electronics

MDPI

Article

FPGA-Based Chaotic Image Encryption Using Systolic Arrays

Furkan Ciylan [1,*], Bünyamin Ciylan [2] and Mehmet Atak [3]

[1] Cyberdes Microelectronics, Ankara 06378, Turkey
[2] Department of Computer Engineering, Technology Faculty, Gazi University, Ankara 06560, Turkey; bciylan@gazi.edu.tr
[3] Department of Industrial Engineering, Gazi University, Ankara 06560, Turkey; matak@gazi.edu.tr
* Correspondence: furkan.ciylan@cyberdes.com; Tel.: +90-534-050-1484

Abstract: Along with the recent advancements in video streaming, concerns over the security of transferred data have increased. Thus, the development of fast and reliable image encryption methodologies has become an emerging research area in the field of communications. In this paper, a systolic array-based image encryption architecture is proposed. Systolic arrays are used to apply the convolution operation, and a Lü–Chen chaotic oscillator is used to obtain a convolutional filter. To decrease resource consumption, a method to fuse confusion and diffusion processes by using systolic arrays is also proposed in this paper. The results show that the proposed method is highly secure against some differential and statistical attacks. It is also shown that the proposed method has a high speed of encryption compared to other methods.

Keywords: Lü–Chen (2002) chaotic system; FPGA; chaotic systems; image encryption; convolution; systolic array

check for updates

Citation: Ciylan, F.; Ciylan, B.; Atak, M. FPGA-Based Chaotic Image Encryption Using Systolic Arrays. *Electronics* **2023**, *12*, 2729. https://doi.org/10.3390/electronics12122729

Academic Editors: Esteban Tlelo-Cuautle, Everardo Inzunza-González and Walter Leon-Salas

Received: 27 April 2023
Revised: 12 June 2023
Accepted: 13 June 2023
Published: 19 June 2023

1. Introduction

Data security and the latest advances in the video streaming field have gained importance. Secure and real-time image transfer is the focus of much multimedia research. Since digital images have low entropy, high redundancy, and strong pixel correlation, encrypting image data using traditional methods is not efficient. Thus, designing new cryptographic algorithms using chaotic systems has become an emerging field of research.

A cryptographic system uses keys to encrypt and decrypt data. There are two different cryptographic algorithm categories. These are symmetric encryption and asymmetric encryption [1]. In symmetric encryption methods, both the sender and receiver must have the same key pair. Although these types of encryption methods have some advantages over asymmetric encryption methods, such as simplicity and the requirement of fewer resources, the key exchange process is a major problem [2]. In asymmetric encryption methods, a public and private key pair are used. Everyone can know the public key, but the private key should be kept a secret. The private key cannot be derived from the public key.

Traditional asymmetric algorithms such as elliptic curve cryptography (ECC) [3] or Rivest–Shamir–Adleman (RSA) [4] and even symmetric algorithms such as Advanced Encryption Standard (AES) [5] are very resource- and energy-consuming methods of image data encryption, which affects the performance of real-time streaming applications. This situation limits the use of those cryptographic algorithms in constrained environments. The problem typically applies to the Internet of Things (IoT) and urges the introduction of new lightweight cryptographic applications.

Chaotic maps are used to create chaotic sequences. It is not possible to predict or analyze chaotic sequences [6,7]. The complexity of the structure of a chaotic sequence is very high. Due to these advantages, chaotic systems have the potential to be actively used in encryption systems to increase overall system security. Two steps are generally followed to create a chaotic encryption method. These are scheme scrambling and diffusion.

The implementation of cryptographic algorithms on Field Programmable Gate Arrays (FPGAs) has become an important field of research due to their highly parallelized high-speed computing capabilities [8]. Due to their reprogramming capability, it is possible to use them to rapidly develop and create Application-Specific Integrated Circuit (ASIC) prototypes. This method is applied to the FPGA since it is possible to reach higher encryption speeds due to its highly parallelized architecture.

Today, video streaming systems can stream videos at over 120 frames per second. Along with these high speeds, some industries, such as the defense industry and video streaming platforms, also need a secure end-to-end encryption methodology. However, when the performance levels of today's chaotic encryption systems are considered, reaching those frame rates without high latency is impossible. Thus, a high-speed encryption methodology is required in this field to achieve up-to-date streaming technology.

One of the main problems in the image encryption field is that building a real-time solution with high security is difficult. It is possible to reach a high level of security with chaotic systems, but they still require significant time to encrypt images. The main motivation of this paper is to solve this issue. With the proposed approach, creating real-time encryption solutions with higher security is now possible. The proposed approach uses systolic arrays to ensure that a continuous image encryption pipeline is established to reach a high level of security when streaming images.

In this work, a new architecture that brings together the convolution operation with systolic arrays and chaotic oscillators is presented. It is shown that the proposed architecture can successfully encrypt images at higher speeds. Thus, implementing the proposed architecture makes it possible to create safe and low-latency video streaming pipelines. Along with this, a new architecture for further developments in the chaotic encryption field is proposed.

This paper proposes a novel FPGA-based chaotic encryption model using the convolution operation with systolic arrays over the Lü–Chen chaotic oscillator proposed by [9].

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the proposed method. Section 4 describes the experiments and the evaluation of the results. The discussion is presented in Section 5. The conclusion is given in Section 6.

2. Related Works

The fundamentals of chaos theory were proposed by Henri Poincaré [10]. Lorenz then researched the application of chaos theory to weather systems [11]. In his research, he tried to reuse outputs gathered from a mathematical model by rounding them. By applying this technique, he aimed to achieve results faster, but in the end, he observed unpredictable results. Li et al. used the term chaos to explain results such as those obtained by Lorenz [12]. Leon O. Chua built an analog circuit and applied a chaotic system [13,14]. The continuous autonomous Chua circuit set an important example for later research conducted in this field [15,16].

In 2007, Zhang et al. proposed a chaos-based image encryption algorithm [17]. To improve the security and speed of this method, standard maps, cat maps, and Baker maps were used [18,19]. E. Yavuz et al. have also proposed an image encryption algorithm using chaos [20]. They proposed applying confusion and diffusion principles to an image using two independent chaotic functions. This method is preferred over advanced scrambling and diffusion methods in some studies. M. François et al. proposed a new approach by using symmetric chaotic encryption [21]. The method depends on the bitmap permutation process, resulting in higher computational complexity. K.-W. Wong et al. proposed using a look-up table and a combination of swapping methods instead of a 1D chaotic map in the diffusion process [22]. X. Zhang and Z. Zhao proposed a two-way diffusion method. In this method, they used pixel values from top to bottom and right to left in addition to the standard diffusion process [23]. This helped to reduce time complexity by reducing the number of required rounds. J.-X. Chen et al. proposed a continuous diffusion method. They applied complementary diffusion in addition to the standard diffusion process [24]. This

method reduced the time complexity by increasing security in fewer rounds. X. Wang et al. proposed a chaotic block image encryption method. This method was based on the dynamic random growth technique [25]. In this method, they applied an intermediate parameter as the initial conditions of a chaotic map and generated random data pruned against plaintext attacks.

Studies on FPGA-based high-speed methods have gained importance, along with the recent advancements in real-time streaming applications and increasing demand for encrypted video streams. Koyuncu et al. proposed a new design using a Burke–Shaw Chaotic system with an RK5-Butcher algorithm on Virtex-6 FPGA [26]. Alcin et al. used the Pehlivan–Uyaroglu chaotic system along with the Artificial Neural Network approach on a Virtex-6 FPGA [27]. Koyuncu et al. proposed a Pehlivan–Wei chaotic system design. In this design, they used Euler, RK4, and Heun algorithms instead of the Virtex-6 FPGA [28]. Tuna et al. proposed using a chaotic system design that uses the Heun algorithm on a FPGA [29]. Akgul et al. proposed a 3D chaotic system design using the RK4 algorithm [30]. Two fractional-order 4D chaotic system designs on the Kintex 7 FPGA have been proposed by Rajagopal et al. [31]. A multi-butterfly chaotic system design has been proposed by Lai et al. [32]. Tlelo–Cuautle et al. have proposed a multi-scroll chaotic oscillator design on the Cyclone IV FPGA [33]. A chaotic chameleon system with an RK5B algorithm on a Virtex-6 FPGA has been proposed by Rajagopal et al. [34]. Aceng S. et al. have proposed a novel 3D chaotic system with line equilibrium to encrypt images [35].

3. Materials and Methods

The system proposed in this study is an end-to-end parallel pipeline. Thus, it is possible to extend the pipeline for increased security. Confusion and a diffusion process are applied at each step of this pipeline. The standard methodology for applying the diffusion process to a FPGA is very costly. Thus, a new method to combine confusion and diffusion processes by using the systolic arrays is proposed in this work.

At the beginning of the process, a diffusion process is applied to the input array of pixels. The dimensionality of each array is 3×3. Then, the convolution process is applied to the array.

