Journal of Applied Mathematics & Bioinformatics, vol.5, no.3, 2015, 15-35
ISSN: 1792-6602 (print), 1792-6939 (online)
Scienpress Ltd, 2015

Secure Text Encryption Based on Hardware Chaotic Noise Generator

Christos K. Volos[1] **and Antonios S. Andreatos**[2]

Abstract

In recent years, the sharing of information has become more prevalent than the past due to the rapid development of Internet and mobile communication technologies. However, in open networks, there is a potential risk of making sensitive information, such as military orders vulnerable to unauthorized interceptions. The development of robust cryptographic schemes is thus essential to the provision of information security. During the last decade an interesting relationship between cryptography and chaos theory has been developed. Consequently, several chaos-based cryptosystems have been put forward. In this context, this paper presents a novel text message encryption scheme which is based on a Chaotic Random Number Generator. This system is realized by the Arduino, an open-source physical computing platform based on a simple microcontroller board. As a chaotic system, the well-known

[1] Physics Department, Aristotle University of Thessaloniki, Thessaloniki, GR-54124, Greece. E-mail: volos@physics.auth.com

[2] Div. of Computer Engineering and Information Science, Hellenic Air Force Academy, Dekeleia Air Force Base, Dekeleia, Attica, TGA-1010, Greece. E-mails: aandreatos.hafa@haf.gr, aandreatos@gmail.com.

Article Info: *Received* : November 2, 2014. *Revised* : February 2, 2015.
Published online : September 15, 2015.

Chua system is chosen, due to its rich dynamical behavior. The quality of the produced random number sequences is tested by the FIPS-140-2 statistical suite. The satisfactory results from the use of the aforementioned statistical tests, in regard to other previous works, confirm the usefulness of the proposed text encryption scheme.

Mathematics Subject Classification: 68P25
Keywords: Text encryption; chaos; chaotic random number generator; Chua circuit; Arduino microcontroller; FIPS-140-2 test suite

1 Introduction

In the last decades the success of many military operations critically depends on our ability to create confidential channels of communication. Especially in military operations, generals require command and control systems to relay orders down to the chain of command without the fear of enemy interception. So, in these operations secrecy is essential for their success. This secrecy is achieved with the use of cryptography.

Especially, the textual information security can be satisfied with the direct application of many well-established encryption schemes, such as the Data Encryption Scheme (DES), the International Data Encryption Algorithm (IDEA) and Advanced Encryption Scheme (AES) [1]. However, this field is constantly evolving, as a great number of research groups work in this field in order to achieve encryption systems with improved characteristics regarding their safety.

In addition, nonlinear systems and especially systems which excibit chaotic behavior have attracted the interest of the research community, due to the great number of applications in various scientific fields, such as social sciences, ecology, electronic circuits, lasers, chemical reactions, fluid dynamics, mechanical systems etc. [2,3]. Chaotic systems revealed that despite of the knowledge of their evolution rules and initial conditions, their future behavior seemed to be arbitrary and unpredictable.

Due to the aforementioned characteristic, an interesting relationship between chaos theory and cryptography has been developed in the last two

decades. This occurs because many properties of chaotic systems such as: sensitivity on initial conditions or system's parameters, ergodicity, deterministic dynamics and structural complexity can be considered analogous to the diffusion with small change in plaintext or secret key, confusion, deterministic pseudo-randomness and algorithmic complexity properties of traditional cryptosystems [4]. As a result of this close relationship, several chaos-based cryptosystems have been put forward since 1990 and play an important role in military operations because of the significant strategic advantage that these systems provide.

The robustness of many cryptographic systems is more and more based on random number generators. These generators can be classified into three major types: True Random Number Generators (TRNGs), Pseudo-Random Number Generators (PRNGs) and Hybrid Random Number Generators (HRNGs) [5], depending on the source of randomness.

This work is devoted to a novel Chaotic Random Number Generator (CRNG) based on a chaotic system (Chua system [6]), belonging to the well-known double-scroll family. The values of the systems' parameters and the initial conditions are the keys of the proposed cryptographic scheme. The produced random numbers sequences are used to encrypt and decrypt texts, which is a very useful application, especially in the case of military operations, as it was mentioned. The proposed cryptosystem is realized by the Arduino, an open-source physical computing platform based on a simple microcontroller board.

This paper is organized as follows. In Section 2 the definition of chaotic systems in general and the description of the Chua system, which has been used, is given. Section 3 introduces the dynamical systems of double-scroll chaotic attractors in which the chosen Chua circuit belongs. In Section 4 the random number generators are described and the proposed CRNG is presented in detail. Section 5 presents the results of the CRNG obtained by the well-known FIPS-140-2 statistical test suite. Section 6 demonstrates the encryption and decryption process of a text message, which is realized by the Arduino. Finally, conclusion remarks and some thoughts for future work are presented.

2 Chaotic Systems

The term "Chaos" refers to some dynamical phenomena that are considered to be complex and unpredictable. Although it was precluded by Poincaré at the end of the 19th century [7], chaos theory became popular in the second half of the 20th century [8,9] after observations on the evolution of different physical systems. These systems revealed that despite of the knowledge of their evolution rules and initial conditions, their future seemed to be arbitrary and unpredictable. This particular characteristic opened quite a revolution in modern physics, terminating with Laplace's ideas of casual determinism [10].

