CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Volume:2 Number:2 Pages:69-76

A Chaos Based Image Encryption On LabVIEW

Bilal Gürevin[*,1], **Muhammed Yıldız**[§,2], **Emre Güleryüz**[§,3], **Mustafa Ç. Kutlu**[*,4] **and Özcan Sorgun**[*,5]
[*]Sakarya University of Applied Sciences, Faculty of Technology , Mechatronics Engineering, Sakarya, Turkey, [§]Sakarya University of Applied Sciences, Faculty of Technology , Electrical-Electronics Engineering, Sakarya, Turkey.

ABSTRACT In this study, a user interface has been designed for chaos based image encryption applications in LabVIEW. First, a random number generator was designed via a chaotic system that analyzed in LabVIEW. Through to the interface developed in this design, the initial values, system variables and how many bits of random numbers will be generated are entered by the user. Moreover, the phase portraits and time series of the system can be observed by the user through this interface. The random numbers produced were tested on NIST-800-22 to measure the reliability of the system. In addition, an interface design that enables easy encryption can be done by selecting random numbers and images produced with the algorithm developed in LabVIEW. The reliability of the application was verified by passing commonly accepted tests in the literature such as histogram entropy, correlation, UACI, NPCR. While the result values of NPCR test for y, z phases are 0,99573 and 0,99605, the UACI result values are 0,29265 and 0,2939. Results shows that the proposed software is reliable.

KEYWORDS
Chaos,
Chaotic systems,
Image encryption application,
Pseudo random numbers,
LabVIEW.

INTRODUCTION

In addition to the advantages of fast-moving technology, there are disadvantages. While access to many areas over the Internet is faster with ever-evolving digital multimedia platforms, it will be inevitable that major problems will arise with vulnerabilities that can arise when entering these platforms. When look the technology, there are many facial recognition applications have been introduced and their presence of many platforms that contain our banking and personal information, which is important to us, makes information security more and more important (Wang *et al.* 2012). This means that the need for research on measures to be taken against illegal information and data copy problems has increased.

The artificial intelligence on image processing has provided us with studies that will replace confirmation in fast access and important operations, while also guiding the encryption efforts to ensure the security of the data. With the healthy implementation of the method to be used in encryption, trust in these applications will progress at the right rate.

Today, cryptography is used in various fields such as statistics, numerical analysis, simulation, and modeling . In recent years, researchers have shown an increased interest in chaotic random number generation, which is one of the safest way of encryption. The random numbers obtained in these studies vary according to the application areas. These include FPGA-based random number generation and applications (Koyuncu *et al.* 2020; Tuna *et al.* 2019a; Tuna and Fidan 2018). Apart from these, random number generator studies with various methods signal hiding, image encryption, encrypted communication studies with random numbers are also available (Alcin *et al.* 2019; Tuna *et al.* 2019b; Akgül 2015; Pareschi *et al.* 2010; Guyeux *et al.* 2010; Avaroğlu and Türk 2013; Buchovecká *et al.* 2017; Lynnyk *et al.* 2015; Çavuşoğlu *et al.* 2014; Uhl and Pommer 2004).

The science of chaos is the most complex behavior of dynamic systems, and it used to form nonlinear events. The chaotic encryption method is the most intuitive and effective way to make images unrecognizable (Chai *et al.* 2017). The feature that makes this method of application safe is that the system of equations in which the numbers used in encryption are produced has sensitivity to the initial values, as well as showing complex dynamic properties. These numbers are used in many engineering fields besides cryptology applications (Ozdemir *et al.* 2018; Chai *et al.* 2017). In this study, in contrast to the studies in the field of chaotic cryptology, dynamic analysis of the 3-dimensional chaotic equation,

Manuscript received: 28 July 2020,
Received in revised form: 31 August 2020,
Accepted: 3 September 2020.

[1] bilalsau@gmail.com
[2] yldzmuhammet92@gmail.com (**Corresponding author**)
[3] emreguleryuz61@gmail.com
[4] mkutlu@subu.edu.tr
[5] y180010019@subu.edu.tr

random number generator created with the equation in LabVIEW environment, and image encoding application with random numbers generated were performed. The numbers produced have successfully passed the NIST-800-22 test and the encrypted images histogram, entropy, correlation, UACI, NPCR tests and have increased the reliability of our application. The fact that this application is made with LabVIEW which allows the multimedia to be encrypted and saved in real time.In the next sections, dynamic analysis of the 3D chaotic system, which will be used in random number generation, random number generation and randomness tests in the LabVIEW environment using the chaotic system, image encryption application in the LabVIEW environment with random numbers produced, and finally security analysis of the encrypted image will be included.

THE CHAOTIC SYSTEM AND ITS DYNAMICAL ANALYSIS

The chaotic system is defined by the Equation 1 as below:

$$\begin{aligned} \dot{x} &= ay \\ \dot{y} &= -x + byz \\ \dot{z} &= -x - cxy - dxz \end{aligned} \tag{1}$$

The equation includes four different parameters of the system: a, b, c, d, and x, y, z status variables. The initial conditions of the system are determined as $x(0) = 0.4$, $y(0) = 0.1$ and $z(0) = 0$. The values of system parameters are $a = 1.9$, $b = 1.1$, $c = 11.5$, $d = 0.7$. The chaotic system has given by the following Equation 2:

$$\begin{aligned} \dot{x} &= 1.9y \\ \dot{y} &= -x + 1.1yz \\ \dot{z} &= -x - 11.5xy - 0.7xz \end{aligned} \tag{2}$$

Phase portraits

Phase portraits can be examined in four different ways in a three-dimensional system. The $x - y - z$, $x - y$, $x - z$ and $y - z$ phase portraits of the system were obtained using the Matlab ODE45 function.

