ISSN 2047-3338

Coupling Two Chaotic Systems in Order to Increasing the Security of a Communication System - Study and Real Time FPGA Implementation

Lahcene Merah, Adda ALI-PACHA and Naima HADJ SAID
Department of Electronics, University of Science and Technology of Oran (USTO), BP 1505 El M'Naouer, Oran 31036, Algeria
merahlah@gmail.com, alipacha@yahoo.com, nim_hadj@yahoo.fr

***Abstract*– A Pseudo Random Numbers Generator (PRNG) based on chaotic systems is developed in this paper. The main goal of this paper is to increase the security of a chaotic crypto system especially against known plaintext and chosen plaintext attacks by coupling two chaotic systems, the Hénon map used to generate the chaotic sequence and the logistic map to control a multiplexer that outputs is x(k) or x(k-1) of the Hénon system according to the value generated by the logistic map. In order to generate a strong pseudo random sequence, the output of the multiplexer is then pass through a logical circuit which is also controlled by the logistic map. After the analyses of simulation results, the design will be implemented on FPGA reconfigurable Hardware. The results show that the proposed generator generate a pseudo random sequences that satisfy all the NIST statistical tests, offers a high throughput and has low cost for hardware implementation.**

***Index Terms*– PRNG, Henon Chaotic Map, Logistic Map, Chaos Encryption, FPGA and NIST Tests**

I. INTRODUCTION

WITH rapid development of computer networks and wireless communications, information security becomes one of the major problems for effective use of information technology [1]. By using cryptography, many of security problems were solved and a lot of effective cryptographic PRNGs and algorithms were developed in the recent years, but in the same time there is another path that is in evolution too and threatening cryptographic systems, it is the cryptanalysis. So, strongest and effective crypto system is always a subject of research.

The well-known features of the chaotic systems like strong dependence on initial data, topological transitivity, wide spread spectrum of its signal, etc., directly suggest the idea to use suitable chaos generators to build a new generation of secure encryption methods [2]. Chaotic communication gains more and more concern on its research and application in the communication areas [3], chaotic systems are the nonlinear dynamical systems whose state evolves with time. The future dynamics of these systems are fully defined by their initial conditions. As a result, the behavior of these systems appears random [4].

Recently, many approaches and techniques for the secure transmission of information signals using chaotic dynamics have been proposed [5], for example, chaotic masking [6], [7], chaotic modulation [8] and other. Due to their random aspect, chaotic signals are used as pseudo random number generators on many recent proposed cryptographic approaches (chaotic synchronization), in which the bits of the plaintext XORed with the bits of the chaotic sequence, this kind of encryption called stream cipher. The plaintext can be recovered again by XORing the encrypted message with the same chaotic sequence. In fact, sometimes these kinds of crypto systems cannot resist what known plaintext and chosen plaintext attacks that can successfully be launched against such cryptosystems to recover the system parameters and subsequently eavesdrop on the message transmission. In order to remedy this problem we propose a new approach that based on coupling two chaotic systems, the first called control system (CRS) and the second called the encryption system (ENS). The CRS based on the one dimensional chaotic system of the logistic map and the ENS based on the two dimensional chaotic system of the Hénon map, the principle of functioning of the proposed chaotic generator (PCG) is described in section III.

This paper is organized as follow. In section II we introduce an over view of the chaos theory, its application in the cryptographic and the most knows cryptographic attacks. In section III the PCG is presented. Section IV presents the evaluation of the PCG and section V presents the real time FPGA implementation of our PCG.

II. CHAOS AND CRYPTOGRAPHY

A. Chaotic Systems

Randomness means different things in various fields. Commonly, it means lack of pattern or predictability in events [9], it is an abstract concept that has been very useful but that, strictly speaking, cannot be realized on a computer. Indeed, many models in physics-such as random walks, localization, and percolation-are based on the concept of randomness [10]. Alternatively, a deterministic model takes initial conditions in

and gives out an exact trajectory of the system. If it is a chaotic system, then the trajectories will vary wildly as you shift the initial conditions even slightly [11], this is well property of chaos, which is to say that the very similar initial conditions can generate two completely different sequences after a number of iterations, and we cannot predict the next number by the sequence itself, and the sequence has white noise, which is to say that it distributes equiprobably on the range [12]. We cannot have chaos without determinism. Chaos is not the lack of order. Chaos is order that is very sensitive to initial conditions; it's not random at all.

