Mesopotamian journal of Cybersecurity
Vol.4,No.3, **pp**. 136–151
DOI: https://doi.org/10.58496/MJCS/2024/019; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Design and Practical Implementation of a Stream Cipher Algorithm Based on a Lorenz System

Hayder Mazin Makki Alibraheemi [1, *], Mazen M. A. Al Ibraheemi [1], Zainb Hassan Radhy [2]

[1] *Department of Electronics and Communication Engineering, Faculty of Engineering, University of Al-Qadisiyah, Iraq*

[2] *College of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq*

ARTICLE INFO

Article history

Received 15 Aug 2024
Accepted 25 Oct 2024
Published 18 Nov 2024

Keywords

Chaos

FPGA

Hardware cosimulation

Image encryption

Lorenz attractor

ABSTRACT

Currently, the security of data has gained significant attention in modern life. Researchers have continued to address this issue. This work addresses image encryption in communication systems. It presents a proposed design and implementation of a cryptography system based on the Lorenz chaos oscillator. The paper methodology uses Xilinx System Generator (XSG) and Field Programmable Gate Array (FPGA) technologies to implement the chaotic system. To determine the approach that uses the least amount of FPGA resources while providing effective and efficient performance, the differential equations of the Lorenz chaotic system are solved via the forward-Euler and Runge–Kutta integration techniques. In the XSG environment, a secure communication system is constructed on the basis of the solution of the differential equations. After that, the planned communication system is implemented on the FPGA board and tested to encrypt images (coloured images). The histogram, entropy and other related security analysis factors are calculated and analysed to test the efficiency of the designed system. Six statistical methods were employed to provide a high level of image encryption in this work. Findings have shown that the proposed system generates (with stable, fast and robust performance) pseudorandom bits that can be successfully used to encrypt the data bits. The simulation and FPGA results are in good agreement; however, the security analysis factors prove that the system can be successfully adopted for image encryption purposes in real-time applications.

1. INTRODUCTION

In modern data handling systems, image encryption has gained a highly significant position in maintaining the security and privacy of image details. Currently, the world has periodically witnessed some form of cybersecurity threat (cyber attacks). To address this problem, some styles of image encryption, such as symmetric encryption, asymmetric encryption, and steganography, have been described in the literature. In general, the encryption essentially involves adding an extra file, an extra layer, to the original image or data to serve as a protector against any unauthorized access. The more extensive encryption method involves converting the data into an unregular form that cannot be easily interpreted. This method of encryption is more secure than conventional security approaches, such as passwords or encrypting individual files [1-3].

Therefore, research topics related to data security and privacy are currently highly emerging. Chaotic systems are nonlinear aperiodic, highly sensitive to initial conditions and system parameters, and random with unpredictable behavior [1]– [5]. On the other hand, chaotic nonlinear systems can still be synchronized. These promising features of chaotic systems have pushed them in a leading position to be widely used in message encryption to provide secure transmission channels [6]–[10]. References [11], [12] have shown that communication systems based on chaotic schemes offer promising features of very high speed and robust performance. These findings have also been confirmed by [13]–[15], who mentioned that the features gained by applying chaotic systems support high-speed applications of communication systems, such as real-time encryption systems. Therefore, the use of chaos phenomena in cryptography algorithms has emerged as a possible solution because of its superior properties.

Moreover, in recent years, many encryption algorithms have been developed, especially those used by researchers for image encryption. Two cryptographic phases are employed in [15] to encrypt the image pixels: diffusion for the bit value in phase two and scrambling the pixel position in phase one. These two phases are accomplished by employing a new chaotic map called Nahrain. Reference [16] applied a Nahrain system using FPGA to encrypt and test colored images. The design and implementation of a chaos-based secure substitution permutation network (SPN) are presented, where the proposed

*Corresponding author. Email: hayder.mazen@qu.edu.iq

cryptosystem is realized via FPGA and software tools C/C++, Altera Quartus II and ModelSim. A chaotic block cipher was used in [17] to encrypt 256 blocks of plain-image bits as part of an image encryption technique. In the hybrid domain, the key space of the encryption system can be increased, and as a result, the security level can be increased. In [18], a combination of chaotic maps and a linear feedback shift register (LFSR) was introduced to increase the security level. The skew tent map with a combination of diffusion and permutation algorithms is used together to produce a robust security system with a large key space in [19] for image encryption purposes. In [20], the sum of product (SOP) encryption technique was used on the basis of Boolean algebra to quickly encrypt plain images. The combination of an advanced encryption system (AES) and a chaotic system was presented in [21], where FPGA implementation of a chaotic-based AES image encryption system was carried out via the Verilog hardware description language (HDL) and Xilinx Design Suite. The image encryption technique used in [22] is based on the use of 3D different chaotic maps, such as the Henon map, logistic map, Baker map, and cross chaos map.

In addition, a comparative analysis is carried out in this paper. The encryption speed of the image cryptosystems that are based on chaos is increased via the use of a dynamic updating lookup table, which effectively reduces the number of iterations of the chaotic systems [23]. A combination of a KAA map with multiple chaotic maps was implemented in [24] for colored image encryption purposes, where two keys are generated separately via a 2D logistic sine map, linear congruential map, tent map and Bernoulli map. Then, the two generated keys are diffused via the KAA map. The authors in [25] used a mix of different chaotic maps, such as Tent, Lozi and logistic, to achieve robust and new bit streams that can be adopted in image encryption applications. A cryptosystem based on a four-dimensional hyperchaotic system was studied and implemented in the MATLAB environment in [13] to encrypt 128x128 colored and grayscale images with satisfactory results.

In this paper, the design, simulation and FPGA implementation of a cryptography system based on chaos theory are presented, where the following list concisely highlights the significance and contributions of this work:

- The stream cipher cryptosystem was developed on the basis of a three-dimensional chaos system.
- An adaptive synchronization is designed to provide the necessary synchronization against the channel disturbances.
- FPGA implementation of the overall cryptosystem is achieved via the PYNQ-Z1 board.
- A comparison between the Runge–Kutta and forward Euler integration techniques is performed to address the optimal method that uses the least amount of FPGA resources with the same accuracy and performance.

Section 2 illustrates the chaos-based algorithm used for encrypting the images. This section also includes the numerical integration methods used for solving the dynamical system of the chaotic generator with their implementation in the XSG environment. The forward Euler method and Runge–Kutta method are both used to solve the ordinary differential equation ODE system for comparison purposes. Section 3 presents the results obtained from the XSG environment with the proposed system analysis and its performance. FPGA implementation and hardware cosimulation are presented in section 4. Finally, section 5 presents the conclusions of this paper.

2. CHAOS-BASED IMAGE ENCRYPTION ALGORITHM

The proposed image encryption system is presented in the block diagram shown in Figure (1).

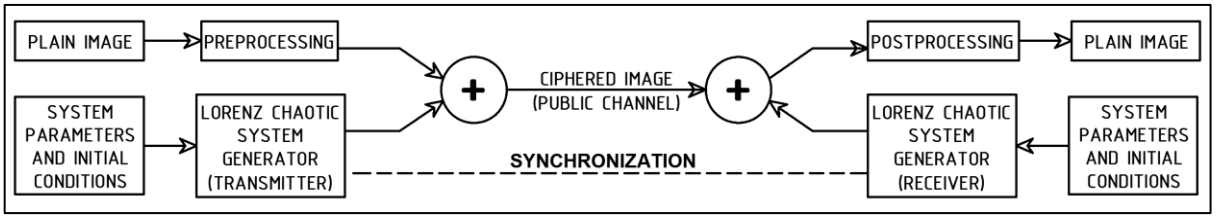

Fig. 1 Image encryption diagram

The encryption system is based on a Lorenz chaotic oscillator with ordinary initial conditions and system parameters. The implementation of the chaotic system is performed via Xilinx system generator blocks, which are configured with 32-bit fixed-point data representation. There are two phases with respect to the proposed scheme, where the first phase is the process of encryption key generation. On the other hand, (and at the same time, the plain image is converted into a bit stream corresponding to the image pixels). The image pixel bit stream is XORed with generated key bits to generate encrypted image pixels that are ready to be transmitted over a public unsecure channel. On the receiver side, the process with the reverse order takes place to recover the plain image pixels. The following subsections illustrate the full design and implementation of the cryptographic system.

