8	0.334538	59/59	RandomExcursionsVariant
8	15	13	12	9	12	13	5	9	4	0.224821	98/100	Serial
9	9	6	13	13	7	12	9	10	12	0.798139	99/100	LinearComplexity

The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 96 for a sample size = 100 binary sequences.

**Figure 12.** Geometrical undersampling: each sequence of components satisfies the NIST test for randomness.

## 4 NOISE-RESISTING CIPHERING

As a first example we propose a novel noise-resisting ciphering based on a large number of uncorrelated chaotic sequences. These cogenerated sequences are actually used in several steps of the ciphering process. Noisy transmission conditions are considered, with realistic assumptions. The efficiency of the proposed method for ciphering and deciphering is illustrated through numerical simulations based on ten coupled chaotic sequences [5]. It can be also adapted to geometric undersampling, provided this undersampling is done in dimension 4.

In this section we detail the noise-resisting ciphered transmission principle, consisting of two steps: the ciphering process and the transmission process (see Figures 13, 14). Both resort to the coupled chaotic pseudo-random generated sequences.

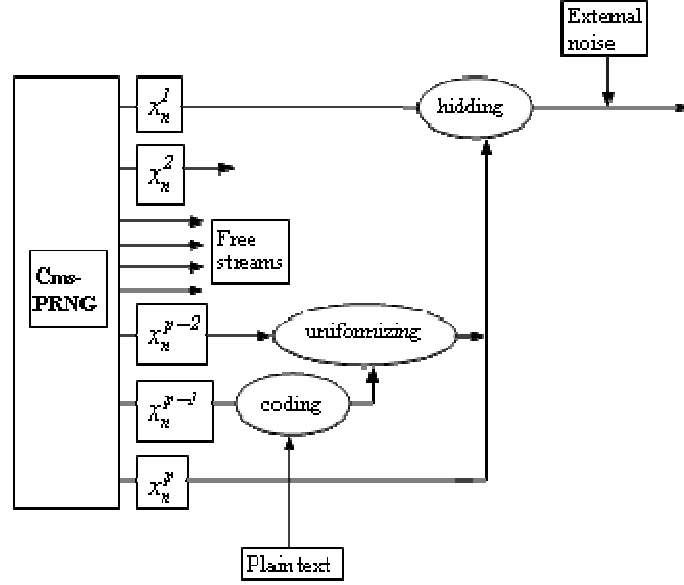
### 4.1. Ciphering principle

We begin with some notations that will be used in the sequel. The *plain text* is denoted  $(t_k)_{k=1,\dots,N}$ : the letters  $t_k$ , for  $k=1,\dots,N$  belong to the alphabet  $\{l_1,\dots,l_\pi\}$  composed of  $\pi$  letters.

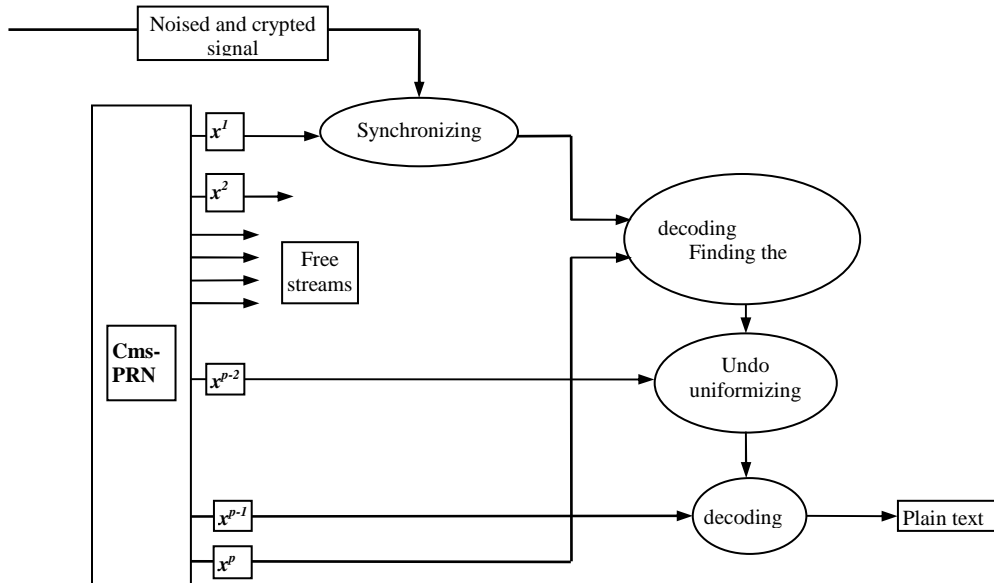
The *ciphered text* is a sequence of real numbers, denoted  $y_k$ ,  $k=1,\dots,N$  and each  $y_k$  belongs to the interval  $J=[-1,1]\subset\mathbb{R}$ . The transmitted signal (at the transmitter side) is denoted  $s_n$  while the received signal is  $\hat{s}_n$  (at the receiver side).

