Real-Time FPGA Implementation of a Switching Chaotic Generators for the Secure Embedded Systems

M.S. Azzaz*, C. Tanougast†, S. Sadoudi*, A. Dandache† and A. Bouridane‡,
*EMP d'Alger, Algerie
†Universit Paul Verlaine de Metz, LICM, France
‡Northumbria University, Newcastle, UK
ms.azzaz@gmail.com, Camel.tanougast@univ-metz.fr

Abstract—**This article proposes a new chaotic key generator for data encryption and its optimized FPGA implementation based on a chaos switching rule between *Lorenz*'s and *Lü*'s no-linear systems. The originality of our approach is that allows a better security communication for embedded application while provides a good tradeoff between performances and hardware resources. Our experimental results have demonstrated the feasibility and the efficiency of our secure solution on Xilinx FPGA Virtex technology.**

I. Introduction

Since recent years, new cryptosystems for secure communications have been developed. These cryptosystems are based on chaos theory since it was shown that these systems can be controlled [1], [2]. Then the synchronization between two identical chaotic systems corresponding one data encryption emission module and one decryption reception module was reported [3]. Therefore, we realized that the key generator based chaos could be useful in secure communication systems because chaos is extremely sensitive to the initial conditions and parameters [2], [4]. In addition, chaotic signals are aperiodic, uncorrelated, broadband, and deterministic and appear random in the time domain [4].

Many methods based on analogue circuits are used to implement chaotic generators such as switched capacitor or analogue CMOS technology [7], [8]. However, these methods exhibit some practical difficulties since the component values vary with age, temperature, etc. [12], [19]. Therefore, it is very difficult to deal with the problem of chaotic synchronization. To overcome this problem, a digital implementation of chaotic generators can be used. In this context, advances in VLSI technology have been employed to the manufacturing of reconfigurable logic including FPGA chips and helped their rapid growth in logic capacity, performance. Recently, several behaviors structure of chaotic systems has been implemented in FPGA technology. Among them, we find *Lorenz*, *Chen* and *Lü* systems [12], [19] that can be used for designing of chaotic hardware key generation for secure communication systems. However, these main chaotic generators are easily identifiable by a simple display of their attractors. Indeed, they are characterized by specific attractors can be used to cryptanalysis. In this context, it becomes important to hide or to develop mechanisms associated with these generators to increase the complexity of cryptanalysis from the visualization and identification of the chaotic signals used for key generator in a cryptosystems.

In this paper, we propose novel structural hardware architecture for a new chaotic generator and its real time implementation using a Virtex Xilinx FPGA [9]. The originality of our approach is based on a chaotic switching between the two *Lorenz*'s and *Lü*'s systems. The structural architecture of our proposed chaotic generator is based on the commutation of the Runge-Kutta method (RK-4) to resolve both *Lorenz*'s and *Lü*'s nonlinear differential equation systems [10]. The interest of our solution is to propose a chaotic system allowing an unidentifiable key generator by a simple analysis of its attractors while proposing optimized architecture and hardware implementation given a very useful and attractive trade off between high speed, low area cost and data security transmission for an embedded system. Then, this proposed architecture can be used as hard key chaotic generator in a self-synchronizing stream cipher encryption [11]. Moreover, it can be also used for the implementation of new others chaotic systems following switching rules adapted from other existing chaotic generators such as Chua's system, Colpitts' system, Rössler's system and Chen's system [12], [13], [14], [15].

This paper is organized as follows. Section II gives a short description and characterization of the *Lorenz*'s and Lü's chaotic models. Section III describes the proposed architecture of our new chaotic system including the simulation results. The hardware implementation results on Virtex-II Xilinx FPGA technology and the performance evaluation are presented in Section IV. In this section a real time measurements demonstrates the feasibility and the efficiency of our secure encryption solution are also given. Finally, a conclusion and future work are given in Section V.