2.1 Chaos System Mathematical Description

Mathematically, the chaotic behavior can be obtained by numerically solving the differential equations given in [24]. Those differential equations are adopted two times: the first one is for the master subsystem (transmitter), and the second one is for the slave subsystem (receiver). The chaotic attractor for the master is represented in equations 1 through 3.

$$\frac{dx_m}{dt} = \sigma(y_m - x_m) \tag{1}$$

$$\frac{dy_m}{dt} = \rho x_m - y_m - x_m z_m \tag{2}$$

$$\frac{dz_m}{dt} = x_m y_m - \beta z_m \tag{3}$$

The chaotic attractor for the slave manipulator is represented in equations 4 through 6.

$$\frac{dx_s}{dt} = \sigma(y_s - x_s) \tag{4}$$

$$\frac{dy_s}{dt} = \rho x_s - y_s - x_s z_s \tag{5}$$

$$\frac{dz_s}{dt} = x_s y_s - \beta z_s \tag{6}$$

2.2 Numerical Integration Methods

Traditionally, nonlinear systems are represented and described by using a set of ODEs. Solving this type of equation requires an integration-based process to obtain the solution, and this integration can be written and formulated as the following equation in 7.

$$y(t) = y(i) + \int_i^t f(x)dx\,\pi r^2 \tag{7}$$

where i denotes the initial state of the system and t represents the solution at the required time. Computing the integration of the formula in Equation (7) can be analytic, but practically, this approach will be very difficult and unsolvable. The most common method used to perform this type of integration is numerical integration, such as the forward Euler and Runge–Kutta methods.

2.2.1 Forward Euler Integration Method

The forward Euler integration method is the simplest first-order method with relatively low accuracy in its computations. The general formula for this method is shown in equation 8 [25].

$$y_{t+1} = y_t + hf(x_t, y_t) \tag{8}$$

where yt+1 is the next time variable value, yt is the current time variable, and f (xt, yt) and h represent the interval of the computational time.

2.2.2 Runge Kutta Integration Method

This is the most commonly used integration method for solving ordinary differential equations because of its high accuracy and high stability. This method requires more computational time. The Runge–Kutta method is formulated and described in the following equations in 9 through 13 [26].

$$y_{t+1} = y_t + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{9}$$

$$k_1 = hf(x_t, y_t) \tag{10}$$

$$k_2 = hf(x_t + \frac{h}{2}, y_t + \frac{k_1}{2}) \tag{11}$$

$$k_3 = hf(x_t + \frac{h}{2}, y_t + \frac{k_2}{2}) \quad (12)$$

$$k_4 = hf(x_t + h, y_t + k_3) \quad (13)$$

2.3 Fixed Point Model Implementation

The overall system implementation is described in this subsection, which is divided into three subsections as follows.

2.3.1 Transmitter (Master)/Receiver (Slave) Implementation

The Lorenz chaotic system is designed using the Xilinx system generator XSG with Simulink blocks configured with a 32-bit data format to construct the transmitter and receiver subsystems. The 32-bit fixed-point XSG model is designed by using the Fix32_18 data format, where 1 bit is assigned for a sign, 18 are fractional bits and 13 represent integer bits. The sets of equations in 1 and 2 are solved two times to implement the subsystems. The first method uses forward Euler, and the second method uses the Runge–Kutta integral method. Two solutions are used in this subsection to design the XSG model for the transmitter (master) and receiver (slave) subsystems of the proposed cryptosystem. The XSG model is then used to generate the bit file that will be used later to configure the FPGA board. Figure 2 and Figure 3, shown below, illustrate the 32-bit fixed-point XSG model for the transmitter and receiver subsystems designed with the forward Euler integration method.

On the other hand, the chaotic system is implemented via the fourth-order Runge–Kutta integration method with the same initial conditions and system parameters to determine the most flexible, efficient, and effective solver method for FPGA. Figure 4 illustrates the designed block diagram of the transmitter/receiver subsystems. The design and details of each block are described in Figure 5 and 6. Figure 5 describes the XSG blocks of the K units or parameters that are required to calculate the estimated signal. However, the estimated signal is calculated on the basis of the precalculated K parameters using the XSG blocks, and Figure 6 shows the estimated signal blocks.

The two subsystems are identical and correspond to the blocks, parameters and other coefficients, but the only difference between them is the control signals that appear in the receiver subsystem, which are used mainly to provide the necessary synchronization between the two subsystems.

Fourth-order Runge-Kutta is the widely used integral solver method because of its ease of programming and stability, but this method requires more computational time than the forward-Euler method because of its high accuracy with a low commutative error level.

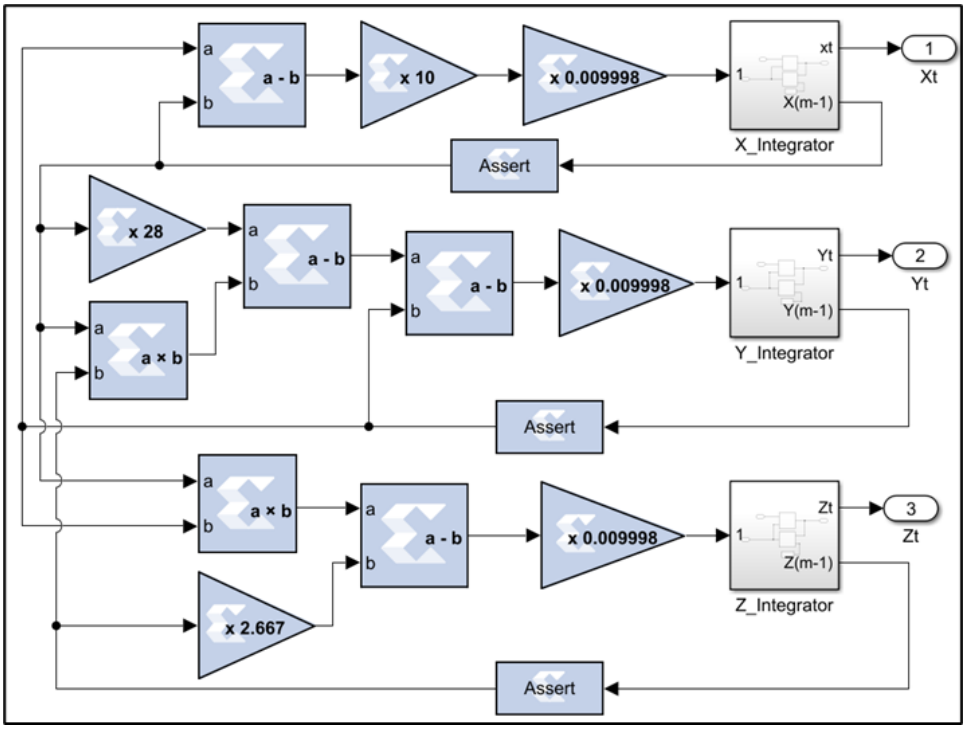

Fig. 2. 32-bit Fixed-point Lorenz Attractors Based on the Forward Euler for the Master Subsystem

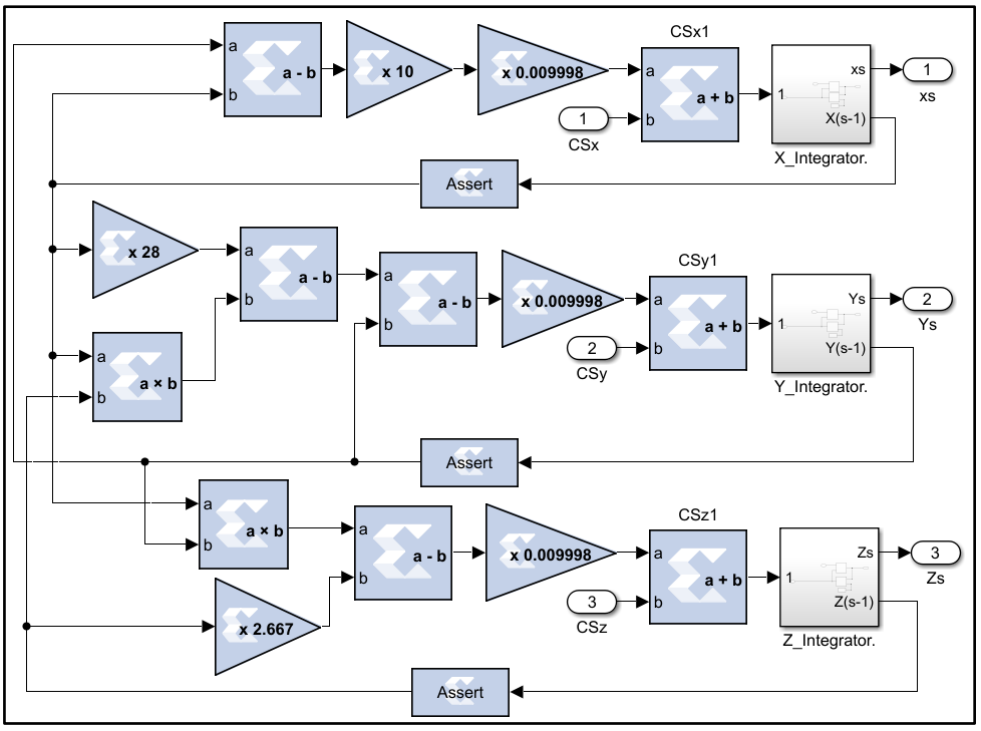

Fig. 3. 32-bit Fixed-point Lorenz Attractors Based on the Forward Euler for the Slave Subsystem