B. Chaos Based Cryptography

Since 1990s, many researchers have found that there exists some interesting relationship between chaos and cryptography [13]. The possibility for self-synchronization of chaotic oscillations has sparked an avalanche of works on the application of chaos in cryptography. The random behavior and sensitivity to initial conditions and parameter settings allows chaotic systems to fulfill the classic Shannon requirements of confusion and diffusion [2]. The chaos-based secure communication has become an important and significant research direction. Now various methods for chaos-based secure transmission of private information signals have been proposed, some popular methods are additive masking, chaotic switching, chaotic parameter modulation, chaos shift keying and chaotic frequency modulation [14].

C. Cryptographic Attacks

Cryptographic attacks are designed to subvert the security of cryptographic algorithms, and they are used to attempt to decrypt data without prior access to a key. They are part of Cryptanalysis, which is the art of deciphering encrypted data [15]. There are many kinds of cryptographic attacks up to the attacker and the used cryptosystem; let us adduce some examples of existing kinds of cryptographic attacks:

- *Known Plaintext attacks*: with this type of attacks, the attacker possesses a version of plaintext and its corresponding cipher text. He seeks to discover a correlation between the two.
- *Ciphertext-Only Attacks*: the attacker has access to a ciphertext but he doesn't know its corresponding version of plaintext.
- *Chosen Plaintext attacks:* the attacker can encrypt a plaintext of his choice and studying the result ciphertext.
- *A chosen ciphertext attacks:* the attacker tries to decrypt a ciphertext of his choice and look for its corresponding plaintext.
- *Brute Force Attacks:* During this type of attack, the attacker is trying to bypass security mechanisms while having minimal knowledge about them. It systematically attempts every possible key. With finite key length and high speed computation, this kind of attacks is always successful.

As mentioned in the introduction, many chaotic synchronization based crypto systems are weak against known plaintext, chosen plaintext and chosen plaintext attacks. With these types of attacks, the attacker can get the shape of the chaotic signal used for encryption and thus can found the system parameters.

III. THE PROPOSED CHAOTIC GENERATOR (PCG)

A. The Used Chaotic Systems

Our proposed cryptosystem use two discrete chaotic systems instead one system; the Hénon map and the Logistic map. The Hénon map is a 2-D iterated map with chaotic solutions proposed by the French astronomer Michel Hénon (1976) as a simplified model of the Poincare map for the Lorenz model [16]. The Hénon map is given by:

$$\begin{cases} x_{k+1} = 1 + y_k - a x_k^2 \\ \qquad g \\ y_{k+1} = b x_k \end{cases} \tag{1}$$

The map has been studied in detail for a = 1.4 and b = 0.3, where numerical evidence of chaotic behavior was found. The Fig.4 and 5 shows the attractor of the Hénon map in x-y space and temporal behavior in x-direction (fig.5-a) and y-direction (fig.5-b) for a = 1.4, b = 0.3 and x_0 =0.2 (a and b are the control parameters).

The logistic map is a very simple mathematical model often used to describe the growth of biological populations. In 1976 May showed that this simple model shows bewildering complex behavior. Because of its mathematical simplicity, this model continues to be useful test bed for new ideas in chaos theory as well as application of chaos in cryptography [17]. The simple modified mathematical form of the logistic map is given as:

$$y_{k+1} = r.x_k(1 - x_k) \tag{2}$$

r is the parameter control of the Logistic map, for r = 3.9 and x_0 =0.2, the system (2) has a chaotic behavior as shown is the Fig.5-c (temporal behavior of y_k).

