Turkish Journal of Science & Technology
Volume 12(2), 113-119, 2017

FPGA Implementation of a Chaotic Quadratic Map for Cryptographic Applications

Hidayet OĞRAŞ, Mustafa TÜRK
Technical Education Faculty, Batman University, Batman, Turkey
Electrical and Electronics Engineering, Firat University, Elazig, Turkey
hidayet.ogras@batman.edu.tr

(Geliş/Received:23.02.2017; Kabul/Accepted:06.04.2017)

Abstract

A hardware implementation of a quadratic map through FPGA platform is proposed in this paper. Firstly, a chaotic quadratic map is modeled by using Matlab/Simulink programming and then implemented into the FPGA (Field Programmable Gate Array) to be used for key generation for cryptographic applications. When the quadratic map is in chaotic mode, its output is unpredictable and aperiodic. Besides this, the map has a uniform output distribution and sufficient randomness. These characteristics make the chaotic quadratic map a suitable key generator for cryptography. This paper also reveals the successful real-time implementation of the quadratic map using FPGA for practical applications. Experimental results confirm that the feasibility of the quadratic map is verified under a digital hardware environment.

Keywords: Chaos; Quadratic Map; Implementation; FPGA

Kriptografik Uygulamalar için Kaotik Kuadratik bir Haritanın FPGA Gerçekleştirilmesi

Özet

Bu çalışmada, kuadratik bir haritanın FPGA üzerinden donanımsal gerçekleştirilmesi sunulmuştur. İlk olarak, kaotik kuadratik haritası Matlab/Simulink yazılımı kullanılarak modellenmiş ve daha sonra kriptografik uygulamalar için FPGA (Sahada Programlanabilir Kapı Dizileri) ortamında anahtar üreteci olarak gerçekleştirilmiştir. Kuadratik harita kaotik durumda iken sistem çıkışı tahmin edilemez ve düzensizdir. Ayrıca harita, düzgün bir çıkış dağılımına ve yeterli seviyede rastgeleliğe sahiptir. Bu karakteristik özellikler kaotik kuadratik haritasını kriptografi için uygun bir anahtar üreteci yapmaktadır. Bu çalışma aynı zamanda pratik uygulamalara yönelik olarak kuadratik haritasının FPGA ortamındaki başarılı gerçek zamanlı uygulamasını ortaya koymaktadır. Deneysel sonuçlar kuadratik haritanın uygulanabilirliğini sayısal donanım ortamında göstermiştir.

Anahtar Kelimeler: Kaos; Kuadratik Harita; Gerçekleştirme; FPGA

1. Introduction

Chaos theory in complex systems has been cited increasingly in several different scientific areas especially in engineering science such as secure communication and cryptography. For example, chaos is used for analog and digital communication systems in [1-6]; for image cryptosystems in [7-12] and is applied in electrical power systems in [13-16]. Chaotic systems have similar properties such as sensitivity to initial conditions and control parameters, pseudo-random behavior and mixing with modern cryptography. These fundamentals characteristics can make the chaotic systems a good candidate for the key generation in data encryption algorithms. Many cryptosystems based on the generation of pseudo-random sequences using chaos have been proposed recently for mixing clear messages in information security [17].

Chaos generation in discrete time systems is very easy and simple due to the low complexity, but having high efficiency comparing with analog chaos generators [18]. In fact, analogue chaotic systems typically exhibit some practical difficulties since the component conditions are varying with age, temperature, etc. Furthermore, analog circuit implementations generally require a large chip area for realization. Hence, hardware implementations of the discrete chaotic systems can be a solution to overcome these problems. Many digital realizations of chaotic systems have been reported to be used as key generators in cryptographic applications. For instance, in [19], Henon map as a chaotic generator is implemented in real-time on a FPGA to obtain high frequency at output for chaotic communication. In other study [20], a chaotic map is used as a bit generator and its FPGA implementation is performed successfully for cryptographic applications. Generally, chaotic system is used to generate pseudo-random sequences as key streams to mask information. using Xilinx blocks in MATLAB/Simulink and then Xilinx system generator (XSG) performs the compilation of the design.

