Noise-resisting ciphering based on a chaotic multi-stream pseudo-random number generator

René Lozi, Estelle Cherrier

Noise-resisting ciphering based on a chaotic multi-stream pseudo-random number generator

René Lozi
Laboratory J.A. Dieudonné
UMR CNRS 6621 University of Nice Sophia-Antipolis
France
Email: lozi@unice.fr

Estelle Cherrier
GREYC Laboratory
UMR 6072 CNRS-University of Caen-ENSICAEN
France
Email: Estelle.Cherrier@ensicaen.fr

***Abstract*—In this paper we propose a novel noise-resisting ciphering method resorting to a chaotic multi-stream pseudo-random number generator (denoted Cms-PRNG) detailed in the reference [13]. This Cms-PRNG co-generates an arbitrarily large number of uncorrelated chaotic sequences. These co-generated 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 a Cms-PRNG involving ten coupled chaotic sequences.**

I. Introduction

A. Chaos and cryptography: a brief state of the art

Besides conventional cryptographic methods, relying on number theory, chaos-based cryptography has aroused a great attention for a few years. Indeed, chaotic systems seem to exhibit promising properties, among which one can mention: long-term unpredictability, noise-like behavior, spread spectrum... Two main classes can be distinguished in chaos-based cryptographic methods: those based on continuous-time chaos and those based on discrete-time chaos.
In this paper, we will focus on cryptography with discrete-time chaos, owing to a less computational complexity and no need for synchronization. The first references dealing with chaos cryptography are [16], which proposes a stream cipher based on a one-dimensional chaotic map and [5], in which the secret key is chosen as some parameter of the tent map.

Generally the principle of discrete chaos based cryptography is directly inspired from that of classical cryptography, in the sense that either pseudo-random numbers are generated from a discrete chaotic system to perform stream ciphering or the cipher message follows the chaotic orbit whose initial condition is deduced from the plain message in block ciphering. Then chaos based cryptographic schemes are used in hashing techniques, authentication process, or key-exchange protocols... among other applications.

Most of the methods aimed at designing chaotic cryptosystems resort to pseudo-random number generators (which will be denoted chaotic PRNG in the sequel). Beside standard PRNG (see [9] and the references therein for a thorough study), chaotic systems can be used to generate sequences of pseudo-random numbers. Some works study the random properties exhibited by these chaotic PRNG, while other references discuss about their involvement in cryptography. We mention in this section a rapid survey about chaotic PRNG, among the numerous papers available in the literature.
In the companion references [18], [19], an analysis is performed of the application of a chaotic piecewise linear one dimensional map as random number generator. The parameter values for which the RNG behaves as a Markov information source are analytically and practically studied.
The reference [10] studies the cryptographic properties of a new pseudo-random bit generator based on the coupled map lattice: spatiotemporal chaos is dealt with, stemming from partial differential equation or coupled ordinary differential equations. It means that chaotic properties are exploited both in time and in space. A large number of chaotic oscillators are coupled to produce chaotic orbits whose period is too long to be reached in realistic conditions. In the paper [1] a perturbing orbit technique is used to avoid dynamical degradation due to finite state representation. It shows that this process also increases the cycle length. The authors proposes an analysis of the randomness based on system and signal processing tools. The work [6] proposes a chaotic PRNG based on a simple logistic map and on two coupled logistic maps. This CPRNG is used to encrypt binary plaintexts through bitwise XOR operation, as in standard stream ciphers. The reference [2] proposes a review of RNG features and chaotic systems theory. It studies a family of rational order chaotic maps, obtained as the ratio between polynomials. Then some statistical tests are performed through an invariant measure to evaluate the performances of the corresponding chaotic PRNG. In the reference [4], a particular statistical complexity measure (introduced by [15]) is used to quantify the randomness of any PRNG. This measure exhibits the following interesting property: it is null for totally random process. The proposed method is evaluated on standard well-known PRNG as well as PRNG based on Lorenz chaotic system. This measure is improved in [8]. The same authors present in [7] a quantifier to predict whether a given RNG will pass the Marsaglia DIEHARD test suite [14]. The reference [17]

proposes a method to classify randomizing techniques, through a representation based on the statistical properties (see [15], [7]) of chaotic systems. They define randomizing techniques as suitable manipulations of the generated time series in order to improve their statistical properties. They show that PRNG based on very simple chaotic systems may be greatly improved by resorting to symbolic dynamics. Two techniques are tested, called discretization and skipping.