II. Lorenz's And Lü's Chaotic Systems

In the proposed scheme, *Lorenz*'s and *Lü*'s chaotic systems are employed in key scheming, which are modeled by *Lorenz*'s system [1], [16], which is a famous example of chaotic system. It is represented by nonlinear equation system 1 and Lü's system [5], [6], [15], [18] which is known as a bridge between the *Lorenz*'s and

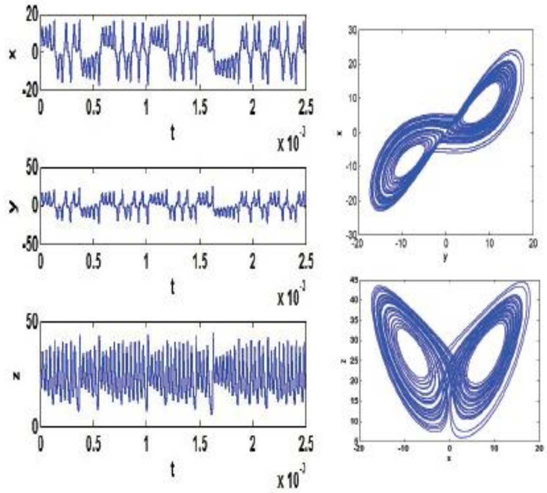

Fig. 1. Matlab Simulation Results of *Lorenz*'s Chaotic Signals and Attractors in Phase Plane: (*x-y*) and (*y-z*).

Chen's systems and is represented by nonlinear equation system 2.

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = -xz + rx - y \tag{1}$$
$$\frac{dz}{dt} = xy - bz$$

$$\frac{dx}{dt} = a(y - x)$$
$$\frac{dy}{dt} = -xz + cy \tag{2}$$
$$\frac{dz}{dt} = xy - bz$$

Solutions of these nonlinear equation systems depend mainly on the initial conditions specified by the initial values of $x = x_0$, $y = y_0$ and $z = z_0$. One numerical solution of these systems can be found with a fourth order *Runge-Kutta* method (RK-4) with the following value coefficients h = 0.01 and *coef* = 1/6 [10]. Indeed, using Matlab simulation tool [17] with Lü's parameters values a = 36, b = 3 and c = 20, *Lorenz*'s parameters values σ = 10, r = 28 and b = 8/3 with initial conditions x_0 = 0, y_0 = 5 and z_0 = 25, gives the corresponding chaotic signals of these two chaotic systems. Figures 1 and 2 relate simulation results of the chaotic signals *x*, *y*, *z* and the two different attractors corresponding to the phase planes (*x-y*) and (*y-z*) of the *Lorenz*'s and *Lü*'s systems, respectively.

III. Proposed Architecture And Modelization

A. RTL Architecture

We propose a new chaotic generator based on a chaotic switching between *Lorenz*'s and *Lü*'s systems. Our proposed architecture consists of the implementation of the RK-4 method to resolve *Lorenz*'s and *Lü*'s nonlinear differential equations system as defined by equations 1 and 2 [1], [15], [16], [18]. An overview of the proposed Register Transfer Level (RTL) architecture for our proposed chaotic generator is given in Figure 3.

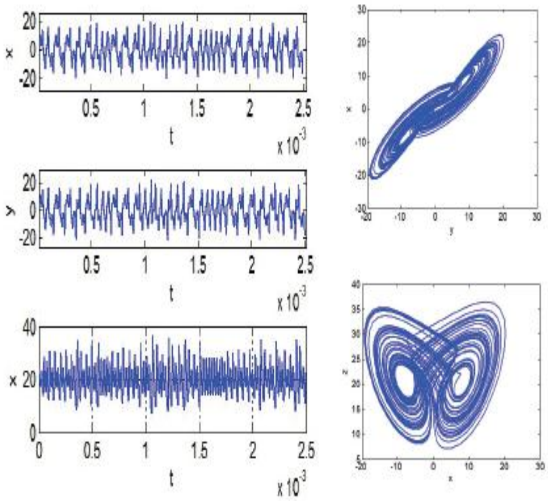

Fig. 2. Matlab Simulation Results of Lü's Chaotic Signals and Attractors in Phase Plane: (*x-y*) and (*y-z*).