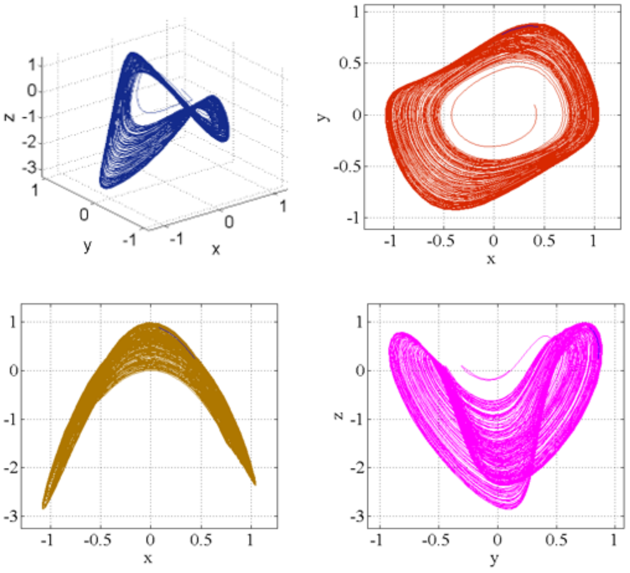


Figure 1 Phase diagrams of the 3D chaotic system

As can be seen in Figure 1, the system has a chaotic structure via the unstable fractal shapes within certain boundaries of the phase portraits (Gürevin 2019).

Lyapunov exponents spectrums

Lyapunov spectrum analysis, such as bifurcation analysis, gives information about the ranges of the system which ranges are chaotic or not. In order for the system to be chaotic, Lyapunov's exponents must be (+, 0,-) (Pamuk 2016).

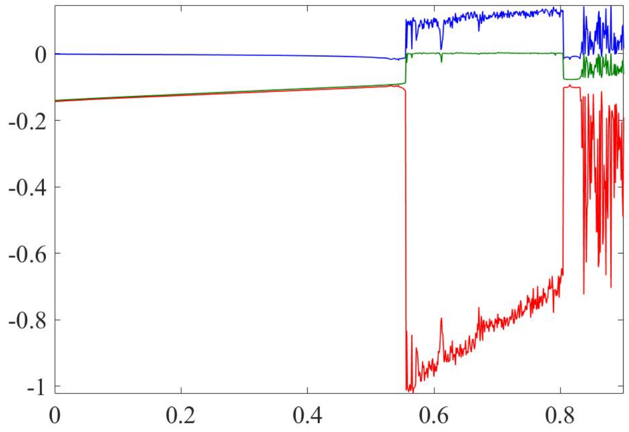


Figure 2 Lyapunov spectrum analysis according to the 'd' parameter of the chaotic system (0-0.9)

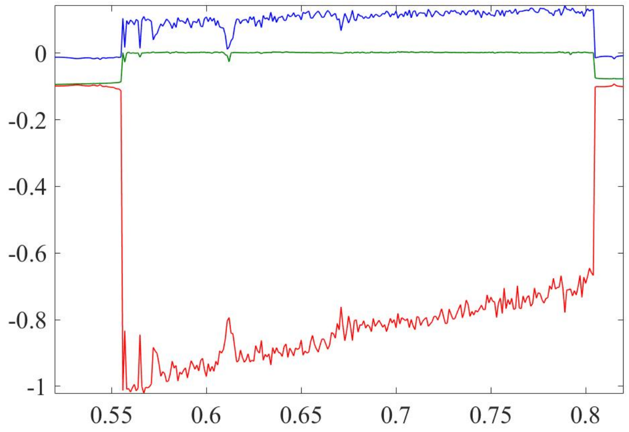


Figure 3 Lyapunov spectrum analysis according to the 'd' parameter of the chaotic system (0.52-0.82)

The $(0 - 0.9)$ range of the Lyapunov spectrum of the equation system, which is drawn by taking the a, b, c constant and variable d parameters into account, is shown in Figure 2, and the detailed version of the range $(0.52 - 0.82)$ is shown in Figure 3.

After 0.55 of the d parameter in the spectrum, Lyapunov's exponentials, λ_1, λ_2, λ_3 received (+,0,-) values, respectively. The intervals examined in the bifurcation diagram and interpreted as chaotic reinforce the chaotic state by overlapping the intervals in the Lyapunov spectrum.

Bifurcation analysis

Another widely used method for examining the chaotic state of the system in dynamic analysis is the bifurcation diagram (Akgul *et al.* 2016). It is also possible to examine whether the system shows chaotic features at the intervals in which the system is examined with the bifurcation diagrams. The bifurcation graphic of the chaotic system used in the article is shown in Figure 4.

