Jestr

Journal of Engineering Science and Technology Review 6 (4) (2013) 9 - 14
Special Issue on Recent Advances in Nonlinear Circuits: Theory and Applications

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

Text Encryption Scheme Realized with a Chaotic Pseudo-Random Bit Generator

Ch. K. Volos[*, 1], I. M. Kyprianidis[2], and I. N. Stouboulos[2]

[1]*Faculty of Mathematics and Engineering Studies,[1] Dept. of Military Science, Hellenic Army Academy, Vari, GR-16673, Greece*
[2]*Physics Department, Aristotle University of Thessaloniki, GR-54124, Greece.*

Received 20 July 2013; Revised 1 September 2013; Accepted 25 September 2013

Abstract

In this work a new encryption scheme, which is realized with a Chaotic Pseudo-Random Bit Generator (CPRBG) based on a Logistic map, is presented. The proposed system is used for encrypting text files for the purpose of creating secure data bases. The Logistic map is the most studied discrete nonlinear map because it has been used in many scientific fields. Also, the fact, that this discrete map has a known algebraic distribution, made the Logistic map a good candidate for use in the design of random bit generators. The proposed CPRBG, which is very easily implemented, uses the X-OR function, in the bit sequences, that are produced by two Logistic maps with different initial conditions and system's parameters, to achieve better results concerning the "randomness" of the produced bits sequence. The detailed results of the statistical testing on generated bit sequences, done by the most well known tests of randomness: the FIPS-140-2 suite tests, confirmed the specific characteristics expected of random bit sequences.

Keywords: Text encryption scheme, Chaotic Pseudo-Random Bit Generator, nonlinear system, Logistic map, FIPS-140-2 suite tests.

1. Introduction

Nowadays, the information security, especially in the Internet, in the mobile networks, and in the military communication systems, depends upon the generation of unpredictable quantities. Examples include the keystream in the one-time pad, the secret key in the DES encryption algorithm, the primes p, q in the RSA encryption and digital signature schemes, the private key a in the DSA, and the challenges used in challenge-response identification systems. In all these cases, the quantities generated must be of sufficient size and mainly must be "random" in the sense that the probability of any particular value being selected must be sufficiently small to preclude an adversary from gaining advantage through optimizing a search strategy based on such probability.

For this reason in the last decades many research teams tried to design devices or algorithms which are capable of generating unpredictable quantities. These devices are called Random Bit Generators (RBGs). As a definition one could say that [1]:

"*A random bit generator is a device or algorithm which outputs a sequence of statistically independent and unbiased binary digits.*"

Many RBGs have been proposed so far. All these can be classified based on the source of the randomness into three major types: True Random Bit Generators (TRBGs), Pseudo-Random Bit Generators (PRBGs) and Hybrid Random Bit Generators (HRBGs) [2].

A TRBG requires a naturally occurring source of randomness, which comes from an unpredictable natural process in a physical or hardware device. However, designing a hardware device to exploit this randomness and produce 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. Also, TRBGs based on natural sources of randomness are subject to influence by external factors, and to malfunction.

For overcoming all these difficulties of obtaining uniform random sequences from TRNG many researchers led to the development of pseudorandom bit generators. A PRBG is a deterministic algorithm which, outputs a binary sequence of length $l >> k$ that "appears" to be random, if a binary sequence of length k is given. The input to the PRBG is called the seed, while the output of the PRBG is called a pseudorandom bit sequence. This bit sequence is not truly random in that it is completely determined by a relatively small set of initial values. PRBGs 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 simulations, e.g., of physical systems with the Monte Carlo method, in cryptography, and in procedural generation.

However, 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

* E-mail address: chvolos@gmail.com

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.

In the last two decades, nonlinear systems, and especially systems which show chaotic behavior, have aroused tremendous interest because of their structural relationship with cryptographic systems. This relationship has been raised because chaos and cryptography have many similar properties, as it is shown in Table 1 [3].

Table 1. Properties of Chaos and its analogous properties of Cryptography.