This work uses a Lü–Chen chaotic system as a chaotic oscillator to generate a randomized sequence of numbers. These numbers then feed forward to a systolic array to apply a convolutional filter for confusion. Due to the nature of the convolution operation, it is possible to create a sequential pipeline for further encryption. At the end of the process, a confused image is generated.

A standard deconvolution process is applied along with the inverse of the diffusion process to decrypt the resulting encrypted image. The number of layers used to encrypt the image directly must affect the time complexity of the decryption phase. Since the encrypted image is a confused and diffused version of the real image, using the wrong number of layers in the decryption phase will result in a meaningless image.

3.1. Lü–Chen Chaotic System

In 1999, Chen proposed a chaotic attractor that is topologically not equal to Lorenz's attractor. Lorenz's attractor is generalized by the $a_{12}a_{21} > 0$ condition with its linear part [36] $A = [a_{ij}]$. However, Chen's system satisfies condition $a_{12}a_{21} < 0$ and belongs to another canonical family of chaotic systems. Jinhu Lu and Guarong Chen later proposed a nonlinear differential equation called the Lü–Chen 2002 chaotic system, which satisfies the $a_{12}a_{21} = 0$ condition [37]. This system is shown in Equation (1).

$$\begin{aligned} \dot{x} &= a \cdot (y - x) \\ \dot{y} &= -x \cdot z + c \cdot y \\ \dot{z} &= x \cdot y - b \cdot z \end{aligned} \tag{1}$$

where a, b, and c are ∈ R. The parameters a and b are fixed and set to a = 36 and b = 3. Parameter c is a variable that influences the system's behavior by generating different types of attractors. Specifically, when c is set within the range of 12.7 to 17.0, the system produces an attractor resembling the Lorenz attractor. The system exhibits a transitory shape in the range of 18.0 to 22.0. Finally, for values of c between 23.0 and 28.5, the system generates an attractor similar to Chen's attractor.

In part a of Figure 1, c is set to 27, so a Lü–Chen attractor is produced. In part b, c is set to 29; in part c, it is set to 20; and in part d, it is set to 15. The solution of the Lü–Chen chaotic system creates an inner loop to generate an organization inside which it is impossible to predict the amplitude and frequency. In this work, the initial parameters are set as follows: parameter a is set to 36, b is set to 3, and c is set to 20. All figures are obtained by using the matplotlib library in Python.

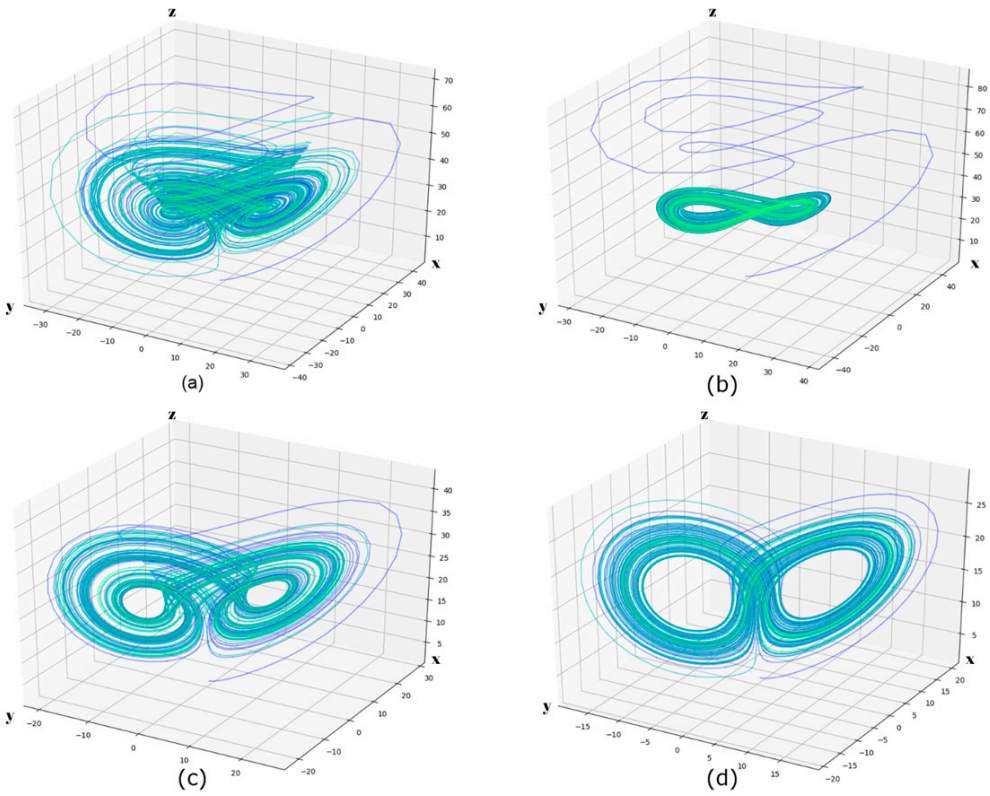

Figure 1. Lü–Chen 2002 attractors where c is set to 27 in (**a**), 29 in (**b**), 20 in (**c**), and 15 in (**d**).

In 2019, M. Tuna et al. proposed a method to model a Lü–Chen chaotic oscillator on a FPGA using the Heun algorithm [9]. We followed the same methodology using Verilog HDL. The equation for the Heun algorithm is given in Equation (2).

$$\begin{aligned} x\left(k^0+1\right) &= x(k) + (a(y(k) - x(k)))\cdot\Delta h \\ x(k+1) &= x(k) + \left(\frac{(a\cdot(y(k)-x(k)))+x\left(k^0+1\right)}{2}\right)\cdot\Delta h \\ y\left(k^0+1\right) &= y(k) + (-x(k)\cdot z(k) + c\cdot y(k))\cdot\Delta h \\ y(k+1) &= y(k) + \left(\frac{(-x(k)\cdot z(k)+c\cdot y(k))+y\left(k^0+1\right)}{2}\right)\cdot\Delta h \\ z\left(k^0+1\right) &= z(k) + (x(k)\cdot y(k) - b\cdot z(k))\cdot\Delta h \\ z(k+1) &= z(k) + \left(\frac{(x(k)\cdot y(k)-b\cdot z(k))+z\left(k^0+1\right)}{2}\right)\Delta\cdot h \end{aligned} \tag{2}$$

In this equation, the intermediate values $x(k^0 + 1)$, $y(k^0 + 1)$, and $z(k^0 + 1)$ are presented. The intermediate values are calculated first since they are required in the next step. Then,

the values of $x(k+1)$, $y(k+1)$, and $z(k+1)$ are calculated. Δh is given as the step size and set to 0.01. The block diagram of the FPGA implementation is shown in Figure 2.

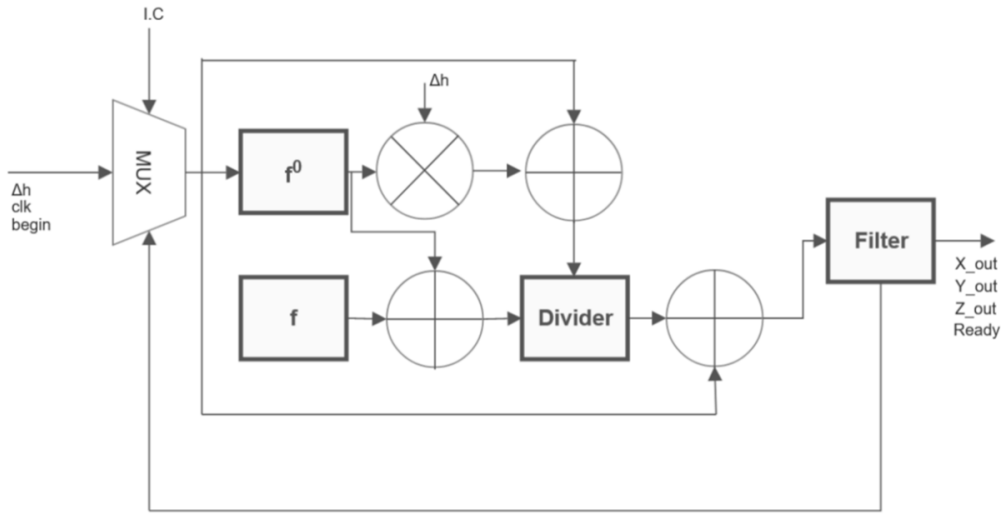

Figure 2. Block diagram of Heun-based Lü–Chen 2002 chaotic oscillator.

A Modulo 255 operation is carried out on each system output value presented in Figure 2 to successfully apply the convolutional filter.

3.2. Systolic Array

A systolic array is a hardware structure designed to increase the speed and efficiency of algorithms that perform the same operation on different data at each time step [38]. It is possible to use systolic arrays to change the standard pipeline architecture. Because of its regularity and scalability, a systolic array is used in this work to apply convolution to the input array of pixels. One of the advantages of the systolic-array-based architecture is its ability to handle high-throughput capacity requirements. This makes them useful in real-time encrypted video streaming applications.