Chaos has been observed in many disciplines such as weather and climate [8], population growth in ecology [11], economy [12], to mention only a few examples. It also has been observed in the laboratory in a number of systems such as electrical circuits [13], lasers [14], chemical reactions [15], fluid dynamics [16], mechanical systems, and magneto-mechanical devices [17]. So, chaos theory provides the means to explain various phenomena in nature and make use of chaotic dynamical systems in many different scientific fields.

In essence, chaos theory studies systems that evolve in time, presenting three particular properties [18]:

- The system must be topologically mixing,
- its periodic orbits must be dense and
- it must be very sensitive on initial conditions.

Firstly, the term "topologically mixing" means that the chaotic trajectory at the phase space will move over time so that each designated area of this trajectory will eventually cover part of any particular region. The second feature of chaotic systems is that their periodic orbits have to be dense, in the sense that, the trajectory of a dynamical system is dense, if it comes arbitrarily close to any point in the domain. Finally, the third and probably the most important feature of chaotic systems, is the sensitivity on initial conditions. This means that a small variation on a system's initial conditions will produce a totally different chaotic trajectory, hence behavior.

In general, a dynamical system describes a physical phenomenon that evolves in time. In mathematical terms, the states of the system are described

by a set of variables and its evolution is given by a set of differential equations with the values of the initial states. This is summarized in Eq. (1),

$$\frac{dX_i(t)}{dt} = F_i(X_i(t), \Lambda) \tag{1}$$

where: $Xi\ (t) \in \Lambda^N$ is the coordinate i of the state of the system at instant time t, that is X is an N-dimensional vector, $i, j = 0, 1, ..., N$ with $N \geq 1$, F is a parametric nonlinear function which describes the evolution of the system and Λ is the vector of parameters that control the evolution of the system. It has been observed that this kind of systems is deterministic, thus the time evolution of X can be calculated with F and Λ from a given initial state X_0. In continuous-time dynamical systems, t takes continuous values, corresponding to the evolution of the system's dynamic behavior in time. On the other hand, discrete-time dynamical systems are described by the following equation:

$$X_{i+1} = F(X_i, \Lambda) \tag{2}$$

Apparently, Eq.(2) describes deterministic systems where the discrete's time evolution of X can be calculated with F and Λ for a given initial state X_0. These systems are also recursive as the next state is calculated from the previous one.

Furthermore, there are two major concepts or terms that are of special interest in the study of nonlinear dynamical systems. The first one is the phase space that is the subspace of Λ^N, where all possible states of the system are confined:

$$U \subset \Lambda^N, F : U \to U \tag{3}$$

where N is the dimension of the phase space or degree of freedom of the system. The evolution of an initial state in space with time is called orbit.

The second one is also another central concept in chaos theory, the attractor. The term attractor refers to the long-term behavior of the orbits, and it represents the region of phase space where the orbits of the system converge after the transitory. The attractor A is a compact region where all orbits converge and where the system gets trapped,

$$A \subset U, A = F(A) \tag{4}$$

Geometrically, an attractor can be a point, a curve, a manifold, or even a complicated set with a fractal structure known as a strange attractor. A brief description of these orbits is [19]:

- Fixed point: it corresponds to a stationary state of the system.
- Limit cycle: it is associated with a periodic behavior of the system. Once the system enters this attractor the states of the system repeat periodically.
- Manifold: there are more than one frequencies in the periodic trajectories of the system. For example, in the case of two frequencies, the attractor is a 2D-torus.
- Strange attractor: it is informally said to have a complex geometric shape with non-integer dimension. Any state in the attractor evolves within it and never converges to a fixed point, limit cycle or manifold. The dynamics on this attractor is normally chaotic, but also strange attractors that are not chaotic exist.

After reviewing the main characteristic of nonlinear dynamical systems, one could define the term chaotic system as a nonlinear dynamical system that has at least a chaotic strange attractor.

3 Systems with Double-Scroll Chaotic Attractors

In this work, the proposed encryption scheme is based on a nonlinear dynamical system capable of producing double-scroll strange attractors, which is an indication of generating chaotic behavior [20]. In Fig. 1 a double-scroll chaotic attractor in x-y-z phase space is shown. Attractors of this type are produced by nonlinear third order dynamical systems in order to have three equilibrium points. One of these equilibria is the origin $(x,\ y,\ z) = (0,\ 0,\ 0)$, while the other two are located at the center of the two attractors.

As it is shown, a typical trajectory of the double-scroll chaotic attractor rotates around one of these equilibrium points, getting further from it after each rotation until it goes back to a point closer to the equilibrium and either repeats the process or directs toward the other equilibrium point and repeats a similar process, but around the other equilibrium point. The important thing

is that in both cases the number of rotations is random. Next, the nonlinear dynamical system which is used at the design of the proposed chaotic random number generator is examined.