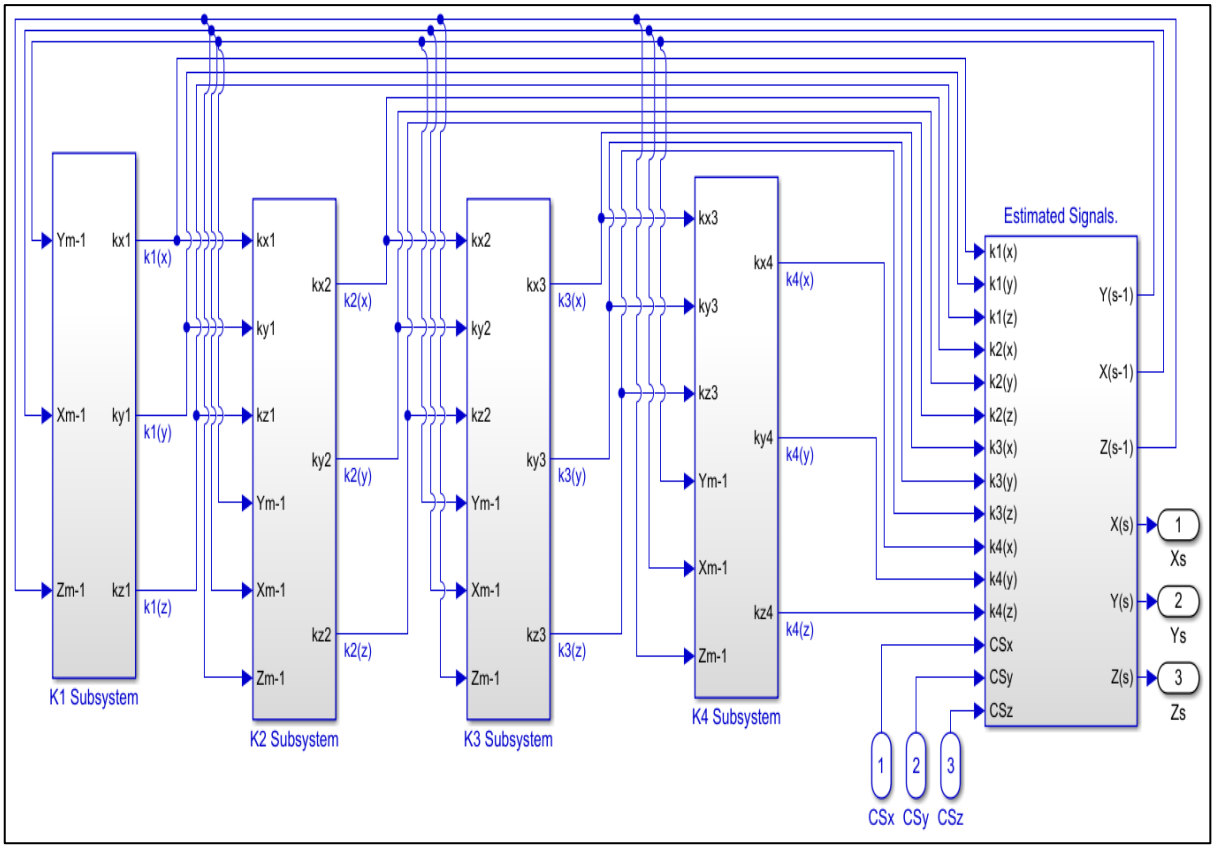

Fig. 4. 32-bit Fixed-point Lorenz Attractors Based on Runge Kutta for the Transmitter/Receiver Subsystem

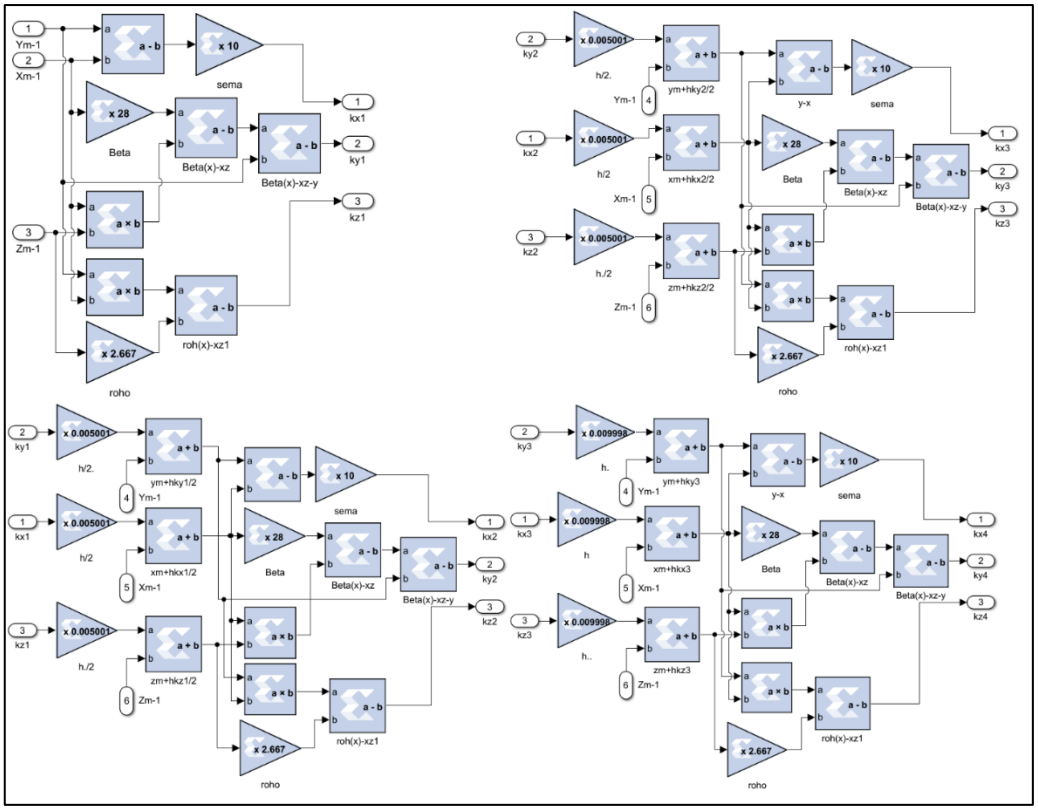

Fig. 5. XSG unit block diagrams of K1, K2, K3, and K4

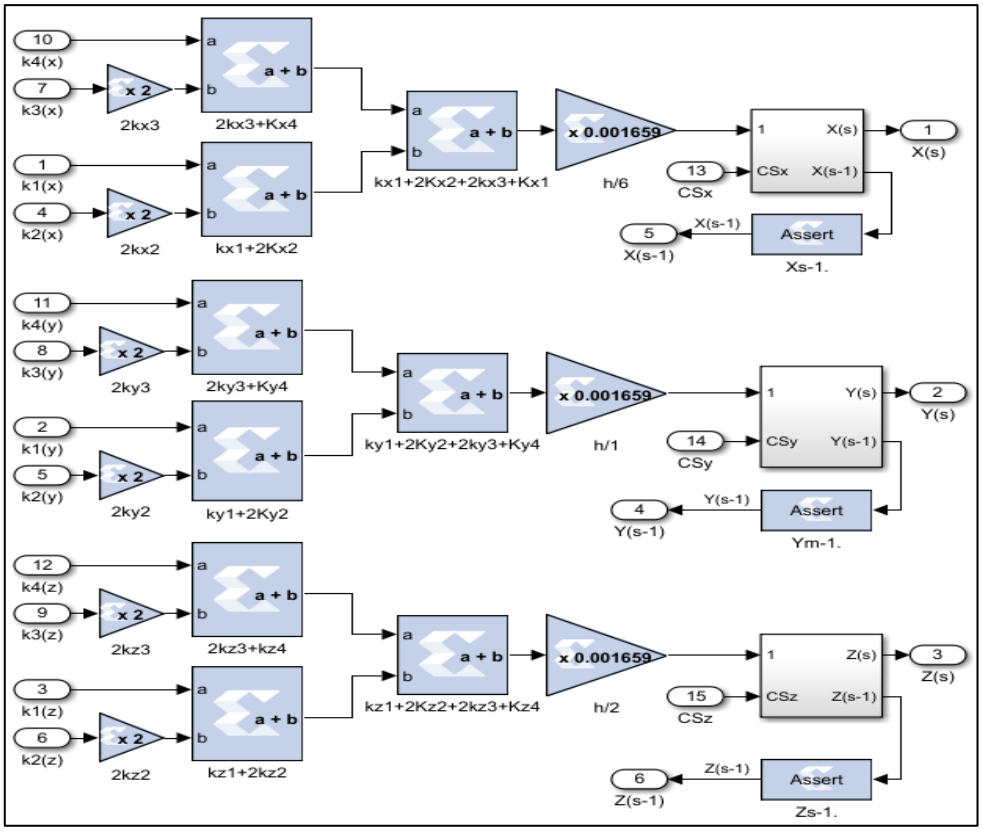

Fig. 6. Estimated signal of the Runge–Kutta method

2.3.2 Transmitter/Receiver Subsystem Synchronization

The synchronization between two components or systems means that the time response (trajectory) of one system or component converges to the same time response of the other system. In terms of chaos and nonlinear systems, this fact was unapplicable because those chaotic systems are aperiodic and have random and unpredictable behavior. In [6], Pecora and Carroll prove that chaotic systems can be synchronized by minimizing the trajectory errors between the master/transmitter and the slave/receiver, which can be implemented practically by implementing the dynamic feedback modulation (DFM) technique, in which the dynamic error between the two subsystems is dynamically calculated via equations 14 through 16. This error should converge to zero after a few milliseconds.