A standard systolic array architecture is shown in Figure 3. For each 3×3 pixel array, weights repeat themselves to apply the same convolutional filter to an image. At each clock cycle, pixels are moved toward the systolic array units. Then, a pixel value is moved between systolic array units delayed by one clock cycle. Since the pixel array size is larger than the filter size, the result of each block is stored in the accumulators and added to the result of the next block.

The architecture of each Processing Element (PE) is shown in Figure 4. It multiplies the inputting pixel value by the related weight and then applies the addition process to the output of the previous unit. It also passes the inputting value to the next unit and outputs a value to the unit under it.

With a standard pipeline architecture, the diffusion process on FPGA is very costly since saving a copy of the diffused version of an image on RAM and the amount of logic required to compute the location changes in parallel is necessary. Thus, a new methodology to combine confusion and diffusion processes is required to decrease the cost of implementation. This work proposes a systolic array-based solution to overcome the problem.

As shown in Figure 5, it is possible to apply convolution-based confusion and diffusion processes using systolic arrays with an additional multiplexer implementation. In the first step of this architecture, pixels are vertically fed through PEs. Then, the outputs of the Lü–Chen 2002 oscillators are fed through the systolic arrays as weight values in the order of Y, Z, and X. Those values are stored in horizontally changed accumulators at the final step. Thus, pixels change places in 2 by 2, and the convolution process is applied to every pixel passed through.

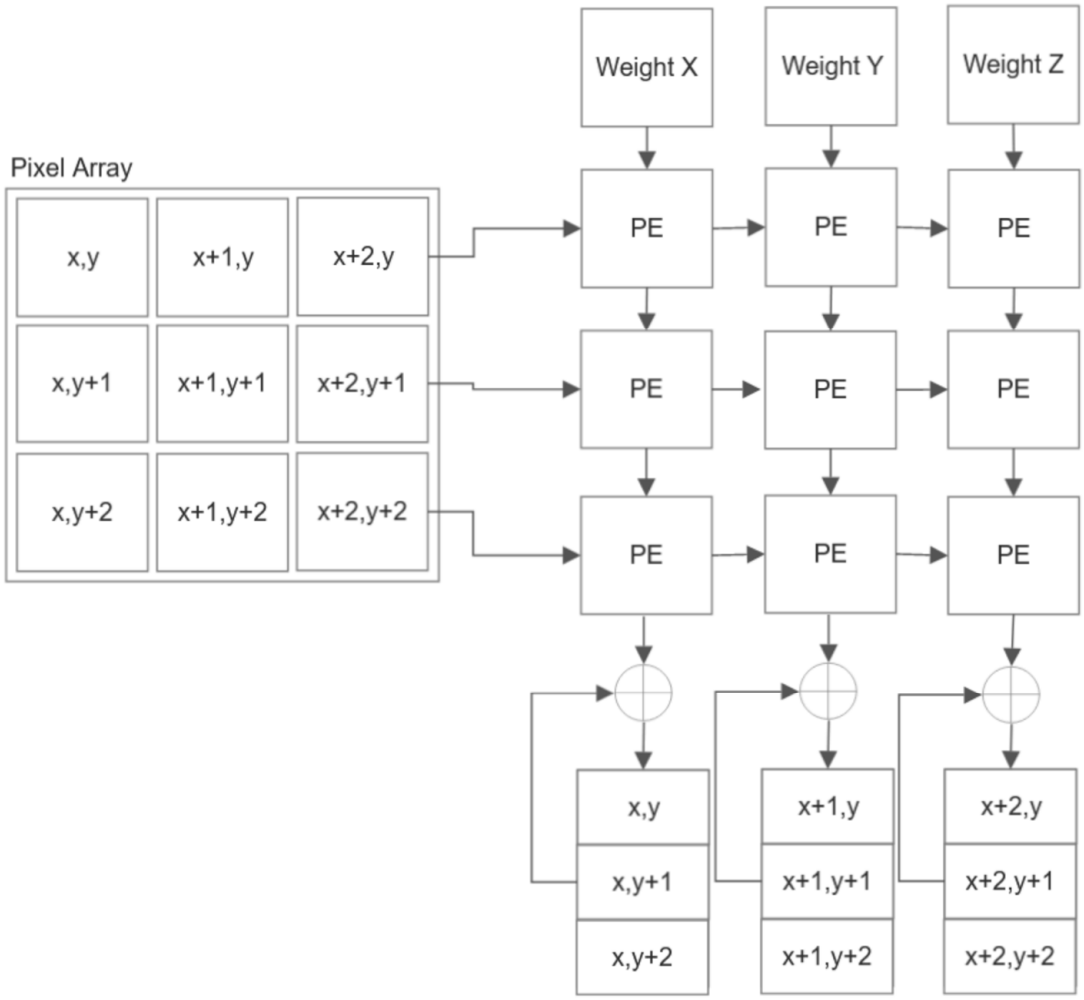

Figure 3. Standard systolic array architecture.

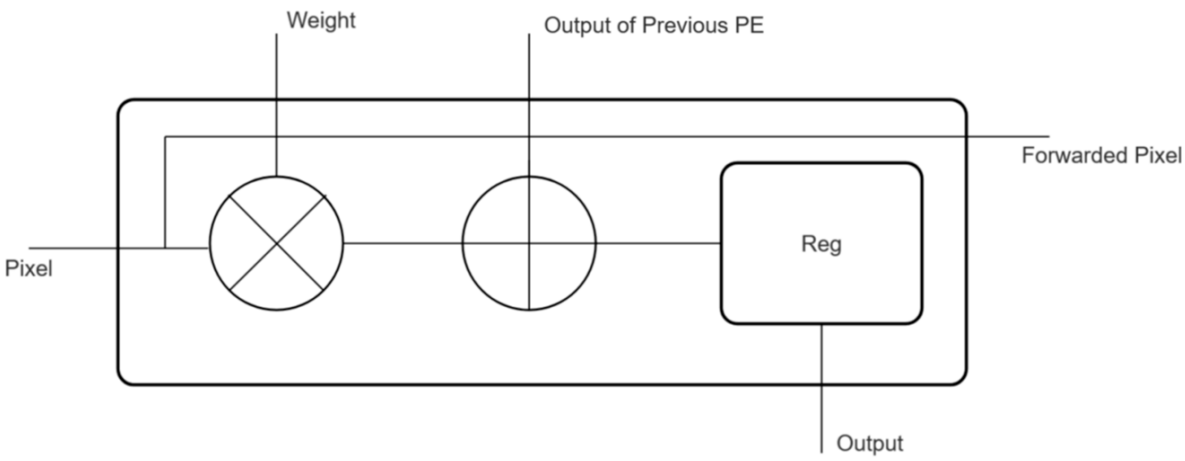

Figure 4. The architecture of a single Processing Element (PE).

3.3. Experimental Design

All experiments were conducted on a Xilinx VC707 FPGA card with a XC7VX485T-2FFG1761 FPGA chip. It has 485,760 logic cells, 8175 Kb distributed RAM, and 37,080 Kb Block RAM. The 32-bit IQ Math format was used to represent floating numbers. The number of resources used by the chaotic oscillator is given in Table 1.

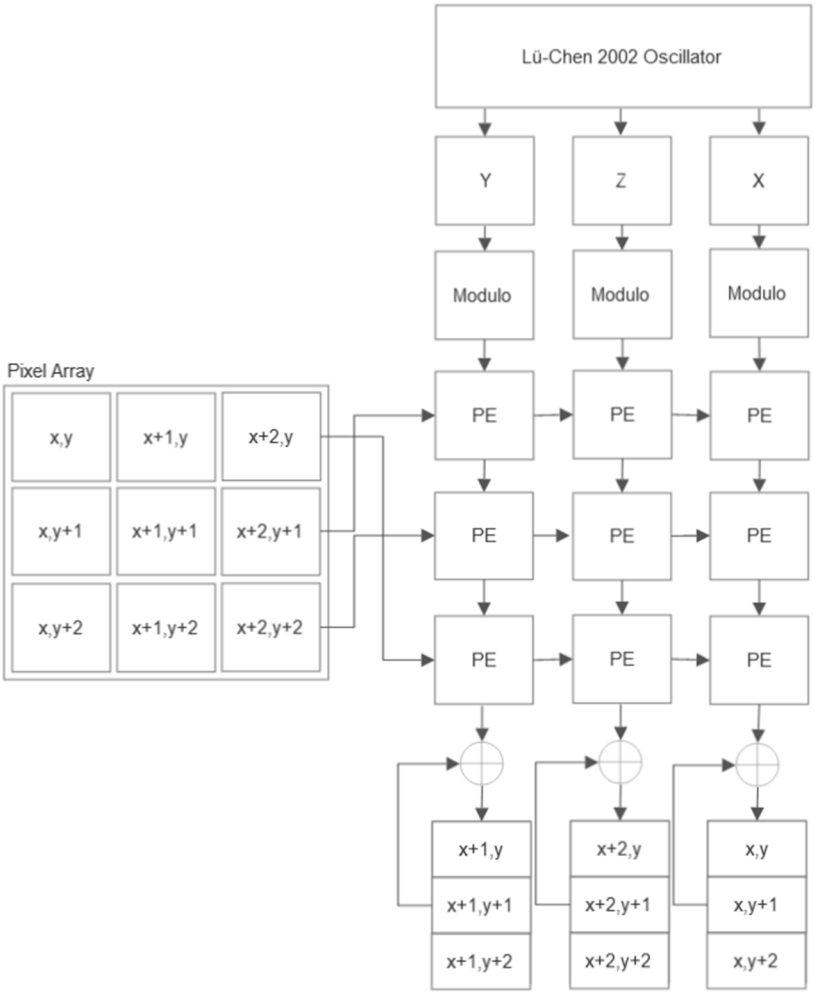

Figure 5. Proposed system architecture.