A novel and efficient technique is presented in the papers [11], [12] to randomize chaotic data. This technique relies on chaotic sampling and mixing, through extremely weak coupling of a piecewise linear chaotic map. The proposed method has been thoroughly studied and characterized and the generated numbers possess good random properties. A new generator on the torus has been proposed in [13]: this generator will be used in the present paper to co-generate uncorrelated pseudo-random sequences to design an efficient and noise resisting chaotic cryptosystem by using the technique of chaotic sampling.

B. Problem formulation

This paper deals with the not so widely treated problem of data transmission in a noisy environment, based on a Cms-PRNG.

The originality of this paper is twofold. First a novel ciphering method is proposed aimed at resisting to a noisy transmission channel. The main idea is to establish, between the transmitter and the receiver, a correspondence between the alphabet constituting the plain text and some intervals defining a partition of $[-1,1]$. Some realistic assumption about the noise boundedness allows to restrict the bounds of the aforementioned intervals in order to precisely resist to the effects of the noise. An extra scrambling resorting to a co-generated chaotic sequence enhances the ciphering process. Then a new chaotic substitution method is developed: considering a chaotic carrier, belonging to the set of co-generated and coupled pseudo-random chaotic sequences, the idea is to randomly/chaotically (in fact, this is determined by a second pseudo-random chaotic sequence) replace some elements of the carrier by a ciphered element (a letter here) of the message. At the receiver end, a copy of the Cms-PRNG, with the same parameters (hence we deal with a symmetrical ciphering method) allows to generate the necessary chaotic sequences and therefore to retrieve the initial message.

The paper is organized as follows. In section II the chaotic pseudo-random numbers generator is detailed. Then section III is dedicated to the ciphering and the transmission processes. The decoding principle is summarized. In the last section some numerical illustrations show the efficiency of the proposed noise-resisting chaotic ciphering.

II. CMS-PRNG DESCRIPTION

The design of the proposed noise-resisting chaotic ciphering requires multiple random sequences. For this purpose consider the system with alternate coupled maps confined to the p-dimensional torus $[-1,1]^p$ recently introduced by Lozi and defined by:

$$\left\{\begin{array}{rcl} x^1_{n+1} &=& 1-2|x^1_n| \\ && +k_1\left(\left(1-\sum_{j=3}^{p}\varepsilon_{1,j}\right)x_n^2+\sum_{j=3}^{p}\varepsilon_{1,j}x_n^j\right) \\ \vdots &=& \vdots \\ x^m_{n+1} &=& 1-2|x^m_n| \\ && +k_m\left(\left(1-\sum_{j=1,j\neq m,m+1}^{p}\varepsilon_{m,j}\right)x_n^{m+1}\right. \\ && \left.+\sum_{j=1,j\neq m,m+1}^{p}\varepsilon_{m,j}x_n^j\right) \\ \vdots &=& \vdots \\ x^{p-1}_{n+1} &=& 1-2|x^{p-1}_n| \\ && +k_{p-1}\left(\left(1-\sum_{j=1}^{p-2}\varepsilon_{p-1,j}\right)x_n^p+\sum_{j=1}^{p-2}\varepsilon_{p-1,j}x_n^j\right) \\ x^p_{n+1} &=& 1-2|x^p_n| \\ && +k_p\left(\left(1-\sum_{j=2}^{p-1}\varepsilon_{p,j}\right)x_n^1+\sum_{j=2}^{p-1}\varepsilon_{p,j}x_n^j\right) \end{array}\right. \tag{1}$$

For more clarity, we also give the equation of the Cms-PRNG in dimension 4:

$$\left\{\begin{array}{rcl} x^1_{n+1} &=& 1-2|x^1_n|+k_1\left((1-\varepsilon_{1,3}-\varepsilon_{1,4})x_n^2\right. \\ && \left.+\varepsilon_{1,3}x_n^3+\varepsilon_{1,4}x_n^4\right) \\ x^2_{n+1} &=& 1-2|x^2_n|+k_2\left((1-\varepsilon_{2,4}-\varepsilon_{2,1})x_n^3\right. \\ && \left.+\varepsilon_{2,4}x_n^4+\varepsilon_{2,1}x_n^1\right) \\ x^3_{n+1} &=& 1-2|x^3_n|+k_3\left((1-\varepsilon_{3,1}-\varepsilon_{3,2})x_n^4\right. \\ && \left.+\varepsilon_{3,1}x_n^1+\varepsilon_{3,2}x_n^2\right) \\ x^4_{n+1} &=& 1-2|x^4_n|+k_4\left((1-\varepsilon_{4,2}-\varepsilon_{4,3})x_n^1\right. \\ && \left.+\varepsilon_{4,2}x_n^2+\varepsilon_{4,3}x_n^3\right) \end{array}\right. \tag{2}$$

with the following extra conditions, to stay on the torus, for all $j\in\{1,\ldots,p\}$:

- if $x^j_{n+1}<-1$ then add 2
- if $x^j_{n+1}>1$ then subtract 2

and $k_i=1$ or $k_i=-1$ for $i\in\{1,\ldots,p\}$.

