True Random Bit Generation From a Double-Scroll Attractor

Müştak E. Yalçın, *Student Member, IEEE*, Johan A. K. Suykens, *Member, IEEE*, and Joos Vandewalle, *Fellow, IEEE*

***Abstract*—In this paper, a novel true random bit generator (TRBG) based on a double-scroll attractor is proposed. The double-scroll attractor is obtained from a simple model which is qualitatively similar to Chua's circuit. In order to face the challenge of using the proposed TRBG in cryptography, the proposed TRBG is subjected to statistical tests which are the well-known Federal Information Processing Standards-140–1 and Diehard test suite in the area of cryptography. The proposed TRBG successfully passes all these tests and can be implemented in integrated circuits.**

***Index Terms*—Chaos, cryptography, random number generator (RNG).**

I. Introduction

RANDOM number generators (RNGs) are used in many areas including computer simulations, Monte Carlo techniques in numerical analysis, test problem generation for the performance evaluation of computer algorithms, statistical sampling, stochastic optimization methods (such as simulated annealing), watermarking for image authentication and cryptography. A good random number generation helps to improve the results in these applications and is vital for cryptographic security. Security issues are coming to the fore today because of increasingly demanding security requirements in many new applications on the internet (such as secure e-mail, e-commerce, e-government). The security of cryptographic algorithms depends on generating secret quantities which are generated by RNGs, for passwords, keys, etc. As noted in [12], the use of pseudo-random processes to generate secret quantities can result in pseudo-security. Random numbers can be generated by a random bit generator which can be defined as a device or algorithm whose output is a sequence of statistically independent and unbiased binary digits [31]. To ensure that a RNG is secure, its output must be statistically indistinguishable from a true random sequence and unpredictable [31], [12].

RNGs can be classified into three classes which are: *true RNGs (TRNGs)*, *pseudo RNGs (PRNGs)*, and *hybrid RNGs (HRNGs)*. A TRNG operates by measuring unpredictable natural processes such as thermal noise from a semiconductor [9], frequency instability of an oscillator [16], elapsed time between emission of particles during radioactive decay [20] and variations in disk drive response times [10], [22]. These are also called hardware-based generators because of the use of the randomness aspect in the hardware. The processes are chaotic or nondeterministic. Furthermore, they produce continuous time analog signals which are often called noise. As is shown in [1], [29], true random bits/numbers can be generated by these physical noise sources.

PRNGs use deterministic processes (also called *deterministic RNGs*) to generate a series of outputs from an initial seed state. PRNGs are much more cost effective and thousands of times faster than hardware based RNGs. However, because the output is a function of the seed state, the actual entropy of the output can never exceed the entropy of the seed [24], [35]. Hence, the randomness level of the pseudo-random numbers depends on the level of randomness of the seed. Therefore, HRNGs use a random generator as a seed generator and expand it. A seed generator is a hardware-based RNG with or without user's interaction, such as random keystrokes, mouse movements, or hard drive seek times.

If one summarizes the present state-of-the-art in RNGs, the conclusion is that PRNGs need a TRNG in order to improve the quality of the output (e.g., the hardware-based Intel RNG [9] included in the Intel 8XX series of chip-sets for generating the random seed [36]). Although commercial cryptographic packages have user's interaction to produce a supposedly random seed and do not need additional hardware, these methods are time consuming, inconvenient for the user and cannot be used with automated scripts [22], [24]. Nevertheless, it is quite difficult to distinguish between a signal coming from a nondeterministic source and a chaotic system ([38] is a recent study which addresses this issue). In this paper, we will group TRNGs into two classes, based on the kind of analog signal they are using and will be called a TRNG either based on a chaotic or on a nondeterministic system (e.g., while the sources of [9]and [20] are known as a nondeterministic system, variations in disk drive response time are known to be a consequence of chaotic air turbulence [16]). The same classification of TRNGs is also made in [29].

The output of chaotic systems can be predicted from the exact initial conditions depending on the first Lyapunov exponent. However, sensitivity with respect to the initial conditions causes unpredictability on the long term. The short-term predictability of chaotic time-series was shown in [11], [5] and has been also

Manuscript received September 5, 2003; revised October 10, 2003. This work was supported in part by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture under Grant IUAP P4–02, Grant IUAP P4–24, and Grant IUAP-V, in part by the Concerted Action Project MEFISTO of the Flemish Community under the FWO Project G.0080.01 and ESPRIT IV 27077 (DICTAM). This paper was recommended by Associate Editor M. Gilli.

The authors are with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven B-3001, Belgium (e-mail: mustak.yalcin@esat.kuleuven.ac.be; johan.suykens@esat.kuleuven.ac.be; joos.vandewalle@esat.kuleuven.ac.be).

Digital Object Identifier 10.1109/TCSI.2004.830683

1057-7122/04$20.00 © 2004 IEEE

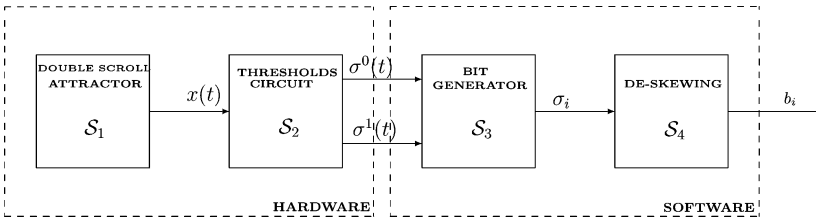


Fig. 1. Proposed TRBG.

addressed in the time-series prediction competition reported in [41]. Nevertheless, short-term predictability of chaotic time-series is seen as a drawback for using chaotic signals in RNGs which are designed for cryptographic applications. A scheme which is taking into account the *predictability time* is studied in [2]. In [18] a noninvertible function is used as a static output of a chaotic system which also improves the randomness for the output signal. In fact, noninvertibility (also called a *one-way function* in cryptography) is very much used in many areas in cryptography such as public key cryptography, authentication [31] and also for generating pseudo-random numbers [37], [31]. Similar ideas have been used by Stojanovski and Kocarev [39] who define a binary generating partition which can be considered as a one-way function, on the chaotic piecewise-linear one-dimensional map for producing random bits. In a similar way, chaotic maps have been used as a binary information source in telecommunications [3], [4].

One should note that when unpredictability is an important issue, direct use of the output of the chaotic system might give an advantage to the opponent if it is short-term predictable. Cryptographic protocols crucially depend on the unpredictability of the secret key which is produced by a RNG. If an attacker can predict the key's value or the number of keys, the protocol can be broken with much less effort than when truly random keys are used. Therefore, it is vital that the secret keys are generated from a true random source. Statistical tests determine whether the sequence possesses a certain attribute that a truly random sequence would be likely to exhibit. Hence any RNG which is proposed for use in a cryptographic protocol, must be subject to statistical tests. There are two very well-known statistical test suites which are Federal Information Processing Standards (FIPS)-140–1 [31] and the Diehard test suite [30]. FIPS-140–1 is issued by the National Institute of Standards and Technology (NIST) and strong statistical tests are given in the Diehard test suite.