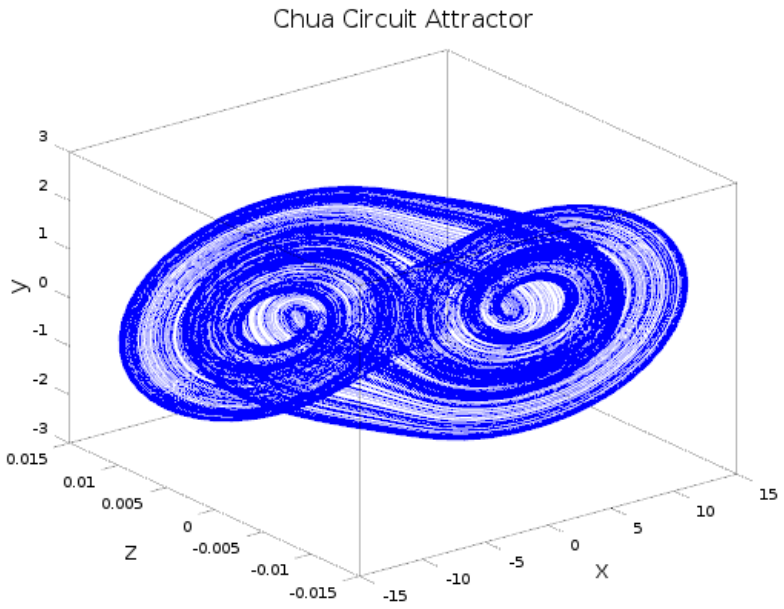


Figure 1: Double-scroll chaotic attractor in x-y-z phase space

3.1 The Chua System

At the beginning of '80s there was a deep desire in the scientific community for an experimental demonstration of chaos with electronic circuits, in order to refute the claim that this phenomenon was a mere mathematical invention. This led the well-known scientist Leon Chua to investigate the possibility of designing an autonomous circuit behaving in a chaotic way [21]. So, he decided to construct an autonomous chaotic circuit with three unstable equilibrium points, which also had the limitation of containing as few as possible passive elements (two capacitors, an inductor and a resistor) and only one two-terminal nonlinear resistor, known as the Chua's diode (NR), with piecewise-linear characteristic having an odd symmetry. In this way arose the Chua circuit (Fig.2(a)), which is the simplest electronic circuit exhibiting chaos. The Chua circuit can be described by the following dimensionless

system of equations:

$$\frac{dx}{d\tau} = \alpha\,[y - x - f(x)] \tag{5}$$

$$\frac{dy}{d\tau} = x - y + z \tag{6}$$

$$\frac{dz}{d\tau} = -\beta y \tag{7}$$

The state parameters $x = v/E$ and $y = v/E$ represent the voltages at the capacitors C_1 and C_2 while $z = i_L R/E$ is the current through the inductor L, as shown in Figure 2(a). The dimensionless time τ is $\tau = t/RC_2$ and the normalized parameters α and β are: $\alpha = C_2/C_1$ and $\beta = R^2C_2/L$ respectively.

The dimensionless form of the nonlinear function $f(x)$ (Fig. 2(b)) is given by the following equation:

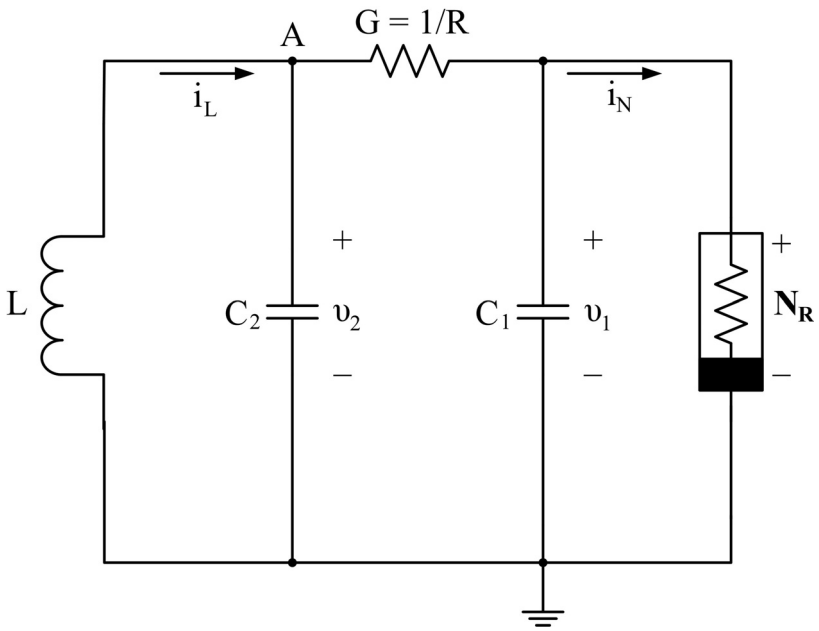
$$\mathrm{f\,(x)} = \begin{cases} bx + b - a, & \text{if } x \leq -1 \\ ax, & \text{if } |x| < 1 \\ bx + a - b, & \text{if } x \geq 1 \end{cases} \tag{8}$$

where $a = G_a/G$ and $b = G_b/G$. G_a and G_b are the slopes of the inner and outer segments and $\pm E$ are the breakpoints of the Chua's diode characteristic curve.

