Design of a New Hardware IP-HLS for Real-Time Image Chaos-Based Encryption

Mohamed Salah Azzaz[1][a], Redouane Kaibou[1][b] Hamdane Kamelia[1], Abdenour Kifouche[1][c]
and Djamel Teguig[2]

[1]*Ecole Militaire Polytechnique, Laboratoire Systèmes Électroniques et Numériques, BP 17 Bordj El-Bahri, Algiers, Algeria*
[2]*Ecole Militaire Polytechnique, Laboratoire Télécommunications, BP 17 Bordj El-Bahri, Algiers, Algeria*

Keywords: FPGA, HLS, Chaos, Encryption, Image, Security, Attack, NIST, DIHARD.

Abstract: This paper presents a new approach for designing a lightweight and efficient prototype of encryption devoted to secure real-time embedded applications. The proposed approach is simple and it is based on two concepts, the first one is related to the design methodology in which it allows a good compromise between performances and time development, by using Vivado High Level synthesis (HLS). In counterpart, and as a second concept chaos-based theory is adopted for the design of a robust stream cipher encryption algorithm with good trade-off between low resources and speed. Simulation and experimental results of image encryption demonstrate that the proposed design presents a good performances in terms of security, low resources and speed. Indeed, the solution can be embedded in many real-time applications namely video encryption.

1 INTRODUCTION

Nowadays, the increase of information transmitted through unsecured public networks pose a great challenge in terms of security in many new technologies such as 5G, 6G and IoT. To overcome this problem, several works are being carried out and others are under-way by academic researchers (Mohanta et al., 2020). Information security is thus becoming one of the major concerns of society. The most used alternative to this problem is cryptography discipline (Touqeer et al., 2021). However, classical cryptography techniques namely the standards *Rivest Shamir Adleman* (RSA), *Advanced and Data Encryption Standards* (AES/DES) are not appropriate for real-time embedded applications because they are computational demanding (Rawat et al., 2019). Thus, current work focuses on other techniques for implementing cryptosystems, that are based on chaotic systems allowing a good trade-off between security with good proprieties in terms of embedded system constraints (Azzaz et al., 2020b). The great motivation of using chaotic signals to secure information is confirmed by large scientific research community of this domain, because of the intrinsic properties such as aperiodic behaviour, determinism, long term unpredictability and its sensitivity to parameters and initial conditions (Alvarez and Li, 2006), and also the possibility of chaotic synchronization, usually used in communications (Kocarev and Parlitz, 1995; Sun et al., 2022). The use of chaos particularly in the field of communication has been considered as a very promising solution to increase the performance of analog or digital transmission systems. The disadvantage of the analog chaos transmission lies in the low degree of confidentiality and the degradation of the chaotic systems properties (Azzaz et al., 2013c). Hence the special importance assumed by cryptography in the digital chaos transmission, because of data integrity, non-repudiation and authenticity in addition to confidentiality (Azzaz et al., 2020b; Alawida et al., 2019; Bouteghrine et al., 2021; Tanougast et al., 2023). Indeed, chaotic systems have become a current trend for the design of dedicated cryptosystems as they are leightweight and produce chaotic encryption keys in contrast to the existing stream cipher standards using random generated encryption keys (CAESAR project, 2019; eSTREAM, 2014; NIST Lightweight Crypto, 2017). However, according to *Kerckhoffs* principle, the security of cryptographic

[a] https://orcid.org/0000-0001-6207-7626
[b] https://orcid.org/0000-0003-1749-4417
[c] https://orcid.org/0000-0002-6727-4626

Azzaz, M., Kaibou, R., Kamelia, H., Kifouche, A. and Teguig, D.
Design of a New Hardware IP-HLS for Real-Time Image Chaos-Based Encryption.
DOI: 10.5220/0012119400003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 478-485
ISBN: 978-989-758-666-8; ISSN: 2184-7711