The aim of this paper is to show the usage of continuous-time chaotic systems for generating true random bits. We introduce a straightforward method for generating true random bits and take care of randomness in the sense of unpredictability. Due to the difficulty of proving unpredictability in a theoretical way, the proposed true random bit generator (TRBG) is subjected to statistical tests. We demonstrate that the proposed TRBG passes the FIPS-140–1 statistical tests and the full suite of the Diehard test. However, one should notice that none of the statistical tests can prove that the generator is an ideal RNG (the many tests are in fact necessary conditions but not sufficient).

This paper is organized as follows. In Section II, the proposed TRBG is described in general terms. The following sections describe the details of the system block by block. In Section III, the employed chaotic oscillator is described. Section IV explains how a bit sequence is generated by means of the chaotic signal. In Section V, a de-skewing technique is discussed for eliminating bias in the sequence. Furthermore, it is explained how one of the threshold values is set by measuring entropy. Empirical tests and results are presented in Section VI.

II. Double-Scroll-Based TRBG

In this paper, we introduce a new TRBG based on a chaotic system. Fig. 1 shows the proposed TRBG which consists of a hardware and software parts. The software part can also be integrated in the hardware part using standard digital circuits. For practical reasons, this block is implemented here in software. The hardware part includes a chaotic oscillator and thresholds circuit. The chaotic oscillator is a continuous time third order autonomous system and exhibits a *double-scroll* attractor. Double-scroll attractors have been observed in Chua's circuit [7]. Since then, the double-scroll attractor has been addressed by many researchers [28], [6]. Here, the chaotic oscillator has been used as a source for the proposed RBG which is qualitatively similar to Chua's circuit [8]. The double-scroll attractor block which is denoted by $\mathcal{S}_1$ is described in Section III together with its circuit implementation.

The state space of the double-scroll attractor is partitioned into three subspaces by block $\mathcal{S}_2$ of the hardware part. Two of the subspaces are related with the location of the scrolls in state space. The third subspace is a region where the transition between the scrolls occurs. The bit generator (block $\mathcal{S}_3$) is the first block in the software part. Its inputs correspond to the outputs of block $\mathcal{S}_2$. The block $\mathcal{S}_3$ is basically coding the scrolls (e.g., the output of the bit generator is 0110 which means that the state–space trajectory moved from the scroll 0 to the other scroll 1, jumps to itself and then returns to scroll 0). In fact, blocks $\mathcal{S}_2$ and $\mathcal{S}_3$ are discretizing the state variable $x(t)$ in space instead of in time (relying on the ergodicity property of chaos). When the blocks $\mathcal{S}_2$ and $\mathcal{S}_3$ are combined and considered as one block its function is noninvertible. Sampling the chaotic signal in space gives an irregular sampling of the signal in time (Fig. 2). Therefore, it becomes very hard to make a prediction for an opponent in cryptography, both a one-step ahead and long term prediction. In this way, the method presented in this paper is a combination of the two ideas mentioned in the introduction and improves the randomness in the sense of unpredictability [18], [2].

The last block ($\mathcal{S}_4$) in the software part is a de-skewing which is eliminating the bias in the output of the bit generator. This bias depends on the probability density of the chaotic signal itself. Moreover, offset and bandwidth limitations might also effect the bias of the generated sequence [1].

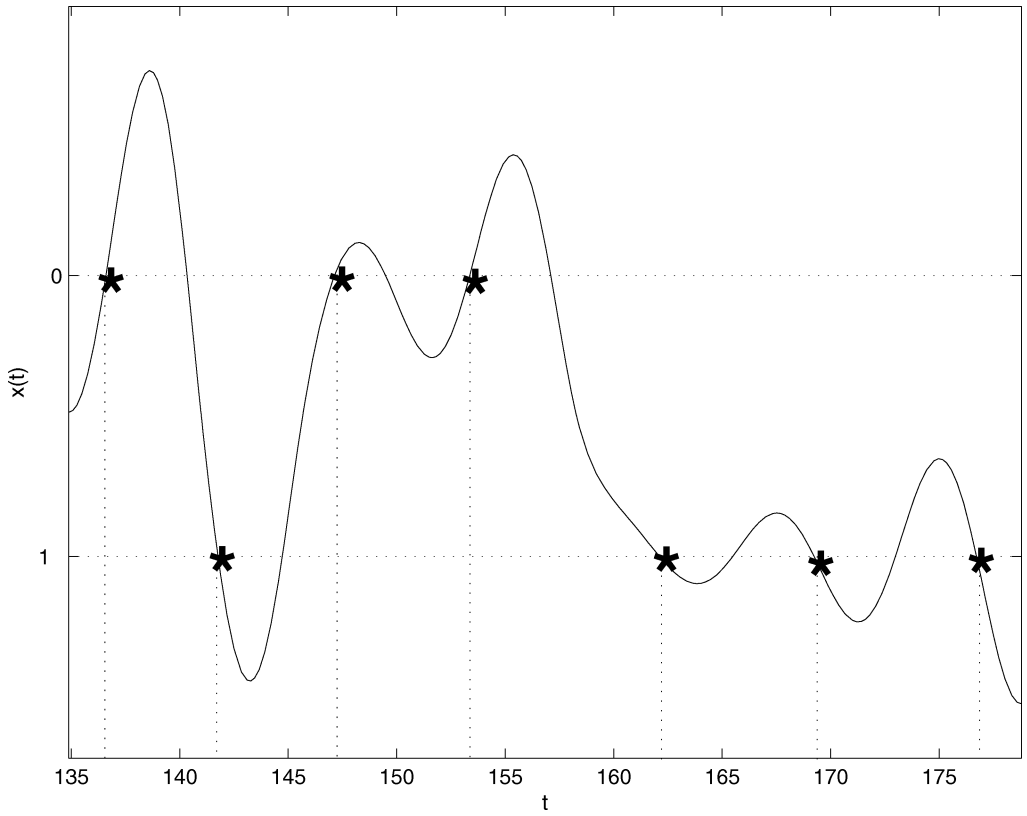


Fig. 2. Sampling the chaotic signal.

III. Double-Scroll Attractor

Recently, in [14] an alternative way for generating double scroll like behavior was proposed by means of a simple circuit implementation. It is given by

$$\mathcal{S}_1: \quad \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}f(\mathbf{C}\mathbf{x}) \tag{1}$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a & -a & -a \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} \quad \mathbf{C} = [0 \quad 0 \quad 1]$$

where

$$f(\zeta) = \begin{cases} 1, & \text{if } \zeta \geq 0 \\ -1, & \text{if } \zeta < 0 \end{cases}$$

and $\mathbf{x} = [x; y; z] \in \mathbb{R}^3$, $\zeta \in \mathbb{R}$. This model produces a double-scroll-like attractor for $a = 0.8$ [14]. By taking a generalization of this nonlinearity, it is also possible to obtain n-scroll attractors, similar to the generalized Chua's circuit [40]. The generalized nonlinearity for (1) is given by

$$f(x) = \sum_{i=1}^{M} g_{(-2i+1)/2}(x) + \sum_{i=1}^{N} g_{(2i-1)/2}(x)$$

with

$$g_\theta(\zeta) = \begin{cases} 1, & \text{if } \zeta \geq \theta \ \theta > 0 \\ 0, & \text{if } \zeta < \theta \ \theta > 0 \\ 0, & \text{if } \zeta \geq \theta \ \theta < 0 \\ -1, & \text{if } \zeta < \theta \ \theta < 0. \end{cases}$$

An experimental confirmation of a 5-scroll attractor for $M = 2$ and $N = 2$ has been given in [45]. Furthermore, families of scroll grid attractors have been obtained with the same model by taking additional nonlinearities with similar characteristics [45], [6]. In this paper, we use $M = 1$ and $N = 0$ in order to obtain a double-scroll attractor. The reason for choosing this model in our application is that the model itself is quite simple. As a result, the implementation can be fully integrated in a CMOS chip with small cost. One of the possible integrations in CMOS chips has been proposed in [15]. Furthermore, two new circuit realizations of this double-scroll attractor have been recently proposed based on MOS-only [34] and log domain p-n-p-only transistors [33]. The latter are very suitable for low-voltage and high-frequency operation and can also be implementated in integrated circuits. Hence, an RNG based on this chaotic system could be included also as a standard part of a system's architecture, as suggested in [12].