B. Coupling of the Two Maps

This section describes step by step the principle functioning of our proposed cryptosystem using the Logistic and Hénon maps (Fig. 3). The design was simulated using both Xilinx suite design environment (Xilinx system generator (XSG), Xilinx ISE) and Matlab/Simulink. Each of the two maps design is presented in Fig. 1 and Fig. 2, all Xilinx system generator blocks are configured with fixed point precision and word length of 32 bits (12 bits for integer part and 20 bits for fraction part). The output of the Logistic map generator (control system - CRS) is then decomposed in 32 bit; the MSB is XORed (using XOR logic function) with its neighbor bit, the results then XORed with the next bit until the LSB. The resulting logical value 0 or 1 (depending on the output of the CRS) then control a multiplexer that outputs the Hénon map generator signals (ENS) x(k) or x(k-1), in other words;

the output of the whole system is x(k) or x(k-1) of the ENS, depending on the logical value generated by the CRS. We kept the same parameters cited in III.A; the different temporal outputs of the system are presented in Fig. 5 (a,b,c and d).

IV. EVALUATION OF PROPOSED CHAOTIC GENERATOR

As shown in the Fig. 5-d, the multiplexer output sequence that seems random, the usefulness of this sequence for cryptographic purposes can be confirmed after evaluating it; the following section will be devoted to the evaluation process.

A. Autocorrelation

The autocorrelation is defined as the mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals. It is the same as calculating the correlation between two different time series (cross-correlation), except that the same time series is used twice - once in its original form and once lagged one or more time periods [18]. The Fig. 6 presents the auto- correlation of the chaotic sequence generated from the proposed generator, it is clear that the sequence has low autocorrelation which implies the no similitude between its samples; this result characterizes the randomness of our sequence.

B. Sensitivity to the Initial Parameters (SIP)

The Fig. 7 presents the cross-correlation between two chaotic sequences generated with two vary close initial condition of the CRS (for x_0=0.2 and x'_0=0.2000001). It is clear that the two signals are completely different although the infinitesimal change on the initial conditions of CRS. The same thing happened on (Fig. 8), when the control parameter (a) of ENS changed from 1.4 to 1.4000001; we get two different chaotic sequences. So, our proposed generator presents a high SIP whatever the change, on the CRS or ENS.

C. Evaluation using the NIST Statistical Tests

In fact, the correlation evaluation of the chaotic sequence generated from the PCG give us a good idea on the degree of randomness of our sequence, but when the desired application area is "the cryptography", this is not sufficient. The pseudo random sequences needed for cryptography must be strong and have special characteristics; these characteristics can be evaluated using the NIST statistical tests (for more information about the NIST statistical tests, please see [19]), in this case the pseudo random number sequence is called cryptographically secure pseudo random sequence (CSPRNS). The NIST tests are 15 tests applicable to the binary sequence of the PCG output signal. The decision about the usefulness of the PCG output sequence for the cryptographic purposes can be taken if all the 15 NIST tests are passed.

The first test is the frequency test; the focus of the test is the proportion of zeroes and ones for the entire sequence. The purpose of this test is to determine whether the number of

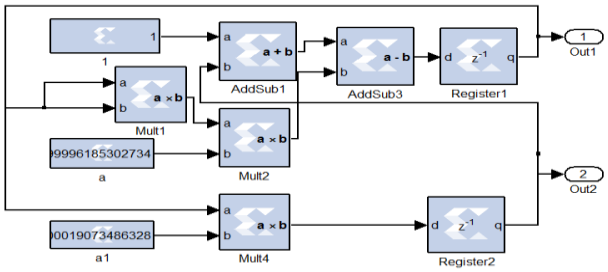

Figure 1: The Hénon chaotic generator (ENS) using XSG

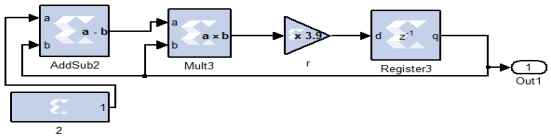

Figure 2: The Logistic chaotic generator (CRS) using XSG

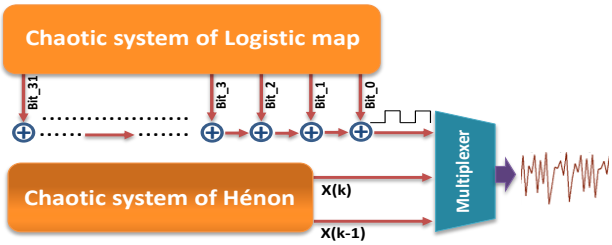

Figure 3: The proposed generator scheme

ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. The test assesses the closeness of the fraction of ones to ½, that is, the number of ones and zeroes in a sequence should be about the same. All subsequent tests depend on the passing of this test [19]. The NIST tests can be evaluated by computing α (called the significance level), α must superior or equal to 0.01.