systems should rely only on the key secrecy. In other words, all other parameters must be assumed to be publicly known. It was reformulated, independently, in *Shannon*'s maxim: "*the adversary knows the system*" (Kerckhoffs, 1883). It is considered today as a fundamental principle by cryptologists. Therefore, it is important to build encryption key generators that meet the security and embedded systems constraints (Fellah et al., 2021; Kifouche et al., 2022). One of the most usual of expression between persons that contains large volumes of information is image. Indeed, its security may be addressed in priority. However, a good numeric circuits in terms of speed and resources may be considered. Recently, the reliable performances of FPGA make them a good solution for real-time embedded applications (Kaibou et al., 2021; Gafsi et al., 2021; Cai et al., 2022). Many novel chaos-based methods for image exist, however, most of them, are only simulated and not validated experimentally (Hasan and Saffo, 2020; Bouteghrine et al., 2021). Some methods are based on Xilinx System Generator tools used for generating HDL code but they are not very optimal in terms of speed and resources (Hagras and Saber, 2020; Gafsi et al., 2023). Some works are based on direct HDL description methods but they require long development time (Azzaz et al., 2013b; Azzaz et al., 2013a; Azzaz et al., 2020b). Currently, a new design approach has been introduced integrating a C/C++ language, HLS (High Level Synthesis) as one of the most used tools to directly transform a C description into hardware IP-Core, described on Verilog or VHDL (RTL: Register-Transfer Level) with a performance comparable to that of manually coded RTL in terms of processing time and resources (Liu et al., 2019). In addition, the development time is generally very short compared to the hardware description based design (Azzaz et al., 2020a; Kaibou et al., 2021; Aissaoui et al., 2022). We will use the HLS tool to design our encryption key generator in the form of an IP-Core ready to be exported to *Vivado* and be used later in an encryption algorithm. In this context, the main contributions of this work are summarized in the following points:

- New architecture of chaos-based generator using a combined continuous/discrete chaotic systems.
- Using a new approach for designing chaos-based encryption system by using *Vivado-HLS*.
- Good security, hardware resources and speed performances by the proposed image cryptosystem.
- FPGA experimental validation of the proposed design on image encryption-decryption.

The remainder of this paper is structured as follows: Section 2 describes the proposed architecture of chaos-based key generator. The proposed image encryption algorithm is presented in Section 3. Security analysis and discussions are given in Section 4. Finally, Section 5 concludes this work.

2 PROPOSED CRYPTOSYSTEM

The proposed cryptosystem in this work is a stream cipher based on One Time Pad (OTP) encryption principle, more suitable for real time applications. The kernel of the later is a new chaos-based generator that makes the robustness of the cryptosystem more efficient against attacks while preserving low computation complexity.

2.1 Proposed Key Generator

The Architecture of the proposed key generator is depicted in Fig. 1. In order to make the generated encryption key sequences of good statistical properties, two rules are used: mixing and disruption.

In the mixing rule three (03) multipliers are used MUX1, MUX2 and MUX3. This allows to select the components x_i, y_i and z_i ($i = 1$ to 4) of the chaotic systems *Lorenz*, *Rössler*, *Chen* and *Lü* according to the selection sw_i (i = 1 to 4), respectively. The MUX4 is used to select the chaotic signals resulting from the first three Multiplexers according to sw_4. The disturbance rule is based on the perturbation of the logistic map trajectories through the signal resulting from the MUX4 to provide a sequences with good confusion propriety.

2.1.1 Mathematical Modelling

Lorenz chaotic system is represented by the Eq. 1. The chaotic behaviour is given by the following parameters and initial conditions, respectively: a = 10, b = 28 et c = 8/3, $x(0)$ = 0, $y(0)$= 5 et $z(0)$= 25.

$$\begin{cases} \dot{x} &= a(y-x) \\ \dot{y} &= bx-y-xz \\ \dot{z} &= xy-cz \end{cases} \tag{1}$$

Rössler chaotic system is represented by the Eq. 2. The chaotic behaviour is given by the following parameters and initial conditions, respectively: $a = 0.2$, $b = 0.2$ et $c = 5.7$, $x(0) = y(0) = z(0) = 0.1$.

$$\begin{cases} \dot{x} &= -(y+z) \\ \dot{y} &= -x+ay \\ \dot{z} &= b+z(x-c) \end{cases} \tag{2}$$

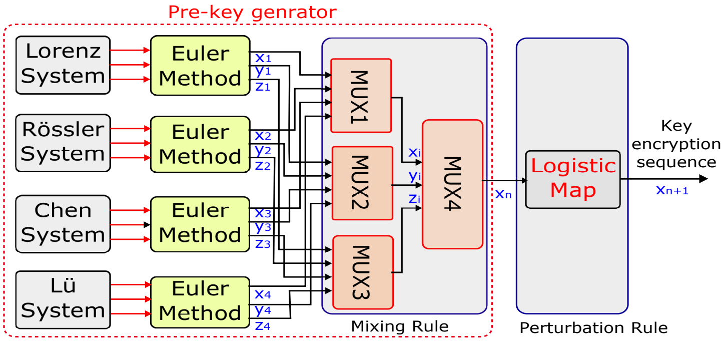


Figure 1: Proposed Encryption Key Generator.