In this paper we consider noisy transmission conditions, which means that  $\hat{s}_n = s_n + \alpha_n$ , where  $\alpha_n > 0$  denotes an unknown additive noise at time  $n$ . We make the following classical assumption: the additive noise is bounded by a known bound  $K$ , which means that

$$\|s_n - \hat{s}_n\| = \alpha_n \leq K, \quad \forall n \geq 0 \quad (33)$$



**Figure 13.** General scheme of the ciphering and the ciphered transmission principle (coding and transmitting).



**Figure 14.** General scheme of the ciphering and the ciphered transmission principle (receiving and decoding).

We detail first how to transform each letter of the plain text  $t_k$  into a real number  $y_k \in [-1,1]$ , with an original noise-resisting method. In a second step, the sequence  $y_k$  will be transformed to obtain a uniform distribution on the interval  $[-1,1]$ .

- Define a partition as follows:

$$[-1,1] = \bigcup_{m=1,\pi} I_m \quad (34)$$

with  $a_m, b_m$  the bounds of the interval each interval  $I_m$ , i.e.:  $I_m = [a_m, b_m]$ .

In fact, owing to the presence of additive noise, not all real numbers inside  $I_m$  can be selected, one must avoid an interval of length  $K$  at each side of the interval  $I_m$ . Therefore some smaller intervals need to be defined.

- Define a sub-interval  $I'_m$  included in the corresponding interval  $I_m$  such that:

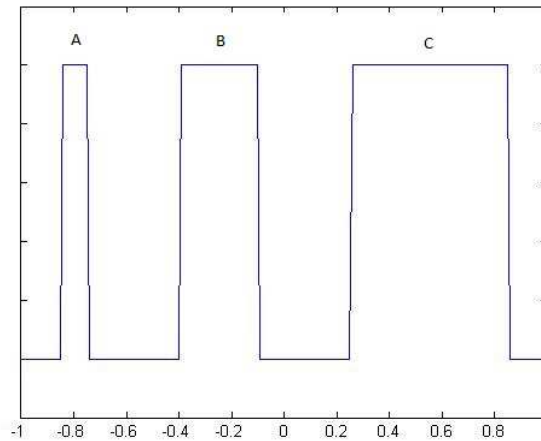
$$I'_m = [a'_m, b'_m] \subset I_m \quad (35)$$

and

$$[a'_m - K, b'_m + K] \subset I_m \quad (36)$$

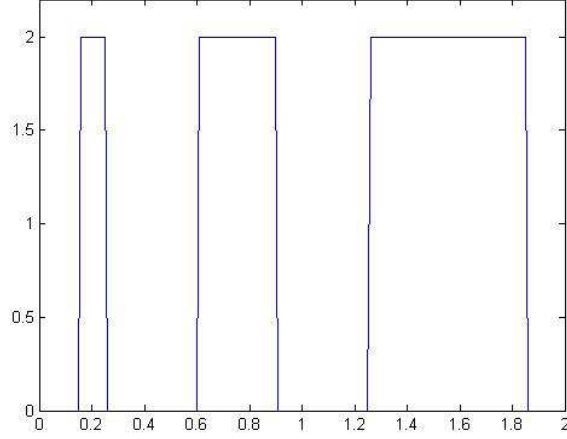
where we recall that  $K$  is the upper bound on the noise, see (33).

Then the coding consists in randomly (i.e. with another pseudo-random sequence generated by (20):  $x_n^{p-1}$ , or the geometric undersampling in dimension 4) choosing for each letter  $t_k$  of the plain text, a real number  $y_k$  inside the interval  $I'_m$  (and not  $I_m$ ) if  $t_k = l_m$ . Each interval  $I'_m$  corresponds to a letter  $l_m$ , for  $m = 1, \dots, \pi$ . Remark that each letter has a frequency of apparition in the plain text, depending on the initial language. Therefore one must carefully choose the length of each interval  $I'_m$  in proportion to the corresponding frequency of the letter  $l_m$ . An illustration is given by Fig. 15 for an alphabet with three letters: the letter  $A$  has a frequency of 10%, the letter  $B$  has a frequency of 30% and the letter  $C$  of 60%.