The Table 1 presents the NIST statistical test results of the PCG output sequence. It is clear from the Table 1 that the sequence cannot be qualified as CSPRNS although its good characteristics shown previously.

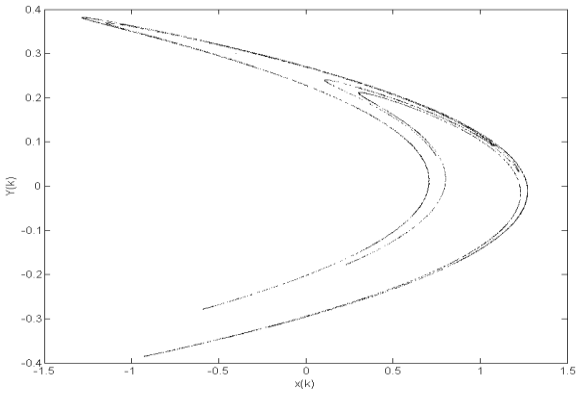

Figure 4: The Hénon map in x-y space

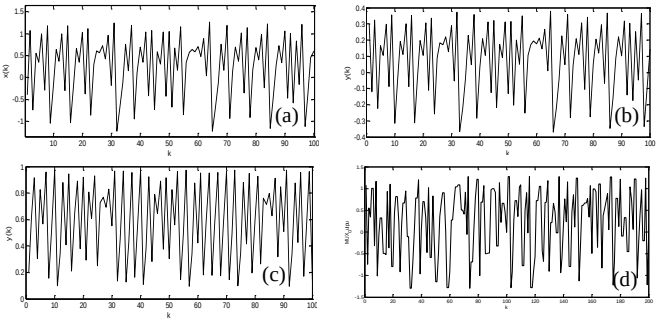

Figure 5: The different PCG outputs

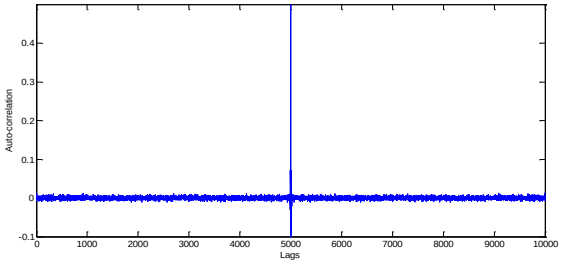

Figure 6: Autocorrelation of the PCG output signal

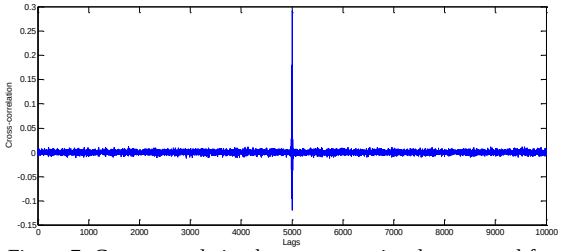

Figure 7: Cross- correlation between two signals generated from the PCG with two very close initial values (variation of x_0 of CRS)

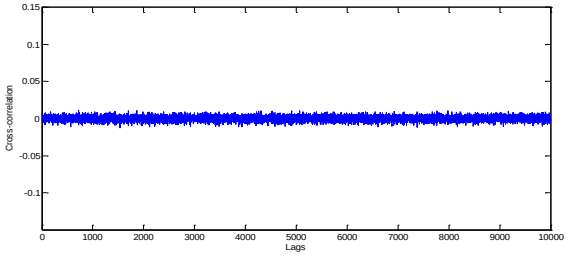

Figure 8: Cross- correlation between two signals generated from the PCG with two very close initial control parameters (variation of a of the ENS)

In order to make our chaotic sequence ideally-distributed and able to pass the NIST statistical tests while keeping their previous characteristics; we pass the chaotic sequence through another circuit; this circuit called "the logical circuit" (LX) and works as follows:

1- We decompose the PCG output sequence to 32 bits.
2- The LSB then Xored with the CRS output, the circuit outputs the result while keeping it to the next step.
3- The previous result is Xored then with the next bit, but this time is inverted, the circuit outputs the result while keeping it to the next operation.
4- We repeat the steps 1 and 2 until the MSB of chaotic sequence.