The rest of the paper is structured as follows: Section 2 briefly introduces the Quadratic map with its dynamical behaviors and some statistical analyses of the map are performed. In Section 3, we realize the digital implementation and hardware-co simulation of the Quadratic map on FPGA. Finally, Section 4 concludes the whole paper.

2. Chaotic Quadratic System

A. Quadratic Map

Quadratic map is a simple discrete system exhibiting chaos and defined by [22],

$$x_{n+1} = r - x_n^{\,2} \tag{1}$$

where $0 < r \le 2$ is called control parameter and $x_n \in (-2,2)$ is the state variable of the system. Quadratic map can show rich dynamic behaviors from a stationary system to a chaotic state. When $r \in (0, 0.74)$, the map behaves in steady state and if $r \in [0.74, 1.5)$, then the map has periodic behavior. When $r \in [1.5, 2]$, the Quadratic map is capable of very complicated behavior which means that the output of the map is aperiodic, non-convergent and very sensitive to initial conditions. Hence, the value of the control parameter specifies the dynamical behavior of the system.

B. Lyapunov and Bifurcation Analyses

Lyapunov exponent checks a sensitivity criterion of the initial condition for a nonlinear dynamical system [23]. In discrete systems,

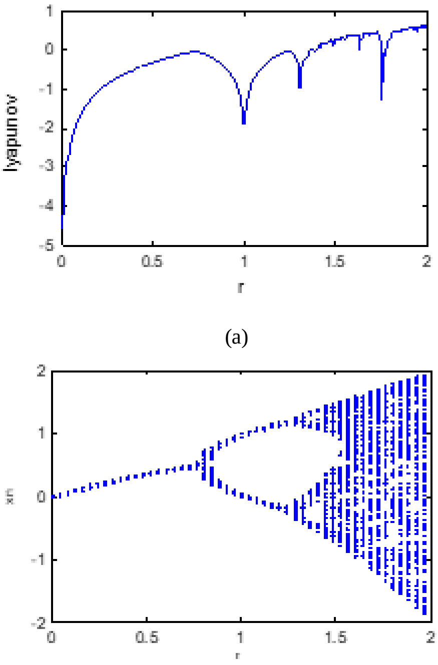
Key streams should be generated randomly and contain enough entropy in order to prevent the key from being guessed. Key sensitivity is also required by secure cryptosystems [21].

In this paper, we consider a quadratic map and present direct real-time implementation into the FPGA as well as hardware co-simulation structure in Simulink. Xilinx ISE (Integrated Synthesis Environment) design software including system generator tool is one of the efficient software technologies, is used to design and implement the chaotic Quadratic map. Firstly, the map equation is modeled by

Lyapunov values are given by the following equation.

$$\lambda = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right| \tag{2}$$

A positive Lyapunov exponent indicates that the orbit of a dynamical system is unstable and chaotic. The dynamical behaviors of a system from a fixed point to a chaos as a function of its control parameter are shown by a bifurcation diagram. Fig. 1 shows the Lyapunov spectrum and the bifurcation diagram of the Quadratic map.



(a)

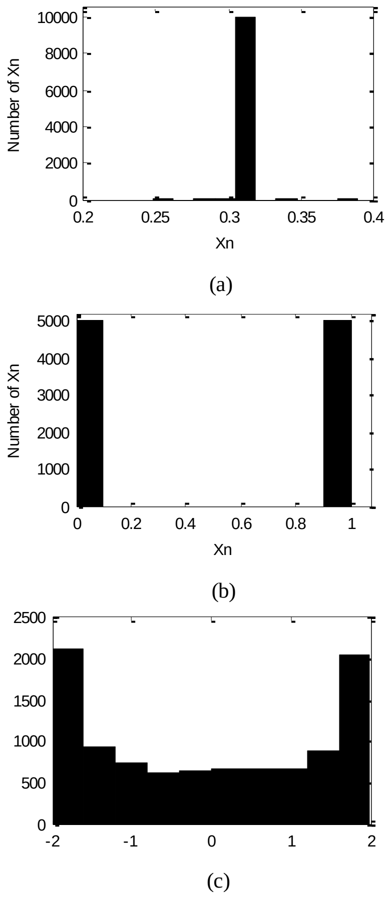
(b)