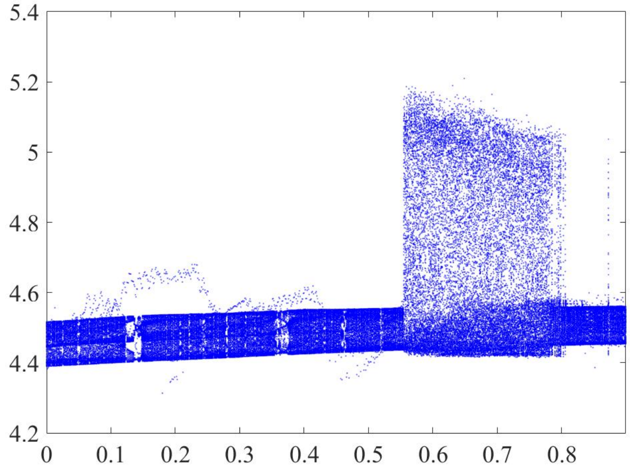


Figure 4 Bifurcation analysis according to the 'd' parameter of the chaotic system $(0 - 0.9)$

In this diagram, while parameter a, b and c are constant, the d parameter is variable. The bifurcation diagram based on the variable parameter was drawn in the range of $0 < d < 0.9$. In the bifurcation analysis, the range d parameter which shows especially chaotic behavior is between $(0.52 - 0.82)$. While these values showed single output up to 0.55, they showed multiple outputs after 0.55. This behavior is defined as the indicator of the chaoticness of the system. The new bifurcation diagram drawn in order to examine the chaotic behavior more clearly in this value range is shown in Figure 5.

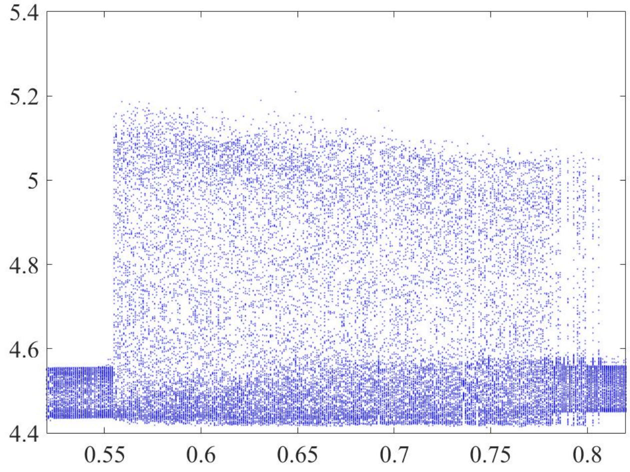


Figure 5 Bifurcation analysis according to the 'd' parameter of the chaotic system $(0.52 - 0.82)$

RANDOM NUMBER GENERATION, RANDOMNESS TESTS AND ENCRYPTION APPLICATION ON LABVIEW

Random number generation on LabVIEW

The pseudo code of the random number generator designed by a custom made LabVIEW program is shown in Algorithm 1. This application is designed according to 3D chaotic systems. In the designed application, the initial condition values and system parameters of the chaotic system are entered. The solution interval and the phase (x, y, z) sampling process are determined and the discrimination process is performed with the Runge Kutta 4 $(RK4)$ method.

The float values obtained in each iteration of the selected phase are converted to 32-bit binary values and LSB (a) entered from these 32-bit binary values are added to the random number sequence by taking the binary value as the number of bits.

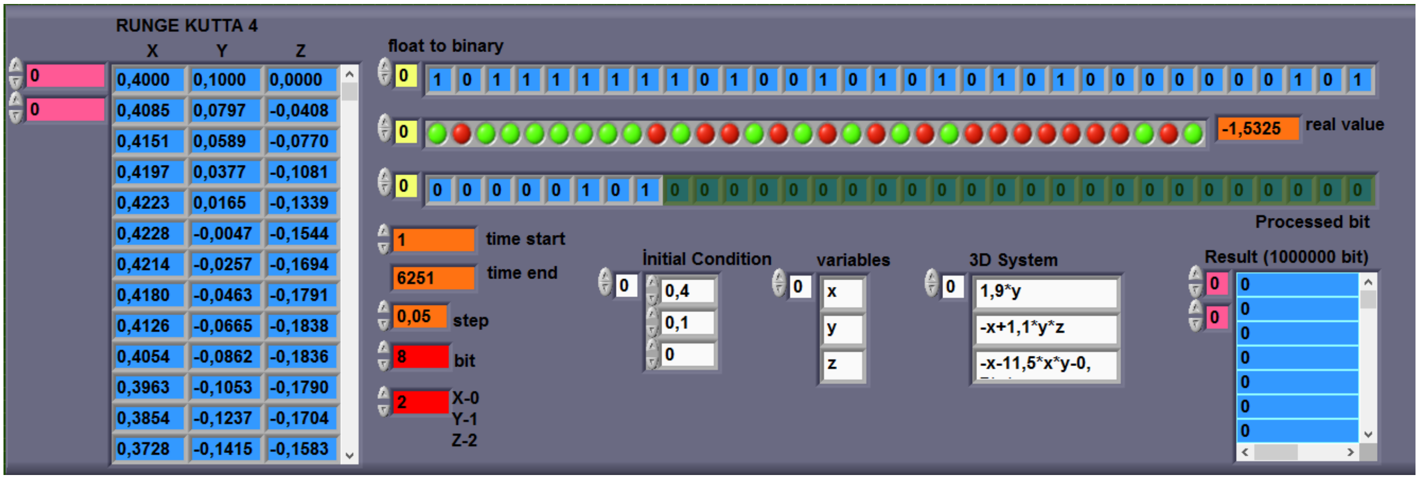
In order to generate a million bit length random number required for NIST-800-22 randomness test, iteration calculation is made by considering variable a. When the specified number of iterations is completed, a random number of one million bits is obtained. The random number is checked in the NIST-800-22 test and analyzed. If the test results are successful, random number generation is terminated. If the result is negative, the process is repeated by reducing the "a" value to obtain more precise results. In this way, the cycle is repeated until random results from random tests are obtained.