Chaos	Cryptography
Ergodicity	Confusion
Sensitivity to initial conditions / system parameters	Diffusion with small changes in plaintext / secret keys
Mixing property	Diffusion with a small change within one block of the plaintext
Deterministic dynamics	Deterministic pseudo randomness
Structural complexity	Algorithm Complexity

As a result of this relationship several chaotic cryptosystems have been presented since 1990 [4-6]. One of the most interesting ways through which chaotic cryptosystems can be realized is via the implementation of Chaotic Pseudo-Random Bit Generator (CPRBG). So, several ideas of designing CPRBG by using nonlinear systems and especially discrete chaotic systems have been proposed by academia and industry [7-14].

The subject of this work is a novel text encryption scheme which is realized with a CPRBG based on two chaotic Logistic maps running side-by-side. The produced, by the CPRBG, bit sequence is a result of the X-OR function in the outputs of the two chaotic Logistic maps which have different initial conditions and system's parameters. The use of two chaotic discrete maps increases the complexity in the random bit generation, as it is confirmed by a well-known statistical test suite, and hence becomes difficult for an intruder to extract information about the system.

This paper is organized as follows. In Section 2, the Logistic map, which is the base of this CPRBG, is presented. Section 3 describes the proposed CPRBG block by block. In Section 4, the statistical tests of FIPS-140-2 which assess the statistical properties of the CPRBG are presented. Also, the results of using the proposed CPRBG in encrypting and decrypting process of text file are presented in Section 5. Finally, Section 6 includes the conclusions of this work.

2. The Logistic Map

The iterative equation,

$$x_{n+1} = rx_n(1-x_n), \quad 0 \le x \le 1 \tag{1}$$

known as the logistic map, is one of the most studied discrete chaotic maps because of its simplicity. Also, it has all the well-known features of chaotic systems (Table 1) and for this reason it possesses great potential for various cryptographic applications such as image encryption [15,16], public key cryptography [17], block cipher [18], and hash function [19].

Furthermore, it was first proposed as pseudo-random number generator by Von Neumann in 1947 [20] partly because it had a known algebraic distribution "a" so that iterated values could be transformed to the uniform distribution. A great number of PRBG, based on various forms of the Logistic equation, has been proposed until today [21-23].

In Eq.(1) the parameter r varies in the interval [0, 4] so that x_{n+1} maps the unit interval into the unit interval. Fig.1 shows the map function of x_{n+1} as a function of x_n, for r = 3.999 and x_0 = 0.5. From this plot the symmetry of the Logistic map about the mid point of the interval [0, 1] is concluded.

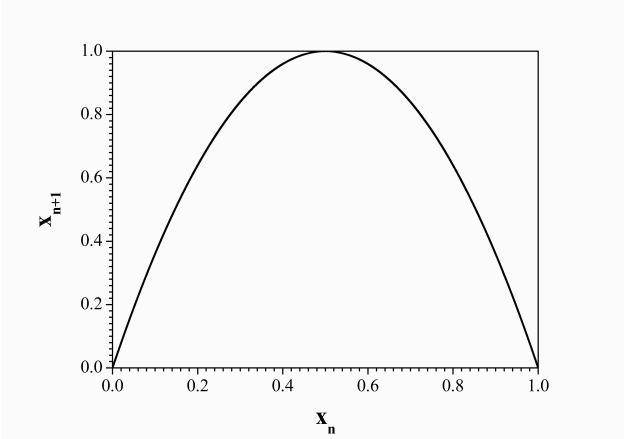

Fig. 1. The map function of the Logistic equation (1), for r = 3.999 and x_0 = 0.5.

Fig.2 serves to illustrate the rich dynamical behavior of Eq.(1) showing the very interesting period-doubling route from periodic to chaotic behavior. This so-called bifurcation diagram is also a very common perspective in nonlinear dynamics, being in this case a plot of the steady-state behavior of Eq.(1) with respect to the bifurcation parameter r.

