2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

A Perturbation-Based Algorithm with Extremely Long Periods of Generated Cycles

Mieczyslaw Jessa

Faculty of Electronics and Telecommunication, Poznan University of Technology
ul. Polanka 3, 60-965 Poznan, Poland
Email: mjessa@et.put.poznan.pl

Abstract–A new algorithm for perturbing pseudo-chaotic orbits, solving the problem of short cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.

1. Introduction

The realization of chaotic maps in finite-state machines leads to serious degradation of chaotic dynamics. The properties of cycles observed in computers depend on the choice of the initial point, and their periods may be unexpectedly short [1, 2]. The problem of dynamical degradation has been addressed in many papers (e.g., [3] and references therein), and it is regarded as one of reasons for the weaknesses of many chaos-based cryptosystems realized in digital machines. Chaotic maps implemented in finite-state machines are known in the literature as pseudo-chaotic maps or digital chaotic maps. Up to now, three basic solutions to improve the properties of pseudo-chaotic maps have been proposed: using a higher finite precision [4, 5], cascading multiple chaotic systems [6] and using a perturbation-based algorithm [7-11]. The first solution does not solve the problem at all, and cascading multiple chaotic systems increases the length of the cycle but complicates digital realization of a chaotic system. In a perturbation-based algorithm, however, a second system is used to perturb the orbits of a pseudo-chaotic map. The use of a perturbation as a method of improving digital chaos was independently proposed by Čermák [7] and Zhou and Ling [8]. It was later improved by Sang et al. [9-10] and adopted for pseudo-chaotic ciphers by Li [11].

In a perturbation-based algorithm, cycles produced by a pseudo-chaotic map are perturbed every Δ iterations, where Δ is an integer. A perturbing sequence comes from another system that produces uniformly distributed pseudo-random sequences. The system can either be realized in the same machine or it can be physically independent of the machine used in computations. An alternative approach is to perturb a control parameter of a chaotic map, but this method seems to be less efficient [3]. The perturbation operation may either be the XOR function of the same groups of bits in a perturbed and pseudo-random sequence or it may be another masking function. The perturbation significantly increases the period of generated cycles. For example, it was shown in [9] that the minimum length m_{min} of generated sequences is

$$m_{\min} = \Delta(2^p - 1), \quad (1)$$

where p is the length of a linear feedback shift register (LFSR) that generates maximal length sequences. Although this method improves the pseudo-chaos observed in digital machines, it has disadvantages. For example, it requires an external system to generate uniformly distributed pseudo-random sequences. The dynamics of the perturbing system are known and are different from those of the pseudo-chaotic map, which is undesirable in cryptographic applications. In this paper, we propose a new method that eliminates both disadvantages. Generated cycles are perturbed with the use of the same pseudo-chaotic map, and the output sequences have extremely long periods, practically independent of the number l of bits used in computations. To change the period of the output cycle, we do not need to redesign the auxiliary generator; it is sufficient to change the size of an auxiliary table that is implemented in the same circuit as the pseudo-chaotic map.

2. The Method

Without losing generality, we consider a one-dimensional chaotic map f, mapping unit interval $I \equiv [0,1)$ into itself:

$$x_n = f(x_{n-1}), \quad n = 1,2,... \quad (2)$$

The perturbing signal (sequence) $\{z_n\}$ is added to a pseudo-chaotic orbit every Δ iterations of f, i.e.,

$$\begin{cases} x_n = z_{n-1} \oplus f(x_{n-1}), & n = 1,2,... \\ y_n = x_n \end{cases}, \quad (3)$$

where $z_n = 0$ for $n \neq (k+1)\Delta$ and $z_n \neq 0$ for $n = (k+1)\Delta$, and $k = 0,1,...$. The symbol $\oplus$ represents the perturbation operation, and y_n is the output signal (perturbed orbit). The novelty of the paper lies in the method of generating $\{z_n\}$ and the analysis of the period of perturbed orbits. The numbers z_n are produced with the use of the shuffling algorithm proposed by Bays and Durham for the multiplicative congruential pseudo-random generator [12,