Algorithm 1 : Pseudo code of LABVIEW random number generation algorithm

```
Input: Parameters and initial condition of chaotic system
Output: Ready tested random numbers
 1: Start: Entering system parameters and initial condition of 3D
    chaotic system
 2: Determination of the value of $\Delta h$
 3: Sampling with determination $\Delta h$ value for RK4
 4: while minimum 1MBit data do
 5:     Solving the 3D chaotic system using RK4 algorithm
 6:     Obtaining time series as float numbers ($x$, $y$ and $z$)
 7:     Convert float to 32 bit binary numbers
 8:     Select LSB-8 bits one of x, y and z phases from RNG
 9: end while
10: Apply NIST-800-22-800-22 Tests for each minimum 1MBit data
11: if (test results == pass)  then
12:     Ready tested random numbers for RNG applications
13: else if test results == false then
14:     Go to step 4
15: end if
16: Exit
```

Figure 6 shows the user interface of the random number generator designed in LabVIEW on the front panel. throughBy means of this panel, equation input, initial conditions, system variables, solution range Δh and phase number to be used in random number generation are produced from the three-dimensional chaotic system. At the same time, sampling values and random number outputs obtained in x, y, z phases can also be examined with this interface.

Figure 7, 8 and 9 show the oscilloscope outputs of random numbers generated from the x, y and z state variables of the chaotic system.



RUNGE KUTTA 4

0
0

X	Y	Z
0,4000	0,1000	0,0000
0,4085	0,0797	-0,0408
0,4151	0,0589	-0,0770
0,4197	0,0377	-0,1081
0,4223	0,0165	-0,1339
0,4228	-0,0047	-0,1544
0,4214	-0,0257	-0,1694
0,4180	-0,0463	-0,1791
0,4126	-0,0665	-0,1838
0,4054	-0,0862	-0,1836
0,3963	-0,1053	-0,1790
0,3854	-0,1237	-0,1704
0,3728	-0,1415	-0,1583

float to binary

0 | 1 0 1 1 1 1 1 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1

0 | -1,5325 real value

0 | 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Processed bit

1 time start

6251 time end

0,05 step

8 bit

2 X-0 Y-1 Z-2

İnitial Condition: 0 | 0,4 | 0,1 | 0

variables: 0 | x | y | z

3D System: 0 | 1,9*y | -x+1,1*y*z | -x-11,5*x*y-0,

Result (1000000 bit): 0 | 0 | 0 0 0 0 0 0 0

Figure 6 Front panel view of photos random number generation design on Labview

Randomness test

The NIST-800-22 test is a nationally accepted randomness test consisting of 16 different tests. In order for random numbers produced in length of 1 million bits to be considered successful, all these tests must pass successfully. Nist-800-22 test was used to perform random tests of random numbers of 1 million bits obtained with a 3D chaotic system in Labview. The success criterion is based on the P-value in the NIST-800-22 test. The p-value determined in each test of 16 different tests in NIST-800-22 is considered to be greater than 0.001 (Akgül *et al.* 2020; Gürevin 2019).

Table 1 NIST-800-22 Test Results

Statistical Tests	P value	P value	P value	Result
	(X_8Bit)	(Y_8Bit)	(Z_8Bit)	
Frequency (Monobit) Test	0,4889	0,906	0,7596	Successful
Block-Frequency Test	0,4198	0,2311	0,3583	Successful
Cumulative-Sums Test	0,4596	0,9662	0,4975	Successful
Runs Test	0,8489	0,8399	0,566	Successful
Longest-Run Test	0,8596	0,625	0,1798	Successful
Binary Matrix Rank Test	0,715	0,3306	0,5173	Successful
Discrete Fourier Transform Test	0,9341	0,2912	0,2551	Successful
Non-Overlapping Templates Test	0,313	3,1517	0,0595	Successful
Overlapping Templates Test	0,5732	0,4911	0,002	Successful
Maurer's Universal Statistical Test	0,2379	0,5083	0,1474	Successful
Approximate Entropy Test	0,6214	0,2826	0,0836	Successful
Random-Excursions Test (x = -4)	0	6,418	0,28	Successful
Random-Excursions Variant Test (x = -9)	0	0,3896	0,6048	Successful
Serial Test-1	0,6692	0,6129	0,4027	Successful
Serial Test-2	0,605	0,3102	0,7874	Successful
Linear-Complexity Test	0,6141	0,5384	0,8781	Successful

Table 1 shows the NIST-800-22 test results of random numbers obtained from x, y and z using 8 bits. While the random numbers obtained from the y and z phases pass the NIST-800-22 test successfully, the random numbers obtained from the x phase could not pass the "Random-Excursions" and "Random-Excursions Variant" tests, and were deemed unsuccessful in the NIST-800-22 test. For this reason, random numbers obtained from y and z phases are used in the encryption applications in the next sections.

Image encryption with Labview

There are two main issues in image encryption. These are pixel mixing (randomness) and spreading (sensitivity) operations. Since neighboring pixel values are close to each other in an unencrypted picture, a meaningful image appears. The more the correlation between the pixels is reduced, the more complicated the image becomes and it becomes unsolvable. Likewise, sensitivity between pixels should be reduced (Akgul *et al.* 2016; Akgül *et al.* 2020; Gürevin 2019). Through to the deterministic complex structure of the chaotic systems and the sensitivity feature in the initial conditions, propagation and mixing processes can be easily provided. Also, since chaos-based encryption method does not require high processing power, it can be easily used in platforms with low processing capacity.