Numerous research results have illustrated the advantages of current feedback op-amps (CFOA) for the implementation of chaotic oscillators such as operating at higher frequencies with design flexibility [13]. Therefore, a CFOA-based circuit realizing the double-scroll attractor has been designed for the purpose of this paper. The circuit diagram is given in Fig. 3. For $C_x = C_y = C_z = C$, $R_1 = R$, $R_2 = R_4 = R/a$, $V_x = ax$, $V_y = ay$, $V_z = z$, and using the normalized quantity $\tau = t/RC$, this circuit realizes the system (1). The same circuit structure has been successfully used in [45] for realizing families of scroll grid attractors. The CFOAs implemented using AD844 from Analog Devices and required nonlinearity is realized by LM311 type comparators. CFOAs and LM311 are supplied under 15-V

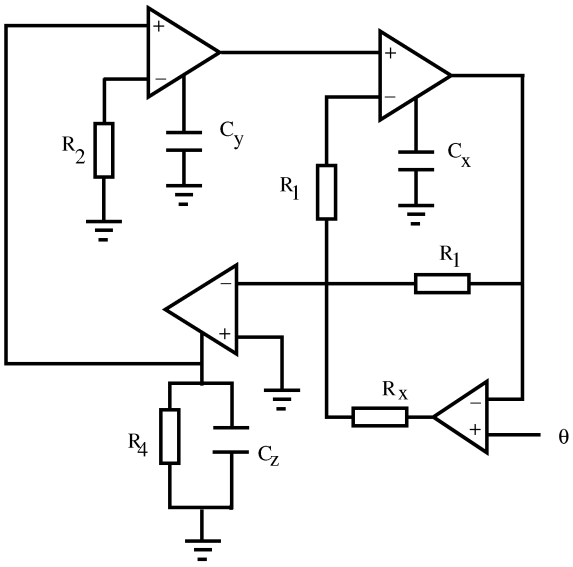


Fig. 3. Circuit realizing the double-scroll attractor.

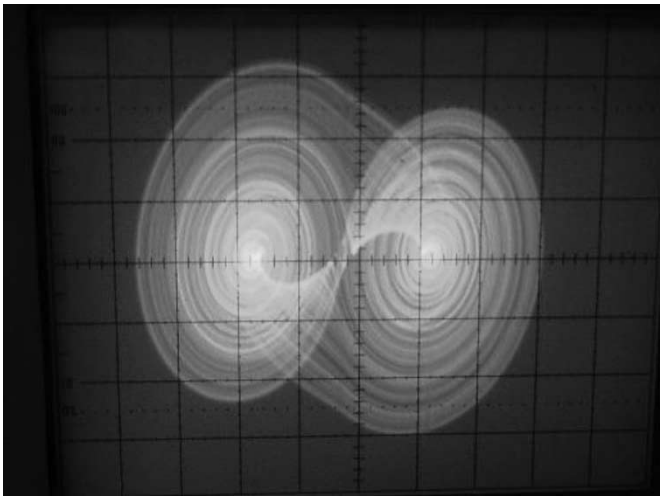


Fig. 4. Double-scroll attractor.

dc and passive component values are taken as $C = 1$ nF and $R_1 = 5.1\ \text{k}\Omega$; the values of other passive components have been assigned during the experiment. On the circuit, θ is measured as -0.51 V, which has been obtained using series connected resistances between ground and -15 V. In order to obtain a symmetric double scroll, the value of R_x is set to 74.8 kΩ, $R_2 = 7\ \text{k}\Omega$ and $R_4 = 11\ \text{k}\Omega$. The corresponding result for these value is shown in Fig. 4.

IV. Bit Generation

A simple way to generate a bit sequence from a chaotic real valued signal is by using a threshold function [43]

$$\sigma_c(x) = \begin{cases} 0, & \text{if } x < c \\ 1, & \text{if } x \geq c. \end{cases} \tag{2}$$

One obtains binary sequences which are referred to as chaotic binary sequences [43]. Because our chaotic system $\mathcal{S}_1$ generates a continuous time signal, we define two thresholds (c_1 and c_2). Then, the state space is divided into three subspaces denoted by $V_0 = \{(x,y,z)|x \leq c_2\}$, $V_T = \{(x,y,z)|c_2 < x < c_1\}$ and $V_1 = \{(x,y,z)|x \geq c_1\}$. Fig. 5 shows the subspaces for the values $c_1 = 0$ and $c_2 = -1$. The block $\mathcal{S}_2$ that realizes the two threshold functions, is described by

$$\mathcal{S}_2: \quad \begin{aligned} \sigma^1(x(t)) &= \begin{cases} 0, & \text{if } x(t) < c_1 \\ 1, & \text{if } x(t) \leq c_1 \end{cases} \\ \sigma^0(x(t)) &= \begin{cases} 0, & \text{if } x(t) > c_2 \\ 1, & \text{if } x(t) \leq c_2. \end{cases} \end{aligned} \tag{3}$$

By discretizing the state space in this way, one can obtain symbolic dynamics that describe the evolution of the system by an infinite sequence of symbols [27], [19]. The two threshold functions are realized by LM311 type comparators (Fig. 6) which are of the same type as the comparators to realize the nonlinearity for the double-scroll attractor. The outputs of the comparators are directly connected to a Personal Computer via a parallel port.

A bit is generated when the signal passes the subspace V_0 or the subspace V_1 region through the subspace V_T. The bit generator block $\mathcal{S}_3$ is described by

$$\mathcal{S}_3: \ \sigma_i(\sigma^0,\sigma^1) = \begin{cases} 0, & \text{if } \sigma^0 = 0, \sigma^1 :_0 \uparrow^1 \\ 1, & \text{if } \sigma^1 = 0, \sigma^0 :_0 \uparrow^1 \end{cases} \tag{4}$$

where $\sigma^0 :_0 \uparrow^1$ shows a raising edge (transition from logic 0 to logic 1) of σ^0 and $i \in \{0,1,2,\ldots\}$. Fig. 7 shows the corresponding output bits for an input of the block $\mathcal{S}_3$. This bit generation method basically aims at characterizing the jumps in signal $x(t)$ from one scroll to the other or staying at the same scroll.

The threshold value c_1 is set to 0 Volt; c_2 is not determined at this point but is considered as a design parameter to further improve the statistical properties of the bit sequence, as will be further explained in the next section. The signals $y(t)$ and $z(t)$ do not directly effect the bit generation. Therefore, when one compares the proposed RBG with a RNG using 1D chaotic map, one recognizes that the proposed RBG brings additional difficulties to the opponent in cryptography.