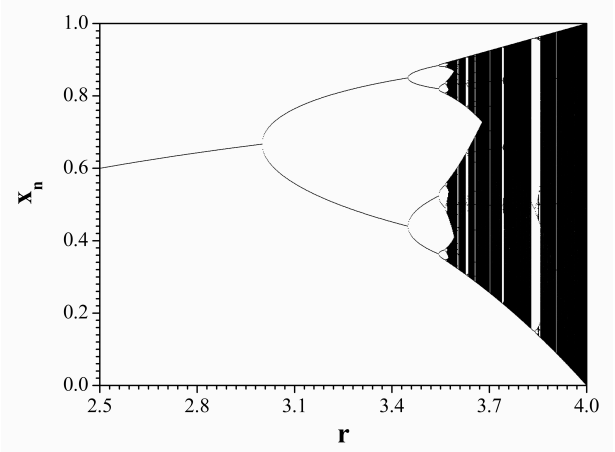

Fig. 2. Bifurcation diagram of x_n vs. r illustrating the period-doubling route to chaos and periodic windows.

As it is shown in Fig.2 the first bifurcation occurs at the value of r = 3, followed by further doublings at shorter and

shorter intervals of r until the period goes to infinity at r_∞ = 3.5699…., signifying chaos. Also, various periodic windows interspersed beyond r_∞, is observed, in which the behavior returns to a normal periodic one, quickly followed again by bifurcations to an infinite period.

So, for $r_\infty > 3.5699\ldots$ the Logistic map shows a strange complex behavior (the so-called chaotic behavior) where map function never repeats its history. This is evident from Fig.3 where no periodicity arises, for r = 3.999 and $x_0 = 0.5$.

Finally, in Fig.4 the well-known Lyapunov exponent:

$$\lambda = \lim_{n\to\infty} \sum_{i=1}^{n} \ln\left|f'(x_i)\right| \tag{2}$$

where,

$$f'(x) = r - 2rx \tag{3}$$

as a function of parameter r, is displayed. As it is known from the nonlinear theory a positive Lyapunov exponent indicates chaos. So, Fig.4 confirms the Logistic map's dynamical behavior as found from the bifurcation diagram (Fig.2).

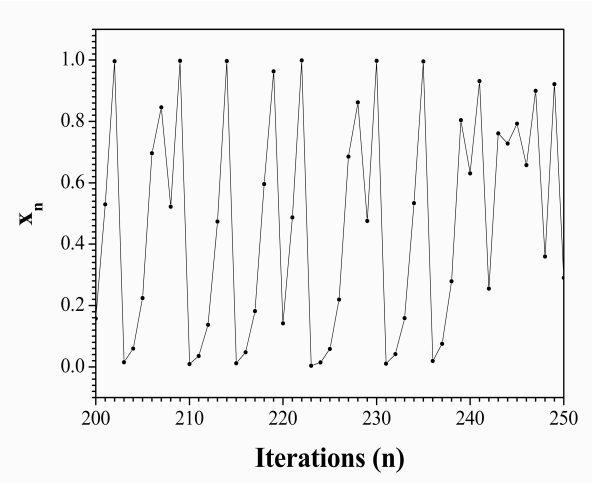

Fig. 3. Variable x vs. n, for r = 3.999 and $x_0 = 0.5$.

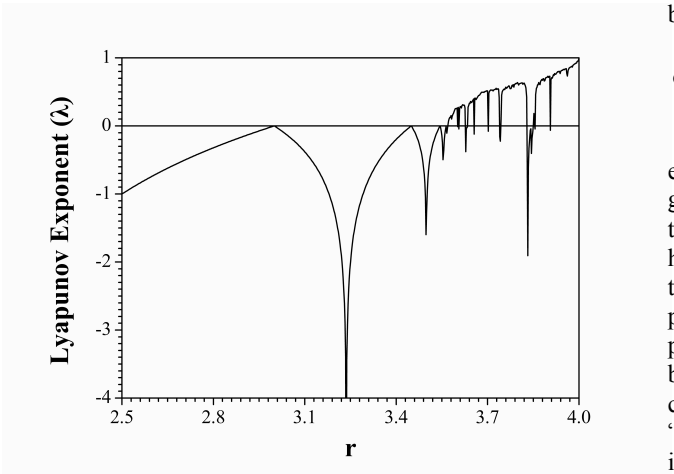

Fig. 4. Lyapunov exponent (λ) vs. parameter r.