$$e_x = \frac{dx_m}{dt} - \frac{dx_m}{dt} = \sigma(y_m - x_m) - \sigma(y_s - x_s) \tag{14}$$

$$e_y = \frac{dy_m}{dt} - \frac{dy_s}{dt} = (\rho x_m - y_m - x_m z_m) - (\rho x_s - y_s - x_s z_s) \tag{15}$$

$$e_z = \frac{dz_m}{dt} - \frac{dz_s}{dt} = (x_m y_m - \beta z_m) - (x_s y_s - \beta z_s) \tag{16}$$

This error value is multiplied by the gain factor (usually 10, 15, 20 or 25), as shown in equations 17 through 19. The error signal in each axis is added to the slave subsystem to provide the necessary synchronization.

$$cs_x = G * e_x \tag{17}$$

$$cs_y = G * e_y \tag{18}$$

$$cs_z = G * e_z \tag{19}$$

The system of equations in 4, 5, and 6 will be modified to be the equations in 20 through 22 that will be used through the system implementation. Figure 7 below illustrates the implementation of the synchronization circuitry in XSG blocks.

$$\frac{dx_s}{dt} = (y_s - x_s) + cs_x \tag{20}$$

$$\frac{dy_s}{dt} = \rho x_s - y_s - x_s z_s + cs_y \tag{21}$$

$$\frac{dz_s}{dt} = x_s y_s - \beta z_s + cs_z \tag{22}$$

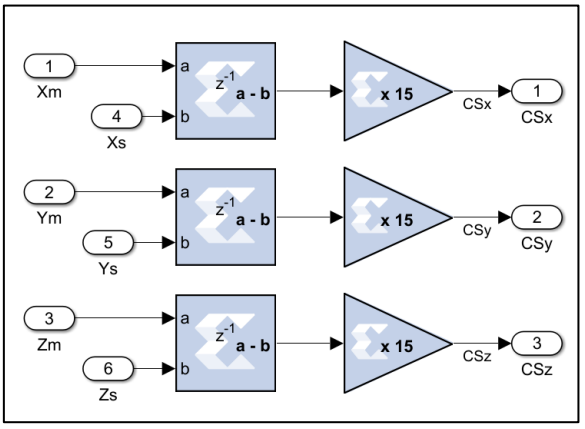

Fig. 7. Control subsystem

2.3.3 Preprocessing and Postprocessing

Figure 8, shown below, illustrates the preprocessing subsystem, which consists of MATLAB/Simulink blocks: Matrix Transpose, Reshape, to frame, and Unbuffer. The combination of these blocks is used to convert the image matrix into serial

samples, each of which contains 8 bits, as a prior stage for encryption. These parallel bits (samples) are then converted into a serial bit stream via parallel-to-serial conversion to encrypt the theme via the XOR operation with the x-dynamics of the chaotic system.

On the other hand, Figure 9 shows the block combination that constructs the postprocessing subsystem that operates in reverse mode to the preprocessing subsystem, where the serial bits are combined together to form serial samples each of 8 bits by means of serial to parallel conversion. These serial samples are then combined to construct the image matrix again. The postprocessing subsystem consists of Buffer, Reshape, Matrix Transpose, and Unit8 MATLAB/Simulink blocks.

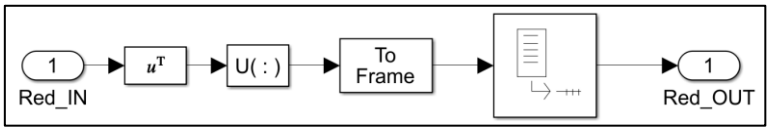

Fig. 8 Preprocessing XSG blocks

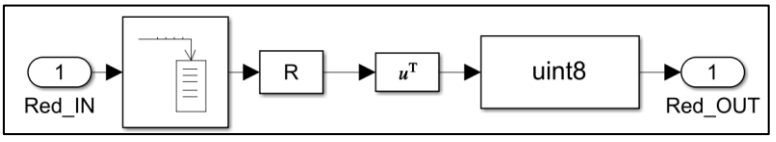

Fig. 9. XSG block postprocessing

2.3.4 Overall System Implementation

The overall encryption system is presented in Figure 10 and Figure 11. However, Figure 10 shows the implementation using the forward-Euler integration method, whereas Figure 11 shows the Runge–Kutta integration method implementation. The three subsystems are clearly presented in the figures below. The image can be called from any location from the PC by using the MATLAB/Simulink block (image from the file). Then, the image matrix is converted into serial samples of data, which enter the XSG domain by using a gateway in the block. Image binary bits are encrypted via the XOR operation with X chaotic dynamics at the transmitter system. On the other hand, the receiver reverses the operations to recover the original image matrix bits, as shown in Figures 10 and 11 [27][28] [29][30].

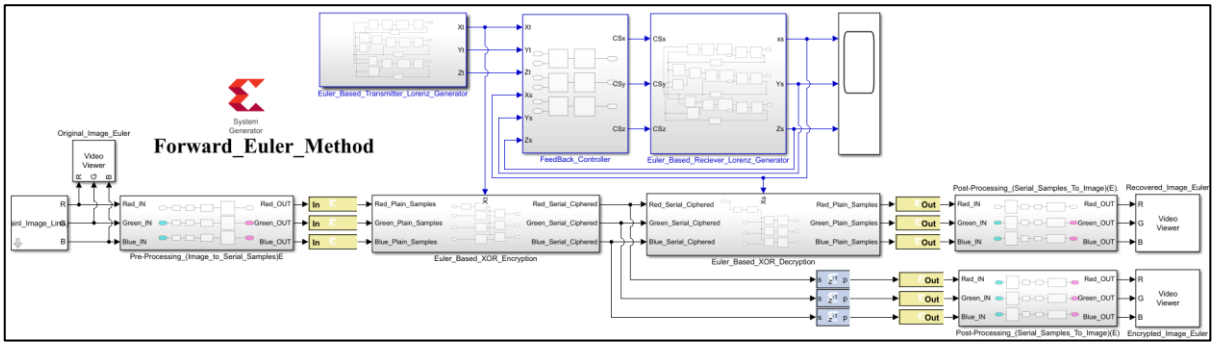

Fig. 10. General XSG block diagram of the proposed image cryptosystem-based on the Euler method

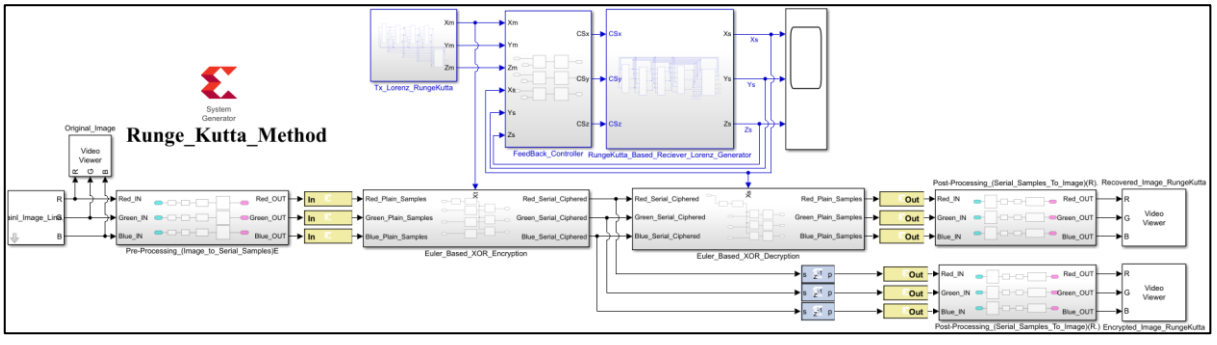

Fig. 11. General XSG block diagram of the proposed image cryptosystem-based on the Runge–Kutta method

3. RESULTS AND DISCUSSION

The dynamic behavior of the master and slave subsystems is approximately identical (using both integration methods) because the dynamic feedback modulation (DFM) technique provides the necessary synchronization. Figure 12 shows the dynamic behavior of the three components (x, y, and z) for both the master and slave (both of them have identical time responses). Figure 13 compares the X dynamics of the master and slave manipulators. The X-master dynamics are presented in black, whereas the X-slave dynamics are presented in blue. Figure 13 proved that the two dynamics are synchronized and can be used for data encryption. The X dynamics of the two subsystems are adopted for generating pseudorandom bits that are used to encrypt the data bits via the XOR operation [31][32].