Chen chaotic system is represented by the Eq. 3. The chaotic behaviour is given by the following parameters and initial conditions, respectively: $a = 35$, $b = 3$ et $c = 28$, $x(0) = y(0) = z(0) = 1$.

$$\left\{ \begin{array}{ll} \dot{x} & = a(y-x) \\ \dot{y} & = (c-a)x - xz + cy \\ \dot{z} & = xy - bz \end{array} \right. \tag{3}$$

Lü chaotic system is represented by the Eq. 4. The chaotic behaviour is given by the following parameters and initial conditions, respectively: $a = 36$, $b = 3$ et $c = 20$, $x(0) = y(0) = z(0) = 1$.

$$\left\{ \begin{array}{ll} \dot{x} & = a(y-x) \\ \dot{y} & = -xz + cy \\ \dot{z} & = xy - bz \end{array} \right. \tag{4}$$

The multiplexers MUX1, MUX2, MUX3 and MUX4 are described by the Eq. 5, 6, 7 and 8, respectively.

$$MUX1 : \left\{ \begin{array}{ll} \text{if } sw_1 & = 00 \Longrightarrow x_1 \text{ of } \textit{Lorenz} \\ \text{if } sw_1 & = 01 \Longrightarrow x_2 \text{ of } \textit{Rössler} \\ \text{if } sw_1 & = 10 \Longrightarrow x_3 \text{ of } \textit{Chen} \\ \text{if } sw_1 & = 11 \Longrightarrow x_4 \text{ of } \textit{Lü} \end{array} \right. \tag{5}$$

$$MUX2 : \left\{ \begin{array}{ll} \text{if } sw_2 & = 00 \Longrightarrow y_1 \text{ of } \textit{Lorenz} \\ \text{if } sw_2 & = 01 \Longrightarrow y_2 \text{ of } \textit{Rössler} \\ \text{if } sw_2 & = 10 \Longrightarrow y_3 \text{ of } \textit{Chen} \\ \text{if } sw_2 & = 11 \Longrightarrow y_4 \text{ of } \textit{Lü} \end{array} \right. \tag{6}$$

$$MUX3 : \left\{ \begin{array}{ll} \text{if } sw_3 & = 00 \Longrightarrow z_1 \text{ of } \textit{Lorenz} \\ \text{if } sw_3 & = 01 \Longrightarrow z_2 \text{ of } \textit{Rössler} \\ \text{if } sw_3 & = 10 \Longrightarrow z_3 \text{ of } \textit{Chen} \\ \text{if } sw_3 & = 11 \Longrightarrow z_4 \text{ of } \textit{Lü} \end{array} \right. \tag{7}$$

$$MUX4 : x_n = \left\{ \begin{array}{ll} x_i \text{ of } \textit{MUX1} \text{ if } sw_4 & = 00 \\ y_i \text{ of } \textit{MUX2} \text{ if } sw_4 & = 01 \\ z_i \text{ of } \textit{MUX3} \text{ if } sw_4 & = 10 \end{array} \right. \tag{8}$$

The produced signal x_n of the Eq. 8 is used to disturb the logistic map trajectories given by Eq. 9. The output sequence represents the key encryption K.

$$x_{n+1} = r x_n (1 - x_n) \tag{9}$$

To resolve the three dimensional continuous chaotic systems given by the Eq. 10, Euler numerical resolution method, expressed by the Eq. 11, is applied for each system.

$$\left\{ \begin{array}{ll} \dot{x} & = F(t,x,y,z) \\ \dot{y} & = G(t,x,y,z) \\ \dot{z} & = Q(t,x,y,z) \end{array} \right. \tag{10}$$

Where $x(t_0) = x_0$, $y(t_0) = y_0$, $z(t_0) = z_0$, F,G and Q are non linear functions. The parameter h is the descetisation step of *Euler* method.

$$\left\{ \begin{array}{ll} x_{n+1} & = x_n + hF(t_n, x_n, y_n, z_n) \\ y_{n+1} & = y_n + hG(t_n, x_n, y_n, z_n) \\ z_{n+1} & = z_n + hH(t_n, x_n, y_n, z_n) \end{array} \right. \tag{11}$$

Encryption/Decryption Processes

The encryption process consists of using OTP encryption algorithm according to Eq. 12.

$$c(k) = m(k) \oplus K(k) \tag{12}$$

The decryption process consists of using the reverse process of encryption according to Eq. 13.

$$m(k) = c(k) \oplus K(k) \tag{13}$$

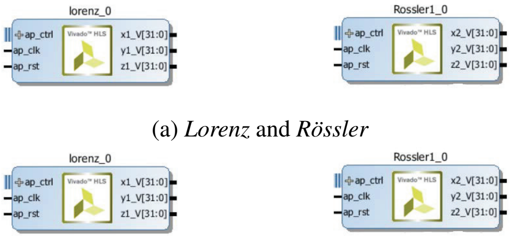
Where $m(k)$, $c(k)$ and $K(k)$ represent the plain-text, the cipher-text and the encryption key, respectively.

3 FPGA IMPLEMENTATION

The proposed FPGA design approach is based on *Vivado-HLS*.