Figure 1 (a) Lyapunov spectrum of the Quadratic map (b) Bifurcation diagram of the Quadratic map

As it is easily observed that when the control parameter is close to 2, then the Lyapunov values are positive and the bifurcation diagram displays complex behavior resulting chaos.

C. Histogram Analysis

Histogram is a graphical display for the frequency distribution of a set of data. A distribution having constant probability for each data is known as uniform distribution. Fig. 2 shows the histogram distribution of the x_n series generated from the Quadratic map with different control parameters. From the graphical results, Fig. 2(a), Fig. 2(b) and Fig. 2(c) demonstrate steady, periodic and chaotic behavior of the Quadratic map, respectively. It is obvious that the histogram has an excellent symmetric property and better uniform distribution when the map behaves chaotically as in Fig. 2(c).



(a)

(b)

(c)

Figure 2. Histogram of x_n series with different control parameter (a) $r = 0.4$ (b) $r = 1$ (c) $r = 2$

D. Checking Chaotic Output

When the Quadratic map is in chaos state, it exhibits complex behavior and generates chaotic sequences at output. Firstly, chaotic output will be checked for homogeneity through central tendency analysis and then the randomness of these sequences will be evaluated by using NIST test. Finally, the entropy of the Quadratic map as a number generator will be determined to measure its uncertainty. If a key generator is used in a cryptosystem, these properties need to be confirmed. In this paper, we chose as $r = 2$ to make the system chaotic and perform the following statistical analyses by using Matlab programming.

1) Homogeneity Analysis

In order to check the chaotic output of the Quadratic map, the following two propositions are considered. First of all, the mean value of the output sequences spreading between (-2,2) should be

$$x_{mean} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} x_k = 0 \qquad (3)$$

and second, the self-correlation of these sequences should be zero as given in the following equation.

$$s(\beta) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{\infty} (x_k - x_{mean}).(x_{k+\beta} - x_{mean}) = 0 \quad (4)$$

According to the above equations, based on 50 simulations with different initial conditions, we have performed 10^6 iterations to get sufficient number of chaotic sequences from the Quadratic map. Then, we got the average mean value 0.000174 and the self-correlation is calculated to be 0.001598. These results are quite good, because both are very close to zero.

2) Randomness Analysis

Randomness means unpredictability and does not follow an intelligible pattern in a sequence of symbols [24]. NIST test is used to determine the degree of randomness of the Quadratic map outputs. NIST includes fifteen tests [25] and each test produces a real *p*-value in [0,1]. If the *p*-value is greater than a significance predefined level such as $\alpha = 0.01$, then the test is passed successfully. When the all statistical tests are passed, then the map is considered as random generator with 99% confidence. NIST uses binary series to test the randomness, but the output of the chaotic Quadratic map is floating-point value. Therefore, the following transformation is used for the output of the map in order to get sequential bit streams.

$$b_n = \begin{cases} 1 \ , \ x_n \geq 0 \\ 0 \ , \ x_n < 0 \end{cases} \quad (5)$$

Here, a threshold level of 0 is selected to produce a bit value "1" or "0" from x_n. We preferred the initial value as $x_0 = 0.123$ to obtain 1,000,000 bits to proceed NIST suite. The results are listed in Table 1.

