Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Engineering

Design and Implementation of a Secure Wireless Hyperchaotic Communication System Using Field Programmable Gate Array (FPGA)

Submitted to the College of Engineering
of the University of Babylon in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy
in Engineering/ Electrical Engineering /Electronics and Communications

By

Hayder Mazin Makki Alibraheemi

2022

Supervised by

Prof. Dr. Ehab A. Hussein **Prof. Dr. Qais Al-Gayem**

2022 A.D **1444 A.H**

ABSTRACT

Image encryption algorithms based on nonlinear systems (such as chaos and hyperchaos) are increasing rapidly because of the wide range characteristics offered by these systems such as unpredictable behavior, extreme sensitivity to initial conditions and system parameters perturbations. The design, simulation, and FPGA implementation of a secure wireless hyperchaotic communication system is presented using five different proposed algorithms. The design and simulation are carried out using Xilinx System Generator (XSG) while the FPGA boards are programmed with Very high-speed integrated circuit Hardware Description Language (VHDL).

The first proposed algorithm is the design of three-dimensional Lorenz nonlinear system generator at the transmitter (Tx) and receiver (Rx) sides to be used as a high-speed chaos switching in order to switch between the designed hyperchaotic carriers where it can be employed in the XOR operation.

The second algorithm involve the combination of different hyperchaotic systems using the XOR logical operation to construct a new multi-dimensional hyperchaotic system. The generated hyperchaotic system is used in the Tx and Rx to encrypt and decrypt the images. Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) between the original and encrypted images are calculated. The results are around 8 for PSNR and $1.3e^4$ for MSE, which indicated that the images are highly different. The number of pixels change rate (NPCR), and unified averaged changed intensity (UACI) are also calculated through this work, the NPCR has a value close to 100%, while UACI is close to 33%, which indicates that the algorithms are highly immune against the differential attacks.

The third proposed algorithm is a cascaded hyperchaotic system generator based on chaos switching, where three different dimensional hyperchaotic systems are used and combined using high speed switch. The output of the speedy switch is controlled by the Lorenz chaotic system (first proposed algorithm). The mixture patterns of the hyperchaotic systems generate completely new binary bit streams that used for secure image-based communications. The UACI, and NPCR results are close to 33% and 100% which satisfied the conditions to stands against the attacks.

The fourth proposed approach involves designing a new version of a five-dimensional hyperchaotic system with a sine wave as an input parameter to the ordinary differential equations to increase the randomness and unpredictability of the developed system. The proposed new system has been adopted to build a secure communication system for image encryption purposes which show a good performance. The calculated MSE and PSNR between the encrypted and plain images are closed to $1.57e^4$ and 6.25 respectively, which means that the images are efficiently encrypted.

The last proposed approach involves constructing a new random binary bit stream generator to be utilized in the communication systems' transmitters and receivers based on the 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} proposed algorithms. The system is meant to exhibit extremely unexpected behavior and randomness, making it ideal for secure image encryption/decryption. The histogram is computed to view the color distribution of the ciphered images which show a flat distribution which indicates efficient performance.

All the designed algorithms are used and tested for RGB and gray scale images with different sizes (equal and unequal dimensions). All the designed algorithms are successfully implemented using FPGA PYNQ-Z1 zynq xc7z020 board with acceptable board resources utilization.

ACKNOWLEDGMENT

I would like to thank the family of the Electrical Engineering department at the University of Babylon especially my supervisors Prof. Dr. **Ehab A. Hussein** and Prof. Dr. **Qais Al-Gayem** for their patience, support, and valuable guidance. This thesis wouldn't have been possible without the priceless supervision of Prof. Dr. **Ehab A. Hussein** and Prof. Dr. **Qais Al-Gayem**.

I would also want to send my heartfelt estimate to all my **teachers** in the Electrical Engineering department for their invaluable advice that opened the knowledge gates for me, and enriched my thinking capabilities.

I would like to express my deepest gratitude to my mother and father **Zainab** and **Mazin** for their huge support and to other family members **Aulla**, **Doaa**, **Tabarek,** and **Yaqeen**, who provide the necessary support to finalize this work.

Also, special thanks go to all my dear friends especially my friend **Akram** for his invaluable assistance.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST ABBREVIATIONS

AES	Advanced Encryption Standard
AWGN	Additive White Gaussian Noise
BRAM	Block Random Access Memory
CORR	Correlation Coefficient
DES	Data Encryption Standard
DFM	Dynamic Feedback Modulation
DSP	Digital Signal Processor
DSS	Digital Signature Standard
E	Entropy
FF	Flip Flops
FPGA	Field Programmable Gate Array
GBUF	Global Buffer
Hwcosim	Hardware Co-Simulator
IDEA	International Data Encryption Algorithm
IO	Input Output
JTAG	Joint Test Action Group
LUT	Lock Up Table
LUTRAM	Lock Up Table Random Access Memory
MD5	Message-Digest Algorithm

MSE	Mean Square Error
MMCM	Mixed Mode Clock Manager
NPCR	Number of Pixels Change Rate
ODE	Ordinary Differential Equation
PSNR	Peak Signal to Noise Ratio
PSRG	Pseudo-Random Bit Generator
RC4	Rivest Cipher 4
RGB	Red Green Blue
RSA	Rivest, Shamir, Adleman
SHA	Secure Hashing Algorithm
TKIP	Temporal Key Integrity Protocol
TTP	Trusted Third Party
UACI	Unified Averaged Changed Intensity
VHDL	Very High-Speed Description Language
XSG	Xilinx System Generator

LIST SYMBOLS

$\dot{x}, \dot{y}, \dot{z}$	Lorenz System State Variables
x_0, y_0, z_0	State Variables initial Values
$x_m(t)$	Master State Variable in X Direction
$x_s(t)$	Slave State Variable in X Direction
e_x, e_y, e_z	Master/Slave Error Signals
α, ρ, β	Lorenz Chaotic System Parameters
a, b	Rossler Chaotic System Parameters
ϕ	Electrical response of the nonlinear resistor of the Chua electronic circuit
a, b, c	Chua Chaotic System Parameters
K, L	Rucklidge Chaotic System Parameters
α, β, γ, a, b, I_0	Nien Chaotic System Parameters
β	Logistic Map Bifurcation Parameter
K1, K2, K3, K4	Runge-Kutta Slope Approximation
Y_{t+1}	Approximated (Estimated) Signal
λ	Continuous Time Lyapunov Exponents
p(i)	Bits Probability
Cs	Synchronization Control Signals
G	Signal Gain

CHAPTER ONE

Introduction

Chapter One: General Introduction

1.1. Motivation

The last few decades have been known by the exponential growth in personal digital based network communication devices, such as personal computers, digital signatures, electronic trade, and multimedia communication devices. This increase in personal devices led to a massive increase in the transmitted information and data over the Internet and public channels. The increase in the amount of communicated data and information over the public channels, a rapidly growing need for data protection and safeguard to ensure data privacy, and confidentiality and to prevent fraud. The development of the Internet and multimedia technologies, massive efforts have been spent by researchers to continuously develop a new and robust cryptographic systems to meet the security requirements for secure data transmission over public channels [1].