3. The Proposed CPRBG

In Ref.[21] the generalized Logistic map as a pseudo-random bit generator has been used. Choosing the mean of the x_n values the author assures the generating of the same numbers of bits according to the following formula:

$$b_n = \begin{cases} 0, & \text{if } x_n \le \bar{x} \\ 1, & \text{if } x_n > \bar{x} \end{cases} \tag{4}$$

where $\bar{x}$ denotes the mean value and b_n is the bit generated by the n-th iteration of the map.

The proposed CPRBG of this work (Fig.5) is based on two Logistic Maps (LM) of Eq.(1), starting from random independent initial conditions: $(x_0, y_0) \in (0, 1)$ and $x_0 \neq y_0$ and using the Eq.(4).

$$\begin{cases} \text{LM1}: \; x_{n+1} = r_1 x_n (1 - x_n) \\ \text{LM2}: \; y_{n+1} = r_2 y_n (1 - y_n) \end{cases} \tag{5}$$

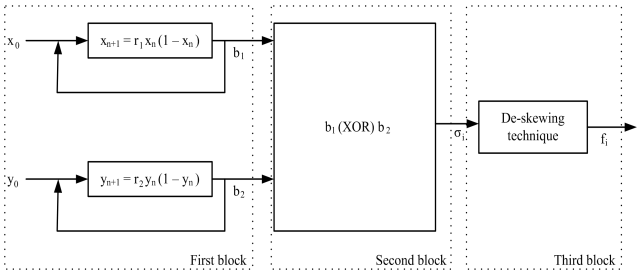

Fig. 5. The schematic block diagram of the proposed Chaotic Pseudo-Random Bit Generator.

These two iterative maps consist the first block of the proposed CPRBG. The set of initial conditions (x_0, y_0) and systems' parameters (r_1, r_2) serves as the seed for the CPRBG.

The bit sequence σ_i of the second block, is generated by using the X-OR function, in the bit sequences (b_1 and b_2) which are produced by the outputs of the two Logistic maps. This technique is used for achieving better results concerning the "randomness" of the produced bits sequence by the proposed CPRBG.

$$\sigma_i = b_1 \oplus b_2 \tag{6}$$

The third block of the proposed CPRBG relies on extracting unbiased bits with no correlation from a defective generator with unknown bias. For this purpose various techniques, which are called de-skewing techniques [1], have been proposed. Von Neumann [24] 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 non-overlapping pairs of bits into output bits by converting the bit pair "01" into an output "0", converting "10" into an output "1", while the pairs "11" and "00" are discarded. This technique is very easily implemented but it decreases throughput of generating approximately 1 bit from 4 bit.

4. Statistical Tests

In order to gain the confidence that newly developed pseudo-random bit generators are cryptographically secure, they should be subjected to a variety of statistical tests designed to detect the specific characteristics expected of truly random sequences. There are several options available for analyzing the randomness of the newly developed pseudo-random bit generators. The four most popular options are:
(i) the FIPS-140-2 (Federal Information Processing Standards) suite of statistical tests of the National Institute of Standards and Technology (NIST) [25],
(ii) the DIEHARD suite of statistical tests [26],
(iii) the Crypt-XS suite of statistical tests [27] and
(iv) the Donald Knuth's statistical tests set [28].

In this section the "randomness" of the produced bits sequence, by the proposed Chaotic PRBG, is analyzed by using the most stringent tests of randomness: the FIPS-140-2 suite of statistical tests. The results of the use of the four statistical tests, Monobit test, Poker test, Runs test, and Long run test, which are part of the FIPS-140-2, are presented in details. As it is known, according to FIPS-140-2 statistical tests, the examined CPRBG will produce a bitstream, $\sigma_i = \sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_{n-1}$, of length n (at least 20,000 bits), which must satisfy the following standards.

• 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) show approximately 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 [25].
• Long Run Test: This test is passed if there are no runs longer than 26 bits.