Algorithm 2 contains the pseudo code, which refers to the encryption of 8 bit images in LabVIEW using random numbers of 1 million bits generated in LabVIEW. First, the image and random numbers to be encrypted are taken into the system. The image sizes are then scaled to the desired rate, and each pixel value is converted to an 8-bit binary level. After determining the total number of pixels (256x256), each consisting of 8 bits in length, a random number of sufficient length is selected. Image pixels converted to 8 bits are transferred to an array. Random numbers are subjected to XOR operation with this sequence created from 8 bit image. 8-bit sequence values obtained as a result of XOR operation are converted to decimal number system. The resulting values are again placed in pixel locations of the image size. In this way, the encrypted image is obtained. Figure 10 shows the output of the 8-bit image and the encrypted image in LabVIEW.

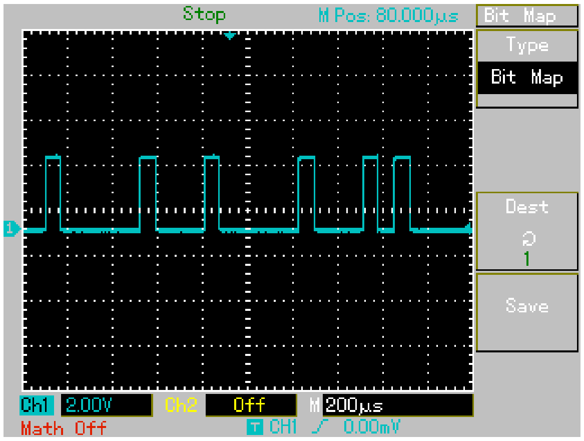


Figure 7 Phase x oscilloscope image

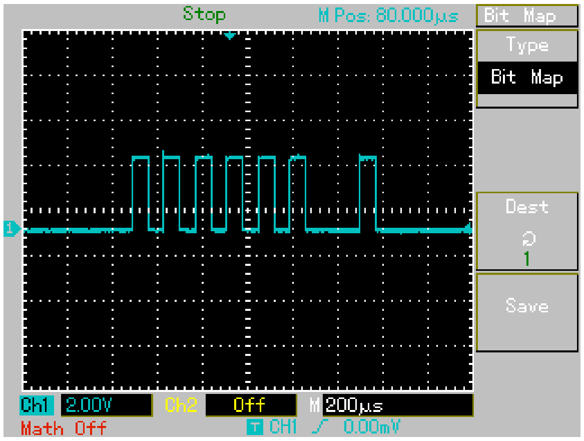


Figure 8 Phase y oscilloscope image

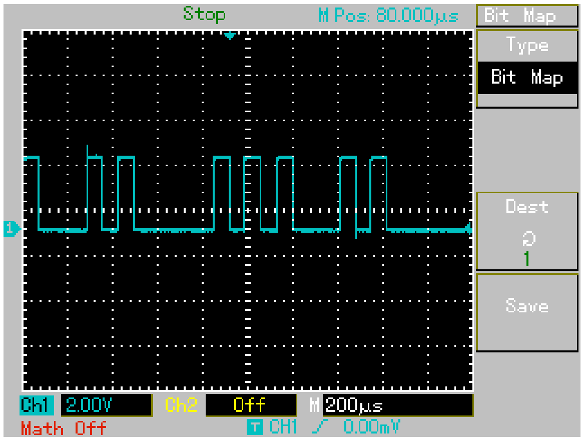


Figure 9 Phase z oscilloscope image

Algorithm 2 : Pseudo code of image encryption algorithm

Input: Ready tested random numbers and image data
Output: Encrypted image data (8 bit)

```
Start: Entering random numbers and image data
Determination of image size ($A = 256$, $B = 256$)
Determination of variables ($k = 0$, $t = 0$)
Convert the image to GrayScale
for i=0 → A-1 do
    for j=0 → B-1 do
        Convert image(i, j) to 8bit
        8bit image(i, j) = image(i, j)
    end for
end for
for i=0 → A-1 do
    for j=0 → B-1 do
        Array_image(0, k) = 8bit_image(i, j)
        k=k+1
    end for
end for
Encrypted_Array = Array_image XOR random numbers
for i=0 → A-1 do
    for j=0 → B-1 do
        Encrypted_image(i, j) = Encrypted_Array (0, t)
        t=t+1
    end for
end for
Exit
```

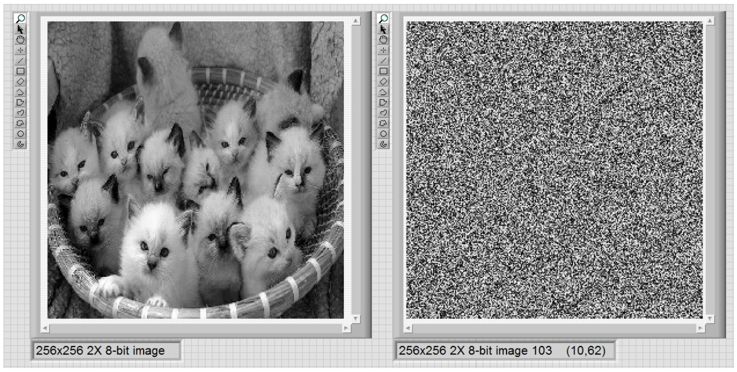


Figure 10 Encrypted 8 bit image and original version

SECURITY ANALYSIS

The fact that the encrypted image is visually invisible does not mean that a good encryption has been made. There are some statistical analysis methods used to measure the reliability of encryption. In this section, security analyzes of the encrypted image obtained from chaos-based encryption were performed. While performing security analysis, histogram, entropy, correlation, UACI (Unified average changing intensity), NPCR (Number of pixel change rate) methods were applied on 8 bit images encrypted with the y and z phase of the chaotic system.