Generally, the term cryptography is referred to the art and science that study the methods of protecting and safeguarding the user's sensitive data and information from unauthorized disclosures during the data transmission through unsecured public channels like the Internet. Basically, all of the cryptographic systems convert the readable (intelligible) original message which known as a plain-message into unreadable (mysterious) message generally known as ciphered-message and vice versa. The conversion process that takes the plain-message to ciphered-message is known as encryption and it can be performed by applying some complex mathematical algorithms to the plain-message with the help of secret data known as the encryption key. In the other hand the decryption operation represents the process that retrieval the plain-message from the ciphered-message by using some mathematical algorithms and secret key[2]. Generally, there are four main types or disciplines

of the cryptography (network security) which are summarized in the chart presented in figure 1.1.

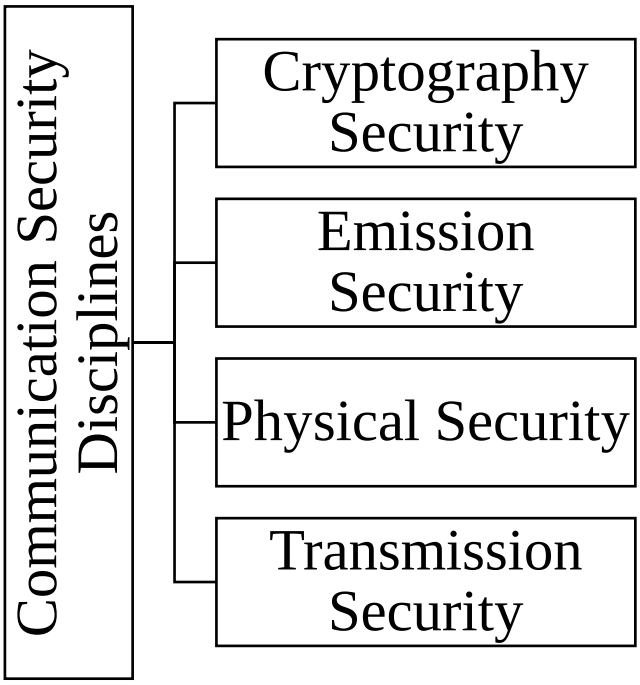

Figure 1-1 Communication Security Disciplines

Over the recent years, considerable researches have been taken to develop new chaotic or hyperchaotic systems and for their promising applications in real-time encryption and communication. In fact, it has been shown that chaotic systems are good candidates for designing cryptosystems with desired properties

1.2. Literature Review of Hyperchaotic Cryptography Systems

Image encryption based on chaos and hyperchaos systems has been taken a huge amount of interest from the cryptographic scientists and researchers, where there is ongoing demand for new robust and complex techniques and algorithms to be used for image encryption cryptographic systems to protect the user's personal images and data. In this section, various approaches, techniques and algorithms that used in the literature for image encryption based on hyperchaotic systems will be discussed.

In 2009, S. Liu, J. Sun, Z. Xu proposed new algorithm for image encryption systems. The proposed algorithm adopts stream cipher techniques, where the key sequence will be generated by using two chaotic logistic maps. The first logistic map will operate in the background to update the system parameters of the second logistic map, which its output will be considered as the keystream that will be used in chaos masking of the plaintext image. The suggested algorithm's performance is tested using histogram, neighboring pixel correlation, and other security analysis parameters, and the findings show that the algorithm is ready for usage in practical applications. [3]

In 2010, Y. Tan and W. Zhou proposed a new evaluation strategy for image scrambling. Firstly, the authors presented an idealized image scrambling is presented in which the histogram constructed in a plain level; next image scrambling is divided into sub-images in order to get a certain histogram sequence. Finally, the relationships between each two histogram sequences are computed in order to evaluate the degree of image scrambling. [4]

In 2010, A. Pande and J. Zambrano introduced the chaotic stream cipher's conception and FPGA implementation. The real-time stream cipher embedded system of the proposed technique implements the modified logistic map. Comparing the modified logistic map to the standard logistic map reveals certain advantages. Up to 16 bits of encrypted data can be delivered by the system's design for each clock cycle. In this design, the Xilinx Vertix-6 FPGA is used. [5].

In 2011, M. Prasad and K. L. Sudha presented pixel scrambling based image encryption system to be used for image cryptosystems. Where the chaotic map sequences will be used to shuffle the positions of the plaintext image to generate ciphertext scrambled image in the encryption side, while the same chaotic sequence will be generated on the decryption side to retrieve the

original pixel positions in order to get the original plain image. Two chaotic maps are adopted in this design, which are Lorenz map and Henon map, to figure out the most effective and suitable map for the system design. [6]

In 2011, Z. Yong proposed a new encryption algorithm based on the chaotic logistic map and cheat image message for designing image cryptosystems, where the proposed algorithm provide a novel confusion and diffusion for the input plaintext image. The secret key to the system is represented by the initial conditions and logistic map control parameters. The algorithm uses one of the most used images in the public network as a cheat image in corporation with a chaotic sequence that generated from the chaotic map to encrypt and decrypt the images. [7]

In 2011, Z. Tang and X. Zhang proposed an image encryption algorithm based on Arnold chaotic transformation and random strategies and has no relations or limitations on the size of the plaintext image. The algorithm consists of three phases, where in the first phase the generation of random numbers or sequences from the chaotic map is taking place. In the second phase the plaintext image will be divided into random squares. Finally, the phase of scrambling in which each image pixel of each block will be scrambled based on generating sequences in the first phase. [8]

In 2012, I. S. I. Abuhaiba, and H. A. AbuGhali suggested a two-dimensional chaotic map-based symmetric key cryptosystem and external 128-bit secret key. The proposed algorithm consists of a set of operations, such as swap, linear shift and XOR, that used to convert the plaintext image to ciphertext image and vice-versa. Theoretical analysis of the results that obtained from the proposed system verify the superiority to previous cryptosystems and it can cope with different types of attacks. [9]

In 2013, O. M. Abu Zaid proposed an image-based cryptosystem using Henon chaotic map, Baker map and Arnold cat map. In this proposed algorithm a combination of shuffling pixel positions and changing the pixel values is performed to hide or dissipate the relationships between the plaintext images and ciphertext images. The key sequence that generated from the chaotic maps are adopted for pixels shuffling and changing values. [10]

In 2013, S. Sadoudi, and A. Dandache designed and implemented secure hyperchaotic wireless communication system based on FPGA platform, where image of size 128×128 pixels is encrypted using chaos masking, the encrypted image has been sent wirelessly using the Zigbee wireless protocol. In the other hand the transmitted image is received and decrypted back in the receiver side. The synchronization between the transmitter and receiver is implemented through the use of dynamic feedback modulation technique DFM. The implementation results proof that the system can be used in real world wireless communication applications such as the wireless sensor networks. [11]