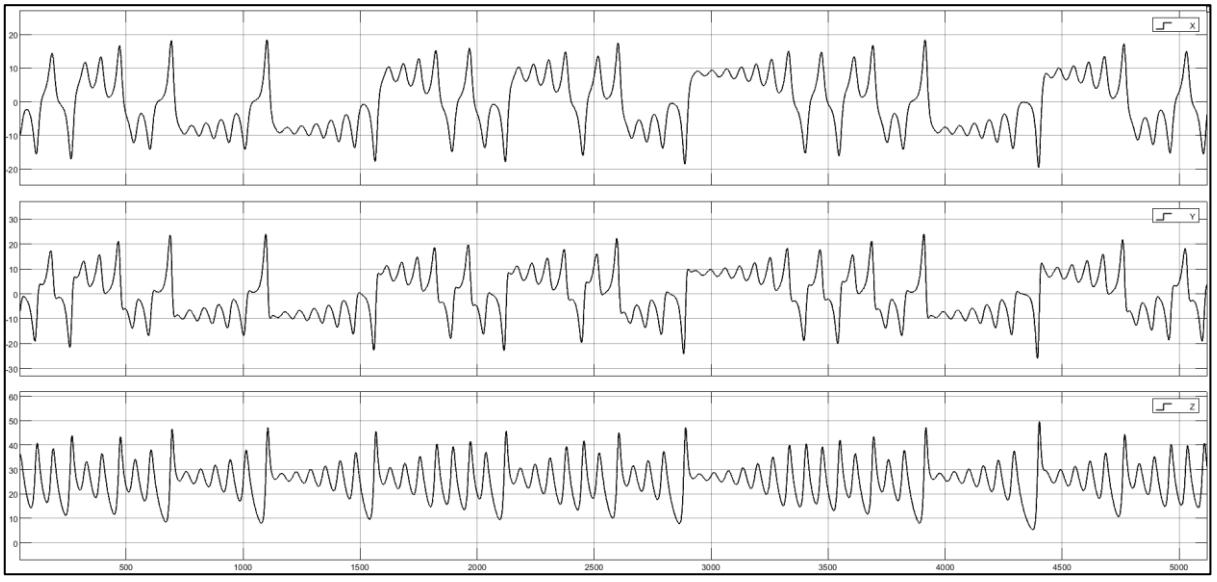

Fig. 12. X, Y, and Z Dynamical Behavior of the Master System

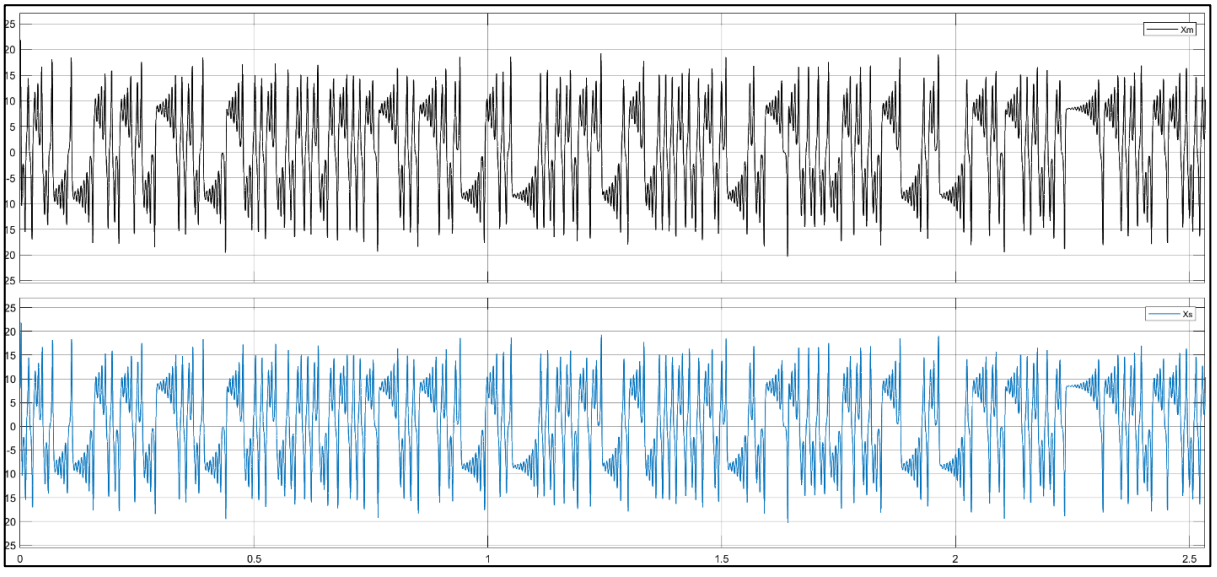

Fig. 13. X-Dynamics of Master and Slave

Figure 14 depicts the 3D plot for the Lorenz trajectory. The left trajectory is solved via the forward Euler integration method, whereas the right trajectory is solved via the Runge–Kutta method. Since the two trajectories are approximately identical, both integration methods are suitable for solving the nonlinear systems described via ODEs [33].

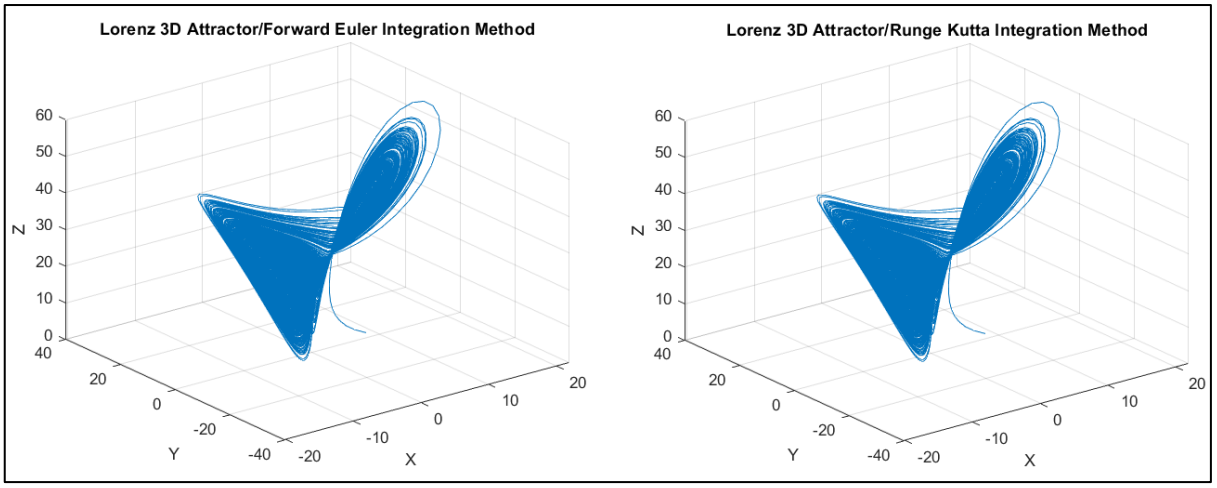

Fig. 14. 3D Lorenz trajectories obtained via the forward Euler and Runge–Kutta methods

The proposed encryption system is tested and tested via different image sizes, where it shows stable, fast and robust performance. Figure 15 shows the different sizes of colored plain images used to test the proposed cryptosystem. The plain-colored images are first converted into binary form, and then the binary data are XORed with X-dynamics in the master subsystem to obtain the encrypted images. On the other hand, the encrypted image is converted back into a plain image at the slave subsystem via identical X-dynamics with the XOR operation, as shown in Fig. 15 above. The security analysis of the proposed cryptographic system and the encrypted images is performed by using six statistical methods, which are as follows: image histogram, peak signal-to-noise ratio (PSNR), entropy, correlation, number of pixels change rates (NPCRs), and unified averaged changed intensity UACI.

3.1 Histogram analysis

A histogram is a graphical view that shows the level of the image data distribution. Once the histogram levels of the plain and encrypted images significantly differ, the encrypted images do not present any evidence or valid information for the attacker to perform any statistical attacks on the cryptographic systems. The histogram levels of the plain and encrypted images are presented in Figure 15 above for the proposed cryptographic system. As indicated by the distribution levels, there is a significant difference between the plain and encrypted image histograms, which indicates that the attacker will not be able to start any statistical-based attacks because there are no valid statistical data in the encrypted images.