The output of 32 bits then again undergoes to the same steps 1, 2, 3 and 4, the Fig. 10 illustrates clearly the principal functioning of the logical circuit. In this case, the output sequence of the whole system is ideally distributed and can pass all the statistical tests as shown on the Table 2.

D. Resistance Against Attacks

As mentioned in the introduction, the classical method to use chaotic in stream cipher cryptography is to combine the plaintext with the chaotic signal, the known plaintext and chosen ciphertext attackers can get the shape of the chaotic signal used for encryption and thus can found the system parameters. In our proposed generator these kinds of attacks cannot break the system because even if the attacker knows the shape of the sequence, he can't compute the parameters. That due to the complexity of composition of the sequence, it was generated from the logical circuit which is controlled by the output of the CRS and the chaotic sequence that generated from the multiplexer which is also composed of x(k) or x(k-1) of the ENS; ordered with a specific way depending of the output of the CRS, so it is clear that the reverse operation is impossible.

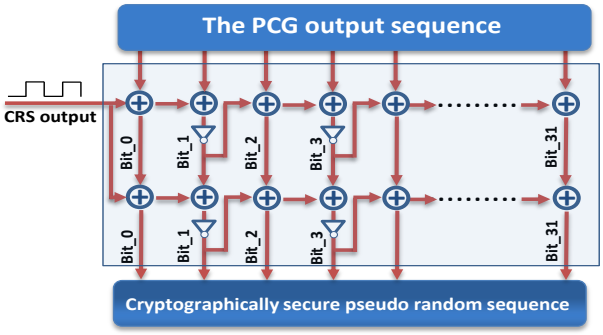

Figure 9: Schema of the logical circuit (LC)

TABLE 1: NIST STATISTICAL TEST RESULTS OF PCG OUTPUT

Statistical Test	Status	P_value
Frequency	FAIL	0.00000
Block Frequency (m = 256)	FAIL	0.00000
Cusum-Forward	FAIL	0.00000
Cusum-Reverse	FAIL	0.00000
Runs	FAIL	0.00000
Long Runs of Ones	PASS	0.49799
Rank	FAIL	0.00000
Spectral DFT	FAIL	0.00000
NonOverlapping Templates (m = 9, B=000000001)	PASS	0.29698
Overlapping Templates (m = 9)	FAIL	0.00000
Universal	FAIL	0.00884
Approximate Entropy (m = 10)	FAIL	0.00000
Random Excursions (x = +1)	FAIL	0.00000
Random Excursions Variant (x = -1)	FAIL	0.00000
Linear Complexity (M = 500)	PASS	0.98735
Serial (m = 16, $\nabla\Psi_m^2$)	FAIL	0.00000

V. THE REAL TIME FPGA IMPLEMENTATION OF THE PCG

FPGA technology is a growing area of research that has the potential to provide the performance benefits of ASICs and the flexibility of processors; it is a reconfigurable hardware device that seems to combine the advantages of software and hardware implementations. Furthermore, there are potential

TABLE 2: NIST STATISTICAL TEST RESULTS OF LC OUTPUT

Statistical Test	Status	P_value
Frequency	PASS	0.10183
Block Frequency (m = 256)	PASS	0.01941
Cusum-Forward	PASS	0.12492
Cusum-Reverse	PASS	0.06108
Runs	PASS	0.51225
Long Runs of Ones	PASS	0.76240
Rank	PASS	0.76478
Spectral DFT	PASS	0.94147
NonOverlapping Templates (m = 9, B=000000001)	PASS	0.70248
Overlapping Templates (m = 9)	PASS	0.07434
Universal	PASS	0.60040
Approximate Entropy (m = 1)	PASS	0.21134
Random Excursions (x = +1)	PASS	0.78037
Random Excursions Variant (x = -1)	PASS	0.38949
Linear Complexity (M = 500)	PASS	0.36831
Serial (m = 16, $\nabla\Psi_m^2$)	PASS	0.90777

advantages of reconfigurable hardware in cryptographic applications: algorithm agility, algorithm upload, architecture efficiency, resource efficiency, algorithm modification, throughput, cost efficiency [20], [21].