In 2014, G. Hanchinamani and L. Kulakarni developed a novel method for image encryption based on a pseudo-Hadamard transform and a 2-D Zaslavskii map. Permutation and diffusion are the two processes that make up the encryption process. Key-sensitivity, key-space, statistical, entropy, differential, and performance analysis are used to extensively examine the security and performance of the proposed approach. [12]

In 2015, J. Zhang proposed a new algorithm in which the combination of a chaotic map with a hyperchaotic system is presented. The proposed algorithm has been adopted for image encryption, where the Arnold Cat chaotic map is used in combination with four dimensional Lorenz hyperchaotic systems. The Arnold Cat map is used to scramble the pixel position, while the hyperchaotic Lorenz system is used to change the pixel value by means of hyperchaos

masking. An extensive analysis has been performed for the scheme to test its performance, the analysis results were satisfactory and the system could be used for image encryption effectively. [13]

In 2016 A. K. Abdul Hassan presented a new hyperchaos system with high capacity, security, and efficiency that is based on Hénon and Logistic maps. The key for diffusion in an image encryption algorithm is generated using the suggested hyper chaos system. The security analysis of the proposed hyper chaotic system-based picture encryption method demonstrates that it has a wide key space, strong encryption key sensitivity, and strong statistical properties. Different applications are suitable for encryption and decryption times. [14]

In 2017, Z. Yong proposed new image encryption system based on four dimensions hyperchaotic oscillators. The proposed algorithm is consisted of two diffusion operations, one pixel scrambling operation, and three matrix rotation operations of 180 degrees. The hyperchaotic system is mainly used to generate the key stream or sequence to be used for diffusion processes in XOR operation. The proposed system has large encryption space and fast response which qualifies the algorithm to be used in real world applications. [15]

In 2018, H. Natiq, and N. M. G. Al-Saidi two-dimensional proposed image encryption model based on Henon map and one-dimensional sine map are merged together to construct Sine Henon Alteration Model (2D-SHAM). The new alteration model will be adopted to generate two chaotic matrices S1 and S2, which will be used for image confusion and diffusion. The proposed image encryption is consisting of two rounds, in each round the pixels of the input plain image are shuffled and diffused alternately. Firstly, the rows and columns of the plain image are shuffled sequentially by using the chaotic matrices; secondly the pixels of the shuffled matrix are diffused to ensure that the principles of confusion and diffusion are achieved. In order to maximize the

security, performance, the above procedure will be repeated in the second round. Different security analysis is performed to check the performance of the proposed system, where all the analysis results prove that the system is capable of withstand differential attacks, chosen-plaintext attack and chosen-ciphertext attack. [16]

In 2018, X. Wang, X. Zhu, X. Wu, and Y. Zhang proposed an Image encryption based on one time pad algorithm, where the inputs of the onetime pad are represented by the plain image and the chaotic logistic map. Secure Hash Function (SHA1) and Message Digest Algorithm (MD5) are used to compute the chaotic map sequence and the input image as a preparatory step for the encryption. On the other hand, to ensure the pixel diffusion process, cyclic shift function and piecewise linear chaotic maps are also adopted to provide shift numbers for the used chaotic map. The experimental and simulation results prove that the system is robust and can stand against the well-known attacks. [17]

In 2018, H. Natiq, and N. M. G. Al-Saidi introduces a unique algorithm for systems of colored picture encryption. The algorithm combines one-time keys, DNA sequence operations, and spatiotemporal chaos in a unique way. The proposed algorithm is divided into three parts. In the first phase, key streams are generated utilizing a dynamic system with discrete space, discrete time, and continuous state called the NCP map based CML and the Secure Hash Function (SHA 256). The colorful image will then be divided into its three parts (Red, Green, and Blue), and utilizing the DNA encoding technique, these three images will be randomly transformed into DNA matrices. In order to accomplish row-wise and column-wise permutation, the DNA matrices are afterwards concatenated into a single matrix. To perform DNA addition, subtraction, and XOR operations on these DNA blocks, the permuted DNA matrix is split into three equal-sized blocks. In the final step, the DNA

decoding procedure transforms the DNA matrix back into the decimal matrix. The proposed algorithm can be utilized for real-time encryption systems and is capable of withstanding a variety of security threats, according to the security study. [18]

In 2018, M. A. Al-Khasawneh, S. Hasan, and A. A. Bakar proposed a new image algorithm that's based on the combination of multiple chaotic maps, where logistic map, Henon map, and Gauss iterated maps are used together. Firstly, the key stream will be generated from these chaotic maps as follows, the logistic map output will be considered as an input to the Gauss iterated map which will generate its output stream, in the same time Henon map also will generate its output stream. The output of Gauss iterated map and Henon map will be combined with each other using XOR operation. The XOR operation output represents the generated key stream. In the other hand the plain image will be decomposed into its three image colors (Red, Green, Blue), then the encryption process between the image matrices and the generated keystream will take place using XOR operation. After the ciphered images are generated, the three colors ciphered images are recombined to form a single ciphered image. [19]

In 2019, G. Cheng, C. Wang, and H. Chen adopted a five-dimensional hyperchaotic system to construct novel image encryption system. The proposed system consists of two stages. In the first stage the input plain image is decomposed into its three-color components (R, G, B), then based on the generated hyperchaotic streams a block permutation is used to exchange the pixels between the red, green and blue image matrices. In the second stage the initial condition of the hyperchaotic system is modified to generate a new key stream, the generated key stream, and then used to diffuse the pixel values of the formatted images. Statistical analysis of the proposed system is carried out,

where the results proof that the system can cope with different attacks, chosen plaintext attack, entropy, and noise attacks. [20]

In 2019, X. Zhang, L. Wang, Z. Zhou, and Y. Niu presented a revolutionary design for image encryption techniques based on the space-filling property of the Hilbert curve and the infinite property of the H-geometric fractal shapes. First, the secure hash algorithm 3 is used to determine the hash value of the plain image (SHA3). Rossler chaotic system and piecewise linear chaotic map (PWLCM) initial values are then set using the hash value. The pixels of the input plain image, on the other hand, are subsequently scrambled using the created chaotic sequence, altering their location and value. The Hilbert cure and H-fractal are combined in the second encryption phase to scramble and alter the pixel position and pixel values. Finally, to maximize the security level of the proposed algorithm the confusion and diffusion characteristics should be enhanced and this can be performed by adopting ciphertext feedback. This algorithm possesses a high security level due to its complexity and cascaded encryption phases, so it can be used for highly sensitive information transmission such as military and judicial purposes. [21]