Histogram analysis

Histogram analysis is one of the analysis methods used as a secure encryption criterion in encrypted images. The proximity of the histogram distribution indicates the encrypted images resistance to deciphering (Akgül 2015; Akgul *et al.* 2016; Akgül *et al.* 2020; Gürevin 2019; Pamuk 2016; Keleş 2012). Figure 11 shows histogram distributions of 8-bit source image and 8-bit images encrypted with y - z dimensions. As seen in the figure, while the source image histogram distribution is in a scattered structure, the histogram distributions of the encrypted images are close to each other. When looking at histogram distributions, it can be interpreted that encryption is good.

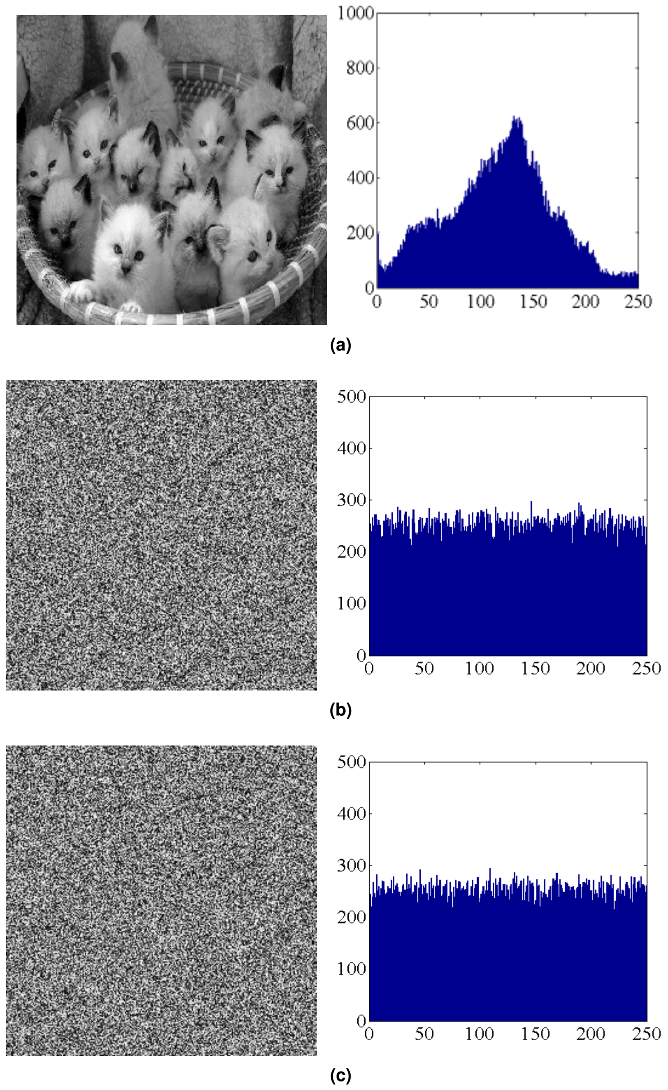


Figure 11 a) 8 bit reference image and histogram analysis **b)** encrypted image with y phase and histogram analysis **c)** encrypted image with z phase and histogram analysis

Entropy and correlation analysis

Another method used as security analysis in the encrypted image is the entropy and correlation coefficients. In the literature, the entropy value of a well-encrypted image is considered to be 8, while correlation values are expected to be as close to 0 as possible (Akhshani *et al.* 2010).When looking at Table 2, the correlation values of the source image are close to 1, while the correlation values of the encrypted images are close to 0. Again, while the entropy value of the source image is below 8, the entropy values of the encrypted images are close to 8. Considering the correlation and entropy coefficients, it can be interpreted that encryption is good. Another application used as a measure of success in image encryption is histogram maps. The correlation distributions of unencrypted images are concentrated diagonally. A well-encrypted image should have a homogeneous distribution of correlation distributions (Sivakumar and Li 2019). When look the correlation maps of the source image in Figure 13, there is a concentration in the diagonal. On the other hand, when the correlation maps of the images encrypted with the y and z phases in Figure 12 are examined, it is seen that there is a homogeneous distribution. Considering the correlation distributions, it can be interpreted that encryption is good.