In this paper, one of recent families of FPGAs technology was used; it is the SPARTAN 6 XC6SLX45 chip from XILINX, embedded on ATLYS complete circuit board from DIGILENT INC. The hardware implementation VHDL code was generated automatically from the XSG blocks, then after certain steps using the ISE tool a bitstream file also generated and targeted to the FPGA chip. The Fig.10 and Fig.11 present the timing and design summary of the PCG FPGA implementation.

In order to visualize real time outputs of the PCG FPGA circuit, we use a DAC (digital to analog converter), in this case we used the Pmod-DA1 chip that designed by DIGILENT INC and has 4 analog outputs. The Pmod-DA1 chip contains two AD7303 8-bit D/A converter chips from ANALOG DEVICES INC that convert up to one MSa/s.

As we can see from Fig. 10, our design has low cost implementation and the consumed FPGA resources are low. The maximum estimated frequency is f =25.456 MHz, since the PCG gives 32 bits of data key per clock cycle, it is easy to compute the throughput:

Throughput = output word length × f
= 32 bits × 25.456 MHz = 814 Mbps

This is a high throughput and may be covers the most actual cryptography applications requirements.

The Fig. 14, Fig. 15, Fig. 16 and Fig. 17 present the different outputs of the PCG viewed on the oscilloscope. It seems clearly that all the results are identical to those obtained with simulation.

```
Timing summary:
---------------

Timing errors: 0  Score: 0  (Setup/Max: 0, Hold: 0)

Constraints cover 3207552630384812000 paths, 0 nets, and 2849 connections

Design statistics:
   Minimum period:  39.283ns{1}   (Maximum frequency:  25.456MHz)
```

Figure 10: The timing summary of the PCG FPGA implementation

```
Device utilization summary:
---------------------------

Selected Device : 6slx45csg324-3


Slice Logic Utilization:
 Number of Slice Registers:              97  out of  54576     0%
 Number of Slice LUTs:                  214  out of  27288     0%
    Number used as Logic:               214  out of  27288     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    238
   Number with an unused Flip Flop:     141  out of    238    59%
   Number with an unused LUT:            24  out of    238    10%
   Number of fully used LUT-FF pairs:    73  out of    238    30%
   Number of unique control sets:         1

IO Utilization:
 Number of IOs:                        34
 Number of bonded IOBs:                33  out of    218    15%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:              1  out of     16     6%
```

Figure 11: The design summary of the PCG FPGA implementation

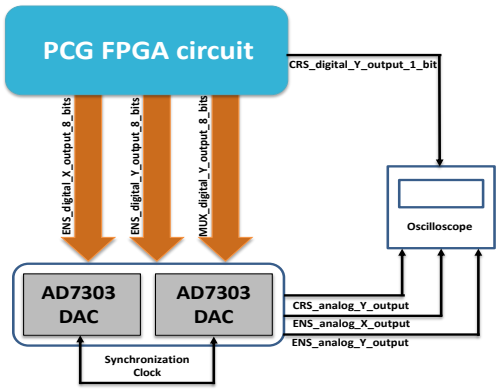

Figure 12: FPGA output digital to analog conversion scheme

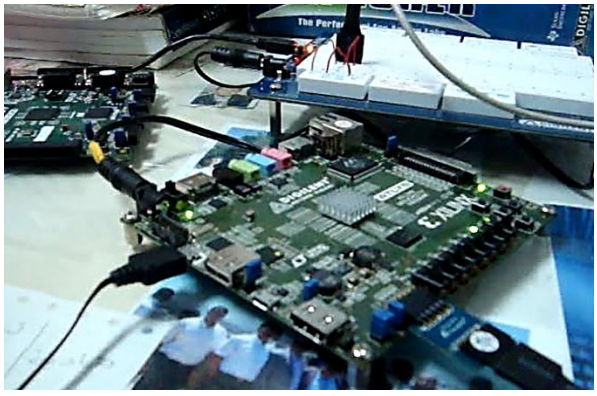

Figure 13: Atlys board with Electronic Explorer Board used for implementation and real time visualization

.