In 2019, H. A. Abdullah and H. N. Abdullah proposed a new chaotic map known as Nahrain. This chaotic map has been adopted for image encryption system by means of pixel position scrambling and pixel value change. The proposed system is simulated using Matlab environment, Altera Quartus, and ModelSim. In the other hand the system is also implemented using the FPGA evaluation board. The NPCR and UACI tests as well as the information entropy are measured for the encrypted images, where NPCR of 99.76% is obtained which is too close from the ideal value (100%), in the other hand the information entropy of 7.9964 is also obtained that is very close from the entropy ideal value of (8). [22]

In 2019, T. M. Hoang presents a novel image encryption system based on the perturbation of the nonlinear systems dynamics, where the logistic map is perturbed dynamically by changing the control parameter bit level value after each iteration during the encryption/decryption process. This perturbation in the logistic map dynamics leads to expand the key space up to 378 bits and this is a very large space compared with other similar cryptosystems. Consequently, this huge number of the key space makes the proposed cryptosystem more robust against the statistical attacks and can resist the brute force attacks. [23]

In 2019, F. S. Hasan and M. A. Saffo proposed a new chaotic algorithm that is dedicated for stream cipher image encryption system. In this algorithm three chaotic map, which are Logistic map, Lozi map, and Tent map, are solved numerically and then combined together with XOR operation to obtain a new Pseudo Random Bit Generator system (PSRG). The new binary random sequence will be used for image encryption by changing the image pixel value. Fixed point 32-bit mathematical implementation has been adopted in this design, where the new algorithm is designed, simulated using Matlab platform. The simulated algorithm has been implemented using the FPGA evaluation board. Different randomness tests are used to test the performance of this design such as histogram, correlation, and some differential attacks (such as NPCR and UACI). The test results show that the system is operating effectively and it can cope with attacks. [24]

In 2019, J. Wu, J. Shi and T. Li proposed an image encryption algorithm based on DNA level diffusion, hyperchaotic systems, and pixel level filtering with kernels of variable shape and parameters. The proposed algorithm consists of four stages. Firstly, hyperchaotic sequence will be generated to provide the hyperchaotic sequence to be used in the subsequent operations. Next, in the second stage the input image pixel value will be changed by means of dynamic filtering. In the third stage, the pixels will be scrambled. To modify the

locations of the pixels, global bit-level scrambling will be used. The resulting bit stream will then be encoded into DNA level data. To modify the values of the pixels, DNA level diffusion is then used once more. The proposed algorithm proof that it can achieve high security level and it can cope with different attacks. [25]

In 2019, H. Liu, Y. Zhang, A. Kadir, and Y. Xu proposed a new technique to mitigate the dynamical degradation and enhance the randomness of the chaotic systems by injecting a pulse into the control parameters of the Lu hyperchaotic system. The hyperchaotic system with its injection will be adopted for image encryption, where three-bit wise operations are used to encrypt the three image layers rapidly with generated sequences from the hyperchaotic system. The systematic analysis of the proposed algorithm is taking place, where the NPCR test of the encrypted image is 99%, which close to the optimal NPCR value (100%), while the UACI is approximately 33.3 which also acceptable value. On the other hand the key space of the proposed algorithm is approximately 10^{84} ($>2^{256}$) which means that the system can resist the brute force attack effectively. [26]

In 2019, S. Zhu and C. Zhu proposed a new image encryption algorithm based on five dimensional hyperchaotic maps. The hyperchaotic map has been constructed by the combination of logistic map and 3 dimensional Lorenz discrete maps. The algorithm has two phases, the first phase consists of image pixels scrambling in which pixel positions will be changed based on the generated chaotic sequence. In the second phase, two diffusion rounds will be carried out to change the pixel values and to strengthen the algorithm against chosen plaintext attacks. Many security analysis tests are carried out to test the performance of the proposed algorithm. The test results demonstrate the practicability of the cryptosystem. [27]

In 2020 M. Hanif, R. A. Naqvi, S. Abbas, M. A. Khan, and N. Iqbal presented a totally novel image encryption algorithm to be used for the systems that encrypting bunch of images at the same time. The proposed algorithm accepts an N number of plain images as input, then the algorithm randomly choosing two images from the bunch of input images, from the selected images a randomly two rows is selected and then swapped. In the other hand, two columns also randomly select from the images and swapped. The process of choosing random images, choosing random rows and choosing random columns has been repeated for many times in order to ensure the principle of pixels diffusion and confusion in the ciphered images. The intertwining logistic map and the improved piece wise linear chaotic map have been adopted in this algorithm for the purpose of generating random number streams to be used for the process of randomly choosing images, rows and columns for swapping operation. The security analysis of the proposed algorithm vividly proves its robustness and resistance against different cyberattacks and it can be used for real world sensitive communications. [28]

In 2022, I. Yasser, A. T. Khalil, and M. A. Mohamed proposed an image encryption model for cloud-based Internet-of-health-systems. The proposed model utilizes novel chaotic map. The image pixels are permuted and diffused based on the value of the new chaotic sequences generated from the chaotic map. Different statistical tests have been carried out to test the performance of the proposed system such as UACI, NPCR, MSE and other tests, which shows that the system is strongly enough to cope with new cyber-attacks. [29]

In 2022, J. Arif et al proposed a novel image encryption system based on chaotic attractors. The proposed system is consisting of a successive permutation and substitution operations with use of substitution box (S-Box). The efficiency of the proposed system has been proofed to be resilient and flexible against the statistical and differential attacks. [30]

In 2022, B. Balakrishnan, and D. Mubarak proposed a chaos-based image encryption system consist of two consecutive set of confusion and diffusion operations. Two-dimensional Logistic Adjusted Sine Chaotic Map is utilized in the proposed technique (2D-LASCM). The proposed system is derived from the Sine wave and the Logistic map. The confusion and diffusion operations are driven form a shuffle indexed matrices that constructed from the 2D-LASCM. The proposed system proof that the encrypted images have high encryption efficiency and high ability to cope with statistical and differential attacks. [31]

In 2022, Ya Wang, and Xinyu Li presented a new a new 5-D hyperchaotic system with high order nonlinear terms. The maximum Lyapunov exponent are close to 2, and there was a better permutation entropy index while a valid chaotic sequence could be generated in three cycles in the FPGA. [32]

1.3. Problem Statement

As aforementioned in the literature review, it is obvious for the researchers that there are some cryptographic requirements that are not being covered yet such as:

1. The need for designing high randomness, unpredictable generator for the bit stream generation.
2. Developing a new hyperchaotic system generator to enhance the performance of the encryption/decryption algorithms.
3. Design an embedded system based on FPGA platform to implement the encryption machinery and decryption machinery.
4. Developing high speed, immunity to noise synchronization circuit to provide the necessary synchronization between encryption and decryption machineries.

1.4. Research Objectives

The main aim of this work is to design an efficient, robust, and high-speed an image encryption algorithm based on hyperchaotic oscillators for with a high degree of randomness and unpredictability. The main objective of this work could be summarized as follows:

1. Proposing, designing and FPGA implementation of an efficient and high randomness encryption algorithm based on hyperchaotic systems for image encryption purposes.
2. Designing and FPGA implements a synchronization method to provide the necessary synch between the encryption and decryption subsystems.
3. Employing the designed algorithm for secure image communication and implementing the overall system in FPGA board.