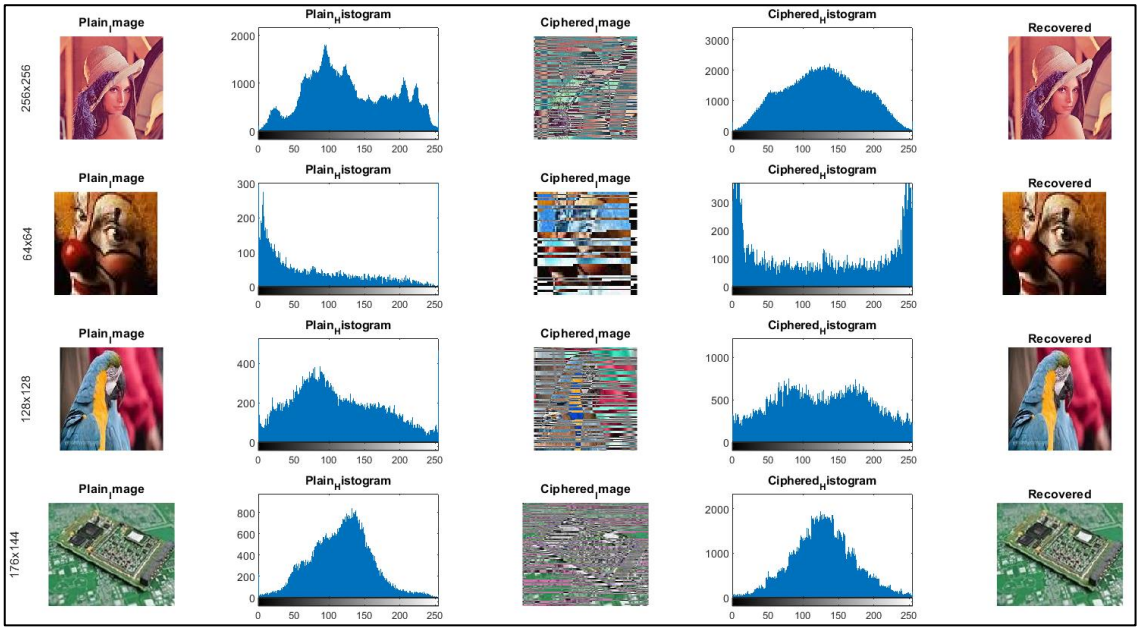

Fig. 15. Plain images, encrypted images, and histogram analysis

3.2 Peak signal-to-noise ratio (PSNR) and mean square error (MSE)

The mean square error (MSE) and peak-to-signal ratio (PSNR) tests are carried out in this subsection. The MSE test measures the difference between two input images, where higher MSE values indicate that the images are highly different, whereas a zero MSE value indicates that the two images are identical. On the other hand, the PSNR test measures the peak signal-to-noise ratio of two images, which reflects the quality of the images; a higher PSNR indicates higher-quality images. The MSE and PSNR can be calculated via the formulas given in 23 & 24.

$$MSE = \frac{\sum_{i,j} I_1(i,j) - I_2(i,j)}{m*n} \tag{23}$$

$$PSNR = 10log \frac{R^2}{MSE} \tag{24}$$

Table I shows the PSNR and MSE comparisons between the original image, ciphered image and recovered image of different sizes. The PSNR and MSE values between the original images and the recovered images indicate that the two images are identical. On the other hand, the comparison of the MSE and PSNR values between the ciphered and original images indicates that the images are significantly different, with a high level of error between them, which means that the proposed cryptosystem is significantly strong against statistical attacks.

TABLE I. TABLE I PSNRS AND MSES OF THE PROPOSED CRYPTOSYSTEM

Image Size	Cipher Image		Recovered Image	
	PSNR	***MSE***	***PSNR***	***MSE***
256 x 256	10.0333	6.4529 e+03	Inf	0
64 x 64	6.2132	1.5551 e+04	Inf	0
128 x 128	8.6375	8.8988 e+03	Inf	0
176 x 144	12.0749	4.0327 e+03	Inf	0

3.3 Adjacent Pixel Correlation Coefficients

The correlation coefficient is a mathematical technique that is used to compute how two factors are related. In terms of image processing, if the correlation coefficient values of two images reach or approach zero, the correlation of neighboring pixels in the plain image and ciphered image can be neglected and removed, and as a result, the statistical attack will be restrained. The correlation coefficient of two matrices can be calculated via the formula 25 shown below. The correlation coefficients between the plain image and the ciphered images (with different sizes) are calculated and presented in Table II below.

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(A_{mn} - \bar{A})^2 \sum_m \sum_n (B_{mn} - \bar{B})^2}} \tag{25}$$

TABLE II CORRELATION COEFFICIENTS

Image Size	Colour Layer	Correlation
256x256	Red	-0.006
	Green	-0.006
	Blue	-0.006
64x64	Red	0.006
	Green	0.006
	Blue	0.006
128x128	Red	-0.0447
	Green	-0.0447
	Blue	-0.0447
176x144	Red	0.0536
	Green	0.0536
	Blue	0.0536

3.4 Entropy

The entropy is a measure of information content and can be explicated as the average of the uncertainty of the source of the information. The image entropy can be calculated via the formula presented in 26. The image entropy of the proposed cryptosystem is calculated and presented in Table 3 for different image sizes and for the three-color layers [34].

$$Entropy\ (s) = \sum_{n=0}^{2^N - 1} P(s_i) \times \log_2 \frac{1}{P(s_i)} \tag{26}$$

where P(s) represents the probability of the symbol si. N denotes the number of bits.

3.5 Number of Pixel Change Rate NPCRs

With respect to differential attacks, the number of pixels changed rate (NPCR) and the unified average change intensity (UACI) are the two most common factors used to estimate the robustness and strength of the image-based cryptosystem or algorithm. Higher UACI and NPCR values indicate higher system resistance and strength against differential attacks. The NPCR quantity can be calculated via the formula in 27. It is clear that the NPCR quantity is concentrated on the absolute number of pixels whose value changes during the attack. The results of the NPCRs of different image sizes and different channels are presented in Table 3, which shows that the system is robust against differential attacks [35] [15].

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M*N} * 100\% \tag{27}$$

$$D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, C_1(i,j) = C_2(i,j) \end{cases}$$

3.6 Unified Averaged Changed Intensity UAIC

The UACI quantity can be calculated via the formula in 28, where this quantity is highly concentrated on the average differences between two images. Table III below shows the UACI values for different image sizes and for the three color channels [35].

$$UACI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |C_1(i,j) - C_2(i,j)|}{255*M*N} * 100\% \tag{28}$$

TABLE III ENTROPY, NPCR, AND UACI RESULTS

Image Size	Colour Layer	Entropy	NPCR	UACI
256x256	Red	7.8138	99.52	41.5
	Green	7.7266	99.19	17.19
	Blue	7.5924	99.15	13.81
64x64	Red	7.5898	98.71	11.2
	Green	7.3368	98.46	18.2
	Blue	6.7945	96.39	24.4
128x128	Red	7.7498	98.46	27.4
	Green	7.7584	98.85	49.7
	Blue	7.8446	99.14	44.2
176x144	Red	7.6641	98.99	55.7
	Green	7.3966	98.73	25
	Blue	7.4762	98.79	46.1

3.7 Key space

One of the main aspects related to data security and cryptosystem design is the cryptographic system key space. Cryptosystems with relatively large keys can provide more secure data that are stronger and more robust against brute force attacks. The proposed image encryption system has six secret keys represented by the initial value of the chaos system (xo, yo, zo) as well as the system parameters (β, ρ, σ). The chaotic parameters and the initial states require 32 bits to represent them; hence, the key space of the proposed system is $(2^{32})^6=2^{192}$. Therefore, this key space is suitable for image encryption and effective against brute force attacks since it is greater than 2^{100} [36]. Table IV shows a comparison between the proposed system and a traditional cryptosystem that is widely used with respect to system key space.

TABLE IV KEY SPACE COMPARISON

Encryption Algorithm	Key space
Proposed System	2^{192}
Reference [39]	2^{45}
Reference [40]	2^{199}
Reference [41]	2^{203}
Reference [42]	2^{149}
Reference [43]	2^{200}

4. FPGA IMPLEMENTATION AND HARDWARE COSIMULATION

The system generator is used throughout this paper to obtain the VHDL code for the chaos system generator. The obtained code is then used to configure the FPGA PYNQ-Z1 evolution board. The dynamical equations of the chaotic system are solved two times: the first with the forward Euler integration method and the second with the Runge–Kutta integration method. The purpose of this is to determine a method that uses a small amount of FPGA board resources. Figure 16 depicts the FPGA implementation via both methods, where the colored image is called from its location on the PC, which is converted into serial samples that are sent to the FPGA board through the JTAG link for encryption purposes. The ciphered image is then sent back to the PC to display it, as shown in figure 16.