Table 1 Results of the NIST test

Test Name	*p-value*	*Result*
Frequency	0.8524	Passed
Block frequency	0.3093	Passed
Runs	0.4939	Passed
Long runs of ones	0.7852	Passed
Rank	0.9912	Passed
Spectral DFT	0.7204	Passed
Non-overlapping templates (m=9; B=000000001)	0.7659	Passed
Overlapping templates (m=9)	0.7819	Passed
Universal (L=7; Q=1280)	0.1201	Passed
Liner complexity	0.9138	Passed
Serial-1 (m=5)	0.5875	Passed
Serial-2 (m=5)	0.6469	Passed
Approximate entropy (m=5)	0.4142	Passed
Cumulative sums forward	0.4086	Passed
Cumulative sums reverse	0.5553	Passed
Random excursions (x=+1)	0.3511	Passed
Random excursions variant (x=-1)	0.8741	Passed

It is concluded that the chaotic Quadratic map is very stochastic that represents random process and generates output sequences having enough randomness according to the NIST results.

3) Uncertainity Analysis

We use information entropy to determine the uncertainty or disorder of the Quadratic map. Entropy is a measure of uncertainty related to a random event [24, 26]. If H(X) is a random source with *N* length, then its entropy is

$$H(X) = -\sum_{i=1}^{N} p(x_i).\log_2 p(x_i) \quad (6)$$

where $p(x_i)$ represents the probability of x_i. For instance, in a uniform bit stream having equal probability '0' and '1', the entropy will be 1 which is a theoretical result. When the output is certain, then the entropy is zero. The entropy of an practical information source is smaller than the ideal one. Generally, the more uncertain or random the event is, the more entropy it will contain [11].

We have used different initial values and number of iterations in order to generate bit streams using the Eqn. (5) from the chaotic Quadratic map. The entropy results for different conditions of the map are listed in Table 2.

Table 2 Entropy results

Initial value	*n*	*# of '0'*	*# of '1'*	*p(0)*	*p(1)*	*Entropy*
0.2	100	52	48	0.52	0.48	0.998845
-0.315	1000	490	510	0.49	0.51	0.999711
1.27	10,000	4,967	5,033	0.4967	0.5033	0.999968
-0.88354	100,000	49,829	50,171	0.4982	0.5017	0.999991

From the results, when the number of iteration is increased, then the entropy value closes to 1 which means that the uncertainty of the map is becoming greater. Generated for all bit streams, number of zeros and ones are very close to each other resulting uniform distribution in the sequences.

4) Sensitivity Analysis

Quadratic map is highly sensitive to initial value. Thus, arbitrarily small change in the initial value will cause significantly different future output. This property is also acceptable while the map is used as a bit generator. To perform the sensitivity analysis, firstly, we randomly choose an initial value $x_0 = 0.123456788$ and iteration of $n = 50$ to generate a bit sequence (b_1) from the map. Then, a very slight change of 10^{-9} is applied to the first initial value, such as $x_0 = 0.123456789$ to generate another bit sequence (b_2). The last ten elements for both sequences are shown in Fig. 3.

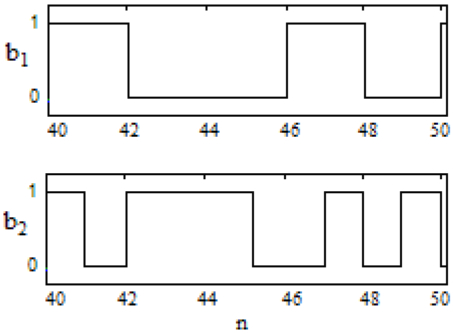


Figure 3 Generated different bit sequences with a slight change of initial value

Fig. 3 states that when a tiny change occurs in the initial value of the chaotic Quadratic map, generated bit sequences are completely different.

3. Digital Implementation

This section describes an approach to the real-time implementation as well as hardware simulation of the chaotic Quadratic map on FPGA. FPGA is a type of programmable chip that can be completely reconfigured for various field applications. Using prebuilt logic blocks and programmable routing resources, FPGAs can be reprogrammed to the required functionality and customized by loading the related configuration data into its internal memory cells. The stored data in these cells determine the logic blocks and reconfigurable interconnects in FPGA. We have used Spartan 3E-XC3S1600E family from Xilinx for the hardware simulation and implementation of the Quadratic map.