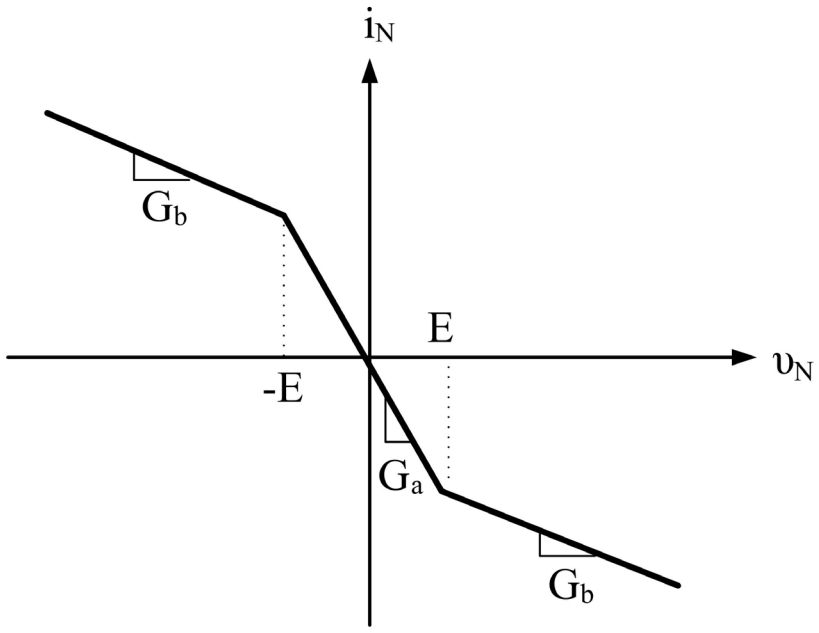
4 Chaotic Random Bit Generators

As already mentioned, the security of many cryptographic systems, especially in the last decade, was mainly based on random number generators. Generators that produce random sequences can be classified into three types [22]:

- True Random Number Generators (TRNGs),
- Pseudo-Random Number Generators (PRNGs) and
- Hybrid Random Number Generators (HRNGs).



(a)



(b)

Figure 2: (a) The Chua circuit and (b) the i_N - υ_N characteristic of the Chua's diode

TRNGs require a naturally occurring source of randomness, which comes from an unpredictable natural process in a physical or hardware device, such as the elapsed time during radioactive decay [23], the thermal and shot noise [24], the frequency instability of an oscillator [25], the variations in disk drive response times [26], the integrating dark current from a metal insulator semiconductor capacitor [27], the movement of a mouse in a PC [28] and the environmental noise [29]. However, designing a hardware device to exploit this randomness and producing a bit sequence that is free of biases and correlations is a difficult task. Additionally, for most cryptographic applications, the generator must not be subject to observation or manipulation by an adversary. Due to the fact that the TRNGs are based on natural sources of randomness, they are subject to influence by external factors which cause malfunctions.

All the above mentioned difficulties of obtaining uniform random sequences from TRNGs led many research teams to the design of pseudorandom number generators. A PRNG is a deterministic algorithm which outputs a number sequence of length $l >> k$ that "appears" to be random, if a number sequence of length k is given. The input to the PRNG is called the seed, while the output of the PRNG is called a pseudorandom number sequence. This number sequence is not truly random in that it is completely determined by a relatively small set of initial values and an algorithm. PRNGs are very important in practice for their speed in number generation, their portability and their reproducibility, and they are thus central in applications such as cryptography, decision making, simulation and data sampling [30,31]. So, a good PRNG for the aforementioned applications should possess three very important characteristics: long period, high speed and randomness.

Nevertheless, it is obvious that in PRNGs due to the fact that the output is a function of the seed state, the actual entropy of the output can never exceed the entropy of the seed. Hence, the randomness level of the pseudo-random numbers depends on the level of randomness of the seed. Thus, HRNGs have been proposed to 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 mouse movements, random keystrokes, or hard drive seek times.

During the last decade several ideas of designing random number generators, by using either continuous or discrete chaotic systems, have been proposed by academia and industry [32-38]. In this work, a step forward to this direction

has been done. The generator is realized by an open-source physical computing platform based on a microcontroller board, in order to study the effectiveness of the proposed method. It was very interesting to find out the strengths and the weaknesses of the specific proposal because the microcontroller gives us the opportunity to use the random number generator in many applications, such as in cryptography, robotics, etc.

The proposed generator is realized by using an open-source Arduino prototyping platform made up of an Atmel AVR ATMEGA 328 microcontroller based on flexible, easy-to-use hardware and software [39]. This platform has 14 digital input/output pins, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. The Arduino is connected to the computer through the USB port and programmed using the language "Wiring" which is similar to C++. The program, which is known as "sketch", is uploaded into the microcontroller using an Open Source Integrated Development Environment, the Arduino IDE. The microcontroller outputs the random number sequences by programming it with the sketches, which implement the proposed generator. The produced number sequenxes by the Arduino were captured to text files by using the HyperTerminal program on Windows. This is the simplest method for writing data from the Arduino to the serial port and save them to a file for processing.