**Figure 15.** Repartition of an alphabet of three letters.

- Once this first step of the coding is achieved, one has to ensure that the ciphered text has a random-like distribution inside  $[-1, 1]$ . With the aforementioned coding alone, this property cannot be ensured, as it can be seen in Fig. 16.



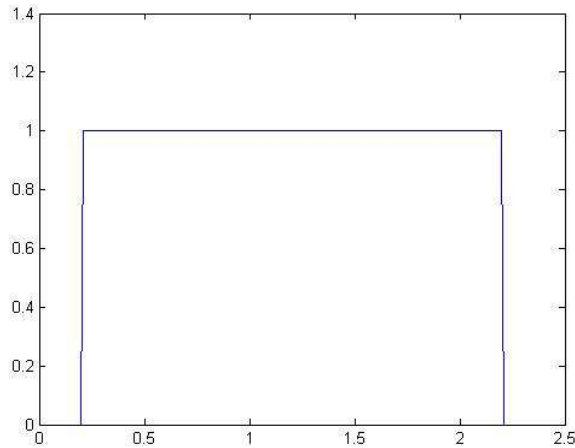
**Figure 16.** Signal to be transmitted without transformation.

Since one needs to leave some holes at the edges of the intervals  $I_m$  to resist the additive noise, the transmitted signal cannot have a random-like repartition. So we propose to transform the ciphered data  $y_k$  before transmitting it.

For all steps  $n \in \mathbb{N}$  such that an encrypted letter is transmitted, we propose to transmit not directly  $y_n$  but:

$$\tilde{y}_n = \begin{cases} y_n + x_n^{p-2} & \text{if } y_n + x_n^{p-2} \in [-1, 1] \\ y_n + x_n^{p-2} + 2 & \text{if } y_n + x_n^{p-2} < -1 \\ y_n + x_n^{p-2} - 2 & \text{if } y_n + x_n^{p-2} > 1 \end{cases} \quad (37)$$

For simplicity of presentation, in the sequel,  $y_n$  will denote  $\tilde{y}_n$ , the ciphered message to transmit.



**Figure 17.** Signal to be transmitted after transformation.

Then the obtained signal to transmit has the desired uniform repartition, as illustrated by Fig. 17.



## 4.2. Transmission principle

We present now how to transmit the ciphered text using substitution method in a new pseudo-random sequence. The transmitted signal is denoted  $s_n$ .

The ciphered text  $y_k$ , defined by (37), is not directly transmitted, it is chaotically hidden in a chaotic carrier signal, as it is explained below.

The ciphering makes use of two coupled chaotic sequences:  $x_n^l$  is used as chaotic carrier, while  $x_n^p$  is used to select the substitution times.

$$s_n = \begin{cases} x_n^l & \text{if } x_n^p < T \\ y_{n(k)} & \text{if } x_n^p \geq T \end{cases} \quad (38)$$

where  $T$  is a predefined threshold. For example, as the  $x_n^p$  are equally distributed on the interval  $[-1,1]$ , if one chooses  $T = 0.8$ , one ciphered letter will be transmitted in average each 10 element of the sequence  $x_n^l$ . If one chooses  $T = 0.98$ , one element over 100 is replaced by a letter.

We do not detail here the sequence  $k(n)$ , it is easily understandable that  $k(n)$  increase of +1 each time  $s_n = y_{k(n)}$  in order to transmit each element of the ciphered sequence  $y_k$ .

## 4.3. Decoding principle

At the receiver end, suppose that the same PRNG defined by (20) is available. The transmitter and the authorized receiver have fixed the same parameters and same initial values, therefore the ciphering is a symmetrical one.

According to the substitution principle defined by (38) and the hypothesis (33) on the additive noise, the received signal can be expressed as:

$$\hat{s}_n = x_n^l + \alpha_n \text{ or } y_{k(n)} + \alpha_n \quad (39)$$

Since the initial conditions of the chaotic pseudo-random number generator (20) are assumed to be public, the receiver exactly knows when  $x_n^p$  is smaller or larger than the threshold  $T$ , so the receiver is able to reconstruct the sequence  $(y_{k(n)} + \alpha_n)$  i.e. the sequence  $y_q + \beta_q$  where  $\beta_q = \alpha_n$  for  $q = k(n)$ .

Since  $\beta_q < K$ , there exists  $m \in \{1, 2, \dots, \pi\}$  such that  $\hat{s}_n \in I_m$ .