3.1 Designed IP-Cores Using HLS

The designed IP-Cores undergo C/RTL Cosimulation to validate the designed hardware of the four chaotic systems to be then exported to the *Vivado* library as depicted in Fig. 2.



(a) *Lorenz* and *Rössler*

(b) *Lü* and *Chen*

Figure 2: HLS-based designed IP-Cores.

Fig. 3 illustrates the bloc design of *pre-key generator* and *Logistic Map* IP-Cores exported to *Vivado*. The 16-bits signals Ind_1 and Ind_2 represent the values of the selections sw_1 et sw_2 and the 32 bits output x_V represents the input x_n of the logistic IP-Core ($xxin\_V$).

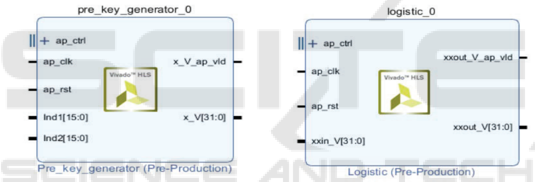


Figure 3: Pre-key generator and Logistic Map.

3.2 IP-Cores HLS Synthesis Results

Synthesis results after place and route on Genesys-2 FPGA performances of the four chaotic systems, *the Pre-key generator* and *Logistic* are shown in Tables 1 and 2, respectively. These results demonstrate that the HLS-based designed IP-Cores allow a good trade-off between FPGA resources consumption and timing and could be integrated as hardware accelerator for the design of chaos based cryptosystem.

Table 1: HLS IP-Cores FPGA Timing Results.

Clock (*ns*)	Target	Estimated	Uncertainty
ap_clk of Chaotic systems	10.00	7.98	1.25
ap_clk of Pre-key Gen.	10.00	80.66	1.25
ap_clk of Logistic Map	10.00	8.43	1.25

Table 2: HLS Resources Consumption Results.

Name	BRAM_18K	DSP48E	FF	LUT
Lorenz	0	14	550	407
Rössler	0	12	555	375
Chen	0	16	610	311
Lü	0	14	478	357
pre-key Gen.	0	65	902	2322
Logistic	0	6	52	96
Available	270	240	126800	63400

3.3 *Vivado-XSim* Simulation Results

Simulation results of the obtained chaotic signals generated from the designed IP-Cores of the four chaotic systems and *Logistic Map* are illustrated in Fig. 4.

The pre-key generator IP-core chaotic sequences are obtained according to two dynamically configured switches sw_1 and sw_2 to visualize the output signals (x, y and z) of the four generators for each value serving as input of the *Logistic Map*.

3.4 Proposed Key Generator

The proposed chaos-based generator RTL is designed under *Vivado* using the exported hardware IP-cores as depicted in Fig. 5. A register of 32 bits (RTL_REG) is used for the pre-key generator output to disturb *Logistic Map* trajectories as illustrated in the simulation results of Fig. 6.

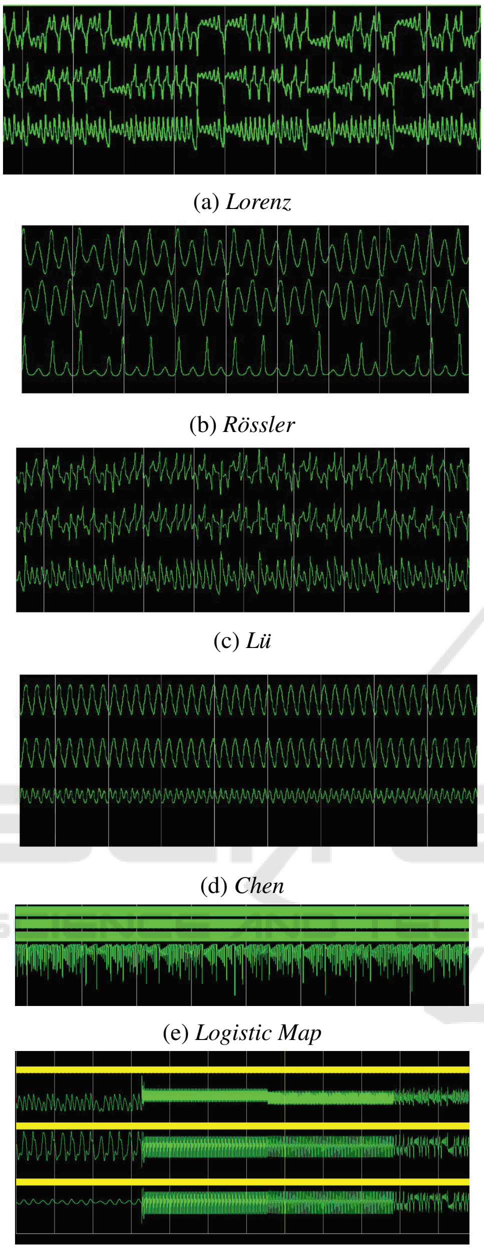
Table 3 illustrates the resources utilization, the timing and the power consumption of the proposed chaos-based generator, theses results demonstrate that the proposed architecture allows a good compromise between resources utilization, the timing and the power, thus, fulfilling the requirements of an embedded system.