It has been shown in reference [3] that the chaotic map (1) co-generates p (p is arbitrarily chosen) uncorrelated sequences of pseudo-random numbers. This is a key property, that allows to use any generated chaotic sequence at any step of the proposed ciphering process: for security purpose,

the transmitter and the receiver can agree to invert the role of the employed sequences. This property is very useful in the proposed method: the transmitter and the receiver can generate exactly the same pseudo-random sequences (from the same parameters and the same initial conditions, this point is discussed just below), but if an intruder intercept one of these sequences, he/she cannot deduce any information about the other coupled sequences. Therefore the proposed ciphering scheme is intrinsically linked with the inherent properties of the Cms-PRNG.

In practice,it is sufficient to choose $p = 10$ (see [3]) to obtain good statistical properties. Among these p sequences, some will be used in the noise resisting transmission or ciphering process. Owing to their property of being non correlated, several sequences issued from the same chaotic generator can be entangled and transmitted at the same time. Another interest for considering a coupled map such as (1) is that it provides many parameters, some of which will be used as secret keys. In the present work, we choose the parameters k_i, $i = 1, \ldots, p$ as secret keys together with some of the $\varepsilon_{i,j}$, $i, j = 1, \ldots, p$. The only condition we impose is that the $\varepsilon_{i,j}$ must belong to the interval $[10^{-15}, 10^{-5}]$ to ensure good random properties for the generated sequences.

Besides, we choose to keep the initial conditions public and to eliminate the N_0 first iterations precisely to avoid the influence of the initial conditions. This means that the ciphered message will be transmitted inside a pseudo random carrier signal once its N_0 first iterations have been removed. In practice, some tests have been performed, showing that N_0 can be set to about 100 iterations. For security requirements, the value of N_0 can be increased at will.

Remark 1. *To bear the fact that the initial conditions are public, the Cms-PRNG must exhibit a particularly strong sensitivity to its parameters, since some of the parameters only are kept secret. Some numerical tests illustrate this sensitivity, reported in the Appendix, at the end of the paper.*

III. Noise-resisting ciphering

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

A. Ciphering principle

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

The *ciphered text* is a sequence of real numbers, denoted y_k, $k = 1, \ldots, N$ and each y_k belongs to the interval $[-1, 1]$.

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

$$||s_n - \hat{s}_n|| = \alpha_n \leq K, \ \forall n \geq 0 \tag{3}$$

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:

$$[-1, 1] = \bigcup_{m=1,\pi} I_m \tag{4}$$

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:

$$I'_m = [a'_m, b'_m] \subset I_m \tag{5}$$

and

$$[a'_m - K, b'_m + K] \subset I_m \tag{6}$$

where we recall that K is the upper bound on the noise, see (3).
Then the coding consists in randomly (i.e. with another pseudo-random sequence generated by (1): x_n^{p-1}) 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 $m = 1, \ldots, \pi$. 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 figure 1 for an alphabet with three letters: the letter A as a frequency of 10%, the letter B has a frequency of 30% and the letter C of 60%.

- 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 figure 2:

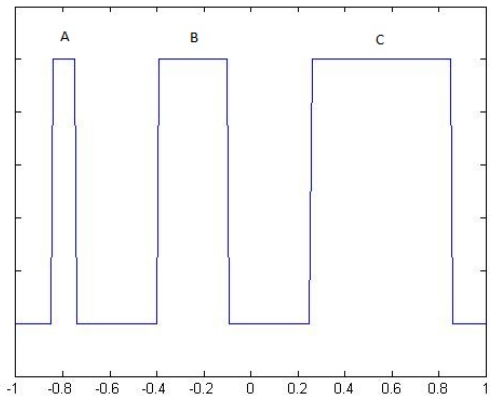

Fig. 1. Repartition of an alphabet of three letters

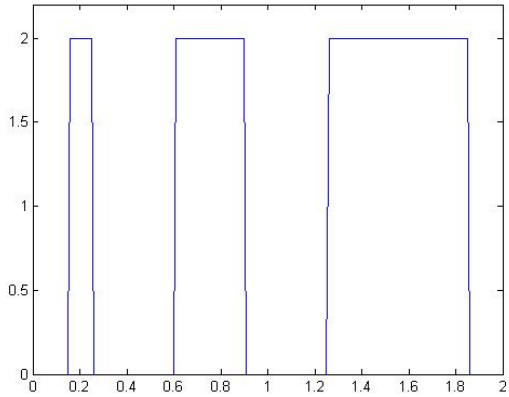

Fig. 2. Signal to be transmitted without transformation

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:

$$\tilde{y}_n = \begin{cases} y_n + x_n^{p-2} & \text{if } y_n + x_n^{p-2} \in [-1,1] \\ y_n + x_n^{p-2} + 2 & \text{if } y_n + x_n^{p-2} < -1 \\ y_n + x_n^{p-2} - 2 & \text{if } y_n + x_n^{p-2} > 1 \end{cases} \tag{7}$$

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

Then the obtained signal to transmit has the desired uniform repartition, as illustrated by figure 3

B. Transmission principle

We present now how to transmit the ciphered text using substitution method in a new pseudo-random chaotic

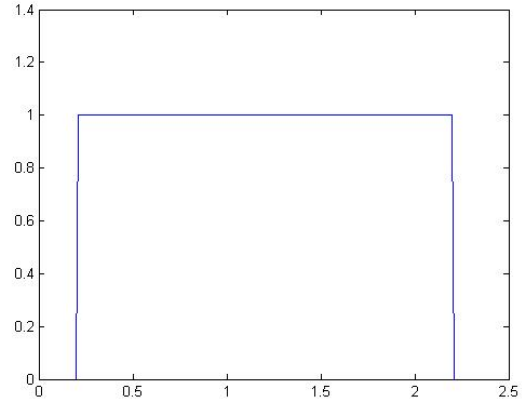

Fig. 3. Signal to be transmitted after transformation

sequence. The transmitted signal is denoted s_n.

The ciphered text y_k, defined by (7), is not directly transmitted, it is chaotically hidden in a chaotic carrier signal, as is explained below.

The ciphering makes use of two coupled chaotic sequences: x_n^1 is used as chaotic carrier, while x_n^p is used to select the substitution times.

$$s_n = \begin{cases} x_n^1 \text{ if } x_n^p < T \\ y_{k(n)} \text{if } x_n^p \geq T \end{cases} \tag{8}$$

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^1. 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.

C. Decoding principle

At the receiver end, suppose that the same Cms-PRNG defined by (1) is available. The transmitter and the authorized receiver have fixed the same parameters, therefore the ciphering is a symmetrical one.

According to the substitution principle defined by (8) and the hypothesis (3) on the additive noise, the received signal can be expressed as:

$$\hat{s}_n = x_n^1 + \alpha_n \text{ or } y_{k(n)} + \alpha_n \tag{9}$$

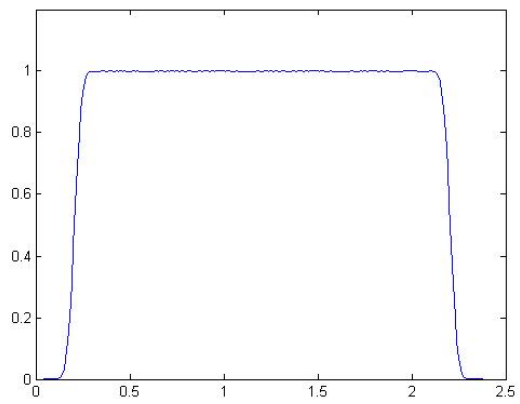

Fig. 4. Received noisy signal

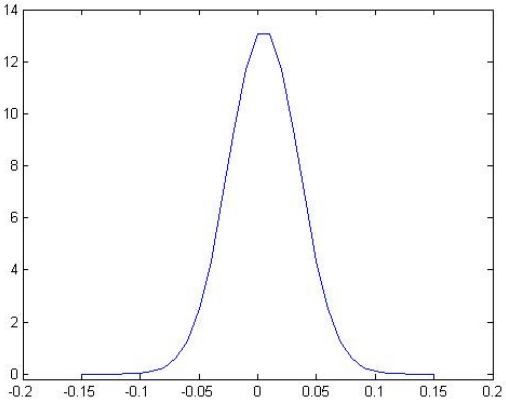

Fig. 5. Distribution of the noise obtained by the Box-Muller formula

Since the initial conditions of the chaotic pseudo-random number generator (1) 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 $\beta_q \leq K$, there exists $m \in \{1, 2, \ldots, \pi\}$ such that $\hat{s}_n \in I_m$.
The receiver also exactly knows the value of x_n^{p-2} and deduce from the rules (7) 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.