1.5. Research Main Contribution

The main contribution of this work presented in the list below:

1. Developing an image encryption method based on a new, high-dimensional hyperchaotic system with high randomness, speed, and predictability and the most positive Lyapunov exponents possible.
2. Designing a multi-dimensional hyperchaotic system-based picture encryption technique that uses various dynamic sizes through the use of XOR mixing.
3. Utilizing the chaos switching technique to design an image encryption algorithm based on parallel high-dimensional hyperchaotic systems and multi-dimensional hyperchaotic systems.
4. Developing a cascaded hyperchaotic system-based image encryption algorithm.

5. Designing an adaptive feedback controller circuit to provide the necessary synchronization between transmitter and receiver systems.
6. FPGA implementation of the proposed encryption/decryption algorithms.
7. Building a transmitter and receiver based on FPGA hardware platform.

1.6. Dissertation Organization

This thesis is organized into five chapters, in addition to the appendices. The organization is explored as follows:

- Chapter one: introduces an overall introduction for the thesis trends, chaos/hyperchaos and a literature survey for the most well-known image-based chaos and hyperchaos cryptography systems, thesis objective and the main contribution.
- Chapter two: introduces a survey about the well-known cryptography systems, chaos/hyperchaos nonlinear systems, their types, their lyapunov exponents, the numerical method used to solve the chaos and hyperchaos systems (nonlinear systems), and finally, a variety of techniques that used to measure the encryption systems performance and quality is presented.
- Chapter three: the design, simulation and FPGA implementation of the proposed algorithms for image are presented in chapter three.
- Chapter four: this chapter presents the simulation and implementation results of the designed algorithms in chapter three with the results of the measuring quality techniques that used to check the performance of the designed algorithms.
- Chapter five: show the conclusions of this thesis and present some suggestions and future works.
- Appendices: it includes all the software codes necessary to run the Field Programmable Gate Array (FPGA) board (VHDL codes) and

the necessary Matlab codes that used to implement the security analysis techniques with some necessary data sheets.

CHAPTER TWO

Cryptography Chaos and Hyperchaos

Chapter Two: Cryptography, Chaos and Hyperchaos

2.1. Introduction

This chapter presents the most important aspects that used in the field of cryptography and network security systems such as: diffusion/confusion principles, block/stream ciphers, symmetric/asymmetric encryptions, security services and purposes. In the other hand, chaos and hyperchaos nonlinear systems are presented, the classes of these systems which can be divided into flow (continues systems) such as Lorenz, Lu, Rossler systems and maps (discrete systems) such as Henon, Logistic, Baker maps. Also, the lyapunov exponents, and general characteristics of these nonlinear systems are presented. Finally, numerical integration methods used for solving the nonlinear systems as well as the encryption quality measuring techniques are presented.

2.2. Cryptography

The term cryptography can be defined as the collection of mathematical techniques and methods that used to protect and safeguard the user's sensitive data and information from unauthorized access during the data transmission through the unsecure public channels (such as Internet). User's original message is called plaintext message (uncoded message), while in the other hand the coded message is called ciphertext. The process that converts the plaintext to ciphertext is known as the encryption process, while the reverse process that convert the ciphertext to plaintext is called decryption process, as shown in figure 2.1. [33]

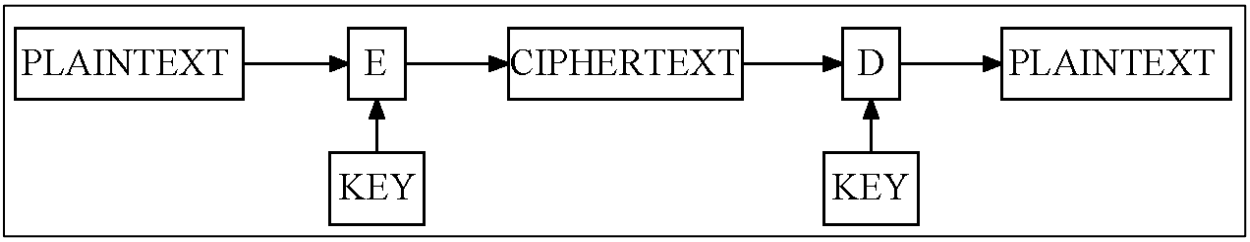

Figure 2-1 Classical Encryption, Decryption System [2]

Cryptographic algorithms are very important and essential part in any communication process, where using these algorithms in the communications provide data integrity, confidentiality, authentication and non-repudiation, which are the most important security services in any transmission process [34].

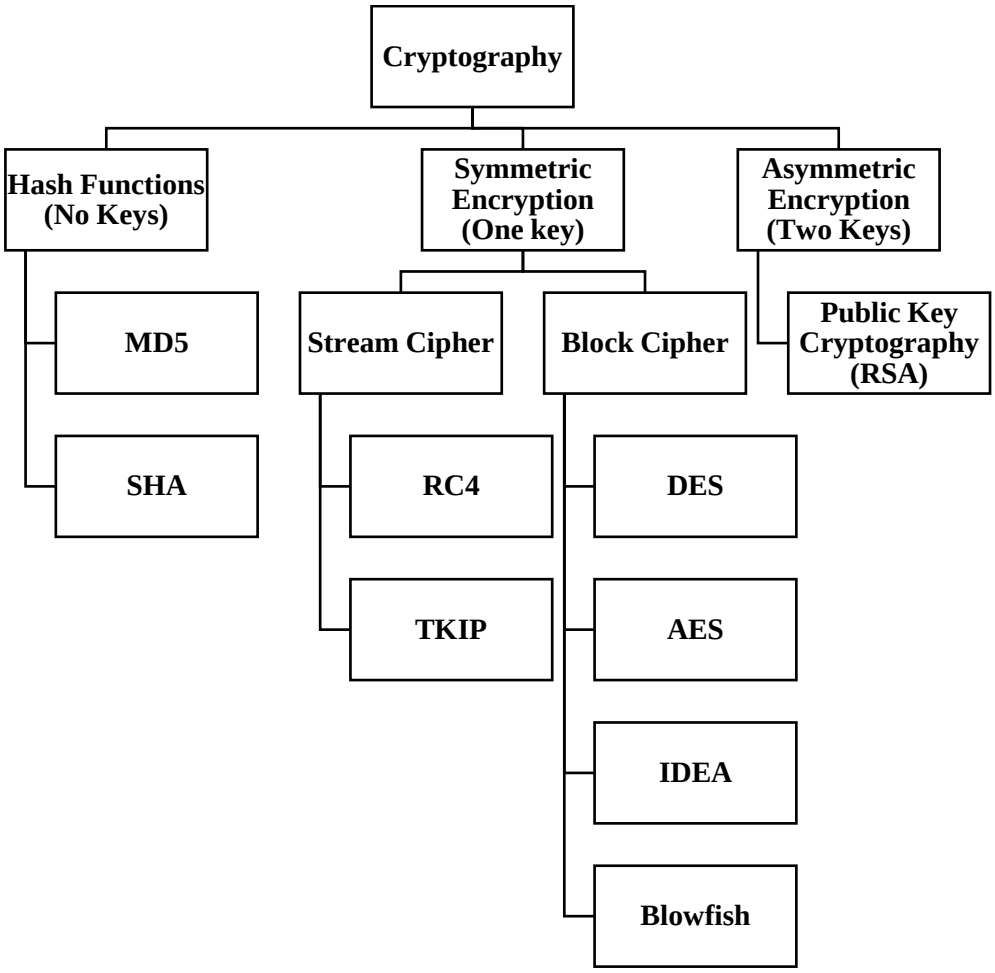

Figure 2-2 Schematic Representation of Cryptography Systems Classification [34]