Table 3: Proposed Generator FPGA Implem. Results.

Resources	Utilisation	Available	Utilisation %
LUT	3029	203800	0.015
Registers	905	407600	2.223×10^{-3}
DSP	38	840	0.045
IO	18	500	0.036
BUFG	1	32	0.031
W. Negative Slack (WNS)	2.015 *ns*		
Total On-Chip Power	0.366 *W*		

3.5 Key Generator Randomness Tests

In order to measure the random aspect of the produced sequences, *ENT*, *NIST* and *DIEHARD* test batteries have been used giving the results of Tables 4, 5, and 6 and thus allowed to be considered as pseudo-random suitable for data security applications.


(a) *Lorenz*

(b) *Rössler*

(c) *Lü*

(d) *Chen*

(e) *Logistic Map*

(f) *Pre-key gen*

Figure 4: HLS designed IP-Cores Simulation Results.

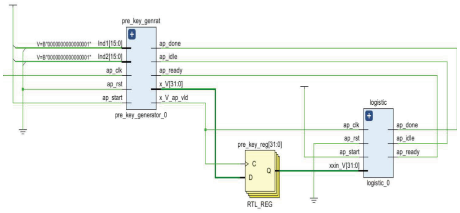


Figure 5: RTL Architecture of the Proposed Generator.

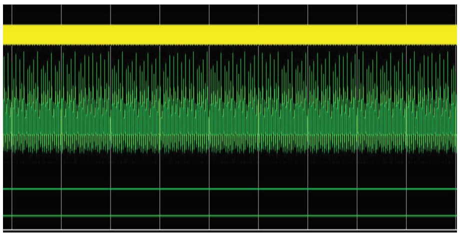


Figure 6: Proposed Generator Simulation Results.

Table 4: *ENT* battery of tests.

ENT tests	values
Entropy	7.999987 bits per byte =8
Chi square test	302.76 > 2.17
Arithmetic mean	127.5685 (127.5 = random)
Monte Carlo value for Pi	3.139690399 (error 0.06 percent)
Serial correlation coefficient	-0.00001218

Table 5: *NIST* battery of tests.

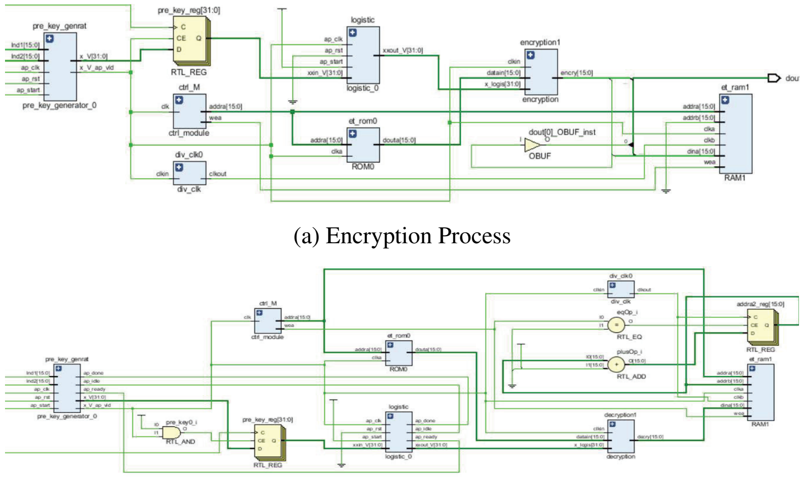
Test	P-Value	Proportion	Test result
Frequency	0.534146	10/10	success
Block Frequency	0.534146	9/10	success
Cumulative Sums (Cusum)	0.534146	10/10	success
Runs	0.350485	10/10	success
Longest Run of Ones in a Block	0.911413	10/10	success
Binary Matrix Rank	0.122325	10/10	success
Discrete Fourier Transform	0.534146	10/10	success
Non Overlapping Template	0.122325	10/10	success
Overlapping Template	0.350485	10/10	success
Universal	0.911413	10/10	success
Approximate Entropy	0.739918	10/10	success
Random Excursions	0.5350485	10/10	success
Random Excursions Variant	0.739918	10/10	success
Serial	0.534146	10/10	success
Linear Complexity Test	0.739918	10/10	success

Table 6: *DIEHARD* battery of tests.