4.1 The Proposed Chaotic Random Number Generator

The chaotic random number generator presented in this work consists of three blocks (Fig.3).

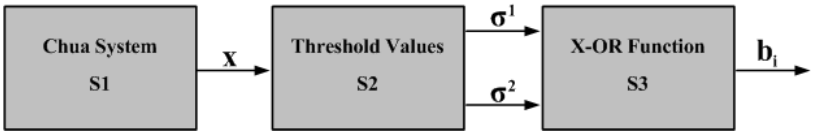


Figure 3: The scheme of the chaotic random number generator

The first block of the generator (S_1) includes the Chua system exhibiting the double-scroll chaotic behavior. In the proposed scheme the signal x, which has the desirable dynamic behavior, is used.

In the second block (S_2) the signal x is sampled with a step n = 500 (i.e., every 500 time steps) by using two threshold values c_1 and c_2. Thus, two different bitstreams σ^1 and σ^2 are produced, which are described by the following equations.

$$\sigma^1(x) = \begin{cases} 0, & \text{if } x < \mathrm{c}_1 \\ 1, & \text{if } x \geq \mathrm{c}_1 \end{cases} \tag{9}$$

$$\sigma^2(x) = \begin{cases} 0, & \text{if } x > \mathrm{c}_2 \\ 1, & \text{if } x \leq \mathrm{c}_2 \end{cases} \tag{10}$$

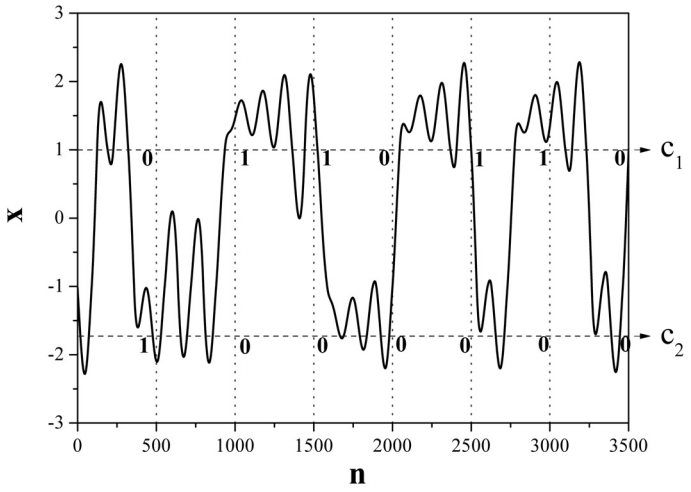


Figure 4: The chaotic signal x, produced by the Chua system, and the way that it is sampled in the second block

Finally, in the last block (S_3) the XOR function is used ($b_i = \sigma^1 \oplus \sigma^2$) and a third bitstream b is produced.

In comparison to other chaotic random number generators [40-42] the proposed scheme has a very simple structure, it is easily implemented either in software or in hardware and it produces an uncorrelated bitstream. Therefore, the use of de-skewing at the output of the proposed generator is not necessary, as opposed to the majority of similar CRNGs [40-42].

5 Statistical Tests of the CRNG

The invention of a foolproof source of random numbers it is not an easy task. In order to gain the confidence that a newly developed random bit generator is cryptographically secure, it should be subjected to a variety of statistical tests designed to detect the specific characteristics expected by truly random sequences.

In the bibliography there are several options available for analyzing the randomness of the newly developed random bit generators. The four most popular options are:

- the FIPS-140-2 (Federal Information Processing Standards) suite of statistical tests of the National Institute of Standards and Technology (NIST) [43],
- the DIEHARD suite of statistical tests [44],
- the Crypt-XS suite of statistical tests [45] and
- Donald Knuth's statistical tests set [46].

In this section the quality of the random number sequences produced by the proposed CRNG will be examined. The FIPS-140-2 test suite has been used. The results of the use of the four common statistical tests, namely, the Monobit test, the Poker test, the Runs test and the Long run test are presented in detail. According to the FIPS tests, the examined RNG will produce a bitstream, $b_i = b_0, b_1, b_2, ..., b_{n-1}$, of length n (at least 20,000 bits), which must satisfy the following conditions.

- Monobit Test: The number n_1 of 1's in the bitstream must be $9,725 < n_1 < 10,275$.
- Poker Test: This test determines whether the sequences of length n ($n = 4$) appear approximately for the same number of times in the bitstream. The bounds of this statistic are then $2.16 < X_3 < 46.17$.
- Runs Test: This test determines whether the number of 0's (Gap) and 1's (Block) of various lengths in the bitstream are as expected for a random sequence (Table 1).

- Long Run Test: This test is passed if there are no runs longer than 26 bits.