Using the fact in information theory that noise has maximum entropy, the systems' parameters (r_1, r_2) and initial condition (x_0, y_0) are chosen such that the measured theoretic entropy [29] of the CPRBG, which is given by the following equation, is maximal.

$$H_n = \lim_{n\to\infty}\left(-\sum_{B^n} P(B^n)\ln P(B^n)\Big/ n\right) \quad (7)$$

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

Furthermore, using the procedure described in Eq.(4) and with the de-skewing technique two bit sequences of length 20,000 bits have been obtained from the outputs of the two Logistic maps of the proposed CPRBG. The measure-theoretic entropy of each Logistic map with respect to system's parameter (r_1, r_2) = (3.999, 3.991) and initial condition (x_0, y_0) = (0.5, 0.4) is calculated to be:
LM1: $H_n = 0.69309$, for n = 3 and $H_n = 0.69270$ for n = 4 and
LM2: $H_n = 0.69297$, for n = 3 and $H_n = 0.69238$, for n = 4.

Then these two bit sequences are subjected to the four tests of FIPS-140-2 test suite. As a result, it has been numerically verified that the bit sequences passed the test suite of FIPS-140-2, in both cases (Tables 2 & 3).

Table 2. Results of FIPS-140-2 test, for the bit sequence produced by the first Logistic map.

Monobit Test	Poker Test	Runs Test	Long Run Test
n_1 = 10,033 (50.165%)	2.3396	B_1 = 2,392 B_2 = 1,242 B_3 = 648 B_4 = 315 B_5 = 154 B_6 = 165	No
Passed	**Passed**	**Passed**	**Passed**

Table 3. Results of FIPS-140-2 test, for the bit sequence produced by the second Logistic map.

Monobit Test	Poker Test	Runs Test	Long Run Test
n_1 = 10,081 (50.405%)	6.8905	B_1 = 2,424 B_2 = 1,197 B_3 = 685 B_4 = 316 B_5 = 161 B_6 = 159	No
Passed	**Passed**	**Passed**	**Passed**

Finally, by using the proposed CPRBG, with the systems' parameters and initial conditions mentioned before, a bit sequence of length 20,000 bits have been obtained from the output of this chaotic pseudo-random bit generator. The measure-theoretic entropy of the CPRBG is $H_n = 0.69310$, for n = 3 and $H_n = 0.69279$ for n = 4, greater than in the case of each Logistic map. Also, the results of FIPS-140-2 test suite (Table 4) have been improved, especially the monobit and the poker test, in regards to the results, which have been found for each Logistic map.

Table 4. Results of FIPS-140-2 test, for the CPRBG.

Monobit Test	Poker Test	Runs Test	Long Run Test
n_1 = 10,014 (50.07%)	2.2899	B_1 = 2,515 B_2 = 1,274 B_3 = 648 B_4 = 308 B_5 = 150 B_6 = 153	No
Passed	**Passed**	**Passed**	**Passed**

5. Text Encryption-Decryption Scheme

As already stated a very important issue nowadays is the secure transmission of information and especially of text messages. For this purpose the proposed CPRBG in encrypting and decrypting process of text file is used. The encryption and decryption scheme is shown in Fig.6.

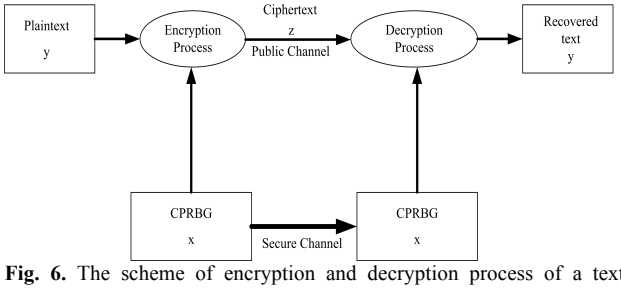

Fig. 6. The scheme of encryption and decryption process of a text message.

The plaintext, which is used, is a text message (Fig.7) saved in a txt file format. Next, each ASCII character of the text file is converted to the decimal equivalent number (y) and all these numbers are saved to a new txt file.

Cryptography (or cryptology; from Greek κρυπτός, "hidden, secret"; and γράφειν, graphein, "writing", or -λογία, -logia, "study", respectively) is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Fig. 7. The plaintext.

The same procedure is followed also in the binary sequence produced by the proposed CPRBG. So, the produced bit sequence is divided into 8-bit numbers, each of them is converted to the decimal equivalent number (x) and all these numbers are saved to another txt file.