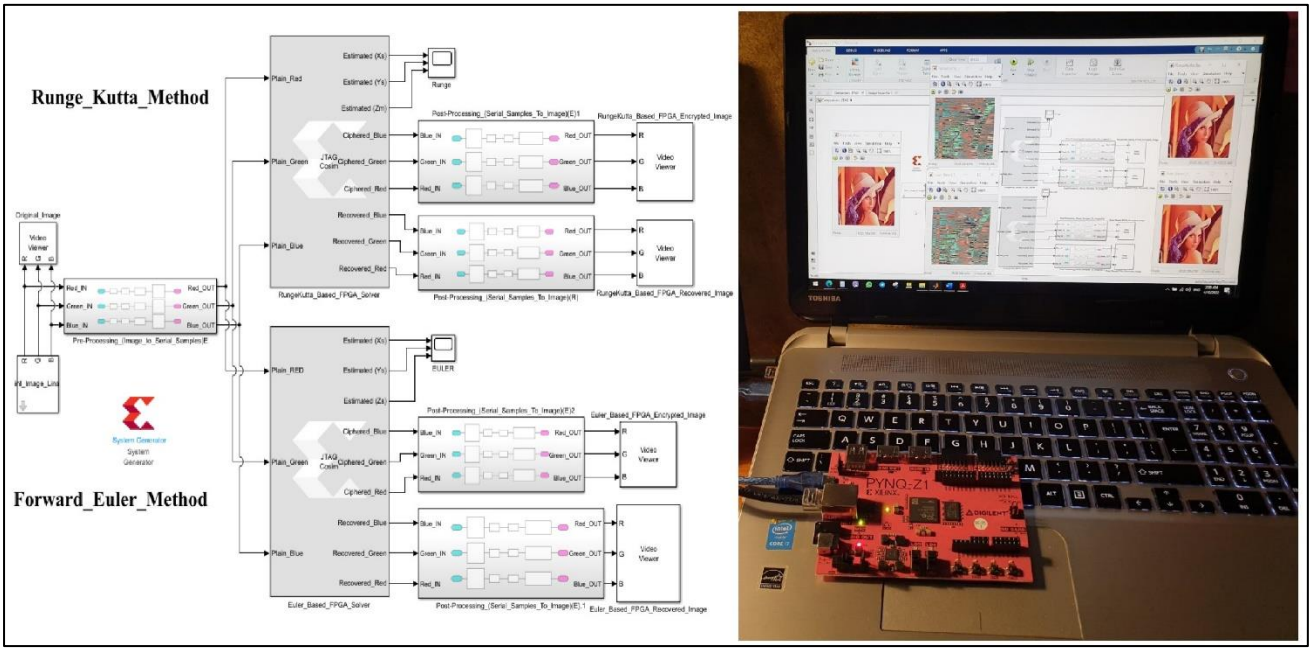

Fig. 16. Real-time hardware cosimulation of image encryption cryptography

Table V shows the FPGA board utilization lists for the Runge–Kutta method and forward Euler method. From the table, it is clear that the Runge–Kutta method consumes more resources than the Euler method does because of its repetition behavior during the solution of the dynamical system [44][45]. Finally, the outputs from MATLAB/Simulink and the FPGA board are clearly identical, which proves that the real-time encryption system operates normally and can be used for real-time applications such as wireless sensor networks (WSNs).

TABLE V FPGA RESOURCES UTILIZED WITH THE RUNGE KUTTA METHOD VS. THE FORWARD EULER METHOD

Resource	Available	Runge Kutta		Forward Euler	
		Utilization	Utilization %	Utilization	Utilization %
Look Up Table (LUT)	53200	5470	10.28	2047	3.85
Look Up Table RAM (LUTRAM)	17400	1	0.01	1	0.01
Flip Flops (FF)	106400	1857	1.75	1472	1.38
Block RAM (BRAM)	140	2	1.43	2	1.43
Digital Signal Processing (DSP)	220	144	65.45	36	16.36
Input Output (IO)	125	1	0.80	1	0.80
Global Buffer (BUFG)	32	4	12.50	4	12.50
Mixed Mode Clock Manager (MMCM)	4	1	25.00	1	25.00

5. CONCLUSION

In this paper, the design and implementation of a cryptographic system based on a Lorenz chaotic attractor for colored image encryption/decryption are carried out. It can be concluded that (in this field of research), it is worth solving the nonlinear equations of the Lorenz attractor by using the forward Euler and Runge–Kutta methods. As a deduction, this analysis facility provides a promising opportunity in simulating and practical implementing the proposed cryptographic system for colored images. However, the Runge–Kutta integration approach adds some form of limitation to the practical analysis method, where it consumes more resources from the FPGA board, causing a considerable amount of timing delay in the encryption/decryption processes. Nevertheless, the results of the simulation and real-time implementation (by the PYNQ-Z1 FPGA evolution board and the PC with the JTAG communication link) show good agreement and appear to be more accurate when the adopted approach is applied. The outcomes are confident and verified by the system testing results, which show highly accepted behavior for different image sizes (256×256, 64×64, 128×128, and 176×144 pixels). Overall, it can also be deduced that the cryptographic system used in this study has the ability to be used for real-time applications such as WSNs, and it is also suitable for current applications because it provides a throughput close to 200 Mb/s.

Funding

The author's paper clearly indicates that the research was conducted without any funding from external sources.

Acknowledgement

The authors wish to thank the educational family in the Electronic and Communication Engineering Department at the College of Engineering at the University of Al-Qadisiyah for their support in completing this research.

References

[1] D. Blackmore, “The mathematical theory of chaos,” Comput. Math. with Appl., vol. 12, no. 3-4 PART 2, pp. 1039–1045, 1986, doi: 10.1016/0898-1221(86)90439-6.
[2] W. Der Chang, “Digital secure communication via chaotic systems,” Digit. Signal Process. A Rev. J., vol. 19, no. 4, pp. 693–699, 2009, doi: 10.1016/j.dsp.2008.03.004.
[3] T. Yang, C. W. Wu, and L. O. Chua, “Cryptography based on chaotic systems,” IEEE Trans. Circuits Syst. I Fundam. Theory Appl., vol. 44, no. 5, pp. 469–472, 1997, doi: 10.1109/81.572346.
[4] A. Soleymani, M. J. Nordin, and E. Sundararajan, “A chaotic cryptosystem for images based on Henon and Arnold cat map,” Sci. World J., vol. 2014, 2014, doi: 10.1155/2014/536930.
[5] V. Milanović and M. E. Zaghloul, “Improved masking algorithm for chaotic communications systems,” Electron. Lett., vol. 32, no. 1, pp. 11–12, 1996, doi: 10.1049/el:19960004.
[6] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” vol. 64, no. 8, pp. 821–825, 1990.
[7] I. A. Kamil, “Self-synchronization in chaotic systems,” Journal of Engineering and Applied Sciences, vol. 7, no. 6. pp. 411–417, 2012, doi: 10.3923/jeasci.2012.411.417.
[8] S. N. Tambe-Jagtap, “A Survey of Cryptographic Algorithms in Cybersecurity: From Classical Methods to Quantum-Resistant Solutions”, SHIFRA, vol. 2023, pp. 43–52, Jun. 2023, doi: 10.70470/SHIFRA/2023/006.
[9] A. A. Elkouny, “Design and implementation of synchronized VHDL Lorenz chaotic encryption system,” 2014 9th Int. Conf. Informatics Syst. INFOS 2014, pp. CNs1–CNs6, 2015, doi: 10.1109/INFOS.2014.7036713.
[10] M. Eisencraft and A. M. Batista, “Discrete-time chaotic systems synchronization performance under additive noise,” Signal Processing, vol. 91, no. 8, pp. 2127–2131, 2011, doi: 10.1016/j.sigpro.2011.01.021.
[11] Y. Zhang, “A chaotic system based image encryption scheme with identical encryption and decryption algorithm,” Chinese J. Electron., vol. 26, no. 5, pp. 1022–1031, 2017, doi: 10.1049/cje.2017.08.022.
[12] M. Transport, “Real Time Implementation of Secure Communication System Based On Synchronization of Hyper Chaotic Systems 2016 , 33 rd NATIONAL RADIO SCIENCE CONFERENCE,” 2016 , 33 rd Natl. RADIO Sci. Conf., no. Nrsc, pp. 159–167, 2016.
[13] H. M. M. Alibraheemi, Q. Al-Gayem, and E. A. Hussein, “Four dimensional hyperchaotic communication system based on dynamic feedback synchronization technique for image encryption systems,” Int. J. Electr. Comput. Eng., vol. 12, no. 1, p. 957, Feb. 2022, doi: 10.11591/ijece.v12i1.pp957-965.
[14] J. S. Lin, C. F. Huang, T. L. Liao, and J. J. Yan, “Design and implementation of digital secure communication based on synchronized chaotic systems,” Digit. Signal Process. A Rev. J., vol. 20, no. 1, pp. 229–237, 2010, doi: 10.1016/j.dsp.2009.04.006.
[15] Q. Yang et al., “FPGA implementation of color image encryption using a new chaotic map,” IEEE Trans. Circuits Syst. I Fundam. Theory Appl., vol. 7, no. 1, pp. 129–137, 2019, doi: 10.11591/ijeecs.v13.i1.pp129-137.