13]. Using the method of Bays and Durham, we initially compute the L elements of sequence $\{x_n\}$. These are written into successive cells of an auxiliary table T with size L. The value of the next iteration, i.e., $x_L = f(x_{L-1})$, is the first number (u_1) of the output stream. The same x_L is used to compute the address t_1 of a cell of T. The number read off from this cell is the second number (u_2) of the output sequence. We generate the next number $x_{L+1} = f(x_L)$ and write it into T in the place of number u_2. Next, we use u_2 to compute the next address t_2. The number read off from the cell with address t_2 is the third number (u_3) of the output stream. We compute $x_{L+2} = f(x_{L+1})$ and write it in place of u_3, etc. [12, 13]. The addresses of the cells of table T are computed from the equation

$$t_N = \lceil L \cdot u_N \rceil, \quad N = 1,2,\ldots, \tag{4}$$

where it was assumed that $u_N \in [0,1)$. The period m_u of $\{u_N\}$ is [13]

$$m_u = O(m_x L!)^{0.5}, \tag{5}$$

where m_x is the period of sequence $\{x_n\}$. Period (5) is achieved for

$$L << m_x << L!. \tag{6}$$

If m_x does not satisfy (6), the period of $\{u_N\}$ either does not change or increases slightly [12]. It should be noted that if shuffling sufficiently improves the properties of pseudo-chaotic orbits, we do not need to apply any perturbation algorithm at all. In other cases, the implementation of a perturbation is a better solution.

The simplest method for perturbing the low-order bits of x_n is the computation of the XOR function of these bits as well as the bits encoding symbols t_N. Generally, such an operation is dangerous because the period of $\{t_N\}$ may be very short compared with the period of $\{u_N\}$. In this paper, we propose another approach that exploits all bits of some elements of sequence $\{\tilde{u}_N\}$, computed for $N = (k+1)c\Delta$, $k = 0,1,\ldots$. Sequence $\{\tilde{u}_N\}$ is obtained from a perturbed sequence $\{y_n\}$ shuffled in table T (Fig. 1).

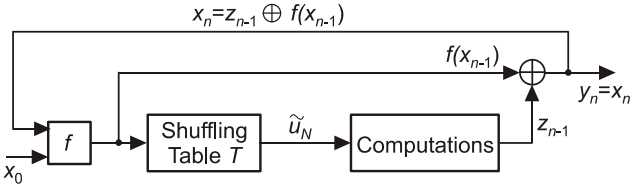

Fig. 1. Self-perturbing of sequence $\{x_n\}$

The constant c is computed as follows. We assume that all numbers are encoded by l bits b_{-j}, $j = 1,2,\ldots,l$ and that l has c divisors. For example, for $l = 16$ the divisors are 1, 2, 4, 8, and 16. Denoting by d one of the divisors, we can divide the sequence of l bits into $c=l/d$ disjoint blocks $B_{N,i}$ ($i = 1,2,\ldots,c$) of bits with length d. Then,

$$\tilde{u}_N = 0.b_{N,-1}b_{N,-2}\ldots b_{N,-l} = 0.B_{N,1}B_{N,2}\ldots B_{N,c} \in I \equiv [0,1), \tag{7}$$

where

$$B_{N,i} = \{b_{N,-[d(i-1)+1]}b_{N,-[d(i-1)+2]}\ldots b_{N,-id}\}. \tag{8}$$

If $N = (k+1)c\Delta$, then we take $\tilde{u}_N$ to produce c digital words $B_{N,i}$. The words are next used in c successive perturbations. A pseudo-chaotic orbit is perturbed by means of signal

$$z_n = \begin{cases} 0 & for \quad n \neq (k+1)\Delta \\ B_{N-1,1+k \bmod c} & for \quad n = (k+1)\Delta \end{cases}, \tag{9}$$

where $B_{0,1+k \bmod c}$ ($k = 0,1,\ldots$) denotes $B_{0,i}$ ($i = 1,2,\ldots,c$), which encodes $\tilde{u}_1$. After the perturbation, the number k is increased by unity and remains unchanged until the next perturbation. It plays the role of the perturbation counter.