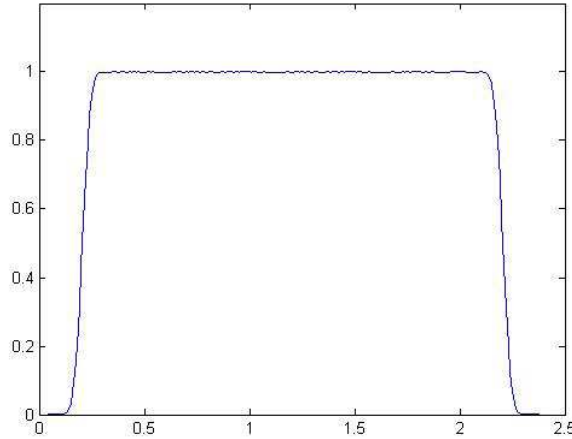
The receiver, also, exactly knows the value of  $x_n^{p-2}$  and deduces from the rules (37) the value  $y_q$ . Then the knowledge of the correspondence between the interval  $I_m$  and the letter  $l_m$  enables the receiver to retrieve the initial message.

## 4.4. Numerical illustration

Now we summarize the main steps of the proposed algorithm:

- 1) Choose the secret parameters  $k_i = 1$  or  $k_i = -1$ , for  $i \in \{1, 2, \dots, p\}$ .

- 2) Define the initial conditions shared by the transmitter and the receiver.
- 3) Iterate the PRNG (20) with the previous initial conditions, both at the transmitter and the receiver side.
- 4) Apply the ciphering and transmission principle as detailed before.



**Figure 18.** Received noisy signal.

The Fig. 18 shows the noisy signal at the receiver side (recall that the transmitted signal is given by Fig. 17). Notice that the Figs. 2 to 4 represent our simulations with  $10^9$  iterations.

## CONCLUSION

We have proposed a new mechanism of undersampling of chaotic number obtained by the ring coupling mechanism of one-dimensional maps. In the case of 2 coupled maps this mechanism allows the building of a PRNG which passes all NIST Test.

This new geometric undersampling is very effective for generating 2 parallel streams of pseudo-random numbers, as we have shown, computing carefully their properties, up to sequences of  $10^{12}$  consecutives iterates of (23) which provides more than  $3.35 \times 10^{10}$  random numbers in very short time. In a forthcoming paper we will test both 3- and 4-dimensional cases.

In addition we have proposed a novel method of noise-resisting ciphering. The originality lies in the use of a chaotic pseudo-random number generator: several co-generated sequences can be used at different steps of the ciphering process, since they present the strong property of being uncorrelated. Each letter of the initial alphabet of the plain text is encoded as a subinterval of  $[-1,1]$ . The bounds of each interval are defined in function of the known bound of the additive noise. A pseudo-random sequence is used to enhance the complexity of the ciphering. The transmission consists of a substitution technique inside a chaotic carrier, depending on another cogenerated sequence. The efficiency of the proposed scheme is illustrated on some numerical simulations.

Cryptography is a wide field of research, in which the brilliant formulae of Srinivasan Ramanujan have been largely used. May be, it will be possible, in a near future, to link such formulae with chaos in the domain of emergent randomness.