V. De-Skewing and Setting Threshold c_2

A natural source of random bits may not give unbiased bits as direct output [23], [21]. Many applications, especially in cryptography, rely on sequences of unbiased bits. There are various techniques to extract unbiased bits from a defective generator with unknown bias. These are called *de-skewing techniques* [31]. These techniques also eliminate the correlation in the output of the natural sources of random bits. This problem is also thought as simulating a fair (unbiased) coin using a biased coin. Von Neumann [32] has probably been the first author to state this problem. He proposed a digital post-processing that balances the distribution of bits. Post-processing converts nonoverlapping pairs of bits into output bits by converting the bit pair 0 1 into an output 0, converting 1 0 into an output 1 and pairs 1 1 and 0 0 are discarded [31]. In [10] the authors used the fast Fourier transform (FFT) for de-skewing in order to obtain uncorrelated and unbiased numbers from the disks timing data. Also, cryptographic hash functions are used as another de-skewing technique by taking advantage of their statistical property [31]. In this paper Von Neumanns technique is used

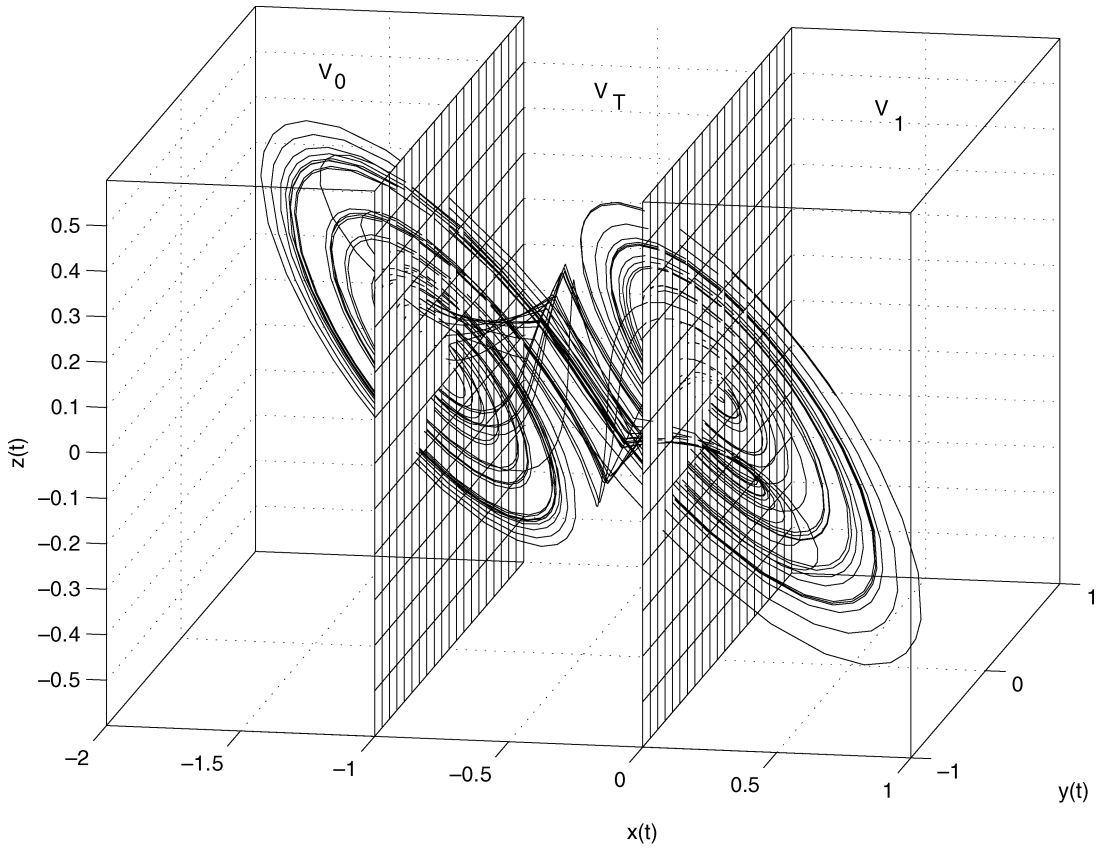


Fig. 5. Discretizing the state space for $c_1 = 0$ and $c_2 = -1$.

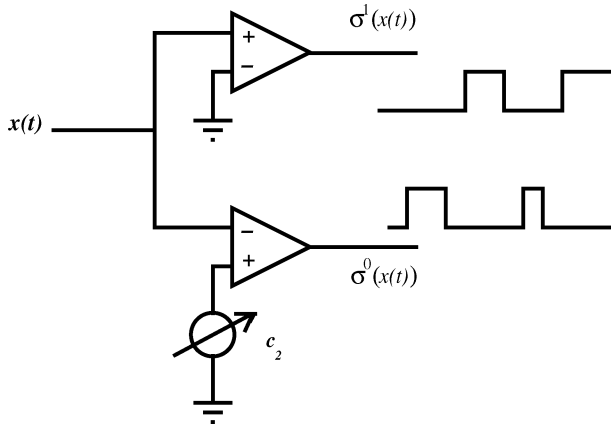


Fig. 6. Additional circuit to generate binary sequences.

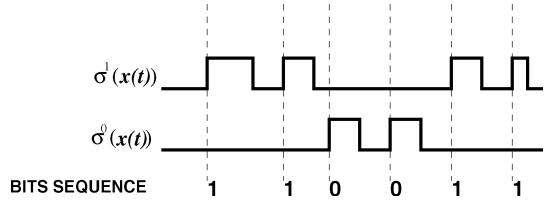


Fig. 7. Bits sequence from the comparator outputs.

because it can easily be integrated into the hardware and it is not decreasing the bit rate too much compared with the other methods. The de-skewing block $\mathcal{S}_4$ which is used in this paper, is described by

$$\mathcal{S}_4: \; b_i(\sigma_i, \sigma_{i-1}) = \begin{cases} 0, & \text{if } \sigma_i = 0 \bigwedge \sigma_{i-1} = 1 \\ 1, & \text{if } \sigma_i = 1 \bigwedge \sigma_{i-1} = 0. \end{cases} \tag{5}$$

In order to obtain an unbiased bit sequence, the bit generator part is combined with a simple subroutine of the de-skewing block $\mathcal{S}_4$. The proposed TRBG has a single output which is denoted by $\mathcal{B} = \{\ldots, b(i-1), b(i), b(i+1), \ldots\}$ and $b(i) \in \{0, 1\}$. Statistical properties of the TRBG can be adjusted by the threshold c_2 value which serves as a control parameter for the TRBG. Using the fact in information theory that noise has maximum entropy, the threshold value c_2 is chosen such that the measured entropy of the TRBG is maximal.