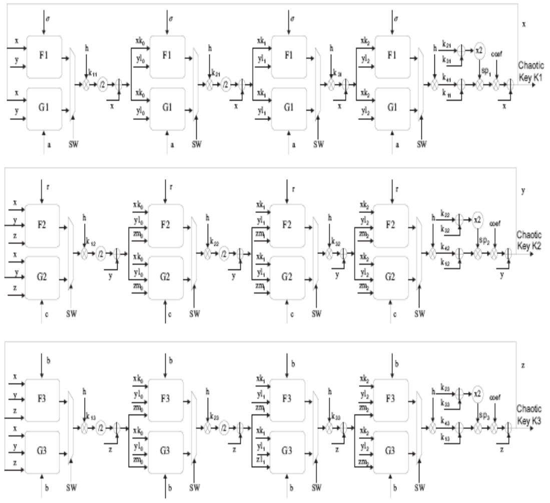

Fig. 3. RTL Architecture of the Proposed Switching Chaotic Generator.

This random key architecture is based both on fixed *Lorenz*'s and *Lü*'s parameters (see Section 2) and consists of the structural feedback of the following main blocks:

- Three functional units' F1, F2 and F3 realize the three equations of the equation system (1).
- Three functional units' G1, G2 and G3 realize the three equations of the equation system (2).

These Units are composed simply by an adder, a subtractor and multiplier logic arithmetic operators in accordance with the set of equations 1 and 2. Our data-path processing architecture for our Switching random keys generator is also depicted in Figure 3.

- The signal SW is the chaotic switching select of the multiplexer where: *SW = x(.)*, and *(.)* is one bit among 32 bits of the generated chaotic key *x*.

B. *Functional Modelization And Simulation*

To test the effectiveness of our solution, we have simulated our RTL architecture of the Switching chaotic generator with *ModelSim* simulator tool [19]. The results obtained are presented in Figure 4. This validation consists to model and describe directly the RK-4 method with (VHDL). In this modelization, we have adopted the implementation based on a finite solution numbers with a fixed point representation of the real data on 32 bits (16Q16). It can be seen that that the functional hardware simulation results are mixed those of the *Lorenz* and *Lü* RK-4 numerical resolution by using *Matlab* simulation tool (see Section II).

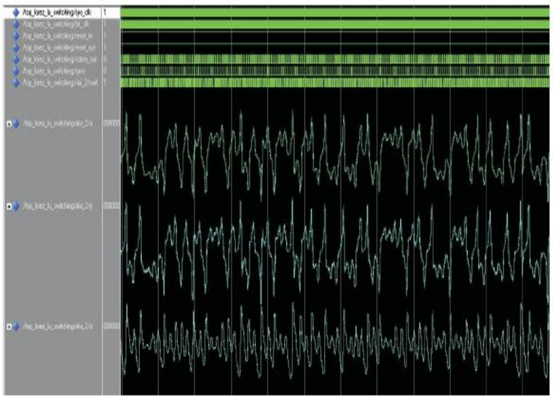

Fig. 4. *ModelSim* Simulation Results of the proposed Switching Chaotic Generator.