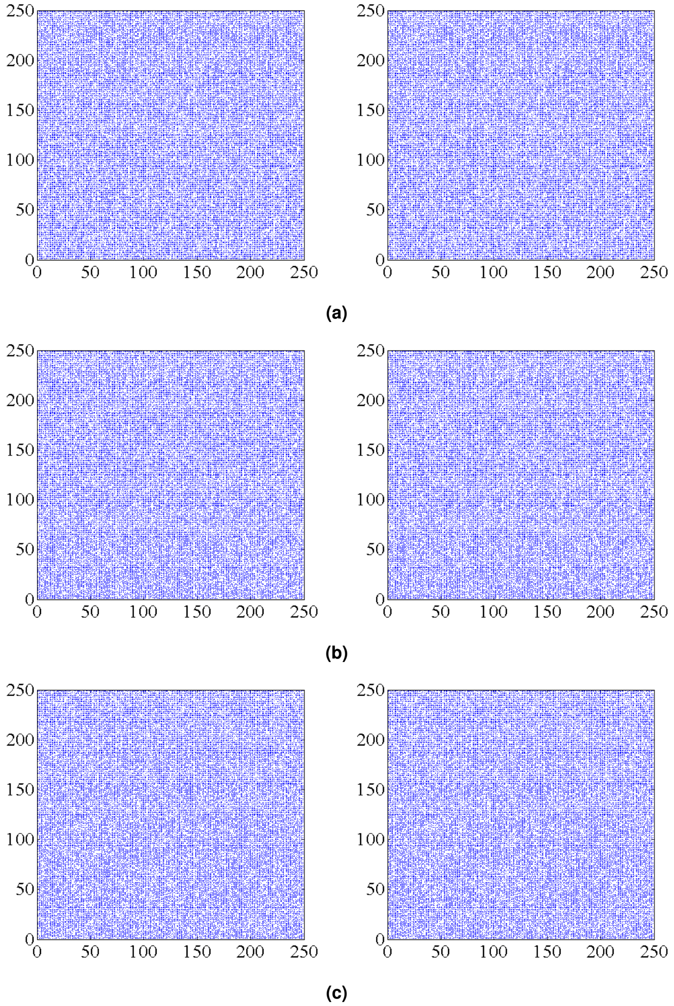


Figure 12 a) diagonal correlation maps of the image encrypted with random numbers generated from the y,z phase **b)** row correlation maps of the image encrypted with random numbers generated from the y,z phase **c)** column correlation maps of the image encrypted with random numbers generated from the y,z phase

Table 2 Values of entropy and correlation

Sample images	Values of horizontal correlation	Values of vertical correlation	Entropy
8 bit reference image	0,91806	0,88837	7,6827
encrypted image with y phase	0,0013293	0,0020999	7,9973
encrypted image with y phase	-0,003312	-0,00016575	7,9976

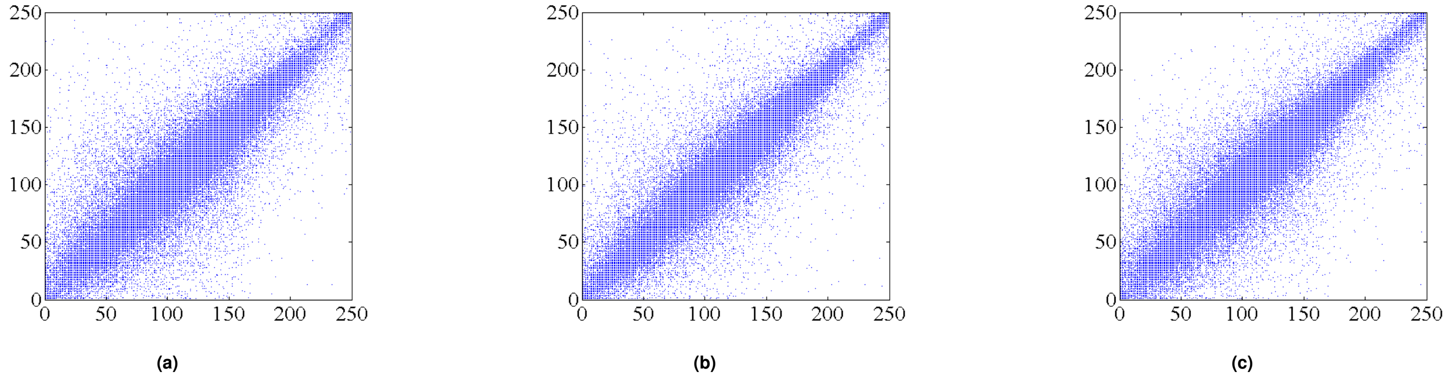


Figure 13 a) diagonal correlation maps of the image encrypted with random numbers generated from the y,z phase **b)** row correlation maps of the image encrypted with random numbers generated from the y,z phase **c)** column correlation maps of the image encrypted with random numbers generated from the y,z phase

Differential attack analysis

UACI and NPCR values are the most common statistics used to measure the strength of the encryption algorithm. UACI states how many percent of pixels change, while NPCR indicates how many percent of pixels change. In the literature, %30 of the UACI value and %99.6 of the NPCR value are stated as a safety criterion against differential attacks (Praveenkumar *et al.* 2015). Considering the values in Table 3, it can be interpreted that the images encoded with the y and z phases are resistant to differential attacks.

Table 3 NPCR and UACI values of encrypted image

Sample images	NPCR	UACI
encrypted image with y phase	0,99573	0,29265
encrypted image with z phase	0,99605	0,2939

CONCLUSION

In this study, random numbers were generated in LabVIEW and image encryption was applied with these numbers. NIST-800-22 test was used to test the reliability of these numbers. As a result of the test, the random numbers generated from the y and z phases of the 3-dimensional chaotic system have passed successfully, while the random numbers generated from x have been unsuccessful. Image encryption application was made with the numbers passed the test. Encrypted images have been successfully passed through histogram, entropy, correlation, NPCR, UACI analysis in the literature, and the reliability of the encryption algorithm has been confirmed. In this study, unlike the random number generation and encryption studies performed before, an interface is designed to generate chaos-based random number generation and image encryption with these numbers. The application has been verified by the analyzes performed properly.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

LITERATURE CITED

Akgül, A., 2015 *Yeni Kaotik Sistemler İle Rasgele Sayi Üreteci Tasarımı Ve Çoklu Ortam Verilerinin Yüksek Güvenlikli Şifrelenmesi*. Ph.D. thesis, Sakarya Üniversitesi.