Xilinx System Generator (XSG) is a high-level design tool and fully integrated in MATLAB/Simulink that enables the use of the model-based Simulink environment for FPGA design. It allows compilation of the design that is captured using Xilinx blocks and generates synthesizable VHDL (Very High speed integrated circuit Hardware Description Language) codes for FPGA programming. All of the downstream implementation steps including synthesis, place and root processes are automatically performed to generate the programming file via XSG. The Quadratic map model has been designed by Matlab/Simulink with XSG which offers the library of fixed-point arithmetic blocks that can be directly implemented into the FPGA. Fig. 4 shows the Quadratic map model created by Xilinx blocks under the Simulink.

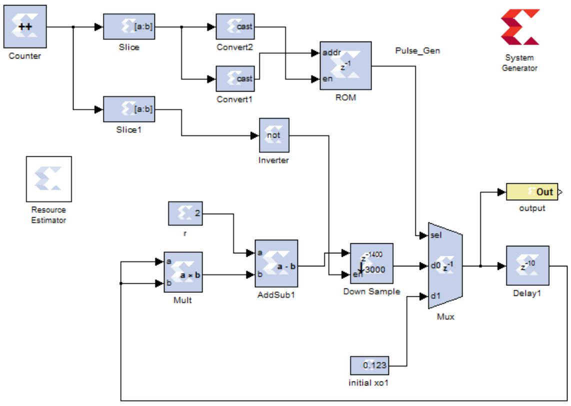


Figure 4 Chaotic Quadratic map model using Xilinx blocks

XSG enables hardware into a simulation, called hardware co-simulation structure that allows incorporating a design running in an FPGA directly into a simulation. Hardware co-simulation compilation targets automatically generate a bit streams and associate it to a block. When this block is simulated in Simulink, then the results for the compiled part are calculated in the hardware. Hence, hardware co-simulation is used to verify that the design actually works in FPGA platform.

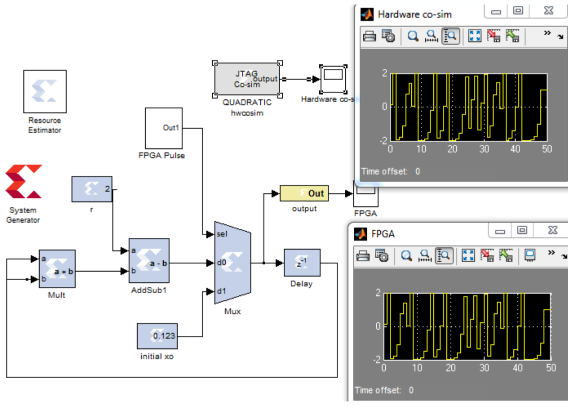


Figure 5 Hardware co-simulation of the Quadratic map

The bitstream download step is performed by using a JTAG cable. We performed the real-time implementation with a fixed-point data type and the real data are represented on 128 bits. Fig. 5 shows the simulation results of the chaotic Quadratic map design with hardware and software in Simulink. It is observed that the hardware-co simulation result is same to the Simulink simulation which means that the realization of the map is performed successfully and the map design actually works in FPGA.

The generated output depends on the initial value of the map that can be directly entered into the design model before the generator starts. We randomly chose the initial value as $x_0 = 0.123$ for the real-time FPGA implementation.

XSG tool automatically generates a synthesizable VHDL codes associated with the design and the created file can be opened with the Xilinx ISE software. PlanAhead tool in ISE is used to assign input and output pin locations in the design. After assigning pins for input and output, then the design is ready to be synthesized in ISE. Successful synthesis creates the programming file of the design. IMPACT tool is used to load the programming file into the FPGA. For real-time implementation of our design, we use the chaotic Quadratic map as an 8-bit number generator to observe the numbers at LED output of the FPGA. First, the output of the map needs to be converted to 8-bit decimal number between 0 and 255. Hence, the following equation is applied to the output of the map.

$$number = \text{mod}(round(x_n \times 10^9), 256) \qquad (7)$$

Here, *round* operation is used to get the nearest integer value and *mod* limits the output between 0 and 255. We assigned 8-bit number for the implementation because our FPGA has eight LEDs at output. Transformation module is also added to the Quadratic map design. For example, if the initial value of the map is 0.123, then the second number generated from the Eqn. (7) will be 88 in decimal or 01011000 in binary. Fig. 6 shows this value at LED output of the FPGA.