.ȷE Pψ□ ε£ E>´□Kn.|;› Ξϋg¦/py)□Ÿ(P3NWƱ□0›ζ•cl g□□΄YR□~□˝- θ l (`¦BŸu□_®δíΞ kb«~ Ïco `N}€Σπ.£ □Ι¨»Tb-Z•X|‰£„”¶%□Πó□ϊ†ά*9□ó“w•R™-T“… *f*τPWψώZύ□E X8’-dZZz Ÿ□Ω ζϋ˝(ζ² uDαv®δ R□ϊ$:zO,«l^□³"©ζ±† ' ζ kR• ΰκΐκΔS‹YAYξ)¤□ ¤b3σζ ‹gvf¤ΐΐβg“Ο0 □ £ *s¥bΩφ 9HφΙβW¦ί{X<o ² 9ι#ΘH§κB|{ρ,ψ§□3 'O ³ ύ”sz),υk□ 1ήζ3ϋ´νt›aohΓ*ΨM*f*_®A κ□%?LξN0ΎYΘ□¥□O□Kιr‹% !/=ξ!•RυUώ¤ψ*Bj‹□S˝¨υYΞνω§ίv□WhZ π¦[•O„ ,Jμ έ¥β mώ□zvδz□□¨ε8}΄E ‘ίκ6^λΛλΥΩE)*f*‘I-lAoK-φ0Bz/j□Aυ 7εΟ ¬P³•@›ζ•Δb□˝ □Δχ¦@ί0[i¤Oε„¬=³²ύ□Z*f*(TBηZLIσ¬□πa[K□ήισ— H9τMάύXω‘©Iμ □ `□†@rg υ□a;QN=ιδ?φ: Y§ ι□0γ ±³ 2□d;‡*f* ΄ERŸΣuΩBβ ²ll□15T*f*έ®SϋU—k™ κόπW4O-—2□μb1ΊΩT”:…³¬Φ□L™ ¥K*<Ξί¦φίΪΖY—□ΛIό ΄‹Uέ¥ϋ©□ X□B΄Y΄EΨά1+□IΦNΔ½□‰:—ώΞ_oFμ°*= ΄E□M□g{km¤□Nk a TΪé5 ωwZD□tEΠ.› □υ#%πΛ’m ΪΥΦ›δ›»Θ} χi¨έ B ΔΛ

Fig. 8. The ciphertext.

Finally, these two files, which contain the sequences of the decimal numbers (x) and (y), produce the ciphertext by using a very simple but effective function:

$$z = \begin{cases} x + y, & \text{if } x + y < 255 \\ x + y - 255, & \text{if } x + y > 255 \end{cases} \quad (8)$$

where, (z) is the decimal number of the ciphertext. By converting each one of the decimal numbers (z) to the equivalent ASCII character, the final ciphertext is produced (Fig.8).

Cryptography (or cryptology; from Greek κρυπτός, "hidden, secret"; and γράφειν, graphein, "writing", or -λογία, -logia, "study", respectively) is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Fig. 9. The recovered text.

Also, the decryption process uses the decimal number sequences of the ciphertext (z) and of the proposed CPRBG (x), in the following function:

$$y = \begin{cases} z - x, & \text{if } z - x \geq 0 \\ z - x + 255, & \text{if } z - x < 0 \end{cases} \quad (9)$$

and converts again each one of the decimal numbers (y) to the equivalent ASCII character of the recovered text (Fig.9).

It should be emphasized that this method can greatly contribute to the security of the transmitted information (text messages) through public channels, as “intrusions” occur often in Internet. The system derives its security from the high chaos to information ratio, which makes it impossible for the ciphertext to be attacked, using any signal processing techniques such as Fourier transform.

Therefore, even if the “intruder” knows the exact encryption and decryption processes, it is impossible to know the chaotic bit sequence derived from the CPRBG, because of the sensitivity of the chaotic system on initial conditions and system's parameters.