Table 1. Resources that are used by chaotic oscillators.

Number of Slice Registers	Number of Slice LUTs	Number of DSP 48E1s
2243	1924	64

As shown in Table 1, resource utilization of the oscillator is minimal (3% LUT utilization) compared to the available resources. The number of resources used by the 300 × 300 systolic array architecture is given in Table 2.

Table 2. Resources that are used by systolic arrays.

Block RAM (Mb)	Number of Slice LUTs	Number of DSP 48E1s
1.12	38,273	63

Since Block RAMs are used instead of distributed RAM, Block RAM usage is given in Table 2. Resources used by the complete system are given in Table 3.

Table 3. Resources that are used by the complete system.

Block RAM (Mb)	Number of Slice Registers	Number of Slice LUTs	Number of DSP 48E1s	Max Frequency (MHz)
1.12	2371	38,273	63	573

It takes 129,600 clock cycles to process a 300 × 300 image. It is possible to process 4421 images per second using a single block. This work used up to five sequential blocks since achieving higher-level security with less resource consumption in five layers is possible. The proposed architecture can process 276 images per second.

4. Results

To demonstrate the proposed method, it was used to encrypt three images, and the results were evaluated using the Correlation Coefficient Test (CCT)—to understand the degree of similarity of adjacent pixels—and the Entropy test.

4.1. Resulting Images

Three images were selected and encrypted to evaluate the proposed approach. Since it is possible to build sequential models, up to five blocks were used to encrypt the images, and the results of each step are shown in Figures 6–8.

4.2. Histogram Analysis

It is possible to create an image's histogram by using intensity values. In this study, 8-bit grayscale images are used so that there are 256 possible intensity values. If the distribution of values is fairly uniform in the encrypted image, it is possible to conclude that the proposed encryption method can successfully encrypt the image.

Histograms of non-encrypted and encrypted images of Lena are shared in Figure 9.

Histograms of the non-encrypted and encrypted images of peppers are shared in Figure 10.

Histograms of the non-encrypted and encrypted images of a raccoon are shared in Figure 11.

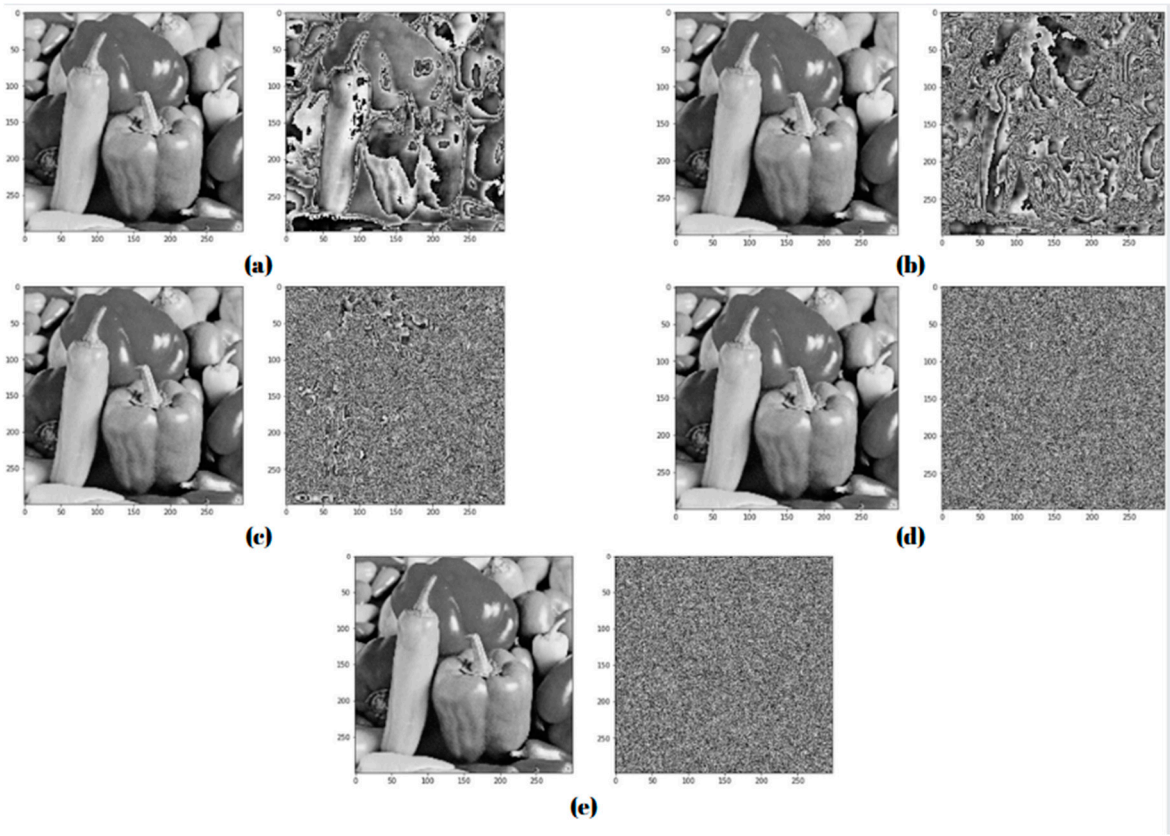

Figure 6. This image of peppers was encrypted using the proposed architecture with (**a**) single blocks, (**b**) two sequential blocks, (**c**) three sequential blocks, (**d**) four sequential blocks, and (**e**) five sequential blocks.

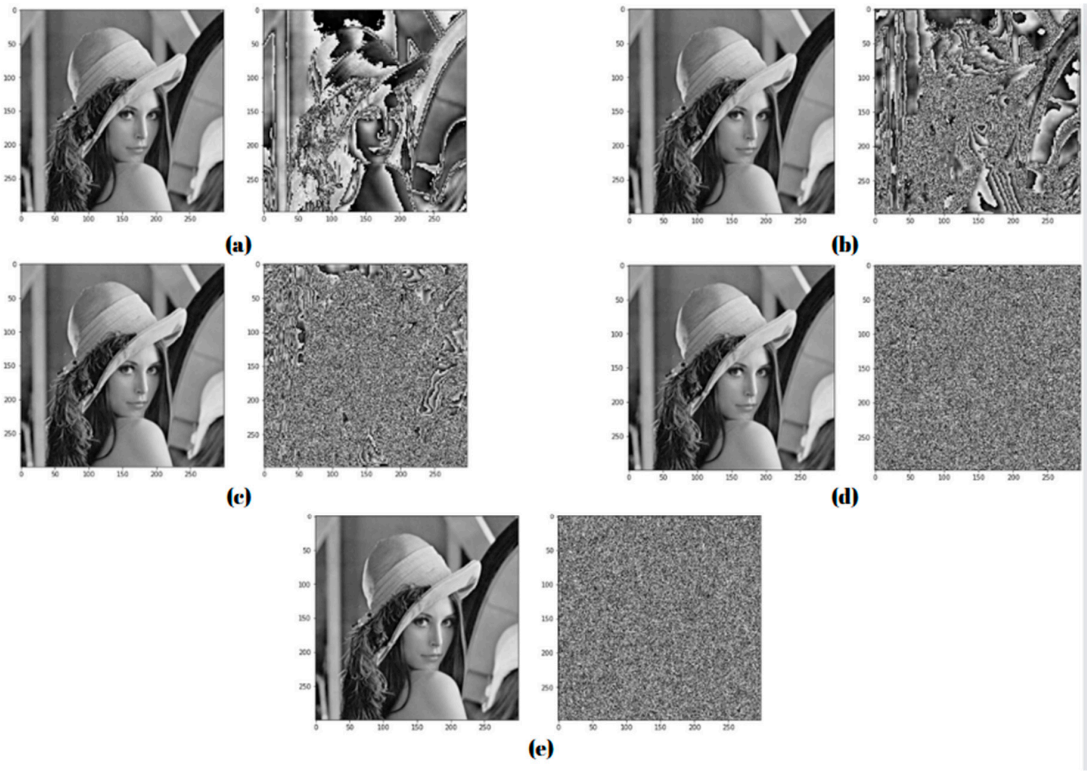

Figure 7. This image of Lena was encrypted using the proposed architecture with (**a**) single blocks, (**b**) two sequential blocks, (**c**) three sequential blocks, (**d**) four sequential blocks, and (**e**) five sequential blocks.