IV. REAL TIME HARDWARE IMPLEMENTATION AND MEASUREMENTS

A. *FPGA Implementation And Results*

An overview of the architecture for a hardware implementation of the proposed Switching chaotic system using Virtex-II Xilinx FPGA technology [9] is depicted in Figure 5. The architecture system consists of three main modules: Control Unit, *Lü* System and *Lorenz* System sub-modules. The Control Unit sub-modules is a Moore state machine which manages and schedules the different operations and functions of our proposed chaotic system. *Lorenz* System and *Lü* System sub-modules generate the random keys using the RK-4 method as described in Section 3 which implements *Lorenz*'s and *Lü*'s nonlinear equation systems defined par the set of equation 1 and 2. A multiplexer controlled by the *x* chaotic signal of the *Lorenz* System sub-module provides the *x*, *y* and *z* chaotic signals of our switching generator from the *Lorenz* System and *Lü* System sub-modules. Once the chaotic signals (*x*, *y* and *z*) with 32 bit word length are obtained, they are converted to analogue format as a sequence of 18 bits using a 18 bits Digital to Analog converter (Codec AC'97) and this process is repeated so that real-time chaotic signals are obtained at the output of the DAC for visualization on an oscilloscope.

Our RTL description of the proposed architecture has been implemented on Xilinx Virtex-II FPGA (XC2V1000) [9] using a VHDL structural description. *ISE 7.2i* of Xilinx tools have been used for this implementation thus allowing evaluating the hardware resource requirements and the associated real time constraints. In order to minimize the area logic resources of the FPGA, which are due mainly to the RK-4 solution method to solve the *Lorenz*'s and *Lü*'s systems, optimization has been carried out. More precisely, we have replaced or approximated most of the multiplication and/or division operations (which are the mist complex arithmetic operations) by the simple left or right shift operations. This has led us to minimize the number of the embedded multipliers/dividers in the implementation.

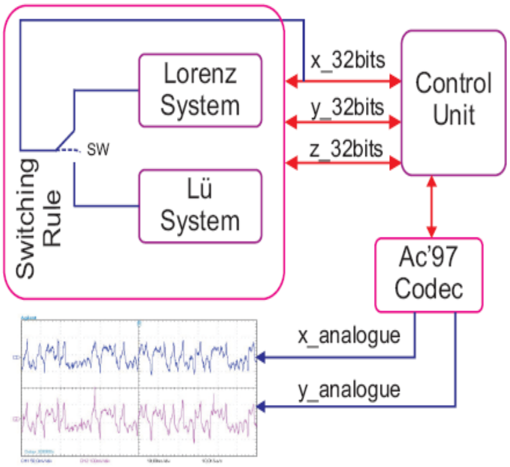

Fig. 5. Digital Hardware Architecture of our Switching Chaotic System.

The results thus demonstrate that our real time switching chaotic generator can be efficiency implemented with FPGA technology. It can be stated that an attractive tradeoff between high speed and low logic resources has been achieved. Indeed, our implementation on a Xilinx Virtex- II device uses only 1316 CLB-Slices, 41 multipliers and no block RAMs.

TABLE I
IMPLEMENTATION RESULTS OF THE SWITCHING CHAOTIC GENERATOR WITH VIRTEX-II-PRO FPGA (2VP30FF896-7)

Device utilization summary FPGA: 2vp30ff896-7	
Number of Slices	1316 out of 13696
Number of MULT 18X18s	41 out of 136
Maximum Frequency	45.744 MHz

In order to evaluate the behavior of the proposed system, it is necessary to use some evaluation metrics. The metrics used for the evaluation results for this system are the throughput rate and the time latency. The throughput rate is defined as the number of bits key in a unit of time for a stream encrypted (or decrypted). In our case and from the performance results (Table I) we have achieved a maximal throughput of 0.82 Gbps. This throughput rate is computed after the initialization phase. Latency is defined as the time necessary to generate a single random key after the start of the generator. In our case latency corresponds to the number of the pipeline stages. Our optimized implementation of the switching chaotic system requires 6 clock cycles to generate one random key. Thus, from the operating clock frequency, we have obtained a time latency of 131 ns.