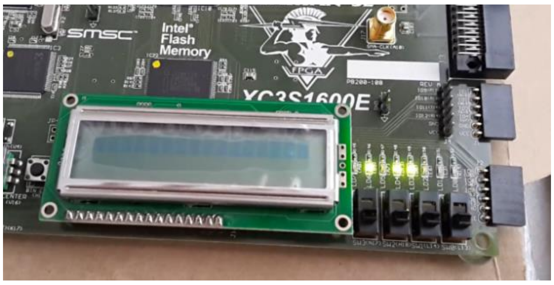


Figure 6 Display of 88 in binary at LED output of the FPGA

Table 3 shows the numbers generated from the chaotic Quadratic system by using MATLAB and FPGA with the same initial value of the map.

Table 3 Generated 8-bit numbers from MATLAB and FPGA

MATLAB (Software)	*FPGA (Hardware)*
192	192
88	88
137	137
189	189
57	57
133	133
12	12
220	220

The amount of FPGA resources and the required by the Quadratic map can be determined by using Resource Estimator block. They are listed in Table 4.

Table 4 Mapping report of the Quadratic map design

Device	**Spartan3E-XC3S1600E**				
Resource Type	***Slices***	***Flip-Flops***	***RAMB 16S***	***LUTs***	***IOBS***
Available	14,752	29,504	36	29,504	250
Used	10,160	10,922	1	18,843	9

4. Conclusion

This paper presents a chaotic Quadratic map and its implementation on a digital hardware. The results of the statistical analyses confirm that the output of Quadratic map can be used as cryptographic keys when the map behaves chaotically. In practice, chaotic Quadratic map can be used as a generator in all scientific fields where the pseudo-randomness and chaos are required. The design of the map as well as hardware co-simulation and real-time implementation are successfully applied to the FPGA platform that encourages its usage for practical applications. This paper can be used as a good guide for anyone who wants to implement digital designs on FPGA without knowing VHDL codes.

5. References

1. Kang, Z., Sun, J., Ma, L., Qi, Y. and Jian, S., (2014). Multimode synchronization of chaotic semiconductor ring laser and its potential in chaos communication. IEEE journal of Quantum Electronics, vol. 50, pp. 148-157.

2. Yang, J., Chen, Y. and Zhu, F., (2015) .Associated observer-based synchronization for uncertain chaotic systems subject to channel noise and chaos-based secure communication. Neurocomputing, vol. 167, pp. 587-595.

3. Eisencraft, M., at al., (2012). Chaos-based communication systems in non-ideal channels. Communications in Nonlinear Science and Numerical Simulation, vol. 17, pp. 4707-4718.

4. Kaddoum, G., Coulon, M., Roviras, D. and Charge, P., (2010). Theoritical performance for asynchronous multi-user chaos-based communication systems on fading channels. Signal Processing, vol. 90, pp. 2923-2933.

5. Zaher, A. A. and Abu-Rezq, A., (2011). On the design of chaos-based secure communication systems, Communications in Nonlinear Science and Numerical Simulation, vol. 16, pp. 3721-3737.

6. Turk, M. and Ogras, H., (2011). Classification of chaos-based digital modulation techniques using wavelet neural networks and performance comparison of wavelet families. Expert Systems with Applications, vol. 38, pp. 2557-2565.

7. Zhu, Z. L., Zhang, W., Wong, K. W., Yu, H., (2011). A Chaos-based symmetric image encryption scheme using a bit-level permutation. Information Sciences, vol. 181, pp. 1171-1186.

8. Patidar, V., Pareek, N. K., Purohit, G. and Sud, K. K., (2011). A Robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. Optics Communications, vol. 284, pp. 4331-4339.