IV. Numerical illustrations

Now we summarize the main steps of the proposed algorithm:

1) Choose the secret parameters $k_i = 1$ or $k_i = -1$ for $i \in \{1, \ldots, p\}$ and $\varepsilon_{i,j} \in [10^{-15}, 10^{-5}]$, for $i, j = 1, \ldots, p$
2) Choose $N_0 \geq 100$
3) Define the initial conditions shared by the transmitter and the receiver
4) Iterate the Cms-PRNG (1) with the previous initial conditions, both at the transmitter and the receiver side
5) Apply the ciphering and transmission principle as detailed before

The figure 4 shows the noisy signal at the receiver side (recall that the transmitted signal is given by figure 3). Notice that the figures 2 to 4 represent our simulations with 10^9 iterations.

Remark 2. *To generate the bounded additive noise in our simulations, we resort to the Box-Muller formula. The process consists of generating a white gaussian noise from two random sequences, uniformly distributed in* $]0, 1]$. *These two random sequences are chosen between the remaining chaotic sequences (that have not been used in another step of the chaotic coding) and their absolute values are considered. The obtained noise is represented in figure 5.*

Notice that in real conditions, the noise naturally affects the transmission through the channel.

V. Conclusion

In this paper 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 co-generated sequence. The efficiency of the proposed scheme is illustrated on some numerical simulations. As further work, some studies should be performed of several sets of unknown parameters, since with the considered CMS-PRNG with 10 states, the number of possible parameters amounts to 90 (the $\varepsilon_{i,j}$ and the k_i). It is also possible to discuss about the opportunity to keep secret the correspondence between the alphabet and the intervals I_m.

References

[1] S.E. Assad, H. Noura, and I. Taralova. Design and analyses of efficient chaotic generators for crypto-systems. In *Proceedings of the Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pages 3–12, Washington, DC, USA, 2008.

[2] S. Behnia, A. Akhavan, A. Akhshani, and A. Samsudin. A novel dynamic model of pseudo random number generator. *Journal of Computational and Applied Mathematics*, 235(12):3455–3463, 2011.

[3] A. Espinel, I. Taralova, and R. Lozi. New alternate lozi function for random number generation. In *Emergent Properties in Natural and Artificial Complex Systems EPNACS, (M. Aziz-Alaoui, A . Banos, C. Bertelle, G. H. E. Duchmap Ed.)*, pages 13–15, Vienna, Austria, 2011.

[4] C.M. González, H.A. Larrondo, and O.A. Rosso. Statistical complexity measure of pseudorandom bit generator. *Physica A*, 354:281–300, 2005.

[5] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori. A secret key cryptosystem by iterating a chaotic map. In *EUROCRYPT'91*, pages 127–136, 1991.

Iteration	x_1	x_2	x_3	x_4
i=2	0.802775649423882687	0.198337199944899456	0.893873727345795399	-0.347193769493635562
i=3	0.407214033713460932	-0.290548081240143419	0.865058835244629565	-0.497163215517021417
i=4	-0.104976034759294692	-0.446154478326659054	0.772719118773404601	0.412887220696436463
i=5	0.343893580914444885	-0.665027721577955644	-0.1325510441039911	0.279201330018808036

TABLE I
CASE 1: $k_1 = 1, k_2 = -1, k_3 = 1, k_4 = -1$

Iteration	x_1	x_2	x_3	x_4
i=2	0.802775649423882687	0.198337199944899456	0.893873727345795399	0.0828063103885205709
i=3	0.407214030273460315	-0.290548084551144059	-0.704941107276218393	-0.362836944247290738
i=4	0.104976187115677827	-0.876155183551102557	-0.772719161018440959	-0.132888008099043914
i=5	0.0861075414927185007	0.0204085218923464214	-0.678326329298663211	0.629247595921321956

TABLE II
CASE 2: $k_1 = 1, k_2 = -1, k_3 = 1, k_4 = 1$

[6] A. Kanso and N. Smaoui. Logistic chaotic maps for binary numbers generations. *Chaos, Solitons and Fractals*, 40:2557–2568, 2009.