Test	P-value	Test result
Birthday spacings test	0.932537	success
Overlapping 5-permutations test	0.551431	success
Binary rank test for 31×31 matrices	0.564486	success
Binary rank test for 32×32 matrices	0.867109	success
Binary rank test for 6×8 matrices	0.461467	success
The tests OPOSO.OQSO and DNA	0.558904	success
Counter the 1's in successive bytes	0.426273	success
Counter the 1's in specified bytes	0.717346	success
A parking lot test	0.142755	success
Minimum distance test	0.426242	success
3D spheres test	0.994113	success
Z-scores	0.249269	success
Overlapping sums test	0.501703	success
Bitstream test	0.890750	success
Runs test	0.826632	success
Craps test	0.77263	success

3.6 Key Space

The key-space is defined as the total number of different keys used in the encryption algorithm. Knowing that our chaos-based generator is composed of five chaotic systems: four continuous-time chaotic systems each of which has 6 parameters (three state variables and three system parameters) coded on 32 bits and a discrete-time chaotic system containing two pa-



(a) Encryption Process

(b) Decryption Process

Figure 7: Proposed Cryptosystem RTL Architecture.

rameters such as: $(n_1, m_1) = (4, 1)$, $(n_2, m_2) = (6, 1)$ and $p = 32$. with n and m are respectively the number of generators used and parameters of each generator and p represents the size of the key in binary.

The key space is calculated as:

$k = 2^{n \times m \times p} = k_1 + k_2 = 2^{n_1 \times m_1 \times p} + 2^{n_2 \times m_2 \times p}$; $k = 2^{768} + 2^{64}$; $k = 2^{100}(2^{7.68} + 2^{-36}) \simeq 2^8 \times 2^{100} \gg 2^{100}$. The resulting key space is big enough than the value presented by (Alvarez and Li, 2006), therefore the later resists to the brute force attack.

3.7 Proposed Chaos-Based Cryptosystem

Fig. 7 illustrates the proposed chaos-based cryptosystem RTL architecture for encryption and decryption processes.

Table 7 shows the resources consumption results of the proposed cryptosystem implementation. It is noticed that the consumed resources is low compared to the available resources of the FPGA. Therefore, the co-design approach adopted is adequate to the required specifications.

Table 7: Cryptosystem Implementation Results.

Ressource	Utilisation	Available	Utilisation %
LUT	1466	203800	7.193×10^{-3}
Registers	507	407600	1.244×10^{-3}
FF	391	407600	9.593×10^{-4}
BRAM	30	445	0.067
DSP	17	840	0.020
IO	18	500	0.036
BUFG	3	32	0.063
W. Negative Slack (WNS)	2.018 *ns*		
Total On-Chip Power	0.342 *W*		

Fig.8 visualizes, on a VGA screen, the FPGA implementation results of encryption/decryption using the proposed chaos-based key generator.

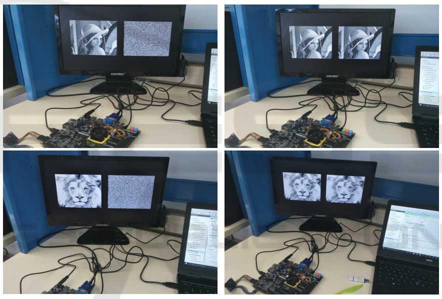


Figure 8: Encrypted and Decrypted Images.

3.8 Cryptosystem Security Analysis

Security analysis is evaluated using statistical and differential analyses.

3.8.1 Statistical Analysis

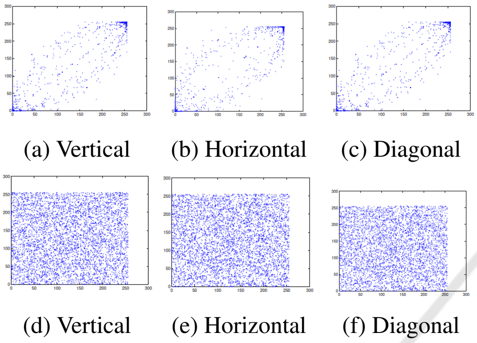
Statistical analysis is performed through correlation and histogram metrics:

Adjacent Pixels Correlation. Fig. 9 shows the correlation (Bouteghrine et al., 2021) of the original and encrypted images neighbouring pixels. They show a strong correlation for the three directions in the original image whereas encrypted images are strongly decorrelated. Therefore the proposed cryptosystem resists correlation attacks.

Histogram. It can be seen from Fig. 10 showing the original and encrypted images histograms, that the

Table 8: Proposed Cryptosystem Performance Comparison.