Table 1: Required intervals for length of runs test

Length of Run	Required Interval
1	2,315 - 2,685
2	1,114 - 1,386
3	527 - 723
4	240 - 384
5	103 - 209
6	103 - 209

Using the fact from the information theory that the noise has maximum entropy, the system's parameters ($\alpha = 15.6$, $\beta = 28.58$, $a = -1.14286$, $b = -0.714286$), the initial conditions ($x_0 = 0.5$, $y_0 = 0.25$, $z_0 = 0.125$), the threshold values ($c_1 = 1$, $c_2 = -1.5$) and the step $n = 500$ are chosen such that the measured entropy of the TRBG is maximum.

For the aforementioned values of system's parameters and initial conditions the measure-theoretic entropy [34] of the proposed CRNG is calculated to be $H_n = 0.693$ for $n = 3$ by using the following equation.

$$H_n = \lim_{n\to\infty} \left(-\sum_{B^n} P(B^n)(lnP(B^n) \right) /n \tag{11}$$

In the above equation $P(B^n)$ is the probability of occurrence of a binary subsequence B of length n.

By using the proposed CRNG, bits sequence of length 20,000 bits has been obtained via a numerical integration of Eqs.(5)-(7), which is subjected to the four tests of FIPS-140-2 test suite. Table 2 presents the results, which verifies that the produced bits sequence passes the FIPS-140-2 tests. It must be stressed that the proposed approach presents better results concerning this test suite in regard to a similar chaotic RNG [35]. Specifically, the increase in measure-theoretic entropy, compared to other works, is due to the use of a higher order nonlinear system, while the improvement of the results of FIPS-140-2 test suite is a consequence of the proposed CRNG's design and the appropriate selection of system's initial conditions and parameters.

Table 2: Results of FIPS-140-2 test, for the chaotic RNG

Monobit Test	Poker Test	Runs Test	Long Run Test
		$B_1 = 2,343$	
		$B_2 = 1,201$	
$N_1 = 9,990$	4.996	$B_3 = 683$	Passed
50.005%		$B_4 = 310$	
		$B_5 = 160$	
		$B_6 = 146$	

6 The Encryption Scheme

The main advantage of the majority of chaotic random number generators referenced here is that they consist software implementations, hence everything is ideal. In this work, a step forward to the realization of a CRNG via a microcontroller is done (i.e., hardware implementation). In general, the microcontroller combines the software programming with the hardware (processor) for doing a specific job. So, it is very interesting to find out the strengths and the weaknesses of the specific proposal because the microcontroller gives us the opportunity for using CRNGs in many applications, such as in robotics, cryptography, etc.

The well-known open-source Arduino prototyping platform was used in this work for realizing the proposed CRNG [39]. Arduino was chosen because, among other advantages, it is low-cost prototyping platform used in a great number of applications.

In this work the microcontroller outputs a random number sequence by programming it with the sketch (Arduino code) which implements the proposed text encryption system. The system consists of a programmed Adruino connected to a PC via a USB cable. The user enters the message in the Serial Monitor text-box and the result appears in the output text-area of the Serial Monitor (Fig. 5).

The procedure followed in this work consists of the following steps:

- **Step 1**: The user types their message in the textbox provided.
- **Step 2**: Each character of the original message (ASCII characters) is converted to the equivalent decimal number (e.g. $A \rightarrow 65$).

- **Step 3**: The CRNG creates a 5-bit random number for each message's character (from 0 [00000] to 31 [11111]).
- **Step 4**: The program adds this 5-bit random number to the corresponding message's character to produce the ciphertext character (another printable ASCII character).

Because we want the ciphertext to be printable, the message cannot use lowercase ASCII characters. The last uppercase ASCII character is z, corresponding to decimal value 90. To this, a chaotic value from 0 to 31 will be added, producing a cipher-character in the range Z (decimal 90) to y (decimal 121). Because the last printable ASCII character is the tilde ("$\sim$", corresponding to 126 decimal), the ciphertext will always have printable ASCII characters. Thus, the cleartext may use all printable ASCII characters from *space* (decimal 20) to Z (decimal 90) except uppercase letters and the characters [‘, {, |, } and $\sim$].

In the receiver, the reverse process takes place. So, an identical CRNG produces the same sequence of pseudo-random numbers and subtracts them from the corresponding ciphertext characters.

To harden the algorithm, we run the RNG for a random number of rounds and throw away the first N random numbers, which is part of the encryption key with the system's parameters and initial conditions.

7 Conclusion

In this work a text encryption scheme based on a chaotic true random number generator was presented. The main element of this CRNG was a microcontroller, which was used as a platform for implementing the proposed CRNG. A well-known dynamical system with a double-scroll chaotic behavior, namely the Chua circuit, was chosen for producing the sequence of random numbers based on a simple but effective RNG scheme. The use of the FIPS-140-2 test suite confirmed the good quality "randomness" of the produced number sequences presenting very good statistical results.