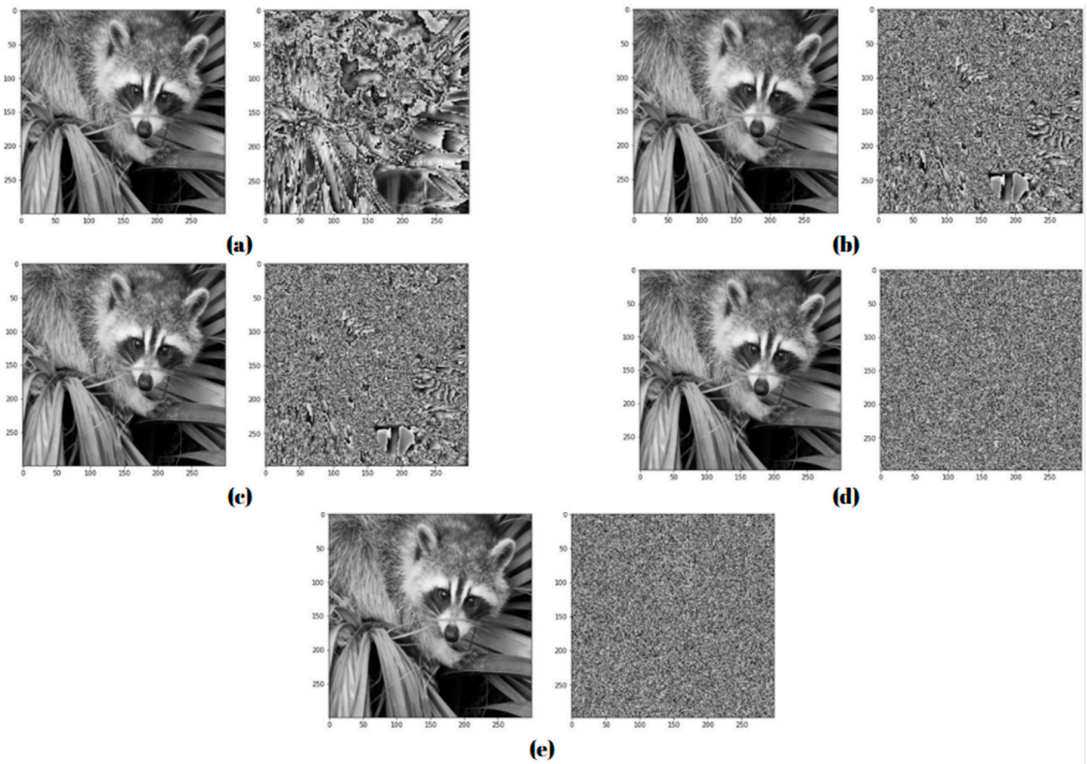

Figure 8. This image of a raccoon was encrypted using the proposed architecture with (**a**) single blocks, (**b**) two sequential blocks, (**c**) three sequential blocks, (**d**) four sequential blocks, and (**e**) five sequential blocks.

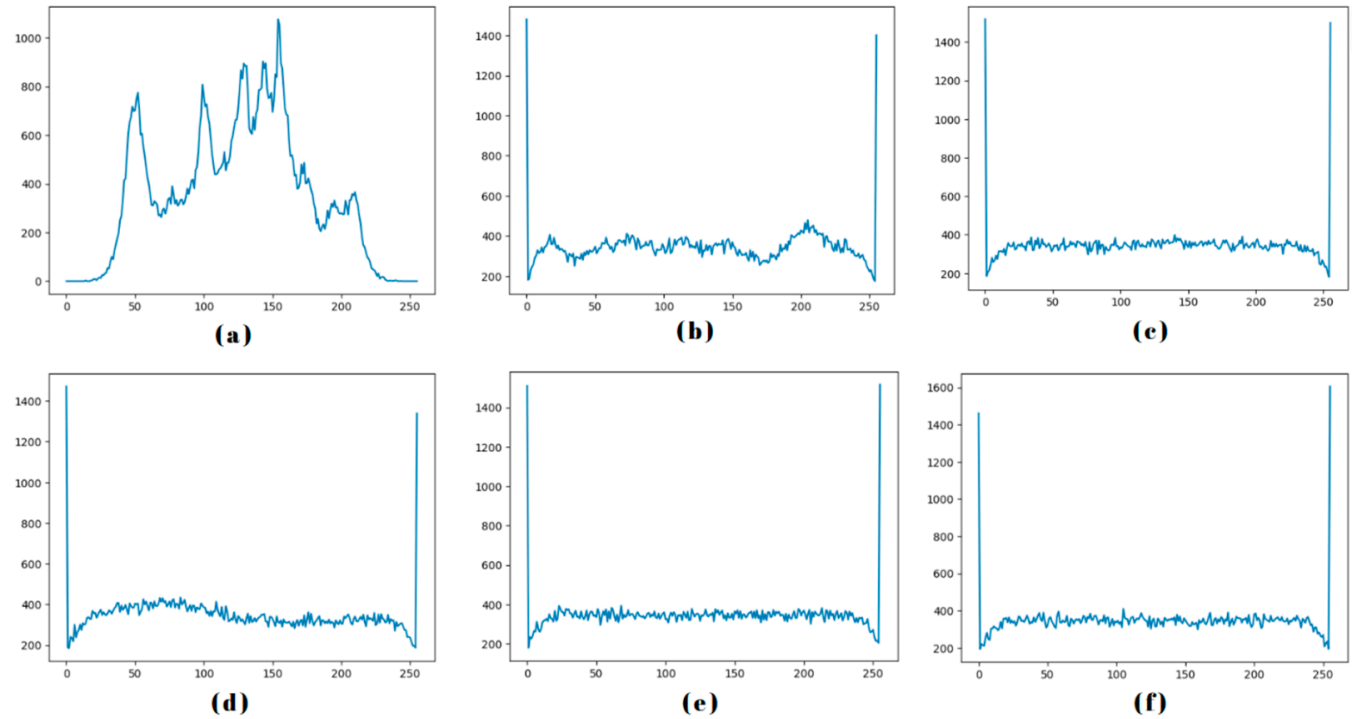

Figure 9. Histogram of the non-encrypted Lena image (**a**), encrypted with one layer (**b**), encrypted with two layers (**c**), encrypted with three layers (**d**), encrypted with four layers (**e**), and encrypted with five layers (**f**).

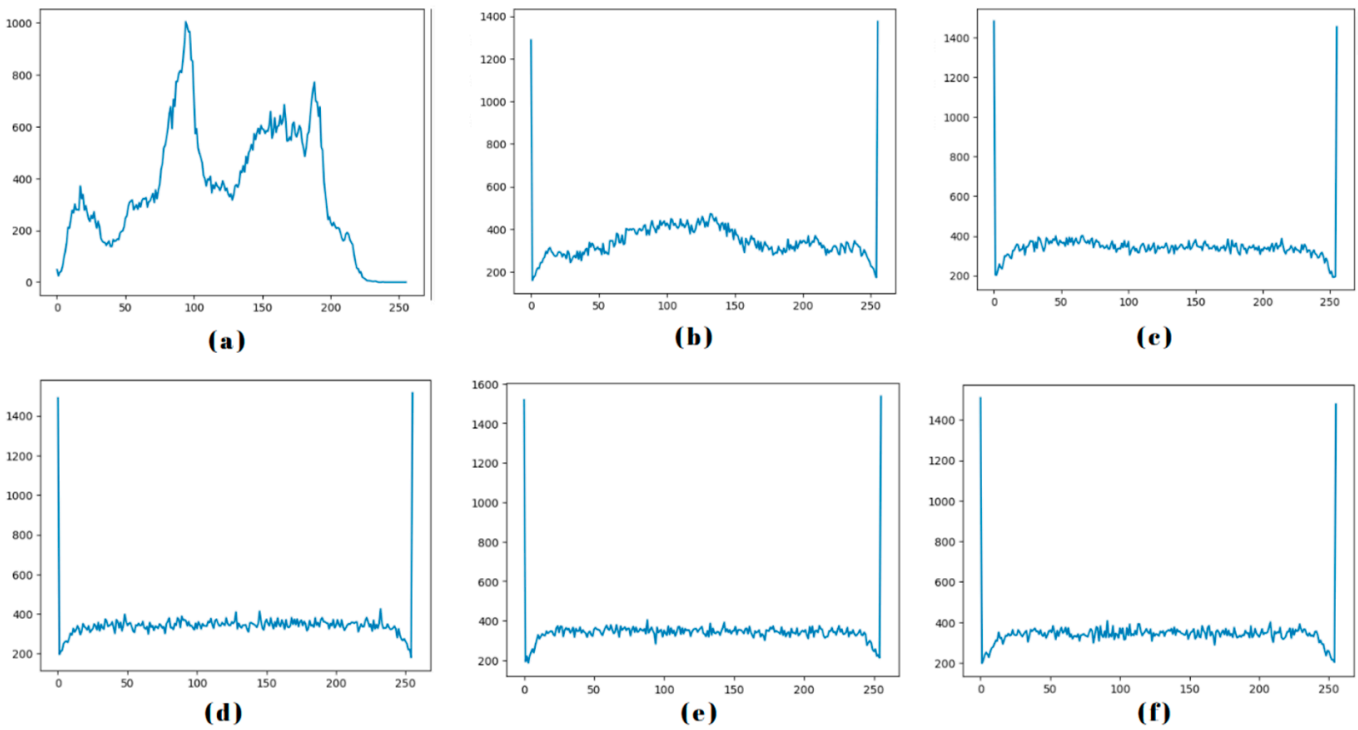

Figure 10. Histogram of the non-encrypted image of peppers (**a**), encrypted with one layer (**b**), encrypted with two layers (**c**), encrypted with three layers (**d**), encrypted with four layers (**e**), and encrypted with five layers (**f**).