## REFERENCES

- [1] E. Araujo, L. dos S. Coelho, "Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system", *Applied Soft Computing*, 8, 1354–1364, (2008).
- [2] M. R. K. Ariffin, M. S. M. Noorani, "Modified Baptista type chaotic cryptosystem via matrix secret key", *Phys. Lett. A*, 372, 5427-5430 (2008).
- [3] M. S. Baptista, "Cryptography with chaos", *Phys. Lett. A*, 240, 50-54 (1998).
- [4] E., Cherrier, & R., Lozi, "Noise-resisting ciphering based on a chaotic multi-stream pseudo-random number generator", in *Proceeding of ICITST, 2011*, Abu-Dhabi, AUE, 11-14 December, 2011.
- [5] A. Espinel, I. Taralova, I., & R. Lozi, "New alternate ring-coupled map for multi-random number generation", *Journal of Nonlinear Systems and Applications*, Vol. 4, n°1, 64-65 (2013).
- [6] P. Gora, A. Boyarsky, Md. S. Islam, W. Bahsoun, "Absolutely continuous invariant measures that cannot be observed experimentally", *SIAM J. Appl. Dyn. Syst.*, 5:1, 84-90 (electronic), (2006).
- [7] S. Hénaff, I. Taralova, I., & R. Lozi, "Dynamical analysis of a new statistically highly performant deterministic function for chaotic signals generation", in *Proceeding of Physcon 2009*, Catania, Italy, 1-4 september, *IPACS open Access Electronic Library*.
- [8] S. Hénaff, I. Taralova, I., & R. Lozi, "Exact and Asymptotic Synchronization of a new weakly coupled maps system", *Journal of Nonlinear Systems and Applications*, 1, 87-95, (2010).
- [9] M. Hénon, "A Two-dimensional mapping with a strange attractor", *Commun. Math. Phys.*, 50, 69-77, (1976).
- [10] O. E. Lanford III, "Some informal remarks on the orbit structure of discrete approximations to chaotic maps". *Experimental Mathematics*, Vol. 7, 4, 317-324, (1998).
- [11] E. N. Lorenz, "Deterministic nonperiodic flow", *J. Atmospheric Science*, 20, 130-141, (1963).
- [12] R. Lozi, "Giga-Periodic Orbits for Weakly Coupled Tent and Logistic Discretized Maps". *Modern Mathematical Models, Methods and Algorithms for Real World Systems*, A. H. Siddiqi, I. S. Duff and O. Christensen (Editors), Anamaya Publishers, New Delhi, India pp. 80-124, 2006.
- [13] R. Lozi, "New Enhanced Chaotic Number Generators", *Indian Journal of Industrial and Applied Mathematics*, vol.1, No.1, pp.1-23, 2008.
- [14] R. Lozi, "Chaotic Pseudo Random Number Generators via Ultra Weak Coupling of Chaotic Maps and Double Threshold Sampling Sequences", In *proceedings of: ICCSA 2009 The 3rd International Conference on Complex Systems and Applications*, University of Le Havre, France, June 29- July 02, 2009.
- [15] R. Lozi, "Complexity leads to randomness in chaotic systems", *Mathematics in Science and Technology: Mathematical Methods, Models and Algorithms in Science and Technology*. Proceedings of the Satellite Conference of ICM, 15-17 August 2010 Delhi, India (A.H. Siddiqi, R.C. Singh, P. Manchanda Editors), World Scientific Publisher, Singapore (2011) pp. 93-125.
- [16] R. Lozi, "Emergence of randomness from chaos", *International Journal of Bifurcation and Chaos*, Vo. 22, No. 2 (2012) 1250021-1/1250021-15.
- [17] R. Lozi, "chaotic mathematical circuitry", In *Chaos, CNN, Memristors and Beyond*, A. Adamatzky, G. Chen (Eds.), World Scientific Publishing, pp. 307-323, (2013).
- [18] R. Lozi, "Can we trust in numerical computations of chaotic solutions of dynamical systems", In *Topology and Dynamics of Chaos*, Ch. Letellier, R. Gilmore (Eds.), World Scientific Series in Nonlinear Science Series A, Vol. 84, pp. 63-98, (2013).
- [19] C. Mira, L. Gardini, A. Barugola, J.-C. Cathala, *Chaotic dynamics in two-dimensional noninvertible maps*, World Scientific Series on Nonlinear Science, Series A - Vol. 20, (1996).
- [20] K. Palmer, *Shadowing in Dynamical Systems: Theory and Applications*, Kluwer Academic Publications (2000).
- [21] S. Y. Pilyugin, "Shadowing in Dynamical Systems", *Lecture Notes in Math.*, Springer, 1706, (1999).
- [22] M. Pluhacek, V. Budikova, R. Senkerik, Z. Oplatkova, I. Zelinka, "Extended Initial Study on the Performance of Enhanced PSO Algorithm with Lozi Chaotic Map", *Advances in Intelligent Systems and Computing*, 1, Volume 192, Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems, pp. 167 - 177, (2012).
- [23] A. Rukhin, et al, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", NIST (2001), <http://csrc.nist.gov/rng/>
- [24] J. C. Sprott, *Chaos and Time-Series Analysis*, Oxford University Press, Oxford, UK, (2003).
- [25] T. W. Tang, A. Allison and D. Abbott, "Parrondo's games with chaotic switching". *Proc. SPIE Noise in Complex Systems and Stochastic Dynamics II*, Vol. 5471, 520-530, Maspalomas, Gran Canaria, Spain, 26-28 May 2004.
- [26] I. Taralova, A. Espinel, & R. Lozi, "Dynamical and statistical analysis of a new Lozi function for random numbers generation", in *Proceeding of Physcon 2011*, León, Spain, 5-8 september, *IPACS open Access Electronic Library*.

- [27] I., Taralova, S. El Hassad, & R., Lozi, “Chaotic Generator Synthesis: Dynamical and Statistical Analysis”, in *Proceeding of ISTP 2012*, London, UK, 10-11 December, 2012.
- [28] J. Viega, “Practical random number generation in software”, in *Proceedings of 19th Annual Annual Computer Security Applications Conference*, 129-140, (2003).
- [29] J. Viega, M. Messier, *Secure programming cook book for C and C++* (O’Reilly, Sebastopol, CA), (2003).