The measure-theoretic entropy [17] of the proposed TRBG with respect to a partition c_2 is given by

$$h^{c_2} = \lim_{n \to \infty} \frac{H_n^{c_2}}{n}$$

where

$$H_n^{c_2} = -\sum_{\mathcal{B}^n} P(\mathcal{B}^n) \ln P(\mathcal{B}^n) \tag{6}$$

with $P(\mathcal{B}^n)$ the probability of occurrence for binary subsequence (words

$$\{\ldots, \{b(i), b(i+1), \ldots, b(i+n-1)\}, \{b(i+n), \ldots, b(i+2n-1)\}, \ldots, \}$$

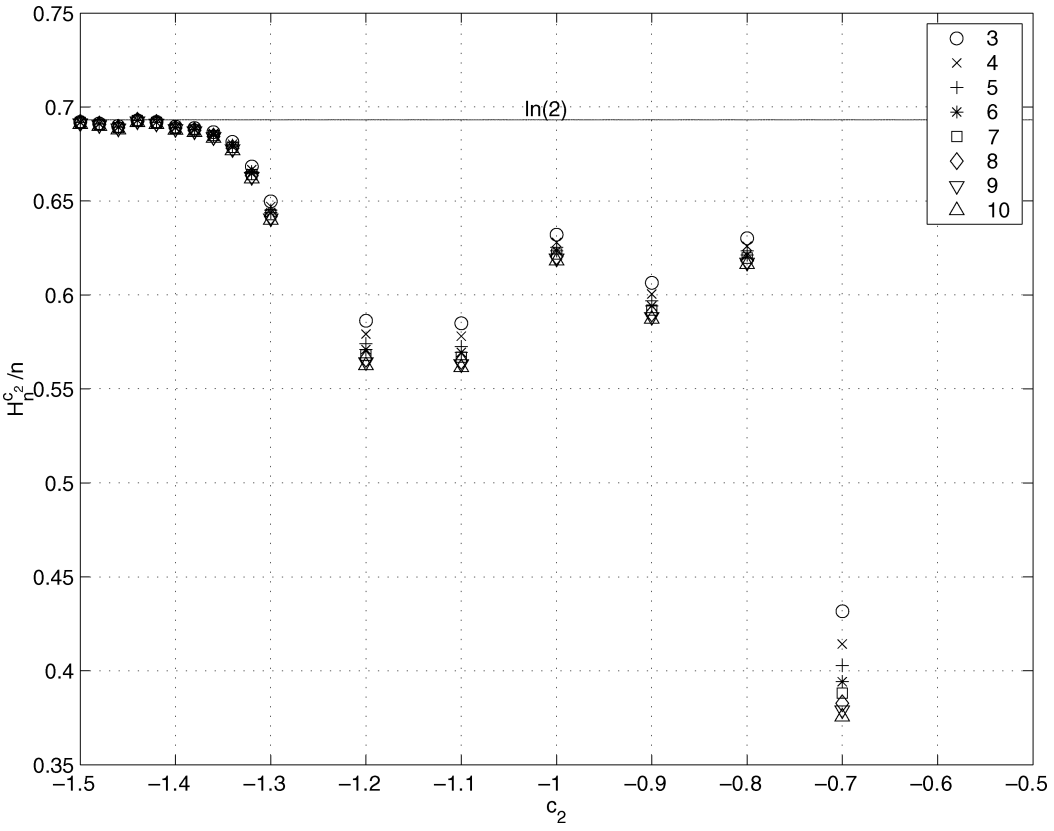


Fig. 8. $H_n^{c_2}/n$ as a function of the threshold c_2 for $n = 3, \ldots, 10$.

of length n[27], [17], [39]. $H_n^{c_2}$ achieves its maximum when the n-word sequences in the output sequence $\mathcal{B}$ of the TRBG are equally distributed.

To find the maximum entropy with respect to the threshold c_2, 1 000 000 bits were generated for 17 candidate thresholds values. Using (6), $H_n^{c_2}/n$ was calculated for $n = 3, 4, \ldots, 14$ (Fig. 8) shows values for $n = 3, 4, \ldots, 10$. (Fig. 9) shows how the proposed TRBG can come close the maximum entropy ($\ln 2$) which might be possible for a TRBG. Threshold c_2 is therefore set to -1.44 Volt.

VI. Statistical Tests

A large number of statistical tests and whole test suites have been proposed to assess the statistical properties of RNGs. Truly random sequences are e.g., likely to exhibit the same number of 0's and 1's in the sequence. A finite set of statistical tests may only detect defects in statistical properties. Furthermore, the tests cannot verify the unpredictability of the random sequence which is demanded in cryptographic applications. In this paper unpredictability of the sequence is a consequence of having a chaotic signal as a source.

In this section, the proposed random bit generator is faced with statistical tests. The statistical tests are applied to a bit sequence obtained after the de-skewing. As introduced in the previous section, de-skewing maintains bias and correlation and it also maintains the other statistical properties. A better result of the statistical test can be obtained, if a strong de-skewing technique is used such as a cryptographic hash function [31], but these are more time-consuming.

Before we present the test results, the tests that are applied in the next subsection are introduced (details of the tests can be found in [31], [25]):

Monobit test: The purpose of this test is to determine whether the number of 0's and 1's in the bitstream are approximately the same.

Poker test: The poker test determines whether the sequences of length m appear approximately the same number of times in the bitstream. The bitstream is divided into k ($k = \lfloor n/m \rfloor$) nonoverlapping parts each of length m. Let n_i be the number of occurrences of the i^{th} type of sequence of length $m = 1, 2, \ldots, 2^m$. The statistic is then

$$X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} {n_i}^2\right) - k \tag{7}$$

which approximately follows a χ^2 distribution [25], [31] with $2^m - 1$ degrees of freedom.

Runs test: A run of bitstream is a subsequence of it consisting of consecutive 0's or consecutive 1's. A run of 0's is called a gap, while a run of 1's is called a block. The runs test determines whether the number of runs of various lengths in the bitstream is as expected for a random sequence. The statistic used is