The proposed text encryption scheme has some advantages over other previous similar works. The use of an open-source physical computing platform

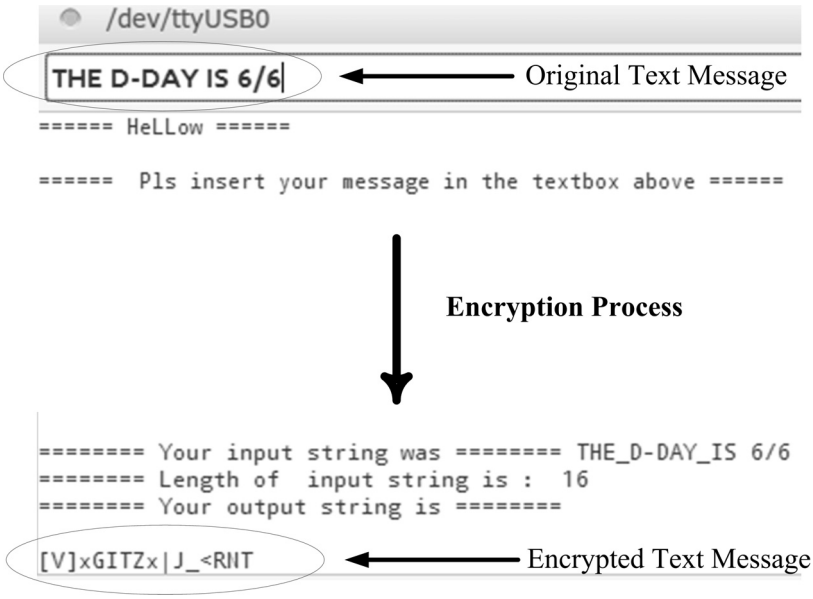


Figure 5: The text encryption process using the Arduino

for implementing the proposed text encryption scheme, is a step forward for making chaotic systems very useful in real-time encryption procedures outside of a software simulation where everything is ideal. Also, the specific use of the Arduino board, a low-cost and portable microcontroller platform, makes this encryption scheme a good candidate for use in many real-world cryptographic applications, such as in military operations. As a future work, we consider the use of an SD card for storing the ciphertext for making the system more useful.

References

[1] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, CRC Press, 2008.

[2] L.D. Kiel and E.W. Elliott, *Chaos Theory in the Social Sciences: Foundations and Applications*, Michigan University Press, 1997.

[3] S. Banerjee, *Applications of Chaos and Nonlinear Dynamics in Engineering*, Vol. 1, Springer, 2011.

[4] G. Alvarez and S. Li, Some Basic Cryptographic Requirements for Chaos Based Cryptosystems, *Int. J. Bifurcat.*, Chaos, **16**, (2006), 2129 - 2151.

[5] T. Shu, *Uniform Random Numbers: Theory and Practice*, Kluwer Academic Publishers, 1995.

[6] L.O. Chua, Chua's Circuit 10 Year Later, *Int. J. Bifurcat. Chaos*, **22**, (1994), 279 - 305.

[7] J.H. Poincarè, Sur le Problème des Trois Corps et les *É*quations de la Dynamique, Divergence Des Séries de M. Lindstedt, *Acta Mathematica*, **13**, (1890), 1 - 270.

[8] E.N. Lorenz, Deterministic Non-periodic Flow, *Journal of the Atmospheric Sciences*, **20**, (1963), 130 - 141.

[9] B. Mandelbrot, *The Fractal Geometry of Nature* , (Ed. 1), W.H. Freeman & Company, ISBN-13: 978-0-7167-1186-5, 1977.

[10] P.S. Laplace, *Trait du Mcanique Cleste. Oeuvres Completes de Laplace*, Gauthier-Villars, Paris, vol. 5, 1825.

[11] R.M. May and A.R. McLean, *Theoretical Ecology: Principles and Applications*, Blackwell, 2007.

[12] C. Kyrtsou and C. Vorlow, *Complex Dynamics in Macroeconomics: A Novel Approach*, In: New Trends in Macroeconomics, C. Diebolt and C. Kyrtsou (eds.), ISBN-13: 978-3-540-21448-9, Springer Verlag, 2005, pp. 223-245.

[13] B. van der Pol and J. van der Mark, Frequency Demultiplication, *Nature*, **120**, (1927), 363 - 364.

[14] L.W. Casperson, *Gas Laser Instabilities and Their Interpretation*, In: Instabilities and Chaos in Quantum Optics II. Proceedings of the NATO Advanced Study Institute, Italy, June 1987 pp. 83 - 98, Springer Verlag, 1988.

[15] R.J. Field and L. Gyrgyi, *Chaos in Chemistry and Biochemistry*, World Scientific Publishing, 1993.

[16] G.L. Baker, *Chaotic Dynamics: An Introduction*, Cambridge University Press, 1996.

[17] F.C. Moon, *Chaotic Vibrations: An Introduction for Applied Scientists and Engineers*, Wiley, 1987.

[18] B. Hasselblatt and A. Katok, *A First Course in Dynamics: With a Panorama of Recent Developments*, University Press, Cambridge, 2003.

[19] K.T. Alligood, T.D. Sauer and J.A. Yorke, *Chaos: An Introduction to Dynamical Systems*, Springer-Verlag, 1997.

[20] T. Matsumoto, L.O. Chua and M. Komuro, The Double Scroll, *IEEE Trans. Circ. Syst.*, **CAS-32**, (1985), 798 - 818.