The analysis of the period of a perturbed sequence is very difficult, but we can find the lower bound of its value. This is sufficient for most applications, such as in cryptography. Let us assume that m_y is the period of $\{y_n\}$ and $m_{\tilde{u}}$ is the period of $\{\tilde{u}_N\}$. If

$$L << m_y << L!, \tag{10}$$

then

$$m_{\tilde{u}} = O(m_y L!)^{0.5}, \tag{11}$$

where

$$m_{\tilde{u}} >> m_y. \tag{12}$$

The period m_y of a pseudo-chaotic sequence perturbed every Δ iterations of f is the least common multiple of period m_p of a perturbing sequence $\{z_n\}$ and Δ,

$$m_y = LCM(m_p, \Delta), \tag{13}$$

If m_p and Δ are relatively prime, then m_y is the product of m_p and Δ. In other cases, we can write that

$$m_y \geq m_p, \tag{14}$$

where $m_p > \Delta$. In the proposed method, the perturbing signal is derived from the shuffled sequence $\{\tilde{u}_N\}$ with period $m_{\tilde{u}}$. The period m_p of the perturbing sequence is $c\Delta$ times shorter than $m_{\tilde{u}}$. Thus, from (14), we obtain that

$$m_y \geq \left\lfloor \frac{m_{\tilde{u}}}{c\Delta} \right\rfloor. \tag{15}$$

The results (12) and (15) are inconsistent. Consequently, the value of m_y increases with the increase in the number of iterations. Longer $\{\tilde{u}_N\}$ yields longer $\{y_n\}$ and longer $\{y_n\}$ produces longer $\{\tilde{u}_N\}$ which, in turn, increases the length of $\{y_n\}$ and so on. The process ends when m_y stops to satisfy (10). The value of m_y becomes constant, and successive numbers begin to repeat with a constant period.

The determination of the exact m_y that can be regarded as significantly smaller than $L!$ is practically impossible. In the technical sciences, it is usually assumed

that a number α is significantly smaller than a number β if $\alpha \le \beta/10$. An exception to this is in the field of metrology, in which the requirement is that $\alpha \le \beta/100$. From a practical point of view, it is safer to assume that m_y stops increasing when

$$m_y = m \ge \left\lfloor \frac{L!}{100} \right\rfloor . \tag{16}$$

For large L this value is significantly greater than $O(m_x L!)^{0.5}$ assessed by Bays and Durham [12]. To obtain periods equal to or longer than $L!/100$, the period m_x of an unperturbed pseudo-chaotic sequence has to satisfy condition (6). For example, if table T has 128 cells, we have the condition that $2^7 << m_x << 128! \approx 1 \cdot 2^{716}$. It is easy to satisfy the condition for the unperturbed orbits of pseudo-chaotic maps.

Let us emphasize that formula (5) was obtained by Bays and Durham under the assumption that $\{x_n\}$ is an approximation of independent and uniformly distributed random numbers from unit interval $I = [0,1)$. Therefore, the analysis of the period of perturbed sequence is valid for maps f that generate sequences satisfying the same assumption. If we apply the proposed self-perturbing method to a map that do not produce uniformly distributed numbers, the smallest period of sequence $\{y_n\}$ may not satisfy inequality (16).

3. Numerical Experiment

In a finite-state machine with a l-bit digital word, a real number is approximated by a rational number. We can perform all computations either traditionally, i.e., with the use of floating-point arithmetic, or we can express the dynamic variables of (2) as $p/2^l$, where $p \in N$, $0 < p < 2^l$.

In the ANSI/IEEE double-precision floating-point arithmetic, where the fractional part of a real number requires 52 bits, we can divide the mantissa into $c = 52/d$ d-bit blocks. The number d is one of six divisors: 1, 2, 4, 13, 26, 52 of $l = 52$. Next, we perturb numbers $\{x_n\}$ every Δ iterations of (2), according to the procedure described in Section 2. Another approach, which is faster and free from errors introduced by the floating-point arithmetic, exploits the transformation of f into the set of integer numbers. It was first proposed in paper [14]. Formula (2) takes the form

$$p_{n+1} = \left\lfloor 2^l \cdot f(p_n / 2^l) \right\rfloor, \quad n = 0,1,\ldots . \tag{17}$$