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i} \tag{8}$$

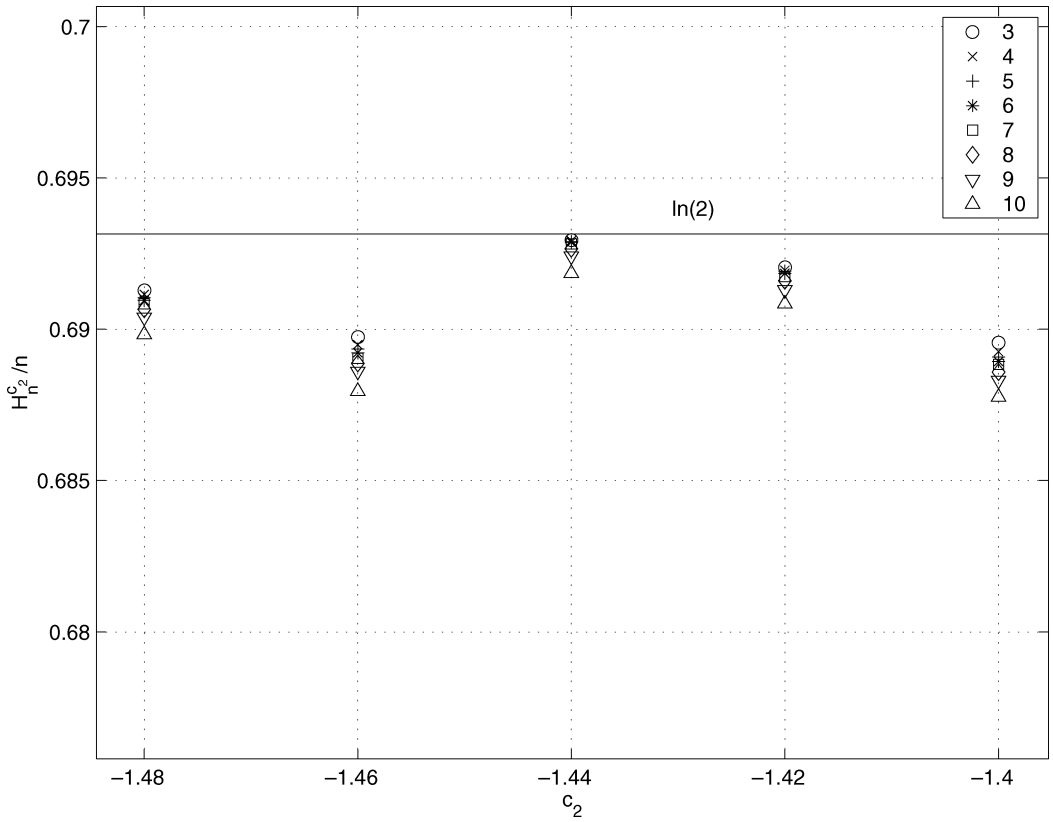


Fig. 9. $H_n^{c_2}/n$ for $n = 3, \ldots, 10$ where $-1.5 \leq c_2 \leq -1.3$.

with $e_i = (n - i + 3)/2^{i+2}$ (the expected number of gaps and blocks of length i) where n is the length of the random sequence and B_i, G_i is the number of blocks and gaps, respectively, of length i in the bitstream for each $i = 1, 2, \ldots, k$. X_4 approximately follows a χ^2 distribution [25], [31] with $2^k - k$ degrees of freedom. It is noted by Knuth [25] that the runs test is a strong test and is used in [44].

A. FIPS 140 – 1 Statistical Tests for Randomness

FIPS are issued by the National Institute of Standards and Technology (NIST). FIPS 140 – 1 covers security requirements for cryptographic modules and specifies recommended statistical tests for RNGs. Here, a single bitstream of length 20 000 bits as output from the RNG is subject to each of the tests. If one of the tests fails, the generator fails the test [31].

- *Monobit test:* The number of 1's (N_1)in the bitstream should satisfy $9654 < N_1 < 10\,346$.
- *Poker test:* The statistic X_3 is computed for $m = 4$. The poker test is passed if $1.03 < X_3 < 57.4$.
- *Runs test* The number of blocks B_i and gaps G_i of length i in the bitstream are counted for each $i = 1, \ldots, 66$. The runs of length greater than 6 are considered to be of length 6. The runs test is passed if the 12 counts B_i and G_i ($i = 1, \ldots, 6$) are each within the corresponding interval given in Table I.
- *Long-run test:* The long run test is passed if there are no runs of length 34 or more.

The proposed TRBG successfully passes all four statistical tests for every run. Table II provides results of five consecutive runs for the FIPS 140 – 1 specified tests.

TABLE I
RUNS TEST TABLE

Length of run	Required interval
1	$2267 - 2733$
2	$1079 - 1421$
3	$502 - 748$
4	$223 - 402$
5	$90 - 223$
6+	$90 - 223$

B. Diehard Test Suite

For further testing, we use the software package called Diehard [30] that has been designed for empirical testing. Diehard includes a battery of about 200 instances of fifteen traditional statistical tests which are birthday spacings, overlapping permutations, ranks of 31×31 and 32×32 matrices, ranks of 6×8 matrices, monkey tests on 20-bit Words, monkey tests OPSO, OQSO, DNA, count the 1's in a stream of bytes, count the 1's in specific bytes, parking lot, minimum distance, random spheres, squeeze, overlapping sums, runs, and craps.

The Diehard test suite has been applied here to 80 million bits (http://www.esat.kuleuven.ac.be/~mey/Ds2RbG/ds2rbg.bin) generated by the system proposed in this paper. It passes the full suite of Diehard. The complete test results can be further verified at http://www.esat.kuleuven.ac.be/~mey/Ds2RbG/Ds2RbG.html.

TABLE II
RESULTS OF FIPS-140–1 SPECIFIED TESTS

Run #	Monobit Test	Poker Test	Runs Test		Long Run Test
1	$N_1 = 9943$	$X_3 = 13.9648$	$B_1 = 2614$	$G_1 = 2540$	No long run
			$B_2 = 1201$	$G_2 = 1271$	
			$B_3 = 629$	$G_3 = 624$	
			$B_4 = 313$	$G_4 = 309$	
			$B_5 = 154$	$G_5 = 154$	
			$B_{6+} = 144$	$G_{6+} = 157$	
Passed Test ?	**YES**	**YES**	**YES**		**YES**
2	$N_1 = 9813$	$X_3 = 17.0496$	$B_1 = 2683$	$G_1 = 2559$	No long run
			$B_2 = 1224$	$G_2 = 1230$	
			$B_3 = 614$	$G_3 = 633$	
			$B_4 = 280$	$G_4 = 354$	
			$B_5 = 146$	$G_5 = 155$	
			$B_{6+} = 140$	$G_{6+} = 155$	
Passed Test ?	**YES**	**YES**	**YES**		**YES**
3	$N_1 = 10011$	$X_3 = 18.1248$	$B_1 = 2536$	$G_1 = 2595$	No long run
			$B_2 = 1269$	$G_2 = 1240$	
			$B_3 = 662$	$G_3 = 610$	
			$B_4 = 291$	$G_4 = 299$	
			$B_5 = 149$	$G_5 = 149$	
			$B_{6+} = 149$	$G_{6+} = 163$	
Passed Test ?	**YES**	**YES**	**YES**		**YES**
4	$N_1 = 10052$	$X_3 = 14.9696$	$B_1 = 2513$	$G_1 = 2482$	No long run
			$B_2 = 1215$	$G_2 = 1277$	
			$B_3 = 607$	$G_3 = 634$	
			$B_4 = 343$	$G_4 = 308$	
			$B_5 = 173$	$G_5 = 153$	
			$B_{6+} = 145$	$G_{6+} = 142$	
Passed Test ?	**YES**	**YES**	**YES**		**YES**
5	$N_1 = 10068$	$X_3 = 24.4224$	$B_1 = 2471$	$G_1 = 2498$	No long run
			$B_2 = 1318$	$G_2 = 1289$	
			$B_3 = 609$	$G_3 = 653$	
			$B_4 = 299$	$G_4 = 284$	
			$B_5 = 162$	$G_5 = 155$	
			$B_{6+} = 161$	$G_{6+} = 141$	
Passed Test ?	**YES**	**YES**	**YES**		**YES**

VII. CONCLUSION

This paper addresses a most encouraging application of chaotic signals in engineering which is its use as a source for RNGs. Although RNGs have been presented as a possible application of chaotic signal for several years, up till now the use of a continuous time signal from a chaotic circuit has never been shown with full empirical tests results. In this paper, we presented a random bit generator which uses a double-scroll attractor from a simple circuit model. The proposed TRBG is subjected to statistical tests using the well-known tests suites FIPS-140–1 and Diehard in cryptography. The proposed TRBG successfully passes all four statistical tests of the FIPS-140–1 test suite for all runs. A strong and more complete test suite Diehard has been applied to 80 million bits. The system successfully passes the full suite of Diehard test. Another essential result of this paper is that the proposed TRBG implementation can be fully integrated. Therefore, it may become a standard part in hardware.