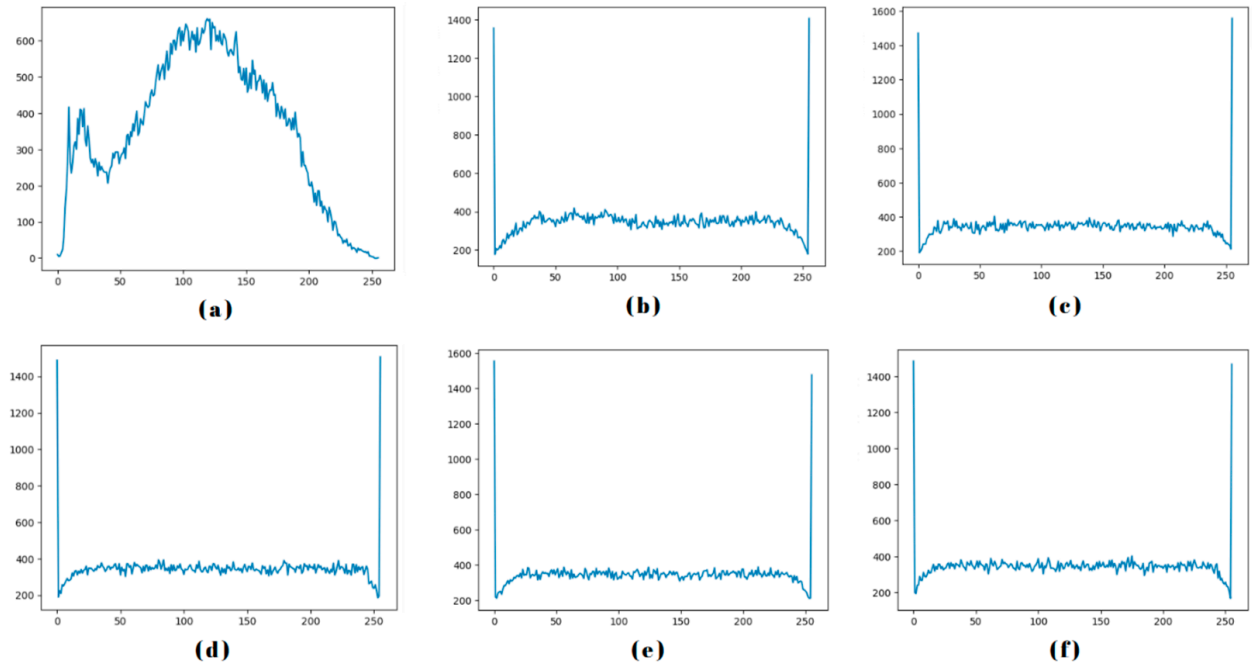

Figure 11. Histogram of the non-encrypted raccoon image (**a**), encrypted with one layer (**b**), encrypted with two layers (**c**), encrypted with three layers (**d**), encrypted with four layers (**e**), and encrypted with five layers (**f**).

As shown in Figures 9–11, the histograms of the encrypted images are uniformly distributed and different from the original image. It can also be seen that the histograms change after the addition of each new encryption layer. These almost-random histograms show that the proposed method can successfully encrypt grayscale images.

4.3. Correlation Coefficient Test

CCT is an important method of statistical analysis used to find the success rate of an image encryption method. It measures the degree of similarity between adjacent pixels; ideally, dependency should not exist between them, and the correlation coefficient should be near 0. CCT is calculated with Equation (3).

$$CC = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2 \times \sum_{i=1}^{N}(y_i - \overline{y})^2}} \tag{3}$$

In this equation, the number of pixels is represented by N; x_i and y_i represent the x and y coordinates of a pixel, respectively; and $\overline{x}$ and $\overline{y}$ represent the coordinates of adjacent pixels.

The results of the CCT for the output of each sequential block when encrypting the pepper image are shown in Table 4.

Table 4. CCT results for the pepper image.

Image	Layer	CCT Horizontal	CCT Vertical	CCT Diagonal
Peppers	1	0.6570	0.6880	0.5669
Peppers	2	0.2994	0.3837	0.2229
Peppers	3	0.0489	0.0936	0.0059
Peppers	4	−0.0093	0.0135	0.0064
Peppers	5	−0.0029	−0.0089	−0.0073

The results of the CCT for the output of each sequential block when encrypting Lena's image are shown in Table 5.

Table 5. CCT results for the image of Lena.

Image	Layer	CCT Horizontal	CCT Vertical	CCT Diagonal
Lena	1	0.5620	0.7273	0.5484
Lena	2	0.3035	0.4431	0.2559
Lena	3	0.0950	0.1804	0.0137
Lena	4	−0.004	0.0230	−0.0019
Lena	5	−0.0069	0.0151	−0.0006

The results of the CCT for the output of each sequential block when encrypting the raccoon image are shown in Table 6.

Table 6. CCT results for the raccoon image.

Image	Layer	CCT Horizontal	CCT Vertical	CCT Diagonal
Raccoon	1	0.4547	0.4883	0.3951
Raccoon	2	0.1110	0.1353	0.0780
Raccoon	3	0.0362	−0.0050	0.0111
Raccoon	4	0.0019	0.0042	0.0028
Raccoon	5	0.0056	0.0042	−0.0091

As shown in Tables 4–6, the correlation between adjacent pixels in an encrypted image decreases. Thus, it is possible to state that the proposed approach can successfully break the correlation between adjacent pixels.

4.4. Image Entropy Analysis

It is possible to use image entropy analysis to measure the unpredictability of an image. Shannon introduced this concept in [39]. It tests whether or not an image has a random distribution of pixel values. The mathematical expression of the method is given in Equation (4):

$$H(K) = \sum_{i=0}^{r-1} P(K_i) log_2 \frac{1}{P(K_i)} \tag{4}$$

where K_i represents the pixel values, $P(K_i)$ represents the probability that a pixel value is found, and r represents the number of symbols (set to 256 for grayscale images). If a grayscale image has pixel values with equal probabilities, $K = (K_0, K_1, K_2, K_3 \ldots K_{255})$, its entropy value is set to 8. In an image that has not been encrypted yet, there is a correlation between pixel values; they are seldom random, and their entropy is generally smaller than 8. If the entropy reaches 8, it can be stated that it has reached the maximum ideal value, and all pixels are distributed randomly.

The results of entropy analysis for the output of each sequential block for the pepper image are shown in Table 7.

Table 7. Entropy results for the pepper image.

Image	Layer	Entropy
Peppers	not encrypted	7.5948
Peppers	1	7.9815
Peppers	2	7.9966
Peppers	3	7.9980
Peppers	4	7.9978
Peppers	5	7.9975

The results of entropy analysis for the output of each sequential block for the image of Lena are shown in Table 8.

Table 8. Entropy results for the image of Lena.

Image	Layer	CCT Horizontal
Lena	not encrypted	7.4391
Lena	1	7.9900
Lena	2	7.9920
Lena	3	7.9978
Lena	4	7.9979
Lena	5	7.9977

The results of entropy analysis for the output of each sequential block for the raccoon image are shown in Table 9.

Table 9. Entropy results for the raccoon image.

Image	Layer	CCT Horizontal
Raccoon	not encrypted	7.6972
Raccoon	1	7.9957
Raccoon	2	7.9977
Raccoon	3	7.9975
Raccoon	4	7.9980
Raccoon	5	7.9980

As shown in Tables 7–9, the entropy of the image approaches the ideal level when encrypted using the proposed method. Thus, it is possible to state that the proposed approach is successful.

4.5. Diffusion Analysis Test

We conducted Rate of Change of the Number of Pixels (NPCR) and Unified Mean Intensity of Change (UACI) tests to measure the resistance against differential attacks. Values between 33.3115 and 33.6156 are considered good for the UACI. Additionally, values between 99.3082 and 99.5906 are considered good for the NPCR, depending on the image size [40]. The NPCR metric can be expressed as Equation (5), and the UACI metric can be expressed as Equation (6):

$$NPCR(\%) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(D(i,j))}{m \times n} \times 100 \tag{5}$$

$$UACI(\%) = \frac{1}{m \times n}[\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\mid I_1(i,j) - CI_2(i,j) \mid}{255}] \times 100 \tag{6}$$

where m and n represent the number of rows and columns of the image, respectively, and are set to 256 in a grayscale image. $D(i,j) = \begin{cases} 1, siCI_1(i,j) \neq CI_2(i,j) \\ 0, siCI_1(i,j) = CI_2(i,j) \end{cases}, CI_1(i,j)\text{etCI}(i,j)$ are the elements at positions i and j of the ciphers of images I_1 and I_2.