In the numerical experiment, we found the period of perturbed cycles produced by the Rényi chaotic map for six initial points. For the Rényi map

$$x_{n+1} = (\lambda \cdot x_n) \mod 1, \quad x \in [0,1], \quad \lambda \in R , \tag{18}$$

where λ is a real number, we obtain

$$p_{n+1} = \left\lfloor 2^l \cdot \left(\lambda \cdot (p_n / 2^l) \bmod 1 \right) \right\rfloor \tag{19}$$

or equivalently

$$p_{n+1} = \left\lfloor (\lambda \cdot p_n) \bmod 2^l \right\rfloor . \tag{20}$$

Because the observed periods can be extremely long, it was assumed in the numerical experiment that all numbers were encoded only by $l = 16$ bits and that the size L of table T was small but satisfied condition (6). Table 1 shows the periods of unperturbed and perturbed orbits. The orbits were produced by a Rényi chaotic map with parameter λ equal to 2.8. Parameter Δ was equal to 500. During the perturbation, the XOR function between z_n and $d = 8$ low-order bits of p_n was computed. Because the perturbation may lead to $x = 0$ and, consequently, to a sequence of zeros, the number $x = 0$ was always replaced by the number $x = 2^{-l}$. Table 1 also contains the smallest period computed from (16) (the numbers in parentheses, below the period found experimentally).

Table 1. The period of unperturbed and perturbed pseudo-chaotic orbits as a function of initial point and the size L of table T

	No pertur-bation	L=9	L=10	L=11
$x_0 = 1/2^{16}$	715	12073000 (3628)	26565000 (36288)	388116000 (399168)
$x_0 = 10/2^{16}$	715	12073000 (3628)	27004000 (36288)	388116000 (399168)
$x_0 = 10^2/2^{16}$	715	68000 (3628)	27004000 (36288)	388116000 (399168)
$x_0 = 10^3/2^{16}$	715	12073000 (3628)	25322000 (36288)	98864000 (399168)
$x_0 = 10^4/2^{16}$	715	68000 (3628)	27004000 (36288)	105575000 (399168)
$x_0 = 6 \cdot 10^4/2^{16}$	715	12073000 (3628)	27004000 (36288)	115570000 (399168)

The period of unperturbed orbits was constant for all initial points considered in the experiment and equal to 715. Consequently, condition (6) is satisfied for $L \ge 9$. During iterations of *f,* the value of m_y increases with the increase in the number of iterations. The process ends when m_y ceases to satisfy (10). To experimentally find the final value of m_y , we have to omit a large number of intermediate states. We first perform some number n_s of initial iterations. For example, for $L = 11$, the effective searching of period 388116000 requires $n_s = O(10^9)$ initial iterations. Number n_s is greater than $L! = 11! = 39916800$. It seems that such a long intermediate state with irregular behavior of the perturbed pseudo-chaotic orbit can be used in many applications instead of a fragment of the periodic sequence.

4. Limitations of perturbation

One of the limitations of perturbation is that perturbation may yield a sequence that terminates in a sequence of zeros or ones. As was shown in Section 3, this

can usually be eliminated without significant computational effort. A much more serious problem is the repetition of long fragments of cycles produced by *f* when *f* is implemented in the computer. If *f* produces many short orbits for different initial conditions, perturbation leads to orbit hopping. Because the orbits are short, we have to make perturbations frequently. If *f* produces long orbits, their number may be small. The probability then rapidly increases that a perturbation will repeat a long fragment of an unperturbed orbit. To overcome this serious disadvantage, we can, for example, combine signals produced by many independent pseudo-chaotic systems. In this paper, we propose to use the shuffling of Bays and Durham to perturbed sequence $\{y_n\}$. This is simpler and requires less computational effort. If the perturbation yields a point that was previously generated and, consequently, a long part of a previously produced sequence repeats, additional shuffling changes the order of appearances of elements of this sequence. This is true if and only if during the perturbation table *T'* does not contain the same numbers and in the same order as for the previous sequence. The number *M* of different contents of *T'* can be computed from the equation

$$M = \left(2^l\right)^{L'}, \tag{21}$$

where *l* is the number of bits that encode numbers and *L'* is the size of the additional table. If numbers written into *T'* are equally probable, the probability *P* that *T'* contains the same numbers and in the same order as for the previous sequence is equal to 1/*M*. For example, if $L' = 64$ and $l = 16$, we obtain $P = 2^{-1024}$.