Akgul, A., S. Hussain, and I. Pehlivan, 2016 A new three-dimensional chaotic system, its dynamical analysis and electronic circuit applications. Optik **127**: 7062–7071.

Akgül, A., M. Z. Yıldız, Ö. F. Boyraz, E. Güleryüz, S. Kaçar, *et al.*, 2020 Microcomputer-based encryption of vein images with a non-linear novel system. Journal of the Faculty of Engineering and Architecture of Gazi University **35**: 1369–1385.

Akhshani, A., S. Behnia, A. Akhavan, H. A. Hassan, and Z. Hassan, 2010 A novel scheme for image encryption based on 2d piece-wise chaotic maps. Optics Communications **283**: 3259–3266.

Alcin, M., I. Koyuncu, M. Tuna, M. Varan, and I. Pehlivan, 2019 A novel high speed artificial neural network–based chaotic true random number generator on field programmable gate array. International Journal of Circuit Theory and Applications **47**: 365–378.

Avaroğlu, E. and M. Türk, 2013 Random number generation using multi-mode chaotic attractor. In *2013 21st Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE.

Buchovecká, S., R. Lórencz, F. Kodỳtek, and J. Buček, 2017 True random number generator based on ring oscillator puf circuit. Microprocessors and Microsystems **53**: 33–41.

Çavuşoğlu, Ü., Y. Uyaroğlu, and I. Pehlivan, 2014 Design of a continuous-time autonomous chaotic circuit and application of signal masking .

Chai, X., Z. Gan, Y. Chen, and Y. Zhang, 2017 A visually secure image encryption scheme based on compressive sensing. Signal Processing **134**: 35–51.

Gürevin, B., 2019 *Mikrobilgisayar Tabanlı Yeni Bir Rasgele Sayı Üreteci Tasarımı ve Şifreleme Uygulaması*. Master's thesis, Sakarya Uygulamalı Bilimler Üniversitesi, bilalsau@gmail.com.

Guyeux, C., Q. Wang, and J. M. Bahi, 2010 A pseudo random numbers generator based on chaotic iterations: application to watermarking. In *International Conference on Web Information Systems and Mining*, pp. 202–211, Springer.

Keleş, C., 2012 *Kaotik haritalar kullanarak görüntü şifreleme*. Ph.D. thesis, Karadeniz Teknik Üniversitesi.

Koyuncu, İ., M. Tuna, İ. Pehlivan, C. B. Fidan, and M. Alçın, 2020 Design, fpga implementation and statistical analysis of chaos-ring based dual entropy core true random number generator. Analog Integrated Circuits and Signal Processing **102**: 445–456.

Lynnyk, V., N. Sakamoto, and S. Čelikovskỳ, 2015 Pseudo random number generator based on the generalized lorenz chaotic system. IFAC-PapersOnLine **48**: 257–261.

Ozdemir, A., I. Pehlivan, A. Akgul, and E. Guleryuz, 2018 A strange novel chaotic system with fully golden proportion equilibria and its mobile microcomputer-based rng application. Chinese Journal of Physics **56**: 2852–2864.

Pamuk, N., 2016 Dinamik sistemlerde kaotik zaman dizilerinin tespiti. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi **15**: 78–92.

Pareschi, F., G. Setti, and R. Rovatti, 2010 Implementation and testing of high-speed cmos true random number generators based on chaotic systems. IEEE transactions on circuits and systems I: regular papers **57**: 3124–3137.

Praveenkumar, P., R. Amirtharajan, K. Thenmozhi, and J. B. B. Rayappan, 2015 Pixel scattering matrix formalism for image encryption—a key scheduled substitution and diffusion approach. AEU-International Journal of Electronics and Communications **69**: 562–572.

Sivakumar, T. and P. Li, 2019 A secure image encryption method using scan pattern and random key stream derived from laser chaos. Optics & Laser Technology **111**: 196–204.

Tuna, M., M. Alçın, İ. Koyuncu, C. B. Fidan, and İ. Pehlivan, 2019a High speed fpga-based chaotic oscillator design. Microprocessors and Microsystems **66**: 72–80.

Tuna, M. and C. Fidan, 2018 A study on the importance of chaotic oscillators based on fpga for true random number generating (trng) and chaotic systems .

Tuna, M., A. Karthikeyan, K. Rajagopal, M. Alcin, and İ. Koyuncu, 2019b Hyperjerk multiscroll oscillators with megastability: Analysis, fpga implementation and a novel ann-ring-based true random number generator. AEU-International Journal of Electronics and Communications **112**: 152941.

Uhl, A. and A. Pommer, 2004 *Image and video encryption: from digital rights management to secured personal communication*, volume 15. Springer Science & Business Media.

Wang, X., L. Teng, and X. Qin, 2012 A novel colour image encryption algorithm based on chaos. Signal Processing **92**: 1101–1108.

How to cite this article: Gurevin, B., Yildiz, M., Guleryuz, E., Kutlu, M. C., and Sorgun, O. A Chaos Based Image Encryption On LabVIEW. Chaos Theory and Applications, 2(2), 69-76, 2020.