Interesting further research directions in this field may include the following.

- One important question might be the bit rate of the proposed RBG which has not been discussed in this paper. Here, we wanted to show that a simple chaotic oscillator can be used for a random bit generator. The bit rate of the proposed RBG depends on the normalized quantity τ of the designed chaotic oscillator. This means that the chaotic oscillator must be designed for a high frequency range when a high bit rate is required. Therefore, this work shows again the importance of high frequency designs of chaotic oscillators for applications like this one.
- Statistical tests have shown that the proposed RBG has good statistical properties. However, unpredictability of the system has not been tested yet. More complex attractors such as n-scrolls or scroll grid attractors may further improve unpredictability at this point.
- One possible cryptanalytic opponent technique might be synchronization of the chaotic system [42]. The aim is ba-

sically then to make a one step ahead prediction of the bits sequence by using an identical chaotic system that is driven by a binary signal coming from the proposed RBG. However, even given a full trajectory of the chaotic signal for reaching synchronization this is a challenge in chaotic communications [26]. By using only binary information where a state variable of the trajectory is located as a driver, it is not easy to synchronize two chaotic systems and eventually to predict the next bit.

REFERENCES

[1] V. Bagini and M. Bucci, "A design of reliable true random number generator for cryptographic applications," in *Proc.1st Int. Workshop Cryptographic Hardware and Embedded Systems*, vol. LCNS 1717, 1999, pp. 204–218.

[2] G. Boffetta, M. Cencini, M. Falcioni, and A. Vulpiani, "Predictability : A way to characterize complexity," *Phys. Rep.*, vol. 356, pp. 367–474, 2002.

[3] S. Callegari, A. Cesaroni, and G. Setti, "Mixed mode unpredictable binary information source exploiting complex dynamics and adaptive median thresholding," in *Proc. 9th Workshop on Nonlinear Dynamics of Electronic Systems (NDES'01)*, 2001, pp. 65–68.

[4] S. Callegari, R. Rovatti, and G. Setti, "Spectral properties of chaos-based FM signals: Theory and simulation results," *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 3–15, Jan. 2003.

[5] M. Casdagli, "Nonlinear prediction of chaotic time series," *Phys. D*, vol. 35, pp. 335–356, 1989.

[6] G. Chen and T. Ueta, *Chaos in Circuits and Systems*. Singapore: World Scientific, 2002.

[7] L. O. Chua, M. Komuro, and T. Matsumoto, "The double-scroll family," *IEEE Trans. Circuits Syst. I*, vol. CAS-33, pp. 1072–1118, Oct. 1986.

[8] L. O. Chua, W. Wu, A. Huang, and G. Zhong, "A universal circuit for studying and genereting chaos-part II: Strange attractors," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 745–761, July 1993.

[9] (1999) The Intel random number generator. Techn. Rep.. Intel Corporation, Portland, OR. [Online]. Available: http://www.intel.com/design/security/rng/techbrief.htm

[10] D. Davis, R. Ihaka, and P. Fenstermacher, "Cryptographic randomness from air turbulence in disk drives," in *Proc. Advances in Cryptology (CRYPTO '94)*, vol. LNCS 0839, 1994, pp. 114–120.

[11] J. Doyne and J. J. Sidorowich, "Predicting chaotic time series," *Phys. Rev. Lett.*, vol. 59, no. 8, pp. 845–848, 1987.

[12] D. Eastlake, S. Crocker, and J. Schiller, "RFC 1750: Randomness recommendations for security," Network Working Group, Tech. Rep., 1994.

[13] A. S. Elwakil and M. P. Kennedy, "High frequency Wien-type chaotic oscillator," *Electron. Lett.*, vol. 34, no. 12, pp. 1161–1162, 1998.

[14] ——, "Construction of classes of circuit-independent chaotic oscillators using passive-only nonlinear devices," *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 289–307, Mar. 2001.

[15] A. S. Elwakil, K. N. Salama, and M. P. Kennedy, "An equation for generating chaos and its monolithic implementation," *Int. J. Bifurcation Chaos*, vol. 12, no. 12, pp. 2885–2896, 2002.

[16] R. C. Fairfield, R. L. Mortenson, and K. B. Coulthart, "An LSI random number generator (RNG)," in *Proc. Advances in Cryptology (CRYPTO '87)*, vol. LNCS 0196, 1987, pp. 203–230.

[17] A. M. Fraser, "Information and entropy in strange attractors," *IEEE Trans. Inform. Theory*, vol. 35, pp. 245–262, Mar. 1989.

[18] J. A. González, L. I. Reyes, J. J. Suarez, L. E. Guerrero, and G. Gutiérrez, "A mechanism for randomness," *Phy. Lett. A*, vol. 295, no. 1, pp. 25–34, 2002.

[19] J. Guckenheimer, "Toolkit for nonlinear dynamics," *IEEE Trans. Circuits Syst. I*, vol. CAS-30, pp. 586–590, Aug. 1983.

[20] M. Guide, "Concept for a high-performance random number generator based on physical random phenomena," *Freq.*, vol. 39, pp. 187–190, 1985.

[21] T. S. Han and M. Hoshi, "Interval algorithm for random number generation," *IEEE Trans. Inform. Theory*, vol. 43, pp. 599–611, Mar. 1997.

[22] M. Jakobsson, E. Shriver, B. K. Hillyer, and A. Juels, "A practical secure physical random bit generator," in *Proc. 5th ACM Conf. Computer and Communications Security*, 1998.

[23] A. Juels, M. Jakobsson, E. Shriver, and B. K. Hillyer, "How to turn loaded dice into fair coins," *IEEE Trans. Inform. Theory*, vol. 46, pp. 911–921, May 2000.

[24] B. Jun and P. Kocher. (1999) The Intel random number generator. Tech. Rep., White Paper. Intel Corporation, Cryptography Research Inc., Portland, OR. [Online]. Available: http://www.intel.com/design/security/rng/CRIwp.htm

[25] D. E. Knuth, *The Art of Computer Programming*. Reading, MA: Addison-Wesley, 1969.

[26] G. Kolumban, M. P. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos-Part II: Chaotic modulation of digital communications," *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 1129–1140, Nov. 1998.

[27] D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[28] R. N. Madan, *Chua's Circuit: A Paradigm for Chaos*. Singapore: World Scientific, 1993.

[29] D. P. Maher and R. J. Rance, "Random number generators founded on signal and information theory," in *Proc.1st Int. Workshop Cryptographic Hardware and Embedded Systems*, vol. 1717, 1999, pp. 219–230.

[30] G. Marsalgia. Diehard. [Online]. Available: http://stat.fsu.edu/~geo/diehard.html

[31] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. (1997) Handbook of applied cryptography. [Online]. Available: http://www.cacr.math.uwaterloo.ca/hac/

[32] J. Von Neumann, "Various techniques used in connection with random digits," in *Applied Math Series*, G. E. Forsythe, Ed. Boulder, CO: National Bureau of Standards, 1951, vol. 12, pp. 36–38.