[7] H.A. Larrondo, C.M. González, M.T. Martín, A. Plastino, and O.A. Rosso. Intensive statistical complexity measure of pseudorandom number generators. *Physica A*, 356:133Ú138, 2005.

[8] H.A. Larrondo, M.T. Martín, C.M. González, A. Plastino, and O.A. Rosso. Random number generators and causality. *Phys. Lett. A*, 352:421–425, 2006.

[9] P. L'Ecuyer. http://www.iro.umontreal.ca/~lecuyer/, 2011.

[10] P. Li, Z. Li, W.A. Halang, and G. Chen. A multiple peudorandom-bit generator based on spatiotemporal chaotic map. *Phys. Lett. A*, 349:467–473, 2006.

[11] R. Lozi. New enhanced chaotic number generators. *Indian Journal of Industrial and Applied Mathematics*, 1(1):1–23, 2008.

[12] R. Lozi. Chaotic pseudo-random number generators via ultra weak coupling of chaotic maps and double threshold sampling sequences. In *3rd Internatioanl Conference on Complex Systems and Applications, (C. Bertelle, M. Aziz-Alaoui, X. Liu Ed.)*, pages 20–24, Le Havre, France, 2009.

[13] R. Lozi. Random properties of ring-coupled tent maps on the torus. *submitted to Discret and Continuous Dynamical Systems Series-B*, 2011.

[14] G. Marsaglia. http://stat.fsu.edu/~geo/diehard.html, 2011.

[15] M.T. Martin, A. Plastino, and O.A. Rosso. Statistical complexity and disequilibrium. *Phys. Lett. A*, 311:126–132, 2003.

[16] R. Matthews. On the derivation of a chaotic encryption algorithm. *Cryptologia*, XIII(1):29–42, 1989.

[17] L. De Micco, C.M. González, H.A. Larrondo, M.T. Martín, A. Plastino, and O.A. Rosso. Randomizing nonlinear maps via symbolic dynamics. *Physica A*, 387:3373–3383, 2008.

[18] T. Stojanovski and L. Kocarev. Chaos-Based Random Number Generators-Part I: Analysis. *IEEE Trans. Circuit Syst. I*, 48(3):281–288, 2001.

[19] T. Stojanovski and L. Kocarev. Chaos-Based Random Number Generators-Part II: Practical Realization. *IEEE Trans. Circuit Syst. I*, 48(3):382–385, 2001.

APPENDIX

In this part, we show the results of some tests we have performed to evaluate the required sensitivity of the trajectories of the Cms-PRNG (1) to the parameters k_i and $\varepsilon_{i,j}$. The same initial conditions are kept, and only one parameter is changed in each test, with respect to the other test. To alleviate the presentation, we give the results in the case $n = 4$. The sensitivity increases with the value of n.

Below are the initial conditions ($i = 1$), kept for all the tests:

$$\begin{cases} x_1 = 0.215 \\ x_2 = x_1 + 0.01777564 \\ x_3 = x_1 + 0.121111556 \\ x_4 = x_1 + 0.3333212212 \end{cases} \quad (10)$$

- Sensitivity to the k_i.

  The following values are chosen for the $\varepsilon_{i,j}$: $\varepsilon_{1,3} = 10^{-7}$, $\varepsilon_{1,4} = 8 * 10^{-7}$, $\varepsilon_{2,4} = 7.7 * 10^{-9}$, $\varepsilon_{2,1} = 4 * 10^{-7}$, $\varepsilon_{3,1} = 5.13 * 10^{-8}$, $\varepsilon_{3,2} = 8 * 10^{-10}$, $\varepsilon_{4,2} = 4 * 10^{-10}$, $\varepsilon_{4,3} = 3.002 * 10^{-7}$.

  The results are presented in tables I and II. Considering these two tests, with the considered set of parameters, choosing the value $N_0 \geq 4$ is sufficient.
- Sensitivity to the $\varepsilon_{i,j}$.

  The same methodology has been tested, with the following values for the k_i: $k_1 = 1$, $k_2 = -1$, $k_3 = 1$, $k_4 = 1$ and only one $\varepsilon_{i,j}$ is slightly varied (from 10^{-15}) from one test to another. For lack of place, the tables with the obtained results are not reported here. Again the value of N_0 can be fixed larger than 4.