B. *Real Time Measurement Results*

The x, y and z real-time chaotic signal measurements of our proposed switching generator, obtained by a direct implementation after optimization, are given in Figures (6.a), (6.b) and (6.c) , respectively. One can compare these real results and that obtained using *ModelSim* (figure 4) simulation tool to ascertain whether these results are similar. The measured real-time attractors ($x-y$) and ($x-z$) are presented in figures (6.d) and (6.e), respectively. These results clearly confirm that the implemented chaotic system work well in the chaotic mode. In addition, these measurements show that the proposed approach provides a new chaotic generator. These results arise from the sample switching between *Lorenz*'s and *Lü*'s systems. Thus, we clearly validate our hardware implementation method and our approach for to develop new chaotic generators based chaotic switching rules between several nonlinear systems.

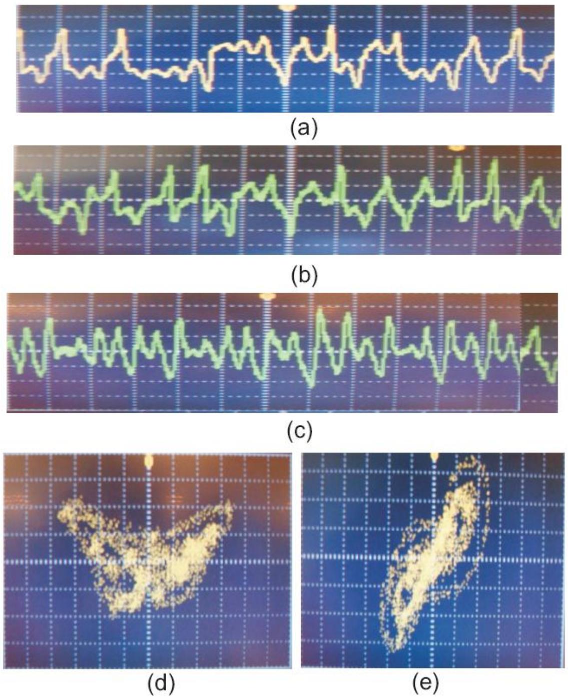

Fig. 6. Real Time Results of Switching Chaotic Generator: (a, b and
c) x, y and z Chaotic Signals, (d) (x-z) Attractor, (e) (x-y) Attractor.

I. Conclusion And Future Work

This paper proposes structural hardware architecture of a new random key generator based on a chaos switching rule between *Lorenz*'s and *Lü*'s chaotic systems for stream encryption systems. This proposed chaotic generator architecture based on the Runge-Kutta method (RK-4) is particularly attractive since it provides low-cost security communication solutions for embedded systems. Indeed, an implementation on a Xilinx Virtex-II technology requires only 1316 CLB-Slices, 41 multipliers and no block RAMs. Our FPGA implementation achieves a throughput rate of 0.82 Gbps at a clock frequency of 45.744 MHz with a low latency time of 131 ns. Ongoing work seeks to further improve the performance of the architecture targeting a full implementation in a pipelined fashion.

However, in this case we will increase the latency time. In addition, our experimental results show the robustness against a cryptanalysis. The real-time chaotic signals generated by our switched system are unidentifiable as switching between *Lorenz*'s and *Lü*'s chaotic samples. We clearly demonstrate the feasibility for the generation of new chaotic systems can be used in an encryption system with low cost while increase the security of communications for embedded applications. Finally, our new approach is very simple, exhibits attractive performances and can be used for the implementation of others chaotic systems following switching rules adapted from other existing chaotic generators such as Chua's system, *Colpitts*' system, *Rössler*'s system and *Chen*'s system.

References

[1] T. E. N. Lorenz, "Deterministic nonperiodic flow", *Journal of the Atmospheric Sciences*, Vol.20, N° 2, pp.130-141, March 1963.

[2] T. Yang, "A survey of chaotic secure communication systems", *International Journal of Computational Cognition*, Vol.2, N°.2, pp. 81-130, June 2004.

[3] P. L. Carroll and L. M. Pecora, "Synchronization in chaotic systems'', *Phy. Rev. Lett.*, Vol.64, N° 8pp.821-824, Feb. 1990.