6. Conclusion

In this paper, a new text encryption process which is realized with a chaotic pseudo-random bit generator was presented. The proposed generator is based on two Logistic maps, with different initial conditions and system’s parameters, running side-by-side. The produced, by the proposed CPRBG, bit sequences was a result of the X-OR function in the outputs of the two Logistic maps. The use of X-OR function has increased the complexity in the random bit sequence, as it was confirmed by the use of a well-known statistical test suite FIPS-140-2. So, the proposed encryption scheme is very robust against interference from an intruder.

References

1. A. J. Menezes, P.C. Van Oorschot, and S.A. Vanstone, Handbook of applied cryptography, CRC Press, (1997).
2. T. Shu, Uniform random numbers: theory and practice, Kluwer Academic Publishers, (1995).
3. G. Alvarez and S. Li, Int. J. Bifurcat. Chaos **16**, 2129 (2006).
4. L. Kocarev, IEEE Circuits Syst. Mag. **1**, 6 (2001).
5. Ch.K. Volos, I.M. Kyprianidis, and I.N. Stouboulos, WSEAS Trans. Circ. Syst. **5**, 654 (2006).
6. Ch.K. Volos, I.M. Kyprianidis, I.N. Stouboulos, Int. J. Multimedia Intelligence and Security **1**, 320 (2010).
7. S. Oishi and H. Inoue, Transactions of the Institute of Electronics and Communication Engineers of Japan E **65**, 534 (1982).
8. V.V. Kolesov, R.V. Belyaev, and G.M. Voronov, J. Communications Technology and Electronics **46**, 1258 (2001).
9. T. Stojanovski and L. Kocarev, IEEE Trans. Circuits Syst. I: Fundamental Theory and Applications **48**, 281 (2001)
10. S. Li, X. Mou and Y. Cai, Pseudo-random bit generator based on couple chaotic systems and its application in stream-ciphers cryptography, In Progress in Cryptology – INDOCRYPT 2001, Lecture Notes in Computer Science, (2001).
11. L. Kocarev and G. Jakimoski, IEEE Trans. Circuits Syst. I: Fundamental Theory and Applications **50**, 123 (2003).
12. S.M. Fu, Z.Y. Chen, and Y.A. Zhou, Computer Research and Development **41**, 749 (2004).
13. L. Huaping, S. Wang, and H. Gang, Int. J. Mod. Phys. B **18**, 2409 (2004).
14. X.M. Li, H.B. Shen, and X.L. Yan, J. Electronics Information Technology **27**, 874 (2005).
15. J. Fridrich, Int. J. Bifurcat. Chaos **8**, 1259 (1998).
16. Q. Zhou, K.W. Wong, X. Liao, T. Xiang, and Y. Hu, Chaos Solitons Fractals **38**, 1081 (2008).
17. R. Tenny, IEEE Trans. Circuits Syst. I **52**, 672 (2005).
18. G. Jakimoski and L. Kocarev, IEEE Trans. Circuits Syst. I. **48**, 163 (2002).
19. Y. Wang, X. Liao, and K. Wong, Information Sciences **178**, 1391 (2008).
20. S.M. Ulam and J. Von Neumann, Bull. Amer. Math. Soc. **53**, 1120 (1947).
21. R. Ursulean, Electronika Ir Electrotechnika **7**, 10 (2004).
22. V. Patidar and K. K. Sud, Informatica **33**, 441 (2009).
23. S. Ahadpour, Y.R. Sadra, and Z. ArastehFard, Int. J. Computer Science Issues **9**, 449 (2012).
24. J. Von Neumann, Various techniques used in connection with random digits, G.E. Forsythe, Applied Mathematics Series, National Bureau of Standards (1951).
25. NIST, Security requirements for cryptographic modules, FIPS PUB 140-2, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf, (2001).
26. G. Marsaglia, DIEHARD statistical tests (http://stst.fsu.edu/pub/diehard, (1995).
27. H. Gustafson, H.E. Dawson, L. Nielsen, W. Caelli, J. Computer Security **13**, 687 (1994).
28. D. Knuth, The art of computer programming: semiemperical algorithms, Addison Wesley, Reading, USA, (1998).
29. A.M. Fraser, IEEE Trans. Inf. Theory **35**, 245 (1989).