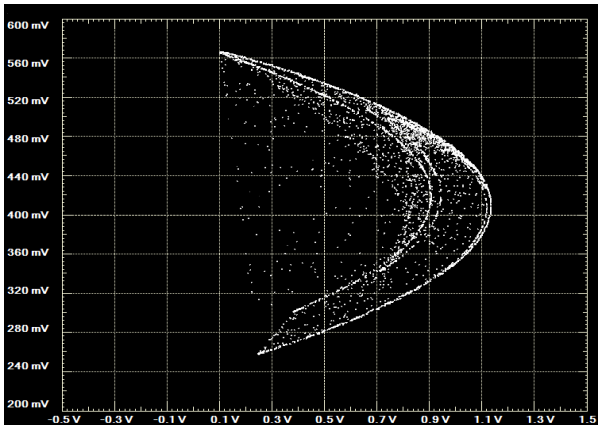

Figure 14: Real time Hénon map in x-y plane

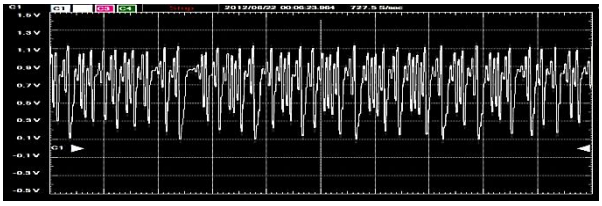

Figure 15: Real time ENS X output signal

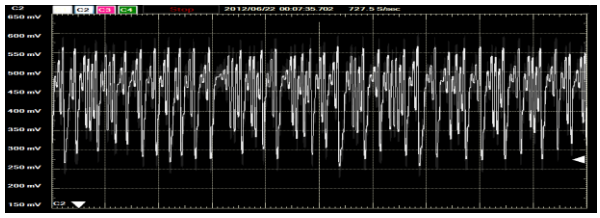

Figure 16: Real time ENS Y output signal

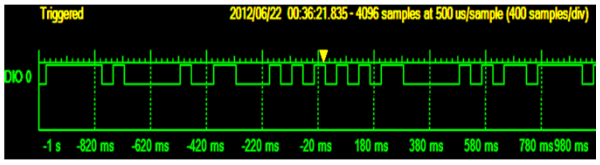

Figure 17 Real time CRS output signal

VI. CONCLUTION

We have seen in this paper a new approach to using the chaotic signals for encryption, the proposed method was based on coupling two simple chaotic signals in order to remedy the weakness of the classical use of the chaotic signals (XORing the chaotic sequence directly with the plaintext). Known plaintext and chosen cipher text attacks are not effective solutions with the new proposed chaotic generator due to the complexity of composition of the encryption sequence. And even the brute force attack is no longer an effective solution because the key (initial parameters) size is relatively large, it can be computed as follow; each parameter has 32 bits, the CRS has two parameters (r and x_0), the ENS has four parameters (x_0, y_0, a and b) so the key has a 192 bits, the number of the possibilities that the key may be take is $2^{192} = 6{,}2771 \times 10^{57}$ which is a big number. The FPGA implementation results show that the proposed scheme has a low cost and offers a high throughput that is sufficient for the most actual cryptographic applications requirements.

REFERENCES

[1] Milos DR UTAROVSKY, Pavol GALAJDAI, "A Robust Chaos- Based True Random Number Generator Embedded in Reconfigurable Switched-Capacitor Hardware," IEEE Conference Publications, Radioelektronika,17th International Conference, 24-25 April 2007, Brno.

[2] Amit Pande, Joseph Zambreno, "Design and Hardware Implementation of a Chaotic Encryption Scheme for Real-time Embedded," IEEE Conference Publications, International Conference on Signal Processing and Communications (SPCOM), 18-21 July 2010,pp. 1- 5, Bangalore.