[4] A. abel and W. Schwartz, "Chaos Communications-Principles, Schemes and Systems analysis'', *IEEE on Inst. for Fun. of Electr. Eng. & EleC.*, Vol. 90, N° 5. pp. 691 – 710, May 2002.

[5] J. Lü and G. Chen, "A new chaotic attractor coined*", Int. Journal of Bifurcation and Choas*, Vol. 12, N°.3, pp. 659-661, 2002.

[6] J. Lü, G. Chen and S. Zhang, "The compound structure of a new chaotic attractor", Chaos, Solitons and Fractals, Vol 14, pp. 669-672, 2002.

[7] T. Matsumoto, "Chaos in electronic circuits'', *IEEE Inst. of Elec. and Elecs Eng.*, vol.75, N°.8, pp. 1033-1046, Aug.1987.

[8] C.Y. Cha and S.G. Lee, "Complementary Colpitts Oscillator in CMOS Technology'', *IEEE Transaction on microwave theory and techniques*, Vol. 53, N° 3, March 2005.

[9] Xilinx, "VirtexII-pro complete Datasheet", *Xilinx*, 2007.

[10] H. William Press, Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling, "Numerical Recipes in C, The Art of Scientific Computing'', *Cambridge University Press*, 1992.

[11] C. Tanougast, S. Weber, G. Millerioux, A. Bouridane and J. Daafouz "VLSI architecture and FPGA implementation of a hybrid message embedded self-synchronizing stream cipher" *4th IEEE International Symposium on Electronic Design, Test and Applications*, pp. 386-389, 2008.

[12] M. I. Sobhy, M. A. Aseeri and A. E. R. Shehata, "Real Time Implementation Of Continuous (Chua And Lorenz) Chaotic Generator Models Using Digital Hardware', *Proc. of the Third International Symposium on Communication Systems Networks and Digital Processing*, pp.38-41 , 1999.

[13] P. Kvarda, ''Investigating the Rössler attractor using Lorenz plot and Lyapunov exponents'', *Radio engineering*. Vol.11. N°.3, pp. 22-23, September 2002.

[14] M. P. Kennedy, “Chaos in the Colpitts oscillator,” *IEEE Trans. Circuits Syst. I,* Vol. 41, pp. 771-774, Nov. 1994.

[15] H.H Chen, J.S. Chiang, Y. L. Lin, C.I. Lee, ‘‘Chaos synchronization of general Lorenz, Lü, and Chen systems’’, *HSIUPING Journal*. Vol.15, pp.159-166, September 2007.

[16] K. M. Cuomo, A. V. Oppenheim, and Steven H. Strogatz, “Synchronization of Lorenz-Based Chaotic Circuits with Applications to Communications’’, IEEE transactions on circuits and systems-11: analog and digital signal processing, vol. 40, N° 10, October 1993, pp. 626 - 633.

[17] Mathworks, Matlab Software, Version 7.3, Mathworks, 2006.

[18] K. Kemih and M. Benslama, “Performance of Chaotic Lü System in CDMA Satellites Communications Systems’’. *Int. Jour. of Information Technology*, Vol. 3, N° 3, March 2006.

[19] Mentor Graphics, « Modelsim SE User’s Manuel, Sofware”, Version 6. 4, *Mentor Graphics*, 2008.

[20] Xilinx, “Xilinx University Program Virtex-II Pro Development System”, *Xilinx*, UG069 (v1.1) April 9, 2008.

[21] M.A. Aseeri, M.I. Sobhi and P. Lee, “Lorenz Chaotic Model Using Field Programmable Gate Array (FPGA)’’, Midwest Symposium on Circuit and Systems, 2002, pp. 686-699.

[22] Xilinx, “Integrated Software Environment (ISE)”, Version 10.1, *Xilinx*, 2008.

[23] Agilent,” Digital Storage Oscilloscope DSO3202A”, *Agilent Technologies*, Inc. 2008, June 25, 2008.