If *L'* does not satisfy condition $L' << m_y << L'!$, then the period of the sequence shuffled in *T*' is comparable to the period of the perturbed sequence $\{y_n\}$. Let us emphasize that additional shuffling does not introduce security to sequence $\{y_n\}$ [15].

5. Conclusions

In this paper, a new algorithm to perturb pseudo-chaotic orbits observed in finite-state machines has been proposed. In comparison with the existing algorithms, this algorithm does not require the perturbing signal to be generated by an external source. The period of the generated cycle depends on the size of the table used to shuffle a perturbed pseudo-chaotic orbit and may be a huge number, independent of the precision of computations. A method for preventing the repetition of long fragments of an unperturbed orbit observed in a perturbed signal has also been considered. The proposed algorithm can be used in applications requiring sequences with extremely long periods, e.g., in cryptography. Another application may be the generation of long-period, pseudo-random sequences in finite-state machines with a small number of states. The achievable periods can be significantly greater than periods obtained for the shuffling algorithm of Bays and Durham. The method is very fast and can be easily implemented in contemporary field programmable gate arrays. The subject of future research should be rigorous analysis of the period of sequence $\{y_n\}$ for arbitrary map *f*. The fact that $\{y_n\}$ and $\{\tilde{u}_N\}$ depend on each other makes the analysis difficult.

References

[1] C. Beck and G. Roepstorff, "Effects of phase space discretization on the long-time behavior of dynamical systems," *Physica 25D*, pp. 173-180, 1987.

[2] P. M. Binder and R. V. Jensen, "Simulating chaotic behavior with finite-state machines," *Phys. Rev. A*, A34, pp. 4460-4463, 1986.

[3] S. Li, G. Chen, and X. Mou "On the dynamical degradation of digital piecewise linear chaotic maps," *Int. Journal of Bifurcation and Chaos,* 2005, vol. 15 (10), pp. 3119-3151, 2005.

[4] D. D. Wheeler, "Problems with chaotic cryptosystems," *Cryptologia*, vol. XIII, pp. 243-250, 1989.

[5] D. D. Wheeler and R. A. J. Mathews, "Supercomputer investigations chaotic encryption algorithm," *Cryptologia*, vol. XV, pp. 140-15, 1991.

[6] G. Heidari-Bateni and S. D. McGillem, "A chaotic direct-sequence spread-spectrum communication system," *IEEE Trans. Commun.,* vol. 42, (2/3/4), pp. 1524-1527, 1994.

[7] J. Čermak, "Digital generators of chaos," *Phys. Lett. A*, (214) vol. 214, pp. 151-160, 1996.

[8] H. Zhou and X. Ling, "Realizing finite precision chaotic systems via perturbation of *m*-sequences," *Acta Electron. Sin,* vol. 25, pp. 95-97, 1997.

[9] T. Sang, R. Wang, and Y. Yan, "Perturbance-based algorithm to expand cycle length of chaotic key stream," *Electronics Letters',* vol. 34 (9), pp. 873-874, 1998.

[10] T. Sang, R. Wang, and Y. Yan, "Clock-controlled chaotic keystream generators," *Electronics Letters,* vol. 34 (20), pp. 1932-1998, 1998.

[11] S. Li, "Analysis and New Designs of Digital Chaotic Ciphers," *Ph. D. Dissertation*, Xi'an Jiaotong University, 2003.

[12] C. Bays and S. D. Durham, "Improving a poor random number generator," *ACM Trans. on Mathematical Software,* vol. 2, pp. 59-64, 1976.

[13] J. E. Gentle, "Random number generation and Monte Carlo methods," Springer-Verlag, New York, 2003.

[14] W. F. Wolff and B. A. Huberman, "Transients and asymptotics in granular phase space," *Z. Phys. B – Condensed Matter*, vol. 63, pp. 397-405, 1986.

[15] M. Dichtl, "Cryptographic Shuffling of Random and Pseudorandom Sequences", Dagstuhl Seminar Proceedings 07021, Symmetric Cryptography 2007, http://drops.dagstuhl.de/opus/volltexte/2007/1014.