Cryptographic systems can be classified into many types according to several factors, all of them controlled by complex mathematical algorithms and keys. Based on the number of keys that used during the encryption/decryption process, cryptographic systems can be classified into three types: No key cryptosystems (such as Hash Function), one key cryptosystems (such as private key or symmetric encryption systems like DES and AES algorithms), and two key cryptosystems (such as public key or asymmetric encryption systems like

RSA algorithm). Figure 2-2 depicts the well-known types of the cryptography systems. Cryptography systems can be characterized based on three independent factors:

- Type of Operation: The type of operations that used to transform the plaintext to ciphertext or vice versa. Essentially, the encryption and decryption algorithms are based on two basic principles, either substitution or transposition. In substitution operation, each element in the plaintext (bit, byte, letter, group of bits, bytes or letters) is mapped into another element, while in the other hand the transposition operation all the plaintext elements are rearranged. The main requirement is that no information be lost (all operations are reversible to get the original data back). [35]
- Number of used keys: The cryptography systems can be classified into three classes according to the number of used keys by both sender and receiver. These classes are, no key cryptography systems (such as Hash functions), one key cryptography systems in which the sender and receiver uses the same key (such as symmetric, conventional encryption or secret-key systems), and two keys cryptography systems in which the sender and receiver uses different keys for encryption and decryption (such as asymmetric, public key encryption and two-key systems). [36]
- Plaintext processing method: Usually the plaintext is processed either a group of elements or one by one element. In group of elements the input is represent one block of element at each time and the output is a block of elements for each input block, this type of processing is called block cipher. In the other hand in one-by-one element processing the system accept only one element as an input, and producing one output element as each time, and this type is known as stream cipher systems. [37]

2.3. Hash Function, Symmetric, Asymmetric Encryption Systems

As aforementioned above, the cryptography systems can be classified into three classes according to the number of keys that used as follow:

2.3.1. Hash Function Cryptography:

It represents one of the most useful algorithms that uses no-key for encryption. The hash function is a mathematical algorithm that accepts an input plaintext data with an arbitrary size and returned a unique hash value (hash codes) as an output with a fixed data size. [17], [38]

2.3.2. Symmetric encryption systems:

This type of encryption system is also known as secret key encryption systems because it adopts only single key for both sides sender and receiver. This type is the most frequently used type because of it offer high data rates (so it called also conventional encryption systems). The conventional encryption system is consisting of five ingredients which are plaintext, ciphertext, encryption algorithm, decryption algorithm and the secure key. As shown in figure 2-3, the conventional encryption system is consisting of encryption machinery and decryption machinery, both operated using the same key. The plaintext message (readable or interpreted) treated as an input to the sender of encryption party, based on the mathematical algorithm and the secret key the plaintext message is translated into ciphertext message (unreadable or uninterpreted). The unreadable message can be transmitted over the public channel to the receiver side. [39] [40]

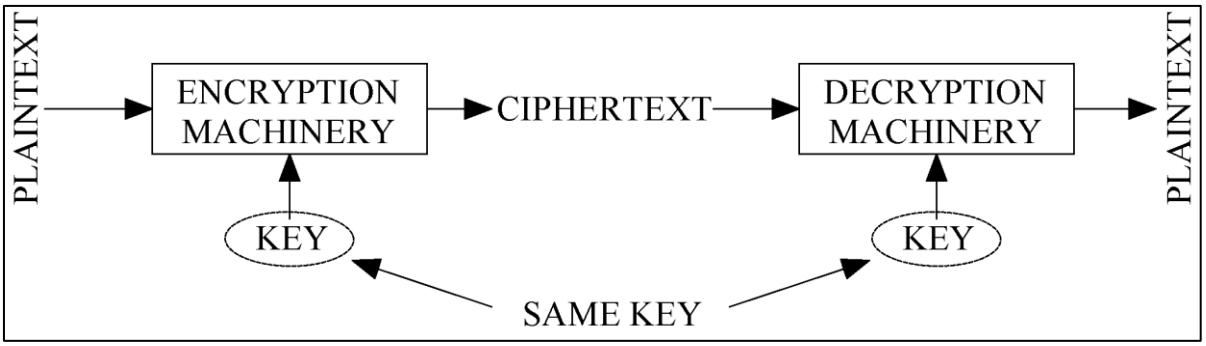

Figure 2-3 Symmetric Encryption System

In the receiver side, the decryption machinery uses the same secret key and mathematical algorithm. The received ciphertext message will be treated as an input and the message will be decrypted. The original plaintext message has been recovered. This system is widely used over the electronic universe due to its high data rates that reach a hundred of megabytes/seconds and the short secret key. There are three basic requirements for designing a symmetric encryption system:

- Designing a strong and robust mathematical algorithm to be used for encryption/decryption process.
- Produce a secret key that will be used for encryption and decryption.
- Develop a mechanism share or distribute the secret key between the sender and receiver or receivers.

Since the same key is used by the encryption and decryption algorithms, the key should be shared between the two parties in a completely high security environment and this considered the main disadvantage of the symmetric key encryption systems especially in the huge networks, where a third party should be used to manage the sharing process between the sender and receivers which known as a Trusted Third Party (TTP). DES, AES, Blowfish and RC4 are an examples of symmetric cryptography systems. [41]

2.3.3. Asymmetric encryption systems:

The asymmetric encryption systems or public-key cryptography is the true revolution in the cryptography and network security field. The key for encryption in public-key cryptography is different from the key for decryption, which is the main distinction between these two types of cryptography systems (but they are related). Each user has two keys: the encryption key, which is known to everyone and is entirely secret; and the decryption key, which is related to the encryption key but is only known to the user. The public key cryptography systems overcome the problems of the key distribution and management presented in symmetric cryptography algorithms but in the other hand the asymmetric cryptography systems required more computational power due to the complexity in the mathematical algorithms and this lead to make the asymmetric encryption systems slower than the symmetric encryption systems. Elliptic Curve, RSA, DSS, and Diffie-Hellman are an examples of public key cryptography systems. [42] [43]

2.4. Block Cipher and Stream Cipher

According to the fashion in which the plaintext is processed, the cryptography systems can be classified into two classes as follows:

2.4.1. Block Cipher Systems:

In this type of cryptography systems, the input plaintext data is divided into blocks of fixed size (typically 64 bits, 128 bits or 256 bits), these blocks of data are processed and encrypted block by block producing a block of ciphertext with same data size.