[16] T. T. K. Hue, C. Van Lam, T. M. Hoang, and S. Al Assad, "Implementation of secure SPN chaos-based cryptosystem on FPGA," 2012 IEEE Int. Symp. Signal Process. Inf. Technol. ISSPIT 2012, no. August 2014, pp. 129–134, 2012, doi: 10.1109/ISSPIT.2012.6621274.
[17] M. Amin, O. S. Faragallah, and A. A. Abd El-Latif, "A chaotic block cipher algorithm for image cryptosystems," Commun. Nonlinear Sci. Numer. Simul., vol. 15, no. 11, pp. 3484–3497, 2010, doi: 10.1016/j.cnsns.2009.12.025.
[18] A. A. Abd El-Latif, X. Niu, and M. Amin, "A new image cipher in time and frequency domains," Opt. Commun., vol. 285, no. 21–22, pp. 4241–4251, 2012, doi: 10.1016/j.optcom.2012.06.041.
[19] M. Garcia-Bosque, A. Pérez-Resa, C. Sánchez-Azqueta, C. Aldea, and S. Celma, "A New Technique for Improving the Security of Chaos Based Cryptosystems," Proc. - IEEE Int. Symp. Circuits Syst., vol. 2018-May, pp. 0–4, 2018, doi: 10.1109/ISCAS.2018.8351195.
[20] B. H. Prasetio, E. Setiawan, and A. Muttaqin, "Image Encryption using Simple Algorithm on FPGA," TELKOMNIKA (Telecommunication Comput. Electron. Control., vol. 13, no. 4, p. 1153, 2015, doi: 10.12928/telkomnika.v13i4.1787.
[21] S. S. H. Shah and G. Raja, "FPGA implementation of chaotic based AES image encryption algorithm," IEEE 2015 Int. Conf. Signal Image Process. Appl. ICSIPA 2015 - Proc., pp. 574–577, 2016, doi: 10.1109/ICSIPA.2015.7412256.
[22] B. Sinha, S. Kumar, and C. Pradhan, "Comparative analysis of color image encryption using 3D chaotic maps," Int. Conf. Commun. Signal Process. ICCSP 2016, pp. 332–335, 2016, doi: 10.1109/ICCSP.2016.7754150.
[23] K. W. Wong, "A fast chaotic cryptographic scheme with dynamic look-up table," Phys. Lett. Sect. A Gen. At. Solid State Phys., vol. 298, no. 4, pp. 238–242, 2002, doi: 10.1016/S0375-9601(02)00431-0.
[24] W. ALEXAN, M. ELKANDOZ, M. MASHALY,E AZAB, A ABOSHOUSHA, "Color Image Encryption Through Chaos and KAA Map," Complexity, vol. 2019, 2023, doi: 10.1109/ACCESS.2023.3242311.
[25] A. Salih, Z. Abdulrazaq, H. Ghanim Ayoub, "Design and Enhancing Security Performance of Image Cryptography," Complexity, vol. 2019, 2024, doi.org/10.21123/bsj.2024.10521.
[26] R. Barboza, "Dynamics of a hyperchaotic Lorenz system," Int. J. Bifurc. Chaos, vol. 17, no. 12, pp. 4285–4294, 2007, doi: 10.1142/S0218127407019950.
[27] M. F. Fathoni and A. I. Wuryandari, "Comparison between Euler, Heun, Runge-Kutta and Adams-Bashforth-Moulton integration methods in the particle dynamic simulation," Proc. 2015 4th Int. Conf. Interact. Digit. Media, ICIDM 2015, no. Icidm, 2016, doi: 10.1109/IDM.2015.7516314.
[28] U. F. Fak and S. Say, "COMPARISON OF RUNGE-KUTTA METHODS OF ORDER 4 AND 5 ON LORENZ EQUATION Emre SERMUTLU 1," pp. 61–69, 2004.
[29] A. Parnak, Y. Baleghi, and J. Kazemitabar, "A Novel Image Splicing Detection Algorithm Based on Generalized and Traditional Benford's Law," Int. J. Eng. Trans. A Basics, vol. 35, no. 4, pp. 626–634, 2022, doi: 10.5829/ije.2022.35.04a.02.
[30] A. Mehdizadeh, M. Mohammadpoor, and Z. Soltanian, "Secured route optimization and micro-mobility with enhanced handover scheme in mobile IPv6 networks," Int. J. Eng. Trans. B Appl., vol. 29, no. 11, pp. 1530–1538, 2016, doi: 10.5829/idosi.ije.2016.29.11b.06.
[31] T. G. Babu and V. Jayalakshmi, "Conglomerate Energy Efficient Elgamal Encryption Based Data Aggregation Cryptosystems in Wireless Sensor Network," Int. J. Eng. Trans. B Appl., vol. 35, no. 2, pp. 417–424, 2022, doi: 10.5829/ije.2022.35.02b.18.
[32] H. Nasiraee, "DoS-Resistant Attribute-Based Encryption in Mobile Cloud Computing with Revocation," Int. J. Eng., vol. 32, no. 9, pp. 1290–1298, 2019, doi: 10.5829/ije.2019.32.09c.09.
[33] N. M. Dyamannavr, N. G. Kurahatti, and A. Christina, "Design and implementation of field programmable gate array based baseband processor for passive radio frequency identification tag," Int. J. Eng. Trans. A Basics, vol. 30, no. 1, pp. 127–133, 2017, doi: 10.5829/idosi.ije.2017.30.01a.16.
[34] M. Ashourian, N. Daneshmandpour, O. Sharifi Tehrani, and P. Moallem, "Real time implementation of a license plate location recognition system based on adaptive morphology," Int. J. Eng. Trans. B Appl., vol. 26, no. 11, pp. 1347–1356, 2013, doi: 10.5829/idosi.ije.2013.26.11b.10.
[35] Y. Karimi, S. Rashahmadi, and R. Hasanzadeh, "The effects of newmark method parameters on errors in dynamic extended finite element method using response surface method," Int. J. Eng. Trans. A Basics, vol. 31, no. 1, pp. 50–57, 2018, doi: 10.5829/ije.2018.31.01a.08.
[36] F. Yu et al., "Analysis and FPGA Realization of a Novel 5D Hyperchaotic Four-Wing Memristive System, Active Control Synchronization, and Secure Communication Application," Complexity, vol. 2019, 2019, doi: 10.1155/2019/4047957.
[37] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI Randomness Tests for Image Encryption," Cyberjournals.Com, 2011.
[38] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," Int. J. Bifurc. Chaos, vol. 16, no. 8, pp. 2129–2151, 2006, doi: 10.1142/S0218127406015970.

[39] W. Wang et al., “An encryption algorithm based on combined chaos in body area networks,” Comput. Electr. Eng., vol. 65, pp. 282–291, 2018, doi: 10.1016/j.compeleceng.2017.07.026.
[40] Z. Bashir, J. Watróbski, T. Rashid, S. Zafar, and W. Salabun, “Chaotic dynamical state variables selection procedure based image encryption scheme,” Symmetry (Basel)., vol. 9, no. 12, 2017, doi: 10.3390/sym9120312.
[41] X. Wu, K. Wang, X. Wang, H. Kan, and J. Kurths, “Color image DNA encryption using NCA map-based CML and one-time keys,” Signal Processing, vol. 148, pp. 272–287, 2018, doi: 10.1016/j.sigpro.2018.02.028.
[42] X. Wang, X. Zhu, X. Wu, and Y. Zhang, “Image encryption algorithm based on multiple mixed hash functions and cyclic shift,” Opt. Lasers Eng., vol. 107, no. December 2016, pp. 370–379, 2018, doi: 10.1016/j.optlaseng.2017.06.015.
[43] M. L. Barakat, A. S. Mansingka, A. G. Radwan, and K. N. Salama, “Hardware stream cipher with controllable chaos generator for colour image encryption,” IET Image Process., vol. 8, no. 1, pp. 33–43, 2014, doi: 10.1049/iet-ipr.2012.0586
[44] H. Omotunde and M. Ahmed, “A Comprehensive Review of Security Measures in Database Systems: Assessing Authentication, Access Control, and Beyond,” Dec. 10, 2023, Mesopotamian journal of Cybersecurity. doi: 10.58496/MJCS/2023/016.
[45] R. M. Al-Amri, D. N. Hamood, and A. K. Farhan, “Theoretical Background of Cryptography,” Dec. 10, 2023, Mesopotamian journal of Cybersecurity. doi: 10.58496/MJCS/2023/002.