	(Hagras and Saber, 2020)	(Maazouz et al., 2022)	(Hasan and Saffo, 2020)	Proposed Work
Keys Generator	Chaotic	Chaotic-AES	Chaotic	Chaotic
Data Size (bits)	32	32	8	32
Keyspace	$10^{135} \times 2^{16}$	2^{356}	2^{256}	2^{800}
FPGA Platform	Spartan-6 X6SLX45	Zybo Z20	SP605 XC6SLX45T	Genesys 2
Data Representation	Fixed Point	Floating Point	Fixed Point	Fixed Point
Maximal Frequency (MHz)	393	666.67	23.356	125.28
Throughput (Mb/s)	1.75	2.7	186.84	4008.9
Test Battery	NIST	-	DIEHARD	ENT-NIST-DIEHARD
NPCR (%)	99.654	99.59	99.62316	99.61
UACI (%)	33.435	33.27	32.68375	33.0481
LUT	294	-	238	3029
Power (mW)	117	-	-	366
Design Approach	XSG/VHDL	C/VHDL	XSG/VHDL	C/VHDL
Development Time	Low	High	Low	Medium
Optimisation Approach	Low	Low	Low	Good


(a) Vertical (b) Horizontal (c) Diagonal

(d) Vertical (e) Horizontal (f) Diagonal

Figure 9: Correlation Analysis.

pixels intensity distribution is similar to that of a uniform law, hence the random character of this distribution for encrypted images. The used algorithm therefore resists against histogram attacks.

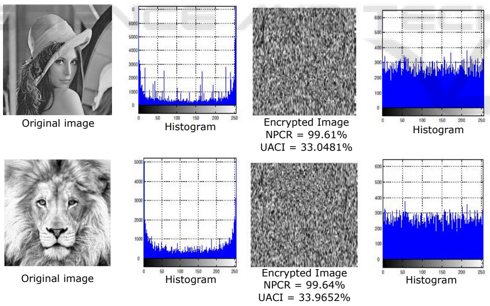


Figure 10: Histogram Analysis.

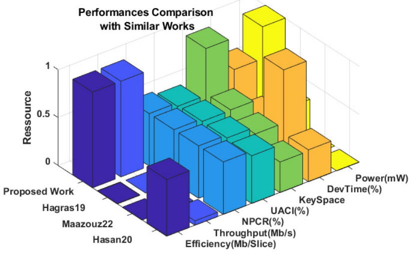


Figure 11: Comparison with Similar Works.

3.8.2 Differential Analysis

Differential analysis has been conducted using Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI) metrics (Bouteghrine et al., 2021) in order to measure the resistance of a cryptosystem against differential attacks. Fig. 10 shows the results for original and encrypted images.

4 COMPARISON WITH SIMILAR WORKS

Fig. 11 and Table 8 present the comparison in terms of resource consumption between the designed generator and those of previous works. It shows smaller development time with more flexibility using the co-design approach providing tested random sequences.

5 CONCLUSION

In this work, the concept of secure encryption by chaos is presented by designing and implementing a new key generator integrated in a chaos-based cryptosystem. The design method has been using HLS co-design offering excellent development time with good hardware performances. The proposed key generator has undergone three different standard battery tests namely *ENT*, *NIST* and *DIEHARD* for generated sequences randomness before being adopted for the developed cryotsystem architecture. Simulation and FPGA implementation results have given good results in terms of encryption/decryption applied on images. The cryptosystem has been evaluated in terms of robustness and resources consumption and has proven to be suitable for real-time data security applications as it has been compared to similar works.

REFERENCES

Aissaoui, N., Kaibou, R., and Azzaz, M. S. (2022). Real-time fpga implementation of digital video watermarking techniques using co-design approach: Comparative study. In *7th Int. Conf. on Image and Sig.Proc. and their App.*, pages 1–6. IEEE.

Alawida, M., Teh, J. S., et al. (2019). An image encryption scheme based on hybridizing digital chaos and finite state machine. *Sig. Proc.*, 164:249–266.

Alvarez, G. and Li, S. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *Int. j. of bifu. and chaos*, 16(08):2129–2151.

Azzaz, M. S., Maali, A., Kaibou, R., Kakouche, I., Mohamed, S., and Hamil, H. (2020a). Fpga hw/sw codesign approach for real-time image proc. using hls. In *1st Int. Conf. on Comm., Control Sys. and Sig. Proc.*, pages 169–174. IEEE.

Azzaz, M. S., Tanougast, C., Maali, A., and Benssalah, M. (2020b). An efficient and lightweight multi-scroll chaos-based hardware solution for protecting fingerprint biometric templates. *Int. J. of Comm. Sys.*, 33(10):e4211.

Azzaz, M. S., Tanougast, C., Sadoudi, S., and Bouridane, A. (2013a). Synchronized hybrid chaotic generators: Application to real-time wireless speech encryption. *Comm. Nonl. Sci. and Num. Simu.*, 18(8):2035–2047.

Azzaz, M. S., Tanougast, C., Sadoudi, S., and Dandache, A. (2013b). Robust chaotic key stream generator for real-time images encryption. *J. of RT Image Proc.*, 8(3):297–306.