2.4.2. Stream Cipher Systems:

This type of cryptography systems relies on serial data principle, where the plaintext data is received bit by bit or byte by byte producing bit or byte of ciphertext data. This way of data encryption is more suitable for real time

applications than the block cipher systems because that the received data are immediately encrypted and transmitted without any delay in the sender node. In other hand the sender in the block cipher systems has to wait until a block of data to be received completely. [34], [41]

2.5. Diffusion and Confusion

The statistics of the repeating letters in any message could provide a useful information for the intruders, where a statistical analysis can be performed to find some information about the algorithm that used in the cryptography system, or the used secret key. In order to frustrate the statistical analysis, two principles are suggested, which are the diffusion and confusion. [44]

In order to achieve the diffusion principle's goal of concealing or dissipating the original message (plaintext) statistical structure into long-range statistics of the ciphertext message, each plaintext symbol must be affected by many ciphertext symbols, which is equivalent to having each ciphertext symbol be affected by many plaintext symbols. The connections between the plaintext and ciphertext communications will be concealed by this process. By repeatedly carrying out successive data permutations (transposition), followed by replacement (representation) operations, diffusion can be implemented.

In order to frustrate and hinder intrusion attempts to determine the encryption key from the ciphertext message statistics, the confusion tries to make the relationship between the statistical structure of the communication system and the secret key as complex as possible. Making the essential application as difficult as possible by utilizing difficult substitution (representation) methods is one way to create confusion.

2.6. Cryptography Purposes (Services)

The employment of a cryptography algorithms to a communication system provides a package of security services that ensures an adequate security level

for both the communicated systems resources and the transmitted messages over the public unsecure channels. The services are: authentication, access control, confidentiality, integrity, non-repudiation, service availability and reliability. [33]

2.6.1. Authentication

The authentication service provides the necessary assurance of the received message origin and the identity confirmation of the two communicated entities [45]. With respect to the communication and networking, there are two types of systems that are in use nowadays, which are the connection oriented and connectionless communications; consequently, there are two authentication services regarding to the types of communications:

2.6.1.1. Peer entity authentication:

This type of authentication is taken place in the connection-oriented networks, where it provides the necessary corroboration of the identity of the two communicated entities. Peer entity authentication is provided during the link establishment between the communicated entities and/or at different periods during the data transfer process.

2.6.1.2. Data origin authentication:

In the connectionless networks the cryptography system authenticates the origin or the source of the received data, where authentication of both entities does not happen. A good example about this type of authentication is the live broadcast of the football matches.

2.6.2. Access Control

In the network security context, the access control principle represents the ability to control, and limit the access to a server system using communication links. In order to achieve this service, each user or entity should be identified

and authenticated using username and password before getting access to the destination system. [46]

2.6.3. Data Confidentiality

This service provides the necessary privacy and protection for the data transmitted over the public channels and the systems resources by a means of data concealment techniques. The confidentiality service protects the data from the unauthorized access using complex mathematical algorithms. [47]

2.6.4. Data Integrity

This service ensure that the messages are kept intact and received as they sent without any duplication, deletion, reordering, insertion, manipulation, or even modification, where the integrity service can identify and address the unauthorized data changes that take place by the unauthorized users to protect the data intact. [47]

2.6.5. Nonrepudiation

The nonrepudiation service prevents the sender and the receiver from denying the transmission or reception of the messages. Where, if a message is sent from transmitter to the receiver, the receiver can prove that the alleged sender in fact who sent that message. In the other hand, the sender can prove that the alleged receiver in fact receive that message.

2.6.6. Service Availability and Reliability

The availability and reliability service ensures that the system resources are accessible and usable by the authorized entities upon their demands. In some kind of attacks, the attacker tries to undermining the capabilities, and the systems resources to prevent the authorized entities from using the data. This service preserves the data available and grant the quality of data for the entities.

2.7. Cryptographic Systems Attacks

The security attacks can be classified into two types according to their effects, which are passive attacks and active attacks. [15]

2.7.1. Passive Attacks

In this type, the opponent or attacker tries to eardrops or monitors the data transmission over the public channel (without affecting the system resources). The ultimate goal of the opponent in this attack is to get information that is being transmitted. Passive attacks can be used to violate the message contents and discover the traffic analysis. [48]

2.7.2. Active Attacks

Active attacks involve a modification for the transmitted data stream or creation of false data stream, where it can be subdivided into four types as follows:

2.7.2.1. Masquerade:

This attack is taken place when one entity pretends to be a different entity.

2.7.2.2. Replay:

The attacker (third entity) in this attack tries to eavesdropping on a communication link and capturing some encrypted messages. These encrypted messages can be retransmitted to produce an unauthorized effect on the communicated entities or systems.

2.7.2.3. Modification of Messages:

Simply, the data stream is transmitted into serial segments, a modification attack is taken place when the attacker altered one or more of these legal segments, reorder them, or even delayed to produce an unauthorized effect. [49]

2.7.2.4. Denial of Service:

This attack inhibits or prevents the normal performance of a specific system due to the unauthorized effects by the attacker. A good example of this attack is disruption of the entire communication network by overloading it with huge number of messages to degrade its performance or even disabling it. [46]

The characteristics of passive attacks are very difficult to identify or uncover, yet they may be readily avoided. Active attacks are the exact opposite. On the other hand, because there are so many different types of network vulnerabilities, it is very challenging to stop active attacks. [50]

2.8. Chaos Theory

Chaos is a nonlinear system that well-known to have unpredictable behavior and extremely sensitivity to the initial conditions and system parameters. More precisely. Chaos systems are described to predicted in the near term and appear to be unpredictable after a while. Chaos system was firstly introduced 1960s by the meteorologist Edward Norton Lorenz during his attempts to simulate the dynamical equations that describe the weather behavior using small digital computer. Lorenz discovered that tiny variations in the initial conditions or system parameters could produce a huge change in the long-term outcome, and this discovery proof that the chaos systems behavior are closer to be deterministic (not random) and too difficult to predict. Chaotic systems could be described as follows:

- Chaos system is a nonlinear phenomenon characterized by a set of differential or difference equations. Chaos systems has a deterministic and unpredictable behavior especially in the long-term instances. Chaotic systems can be represented as a time series or strange attractor as shown in figure 2-4. [51] [52]

- Chaotic systems are extremely sensitive to the initial conditions and system parameters, where if there are two identical chaotic systems with a small or tiny variations between them in the system parameters or initial conditions rapidly this will lead to huge dynamical differences in their states. [11]
- Synchronization of two identical or nonidentical dynamical systems means that the trajectory (behavior) of the first system is converging to the same trajectory of the other one. At the first glance, chaotic systems seem to be a dynamical system that cannot be synchronized since these systems are defying the synchronization (due to their high sensitivity to any slight difference in initial conditions or system parameters). This conception is remained prevalent till the publication of Pecora and Carroll in [53], where the synchronization of two chaos systems has become possible.