[3] Xunzhang, Liya, Ying, Lingling Sun, Zhineng Li, "Research on New Implementation Method of Chaotic Model Based on FPGA," IEEE Conference Publications, 7th International Conference on ASIC'07, 22-25 Oct. 2007, pp. 241 - 244, Guilin.

[4] Musheer Ahmad, Omar Farooq, "Chaos Based PN Sequence Generator for Cryptographic Applications," IEEE Conference Publications, International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), 17-19 Dec 2011, pp. 83 - 86, Aligarh.

[5] Meei-Ling Hung, Cheng-Fang Huang, Jui-Sheng Lin, "Design of Random Digital Sequence Generators and Its Application of Secure Communication," IEEE Conference Publications, International Conference on Fluid Power and Mechatronics (FPM), 17-20 Aug 2011, pp. 889 – 892, Beijing.

[6] Ali Oksasoglu, Tayfan Akgul, "Chaotic Masking Scheme with Linear Inverse System," Physical Review Letters, Vol.75, pp.4596-4597, 18 Dec.1995.

[7] İ. Pehlivan, Y. Uyaroğlu, O. Önal, "Signal Masking Applications Using Chaotic Circuits," 6th International Advanced Technologies Symposium (IATS'11), pp. 360-361, 16-18 May 2011, Elazığ, Turkey.

[8] Ljupco Kocarev, Zbigniew Galias, and Shiguo Lian, "Intelligent Computing Based on Chaos, Studies in Computational Intelligence," Springer, Vol.184, 2009.

[9] Available online at : http://en.wikipedia.org/wiki/Randomness

[10] Chung-Kang Peng and all, "Randomness versus deterministic chaos : Effect on invasion percolation clusters" Physical review, pp.4537, Vol.42, N° 8, 15 Oct 1990.

[11] Available online at : http://www.quora.com/Chaos-Theory/What-is-the-difference-between-chaotic-behavior-and-random-behavior

[12] XINGYUAN WANG, "Two New Chaotic Cryptographies Based On Different Attractor-Portion Algorithms," International Journal of Modern Physics B, World Scientific, Vol. 21, No. 27, 4739-4750, pp. 2007.

[13] Shun-Cheng Hong, Chun-Te Li, Hsi-Kuan Chen and Chin-Hsing Chen, " The Cycle Length, Statistical and Key Sensitivity Properties of a Perturbed Couple Chaotic PRBG," IEEE Conference Publications, International Conference on Multimedia Technology (ICMT), 26-28 July 2011, Hangzhou, pp. 3268 - 3272, ISBN: 978-1-61284-771-9.

[14] Gao Kun , "A Synchronization Controller for the Unified Chaotic System and Its Application in Secure Communication," IEEE Conference Publications, International Conference on Consumer Electronics, Communications and Networks (CECNet), 16-18 April 2011, XianNing, pp. 4700 - 4703, ISBN: 978-1-61284-458-9.

[15] Eric Conrad, "Types of Cryptographic Attacks," available online at : http://www.giac.org/cissp-papers/57.pdf.
[16] Available online at : http://geo.uni-bonn.de/members/hattendorf/html/misc/nonlindyn/henon_map/henon_map.html
[17] Vinod Patidar and K. K. Sud, "A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing," Informatica 33, pp.441-452, 2009.
[18] Available online at : http://www.investopedia.com/terms/a/autocorrelation.asp#axzz1yPu70nME
[19] National Institute of Standards and Technology, U.S Departement of Comerce, " A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," NIST Special Publication 800-22, Revision 1a, April 2010.
[20] Viktor K. Prasanna, Andreas Dandalis, "FPGA-based Cryptography for Internet Security," Online Symposium for Electronic Engineers, November 2000.
[21] Thomas Wollinger, Christof Paar, " Security Aspects of FPGAs in Cryptographic Applications," chapter 1 in "New Algorithms, Architectures, and Applications for Reconfigurable Computing", Wolfgang Rosenstiel and Patrick Lysaght (eds.), Kluwer, 2004.