9. Murillo-Escobar, M. A. et al., (2015). A RGB image encryption algorithm based on total plain image characteristics and chaos. Signal Processing, vol. 109, pp. 119-131.

10. Ye, R., and Guo, W., (2014). An image encryption scheme Multimode synchronization of chaotic semicon based on chaotic systems with changeable parameters," I. J. Computer Network and Information Security, vol. 4, pp. 37-45.

11. Zhu, H., Zhao, C. and Zhang, X., (2013). A Novel image encryption-compression scheme using hyper-chaos and Chinese remainder theorem. Signal Processing: Image Communication, vol. 28, pp. 670-680.

12. Ogras, H. and Turk, M., (2017). A Robust chaos-based image cryptosystem with an improved key generator and plain image sensitivity mechanism. Journal of Information Security, vol. 8, pp. 23-41.

13. Yibei, W., Man, L., Yanting, X. and Hougui, C., (2011). Research on chaos phenomena in power systems. Power engineering and automation conference, vol. 2, pp. 453-456.

14. Yau, H. T., Wang, M. H., Wang, T. Y. and Chen, G., (2015). Signal clustering of power disturbance by using chaos synchronization. Int. J. Electr. Power Energy System, vol. 64, pp. 112-120.

15. Ghasemi, M., Ghavidel, S., Aghaei, J., Gitizadeh, M. and Falah, H., (2014). Application of chaos-based chaotic invasive weed optimization techniques for environmental OPF problems in the power systems. Chaos, Solitons Fract., vol. 69, pp. 271-284.

16. Chen, Q., Ren, X. and Na, J., (2015). Robust finite-time chaos synchronization of uncertain permament magnet synchronous motors. ISA Trans., vol. 58, pp. 262-269.

17. Merah, L., Ali-Pacha, A., Said, N. H. and Mamat, M., (2013). Design and FPGA implementation of Lorenz chaotic system for information security issues. Applied Mathematical Sciences, vol. 7, pp. 237-246.

18. Xue, H., Wang S. and Meng, X., (2013). Study on one modified chaotic system based on Logistic map. Res. J. Appl. Sci. Eng. Technol., vol. 5, pp. 898-904.

19. Aseeri, M. A. and Sobhy,M. I., (2002). A New approach to implement Chaotic generators based on Field Programmable Gate Array (FPGA). Proc. 3rd. Int. Conf. Discrete Chaotic Dynam. Nature Soc., September.

20. Mao, Y., Cao, L. and Liu, W., (2006). Design and FPGA implementation of a pseudo-random bit sequence generator using spatiotemporal chaos. IEEE Proceedings of International Conference on Communications, Circuits and Systems, pp. 2114-2118.

21. Lian, S., Sun, J. and Wang, Z., (2005). Security analysis of a chaos-based image encryption algorithm. Physica A: Statistical Mechanics and its Applications, vol. 351, pp. 645-661.

22. Ramadan, N., Ahmed, H. E., Elkhamy H S. E., and Abd El-Samie, F. E., (2016). Chaos-based image encryption using an improved quadratic chaotic map. American Journal of Signal Processing, vol. 6, pp. 1-13.

23. Hathal, H. M., Abdulhussein, R. A. and Ibrahim, S. K., (2014). Lyapunov exponent testing for AWGN generator system. Communications and Network, vol. 6, pp. 201-208.

24. Marton, K., Suciu, A., Sacarea, C. and Cret, O., (2012). Generation and testing of random numbers for cryptographic applications. Proceedings of the Romanian Academy, vol. 13, pp. 368-377.

25. Rukhin, A., Soto, J., Nechvatal, J. and Smid, M., (2010). A Statistical Test for random and psudorandom number generators for cryptographic applications. NIST Special Publication 800-22 rev1, pp. 2-40.

26. Chen, J. X., Zhu, Z. L., Fu, C., Yu, H. and Zhang, L.B (2015). A Fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. Communications in Nonlinear Science and Numerical Simulation, vol. 20, pp. 846-860.