[21] L.O. Chua, The Genesis of Chua's Circuit, *Archiv fur Elektronik und Ubertragungstechnik*, **46**, (1992), 250 - 257.

[22] T. Shu, *Uniform Random Numbers: Theory and Practice*, Kluwer Academic Publishers, 1995.

[23] M. Guide, Concept for a High-performance Random Number Generator Based on Physical Random Phenomena, *Frequenz*, **39**, (1985), 187 - 190.

[24] W.T. Holman, J.S. Connelly and A.B. Downlatabadi, An Integrated Analog-Digital Random Noise Source, *IEEE Trans Circuits Syst I*, **44**, (1997), 521 - 528.

[25] R.C. Fairfield, R.L. Mortenson and K.B. Coulthart, An LSI Random Number Generator (RNG), *Advances in Cryptology*, Springer-Verlag, **0196**, (1987), 203 - 230.

[26] D. Davis, R. Ihaka and P. Fenstermacher, Cryptographic Randomness from Air Turbulence in Disk Drives, *Advances in Cryptology*, Springer-Verlag, **0839**, (1994), 114 - 120.

[27] G.B. Agnew, *Random Sources from Cryptographic Systems*, In: Advances in Cryptology-CRYPTO?85, pp. 77-81, Springer-Verlag, 1986.

[28] Y. Hu, X. Liao, K. Wong and Q. Zhou, Pseudo-Random Bit Generator Based on Coupled Chaotic Systems and its Application in Stream-Ciphers Cryptography, *Chaos Soliton Fract.*, **40**, (2009), 2286 - 2293.

[29] N.G. Bardis, A.P. Markovskyi, N. Doukas, N.V. Karadimas, *True Random Number Generation Based on Enviromental Noise Measurements for Military Applications*, In: Proc. of 8th WSEAS Int. Conf. on Signal Processing, Robotics and Automation, pp. 68 - 73, 2009.

[30] D. Knuth, *The Art of Computer Programming*, Addison-Wesley Publishing Co., 1981.

[31] S. Park and K. Miller, Random Numbers Generators: Good Ones Are Hard To Find, *Commun. ACM*, **31**, (1988), 1192 - 1201.

[32] S. Oishi and H. Inoue, Pseudo-random Number Generators and Chaos, *Trans. Inst. Electron. Comm. Eng. E*, **65**, (1982), 534 - 541.

[33] V.V. Kolesov, R.V. Belyaev and G.M. Voronov, Digital Random-number Generator Based on the Chaotic Signal Algorithm, *J. Commun. Technol. El+*, **46**, (2001), 1258 - 1263.

[34] T. Stojanovski and L. Kocarev, Chaos-based Random Number Generators - Part I: Analysis, *IEEE Trans. Circ. Syst. Fund. Theor. Appl.*, **48**, (2001), 281 - 288.

[35] S. Li, X. Mou and Y. Cai, Pseudo-random Bit Generator Based on Coupled Chaotic Systems and its Application in Stream-ciphers Cryptography, In: Progress in Cryptology - INDOCRYPT 2001, Lecture Notes in Computer Science, **2247**, pp. 316 - 329, 2001.

[36] L. Kocarev and G. Jakimoski, Pseudorandom Bits Generated by Chaotic Maps, *IEEE Trans. Circ. Syst. I: Fund. Theor. App.*, **50**, (2003), 123 - 126.

[37] S.M. Fu, Z.Y. Chen and Y.A. Zhou, Chaos Based Random Number Generators, *Computer Research and Development*, **41**, (2004), 749 - 754.

[38] L. Huaping, S. Wang and H. Gang, Pseudo-random Number Generator Based on Coupled Map Lattices, *Int. J. Mod. Phys. B*, **18**, (2004), 2409-2414.

[39] Arduino, www.arduino.cc.

[40] Ch. K. Volos, I. M. Kyprianidis and I. N. Stouboulos, Fingerprint Images Encryption Process Based on a Chaotic True Bits Generator, *Int. J. Multimedia Intelligence and Security*, **1**(4), (2010), 320-335.

[41] Ch. K. Volos, I. M. Kyprianidis and I. N. Stouboulos, Image encryption process based on chaotic synchronization phenomena, *Signal Processing*, **93**(5), (2013), 1328-1340.

[42] Ch. K. Volos, Image Encryption Using the Coexistence of Chaotic Synchronization Phenomena, *J. of Applied Mathematics and Bioinformatics*, **3**(1), (2013), 123-149.

[43] NIST, Security Requirements for Cryptographic Modules, FIPS PUB 140-2, (2001). http://csrc.nist.gov/publications/fips/fips140-2/ fips1402.pdf.

[44] G. Marsaglia, DIEHARD Statistical Tests, (1995). http://stst.fsu.edu/pub/diehard.

[45] H. Gustafson, H.E. Dawson, L. Nielsen and W.J. Caelli, A Computer Package for Measuring the Strength of Encryption Algorithms, *Computer Security*, **13**, (1994), 687 - 697.

[46] D. Knuth, *The Art of Computer Programming: Semiemperical Algorithms*, Addison Wesley, 1998.