The results of the NPCR test for the output of each sequential block for the pepper image are shown in Table 10.

Table 10. NPCR results for the pepper image.

Image	Layer	NPCR Result
Peppers	1	99.51%
Peppers	2	99.60%
Peppers	3	99.62%
Peppers	4	99.59%
Peppers	5	99.63%

The results of the UACI test for the output of each sequential block for the pepper image are shown in Table 11.

Table 11. UACI results for the pepper image.

Image	Layer	UACI Result
Peppers	1	23.28%
Peppers	2	25.19%
Peppers	3	25.30%
Peppers	4	25.40%
Peppers	5	25.44%

The results of the NPCR test for the output of each sequential block for the image of Lena are shown in Table 12.

Table 12. NPCR results for the image of Lena.

Image	Layer	NPCR Result
Lena	1	99.72%
Lena	2	99.63%
Lena	3	99.62%
Lena	4	99.58%
Lena	5	99.60%

The results of the UACI test for the output of each sequential block for the image of Lena are shown in Table 13.

Table 13. UACI results for the image of Lena.

Image	Layer	UACI Result
Lena	1	27.10%
Lena	2	24.16%
Lena	3	24.44%
Lena	4	24.59%
Lena	5	24.62%

The results of the NPCR test for the output of each sequential block for the raccoon image are shown in Table 14.

Table 14. NPCR results for the raccoon image.

Image	Layer	NPCR Result
Raccoon	1	99.61%
Raccoon	2	99.60%
Raccoon	3	99.63%
Raccoon	4	99.61%
Raccoon	5	99.59%

The results of the UACI test for the output of each sequential block for the raccoon image are shown in Table 15.

Table 15. UACI results for the raccoon image.

Image	Layer	UACI Result
Raccoon	1	24.31%
Raccoon	2	25.32%
Raccoon	3	25.47%
Raccoon	4	25.43%
Raccoon	5	25.35%

The results of the NPCR and UACI tests prove that the proposed method can generate encrypted images prone to differential attacks.

5. Discussion

Based on the conducted experiments, the proposed approach can successfully encrypt images by highly increasing entropy and decreasing correlation. The NPCR and UACI results show that the proposed method is prone to diffusion attacks. The performance of the system increases with the number of sequential blocks used. It might be possible to increase the system's overall performance in images with more than one dimension (such as RGB images) using a 3D diffusion approach.

The results of the proposed method are compared with other methodologies in Table 16.

Table 16. UACI results for the raccoon image.

Image	Work	NPCR	UACI	Horizontal Correlation	Vertical Correlation	Diagonal Correlation	Entropy
Peppers	[41]	99.61	33.44	0.0139	0.0054	0.0153	7.9992
Lena	[42]	99.59	33.27	−0.0003	−0.0037	0.0020	7.9971
Peppers	[42]	99.64	33.49	−0.0002	0.0020	0.0048	7.9972
Lena	[43]	99.61	33.46	0.0467	−0.0173	−0.0078	7.9970
Peppers	[43]	99.61	33.46	0.0052	−0.0028	−0.0143	7.9971
Lena	[44]	99.65	33.43	0.0016	-	-	7.9978
Peppers	[44]	99.66	33.36	0.0018	-	-	7.9345
Lena	[45]	99.64	33.47	0.0027	0.0012	0.0003	7.9974
Peppers	[45]	99.64	33.59	0.0020	0.0080	0.0008	7.9972
Lena	[46]	99.56	33.50	0.0004	0.0013	−0.0023	7.9978
Lena	[47]	99.60	33.63	−0.0006	−0.0057	0.0009	-
Lena	[48]	49.83	25.01	0.0118	0.0002	0.0148	-
Lena	[49]	-	-	0.0063	0.0056	0.0034	
Lena	Proposed Work (Layer 5)	99.60	24.62	−0.0069	0.0151	−0.0006	7.9977
Peppers	Proposed Work (Layer 5)	99.63	25.44	−0.0029	−0.0089	−0.0073	7.9975

Table 16 shows that the proposed method gives results competitive with state-of-the-art work conducted in this field. The resulting UACI score is lower than the optimal level. Similar results have also been observed in [48]. Both works share the same chaotic oscillator. Thus, it is possible to conclude that the Lü–Chen chaotic oscillator can result in lower UACI levels.

The primary objective of this paper was to decrease the time required to encrypt images for streaming applications by using the parallel processing capabilities of FPGAs and systolic array architectures. To provide a comprehensive analysis, Table 17 compares the times needed to encrypt data when using different methodologies.

Table 17. Encryption times of different methodologies.

Work	Platform	Encryption Time (s)
[41]	MATLAB R2017b, CPU: AMD Ryzen 7 1700, Memory: 8 GB, OS: Windows 10.	0.3949
[42]	FPGA: Zybo Z20 development board equipped with ZYNQ clocked at 666.67 MHz	0.0850
[43]	CPU: Core i7-4720HQ, Memory: 8 GB, OS: Windows10	0.015
[49]	CPU: 2.42 GHz, Memory: 8 GB, OS: Windows 10	1.57
Proposed Approach (5 Layers)	FPGA: VC707 with XC7VX485T-2FFG1761 clocked at 573 MHz	0.0036

It is possible to continue the research by implementing new nonlinear chaotic map approaches, such as the one proposed in [49], on the proposed architecture to improve the

system's security. Another avenue to explore is using fuzzy modeling to model chaotic behavior, as in [50], to decrease the overall computational complexity of the system.

The proposed approach gives state-of-the-art encryption time results and better performance than all other proposed methods. The time required to decrypt the image has not been shared, as it is not realized on a FPGA. This research can be furthered by realizing the required deconvolution process to decrypt the image on a FPGA using a systolic array approach.

In this work, a 2×2 diffusion method was used. It is possible to change this to obtain different results in future research.

The current study used a 32-bit IQ Math approach to represent the oscillator's output. It is possible to use a 64-bit floating point representation to improve security or a 16-bit float type to decrease resource consumption while measuring how the level of security is preserved.

6. Conclusions

This research proposes a FPGA-based chaotic image encryption method that uses systolic arrays. To achieve this, the Lü–Chen chaotic oscillator generates weights for the convolutional filter. Then, the image is fed to the systolic array architecture. The most resource-consuming part of the encryption process is the diffusion process. The confusion and diffusion processes are fused to decrease overall resource consumption and create a fully systolic array-based architecture.

We conducted experiments to show that the proposed method can produce promising results. It can perform very high-speed encryption processes and has a comparable level of security compared to other methods.

The proposed method can achieve higher encryption speeds compared to the other methodologies. Along with this improvement, it may become possible to use chaotic encryption methodologies in low-latency encrypted video streaming applications.

It is possible to increase the number of sequential encryption layers to add an extra layer of protection to the system. Still, the resources of the FPGA are limited, and the total time required for both encryption and decryption processes increases with each new encryption layer.

It is possible to further this research by changing the chaotic oscillator used to achieve higher levels of security. A systolic array-based decryption architecture can also be developed to decrease the total time needed to decrypt the system's output.

Author Contributions: Conceptualization, F.C.; methodology, F.C. and B.C.; software, F.C.; validation, B.C. and M.A.; formal analysis, M.A.; investigation, B.C.; resources, M.A.; writing—original draft preparation, F.C.; writing—review and editing, B.C.; visualization, F.C.; supervision, M.A.; project administration, B.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

References

1. Paar, C.; Pelzl, J. Introduction to cryptography and data security. In *Understanding Cryptography: A Textbook for Students and Practitioners*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–27.
2. Jonathan, K.; Yehuda, L. *Introduction to Modern Cryptography*, 2nd ed.; 2023 Submitted to Journal Not Specified 15 of 16; Version April 27; CRC Press: Boca Raton, FL, USA, 2014.
3. Amara, M.; Siad, A. Elliptic Curve Cryptography and Its Applications. In *International Workshop on Systems, Signal Processing and Their Applications*; WOSSPA: Tipaza, Algeria, 2011; pp. 247–250. [CrossRef]

4. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
5. Daemen, J.; Rijmen, V. *Advanced Encryption Standard (AES)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001. [CrossRef]
6. Zhang, G.; Liu, Q. A novel image encryption method based on total shuffling scheme. *Opt. Commun.* **2011**, *284*, 2775–2780. [CrossRef]
7. Acharya, A.K. Image encryption using a new chaos based encryption algorithm. In Proceedings of the 2011 International Conference on Communication, Computing & Security, New York, NY, USA, 12 February 2011.
8. Sadoudi, S.; Azzaz, M.S.; Djeddou, M.; Benssalah, M. An FPGA real-time implementation of the Chen's chaotic system for securing chaotic communications. *Int. J. Nonlinear Sci.* **2009**, *7*, 1749–3889.
9. Tuna, M.; Alcın, M.; Koyuncu, I.; Fidan, C.; Pehlivan, I. High speed FPGA-based chaotic oscillator design. *Microprocess. Microsyst.* **2019**, *66*, 72–80. [CrossRef]
10. Holmes, P. Poincare, celestial mechanics, dynamical-systems theory and chaos. *Phys. Rep.* **1990**, *193*, 137–163. [CrossRef]
11. Lorenz, E.N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [CrossRef]
12. Li, T.-Y.; Yorke, J.A. Period three implies chaos. *Am. Math. Mon.* **1975**, *82*, 985–992. [CrossRef]
13. Matsumoto, T.; Chua, L.O.; Tanaka, S. Simplest chaotic nonautonomous circuit. *Phys. Rev. A* **1984**, *30*, 1155–1157. [CrossRef]
14. Matsumoto, T. A chaotic attractor from Chua's circuit. *IEEE Trans. Circuits Syst.* **1984**, *31*, 1055–1058. [CrossRef]
15. Tlelo-Cuautle, E.; Munoz-Pacheco, J.M. Simulation of Chua's circuit by automatic control of step-size. *Appl. Math. Comput.* **2007**, *190*, 1526–1533. [CrossRef]
16. Odibat, Z.; Corson, N.; Aziz-Alaoui, M.A.; Alsaedi, A. Chaos in fractional order cubic Chua system and synchronization. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750161. [CrossRef]
17. Zhang, Y.; Wang, Y.; Shen, X. A chaos-based image encryption algo-rithm using alternate structure. *Sci. China Ser. F Inf. Sci.* **2007**, *50*, 334–341. [CrossRef]
18. Chirikov, B.V. A universal instability of many-dimensional oscillator systems. *Phys. Rep.* **1979**, *52*, 263–379. [CrossRef]
19. Arnold, V.I.; Avez, A. *Problemes Ergodiques de la Mecanique Classique*; Gauthier-Villars France: Villiers-sur-Orge, France, 1967.
20. Yavuz, E.; Yazici, R.; Kasapbasi, M.C.; Yamac, E. A chaos-based image encryption algorithm with simple logical functions. *Comput. Electr. Eng.* **2016**, *54*, 471–483. [CrossRef]
21. Francois, M.; Grosges, T.; Barchiesi, D.; Erra, R. Image encryption algorithm based on a chaotic iterative process. *Appl. Math.* **2012**, *3*, 1910–1920. [CrossRef]
22. Wong, K.-W.; Kwok, B.S.-H.; Yuen, C.-H. An efficient diffusion approach for chaos-based image encryption. *Chaos Solitons Fractals* **2009**, *41*, 2652–2663. [CrossRef]
23. Zhang, X.; Zhao, Z. Chaos-based image encryption with total shuffling and bidirectional diffusion. *Nonlinear Dyn.* **2014**, *75*, 319–330. [CrossRef]
24. Chen, J.-X.; Zhu, Z.-L.; Yu, H. A fast chaos-based symmetric image cryptosystem with an improved diffusion scheme. *Optik-Int. J. Light Electron Opt.* **2014**, *125*, 2472–2478. [CrossRef]
25. Wang, X.; Liu, L.; Zhang, Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt. Lasers Eng.* **2015**, *66*, 10–18. [CrossRef]
26. Koyuncu, I.; Ozcerit, A.T.; Pehlivan, I. An analog circuit design and FP-GA based implementation of the Burke-Shaw chaotic system. *Optoelectron. Adv. Mater.-Rapıd Commun.* **2013**, *7*, 635–638.
27. Alcın, M.; Pehlivan, I.; Koyuncu, I. Hardware design and implementation of a novel ANN-based chaotic generator in FPGA. *Opt.—Int. J. Light Electron Opt.* **2016**, *127*, 5500–5505. [CrossRef]
28. Koyuncu, I.; Ozcerit, A.T.; Pehlivan, I. Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dyn.* **2014**, *77*, 49–59. [CrossRef]
29. Tuna, M.; Fidan, C.B. Electronic circuit design, implementation and FPGA-based realization of a new 3D chaotic system with single equilibrium point. *Opt.—Int. J. Light Electron Opt.* **2016**, *127*, 11786–11799. [CrossRef]
30. Akgul, A.; Calgan, H.; Koyuncu, I.; Pehlivan, I.; Istanbullu, A. Chaos-based engineering applications with a 3D chaotic system without equilibrium points. *Nonlinear Dyn.* **2015**, *84*, 481–495. [CrossRef]
31. Rajagopal, K.; Karthikeyan, A.; Srinivasan, A.K. FPGA implementation of novel fractional-order chaotic systems with two equiliriums and no equilibrium and its adaptive sliding mode synchronization. *Nonlinear Dyn.* **2017**, *87*, 2281–2304. [CrossRef]
32. Lai, Q.; Zhao, X.-W.; Rajagopal, K.; Xu, G.; Akgul, A.; Guleryuz, E. Dynamic analyses, FPGA implementation and engineering applications of multi-butterfly chaotic attractors generated from generalised Sprott C system. *Pramana* **2018**, *90*, 6. [CrossRef]
33. Tlelo-Cuautle, E.; Pano-Azucena, A.D.; Rangel-Magdaleno, J.J.; Carbajal-Gomez, V.H.; Rodriguez-Gomez, G. Generating a 50-scroll chaotic attractor at 66 MHz by using FPGAs. *Nonlinear Dyn.* **2016**, *85*, 2143–2157. [CrossRef]
34. Rajagopal, K.; Akgul, A.; Jafari, S.; Karthikeyan, A.; Koyuncu, I. Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses. *Chaos Solitons Fractals* **2017**, *103*, 476–487. [CrossRef]
35. Sambas, A.; Vaidyanathan, S.; Zhang, X.; Koyuncu, I.; Bonny, T.; Tuna, M.; Kumam, P. A Novel 3D Chaotic System with Line Equilibrium: Multistability, Integral Sliding Mode Control, Electronic Circuit, FPGA Implementation and Its Image Encryption. *IEEE Access* **2022**, *10*, 68057–68074. [CrossRef]
36. Vanecek, A.; Celikovsky, S. *Control Systems: From Linear Analysis to Synthesis of Chaos*; Prentice-Hall: London, UK, 1996.
37. Lu, J.; Chen, G. A new chaotic attractor coined. *Int. J. Bifurc. Chaos* **2002**, *12*, 659–661. [CrossRef]

38. Bagavathi, C.; Saraniya, O. Chapter 13—Evolutionary Mapping Techniques for Systolic Computing System. In *Deep Learning and Parallel Computing Environment for Bioengineering Systems*; Academic Press: Cambridge, MA, USA, 2019; pp. 207–223. [CrossRef]
39. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
40. Wu, Y.; Member, S.; Noonan, J.P.; Member, L. NPCR and UACI Randomness Tests for Image Encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun.* **2011**, *2011*, 31–38.
41. Ye, G.; Wu, H.; Jiao, K.; Mei, D. Asymmetric image encryption scheme based on the Quantum logistic map and cyclic modulo diffusion. *Math. Biosci. Eng.* **2021**, *18*, 5427–5448. [CrossRef]
42. Maazouz, M.; Toubal, A.; Bengherbia, B.; Houhou, O.; Batel, N. FPGA implementation of a chaos-based image encryption algorithm. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 9926–9941. [CrossRef]
43. Zhang, Y. Test and verification of AES used for image encryption. *3D Res.* **2018**, *9*, 3. [CrossRef]
44. Hagras, E.A.A.; Saber, M. Low power and high-speed FPGA implementation for 4D memristor chaotic system for image encryption. *Multimedia Tools Appl.* **2020**, *79*, 23203–23222. [CrossRef]
45. Arab, A.; Rostami, M.J.; Ghavami, B. An image encryption method based on chaos system and AES algorithm. *J. Supercomput.* **2019**, *75*, 6663–6682. [CrossRef]
46. Wang, S.; Wang, C.; Xu, C. An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm. *Opt. Lasers Eng.* **2020**, *128*, 105995. [CrossRef]
47. Kaur, G.; Agarwal, R.; Patidar, V. Chaos based multiple order optical transform for 2D image encryption. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 998–1014. [CrossRef]
48. Chen, G.; Mao, Y.; Chui, C.K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [CrossRef]
49. Abu-Ein, A.A. An Effective Chaotic Image Encryption Algorithm Based on Piecewise Non-linear Chaotic Map. *Inf. Sci. Lett. Nat.* **2023**, *12*, 1173–1181.
50. Babanli, K.; Kabaoğlu, R.O. Fuzzy modeling of desired chaotic behavior in secure communication systems. *Inf. Sci.* **2022**, *594*, 217–232. [CrossRef]