[33] S. Özoğuz and N. S. Şengör, "On the realization of NPN-only log-domain chaotic oscillators," *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 291–294, Feb. 2003.

[34] A. G. Radwan, A. M. Soliman, and A. El-Sedeek, "MOS realization of the double-scroll-like chaotic equation," *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 285–288, Feb. 2003.

[35] (1999) Hardware-based random number generation. Tech. Rep., An RSA Data Security White Paper. RSA Data Security. [Online]. Available: http://www.intel.com/design/security/rng/Intel_RNG_v2.htm

[36] (1999) Using RSA BSAFE Crypto-c With Intel Random Number Generator. Tech. Rep., An RSA Data Security White Paper. RSA Data Security. [Online]. Available: http://www.intel.com/design/security/rng/RSA_BSAFE.htm

[37] A. Shamir, "On the generation of cryptographically strong pseudo-random sequences," *ACM Trans. Comp. Syst.*, vol. 1, no. 1, pp. 38–44, 1983.

[38] M. Small and C. K. Tse, "Detecting determinism in time series: The method of surrogate data," *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 663–672, May 2001.

[39] T. Stojanovski and L. Kocarev, "Chaos-based random number generators-part I: Analysis," *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 281–288, Mar. 2001.

[40] J. A. K. Suykens, A. Huang, and L. O. Chua, "A family of n-scroll attractors from a generalized Chua's circuit," *Arch. Elektron. Ubert.*, vol. 51, no. 3, pp. 131–138, 1997.

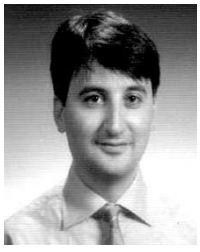
[41] J. A. K. Suykens and J. Vandewalle, "The K.U.Leuven time series prediction competition," in *Nonlinear Modeling: Advanced Black-Box Techniques*, J. A. K. Suykens and J. Vandewalle, Eds., 1998, pp. 241–253.

[42] J. A. K. Suykens, M. E. Yalçın, and J. Vandewalle, "Chaotic systems synchronization," in *Chaos Control: Theory and Applications*, G. Chen and X. Yu, Eds. New York: Springer-Verlag, 2003, vol. 292, pp. 117–136.

[43] T. Kohda and A. Tsuneda, "Statistics of chaotic binary sequences," *IEEE Trans. Inform. Theory*, vol. 43, pp. 104–112, Jan. 1997.

[44] B. Vizvari and G. Kolumban, "Quality evaluation of random numbers generated by chaotic sampling phase-locked loops," *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 216–224, Mar. 1998.

[45] M. E. Yalçın, J. A. K. Suykens, J. Vandewalle, and S. Özoğuz, "Families of scroll grid attractors," *Int. J. Bifurcation Chaos*, vol. 12, no. 1, pp. 23–41, 2001.



Müştak E. Yalçın (S'03) was born in Unye, Turkey, in 1971. He recieved the B.Sc. and M.Sc. degrees in electronics and telecommunications engineering from the Istanbul Technical University, Istanbul, Turkey, in 1993 and 1997, respectively. He is currently working toward the Ph.D. degree in applied sciences from the Katholieke Universiteit, Leuven, Belgium.

His research interests include nonlinear circuits and systems, neural networks and their applications, in particular, cellular nonlinear networks, multiscroll chaotic attractors, and spatio-temporal waves and synchronization.



Johan A. K. Suykens (M'03) was born in Willebroek, Belgium, in 1966. He received the degree in electromechanical engineering and the Ph.D. degree in applied sciences from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1989 and 1995, respectively.

In 1996, he was a Visiting Postdoctoral Researcher at the University of California, Berkeley. He was a Postdoctoral Researcher with the Fund for Scientific Research (FWO) Flanders, Belgium, and is currently an Associate Professor with Katholieke Universiteit Leuven. His research interests are mainly in the areas of the theory and application of neural networks and nonlinear systems. He is author of the books *Artificial Neural Networks for Modeling and Control of Nonlinear Systems* (Norwell, MA: Kluwer, 1995) and *Least Squares Support Vector Machines* (Singapore: World Scientific, 2002) and editor of the books *Nonlinear Modeling: Advanced Black-Box Techniques* (Norwell, MA: Kluwer, 1998) and *Advances in Learning Theory: Methods, Models and Applications* (Amsterdam, The Netherlands: IOS Press, 2003).

Dr. Suykens received the IEEE Signal Processing Society 1999 Best Paper (Senior) Award and several Best Paper Awards at International Conferences. He is a recipient of the International Neural Networks Society 2000 Young Investigator Award for significant contributions in the field of neural networks. In 1998, he organized an International Workshop on Nonlinear Modeling with Time-series Prediction Competition. He has served as Director and Organizer of a NATO Advanced Study Institute on Learning Theory and Practice, Leuven, Belgium,July 2002. He has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS (1997–1999), and since 1998, he is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS.



Joos Vandewalle (F'92) was born in Kortrijk, Belgium, in 1948. He received the electrical engineering degree and the doctoral degree in applied sciences from the Katholieke Universiteit, Leuven, Belgium, in 1971 and 1976, respectively.

From 1976 to 1978, he was a Research Associate and from July 1978 to July 1979, a Visiting Assistant Professor at the University of California, Berkeley. Since July 1979, he has been with the Department of Electrical Engineering (ESAT), Katholieke Universiteit, where he is Full Professor since 1986 and the Head of the SCD division at ESAT, that has more than 120 researchers. From August 1996 to August 1999, he was Chairman of the Department of Electrical Engineering and from August 1999 till July 2002, he was the Vice-Dean, Faculty of Engineering, Katholieke Universiteit. since 1984, he is also an Academic Consultant at the Interuniversity Microelectronics Center, Leuven, Belgium. In the second semester of 2002–2003 he was on sabbatical leave at the I3S laboratory of French NAtional Center for Scientific Research (CNRS) Sophia Antipolis, France. He teaches courses in linear algebra, linear and nonlinear system and circuit theory, signal processing and neural networks. His research interests are mainly in mathematical system theory and its applications in circuit theory, control, signal processing, cryptography and neural networks. His recent research interests are in nonlinear methods (support vector machines, multilinear algebra..) for data processing. He has authored or coauthored more than 200 international journal papers in these areas. He is the co-author of four books and co-editor of five books. He is a member of the editorial board of the *International Journal of Circuit Theory and its Applications*, *Neurocomputing*, *Neural Networks*, and the *Journal of Circuits Systems and Computers*. Since 2001, he is a member of the Advisory Board of the *International Journal on Information Security* (IJIS). Since January 2001, he is Co-editor-in-Chief of *Journal A, the Benelux journal on Automation*.

Dr. Vandewalle received several Best Paper Awards and Research Awards. In 1991–1992, he held the Francqui Chair on Artificial Neural Networks at the University of Liége (Belgium), and in 2001–2002, he held the Chair on Advanced Data Processing techniques at the Free University of Brussels (Belgium). From 1989 to 1991, he was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS and its Deputy Editor-in-Chief from January 2002 to December 2003. He is a member of the Academia Europaea and of the Belgian Academy of sciences and of two Committees of the Fonds voor Wetenschappelijk Onderzoek Vlaanderen (Belgium). He is also Fellow of the Institute of Electrical Engineers, U.K.