Azzaz, M. S., Tanougast, C., Sadoudi, S., Fellah, R., and Dandache, A. (2013c). A new auto-switched chaotic system and its fpga implementation. *Comm. in Nonlinear Sci. and Num. Simu.*, 18(7):1792–1804.

Bouteghrine, B., Tanougast, C., and Sadoudi, S. (2021). Novel image encryption algorithm based on new 3-d chaos map. *Mult. Tools and App.*, 80:25583–25605.

CAESAR project (2019). CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. https://competitions.cr.yp.to/caesar.html. Online; accessed 2023.

Cai, H., Sun, J.-y., Gao, Z.-b., and Zhang, H. (2022). A novel multi-wing chaotic system with fpga implementation and application in image encryption. *J. of RT Image Proc.*, 19(4):775–790.

eSTREAM (2014). eSTREAM: the ECRYPT Stream Cipher Project. https://www.ecrypt.eu.org/stream/. Online; accessed 2023.

Fellah, R., Azzaz, M. S., Tanougast, C., and Kaibou, R. (2021). Design of a simple and low cost chaotic signal generation circuit for uwb applications. *The Europ. Phys. Journal Spec. Topics*, 230(18-20):3439–3447.

Gafsi, M., Amdouni, R., et al. (2023). Hardware implementation of a strong pseudorandom number generator based block-cipher system for color image encryption and decryption. *Int. J. of Circuits. Theory and App.*, 51(1):410–436.

Gafsi, M., Hajjaji, M. A., Malek, J., and Mtibaa, A. (2021). Fpga hw acceleration of an improved chaos-based cryptosystem for rt image encryption and decryption. *J. of Amb. Intell. and Human. Comput.*, pages 1–22.

Hagras, E. A. A. and Saber, M. (2020). Low power and high-speed fpga implementation for 4d memristor chaotic system for image encryption. *Mult. Tools and App.*, 79:23203 – 23222.

Hasan, F. S. and Saffo, M. A. (2020). Fpga hw co-simulation of image encryption using stream cipher based on chaotic maps. *Sens. and Imag.*, 21(1):35.

Kaibou, R., Azzaz, M. S., Benssalah, M., Teguig, D., Hamil, H., Merah, A., and Akrour, M. T. (2021). Real-time fpga implementation of a secure chaos-based digital crypto-watermarking system in the dwt domain using co-design approach. *J. of RT Image Proc.*, 18(6):2009–2025.

Kerckhoffs, A. (1883). *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef.* Librairie militaire de L. Baudoin.

Kifouche, A., Azzaz, M. S., et al. (2022). Design and implementation of a new lightweight chaos-based cryptosystem to secure iot communications. *Int. J.of Info. Sec.*, 21(6):1247–1262.

Kocarev, L. and Parlitz, U. (1995). General approach for chaotic synchronization with applications to communication. *Phy. review letters*, 74(25):5028.

Liu, S., Lau, F. C., and Schafer, B. C. (2019). Accelerating fpga prototyping through predictive model-based hls design space exploration. In *Proc. of the 56th Annual Design Automation Conf.*, pages 1–6.

Maazouz, M., Toubal, A., Bengherbia, B., Houhou, O., and Batel, N. (2022). Fpga implementation of a chaos-based image encryption algorithm. *J. of King Saud Univ.-Comp. and Info. Sciences*, 34(10):9926–9941.

Mohanta, B. K., Jena, D., Satapathy, U., and Patnaik, S. (2020). Survey on iot security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *IoT*, 11:100227.

NIST Lightweight Crypto (2017). Lightweight Cryptography. https://csrc.nist.gov/projects/lightweight-cryptography/round-1-candidates. Online; accessed 2023.

Rawat, A., Sehgal, K., Tiwari, A., Sharma, A., and Joshi, A. (2019). A novel accelerated implementation of rsa using parallel processing. *J. of Discrete Math. Sci. and Crypto.*, 22(2):309–322.

Sun, J., Zang, M., Liu, P., and Wang, Y. (2022). A secure communication scheme of three-variable chaotic coupling synchronization based on dna chemical reaction networks. *IEEE Trans. on Sig. Proc.*, 70:2362–2373.

Tanougast, C., Bouteghrine, B., Sadoudi, S., and Chen, H. (2023). Image encryption using a chaotic/hyperchaotic multidimensional discrete system. In *Recent Adv.in Im. Sec. Tech.: Intelligent Image, Sig., and Video Proc.*, pages 105–125. Springer.

Touqeer, H., Zaman, S., Amin, R., Hussain, M., Al-Turjman, F., and Bilal, M. (2021). Smart home security: challenges, issues and solutions at different iot layers. *The J. of Sup.comp*, 77(12):14053–14089.