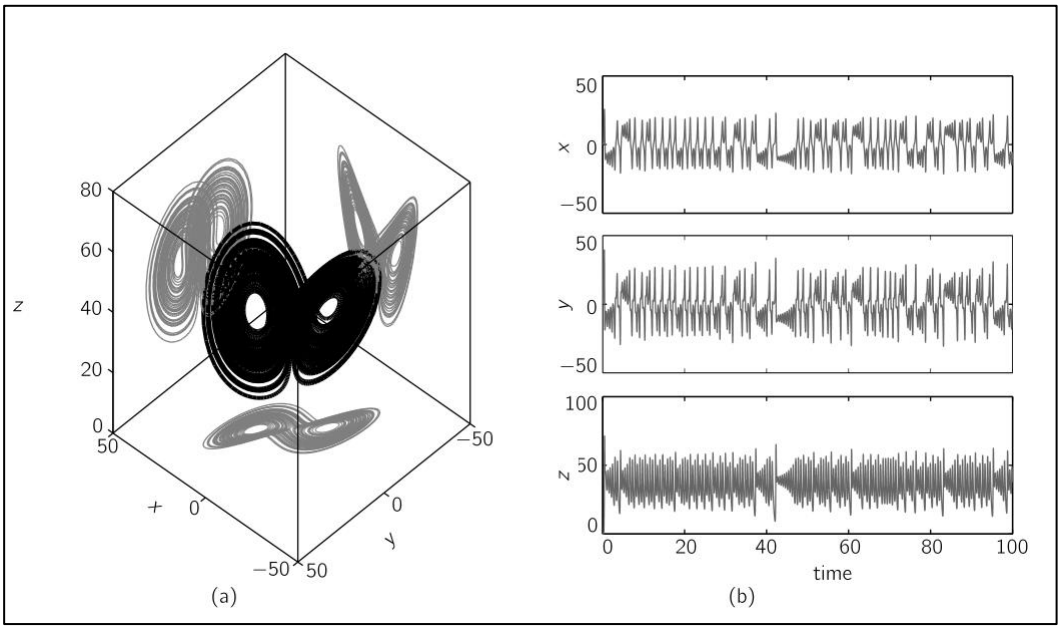

Figure 2-4 a) Strange Attractor Chaotic System, b) Time Series Chaotic System

Due to the properties that the chaotic systems have such as ergodicity, pseudo-random, non-periodicity, and the unpredictability, they are considered a perfect solution for information security technologies such as data (text, video,

images, audio) encryption, digital signature, online trading, and digital watermarking. The power of considering chaotic and hyperchaotic systems the most prominent candidates for cryptography is that their properties that match the diffusion and confusion principles that suggested by Shannon in [44] for a robust cryptographic system design such as, extremely sensitive to initial conditions and system parameters as well as they have unpredictable behavior. Traditionally, the data is modulated and demodulated by using sine carrier signal, while in chaotic systems transmitter sends carrier information signal and the receiver demodulates it to resume the signal. Since chaotic/hyperchaotic signals tend to be random signal in their nature, so modulating the data in these signals will be similar to noise signal, in the other word the intruder would think that the channel contains only noise signals not encrypted signal.

Using chaos and hyperchaos in the field of cryptography has two major directions: the first one is encryption purposes and this direction has three sub-directions, the first one using chaotic/hyperchaotic system as a pseudo-random bit generator and the data will be XORed with output of the generated bits (known as chaos masking) [54], the second sub-direction is using the data signal as an input to modulate the parameters of the chaotic system [55], the last sub-direction is known as chaos switching, in which the data signal is used as a selector to select the carrier signal among different chaotic selectors. [56] in the other hand the second direction is using the chaotic/hyperchaotic system in synchronization

The use of chaotic systems in cryptography has two major directions: - the first one is using the chaotic system for synchronization of secure communication, while the other one is generating stream cipher or block cipher by using chaotic systems.

2.9. Classical Behavior of Chaotic Systems

The behavior of the chaotic and/or hyperchaotic systems can characterized by three main aspects as follows: [57]

- **Space or Domain**: Generally, chaos systems are bounded systems, in the other word their motion is always bounded to a definite region.
- **Periodicity**: Chaos has non-periodic attractor (behavior), where, it has a complex bounded oscillations in which fully reproducible for in the initial conditions.
- **Prediction**: Chaotic system is described to be fully deterministic system, but in the other hand its prediction is almost impossible for long term cases. And this because any small perturbations or inaccuracy in the initial condition or system parameters (even if they are precisely known) will lead to exponential unpredictable and unexpected growth in its behavior.

2.10. Chaotic Systems Types

Chaotic systems can be classified into two classes, the first one is continuous time system known as chaotic flow and the second class is the time discrete system known as chaotic maps.

2.10.1. Chaotic Flow

Chaotic flow systems are considered one of the earliest observations of nonlinear behavior that was made in 1961. Chaotic flow is a time continuous system constructed by a set of differential equations, and described by a time series or strange attractor (trajectory). Chaotic flow strange attractor is referred as a trajectory which is depicted by a smooth and continuous nature. [58]. The following list are the widely used and well-known chaotic flow systems arranged chronologically.

2.10.1.1. Lorenz System

It is one of the most well-known nonlinear continuous time chaotic systems, and it was first proposed in 1963 by MIT meteorology Edward Lorenz. When high temperature (heating process) is applied equally below and low temperature (cooling process) is applied equally above, the Lorenz system was proposed to describe the fluid circulation in a shallow layer of fluid. The fluid is thought to move in two dimensions—horizontal and vertical—in a circular motion with boundary conditions. [59]

The proposed system is described by three coupled nonlinear ordinary differential equations (called coupled because that the right-hand side (RHS) of one equation contains a state variable from other equations) with two nonlinear terms (second order terms, xz and xy) as shown in equation 2.1 to 2.3 below. [59]

$$dx/dt = \alpha(y - x) \tag{2.1}$$

$$dy/dt = x(\rho - z) - y \tag{2.2}$$

$$dz/dt = xy - \beta z \tag{2.3}$$

Where, x, y, and z represent the state variable of the chaotic dynamical system, the initial value of these variables can be 0, but frequently x(0)=10, y(0)=20, z(0)=30 [60]. While in the other hand $\alpha, \rho,$ and β are the system parameters (constant values), the values of these parameters should meet the following conditions: [60]

$$\alpha, \rho, \beta > 0, \alpha > \beta \text{ and } \rho > \frac{\alpha(\alpha+\beta+3)}{\alpha-\beta-1}$$

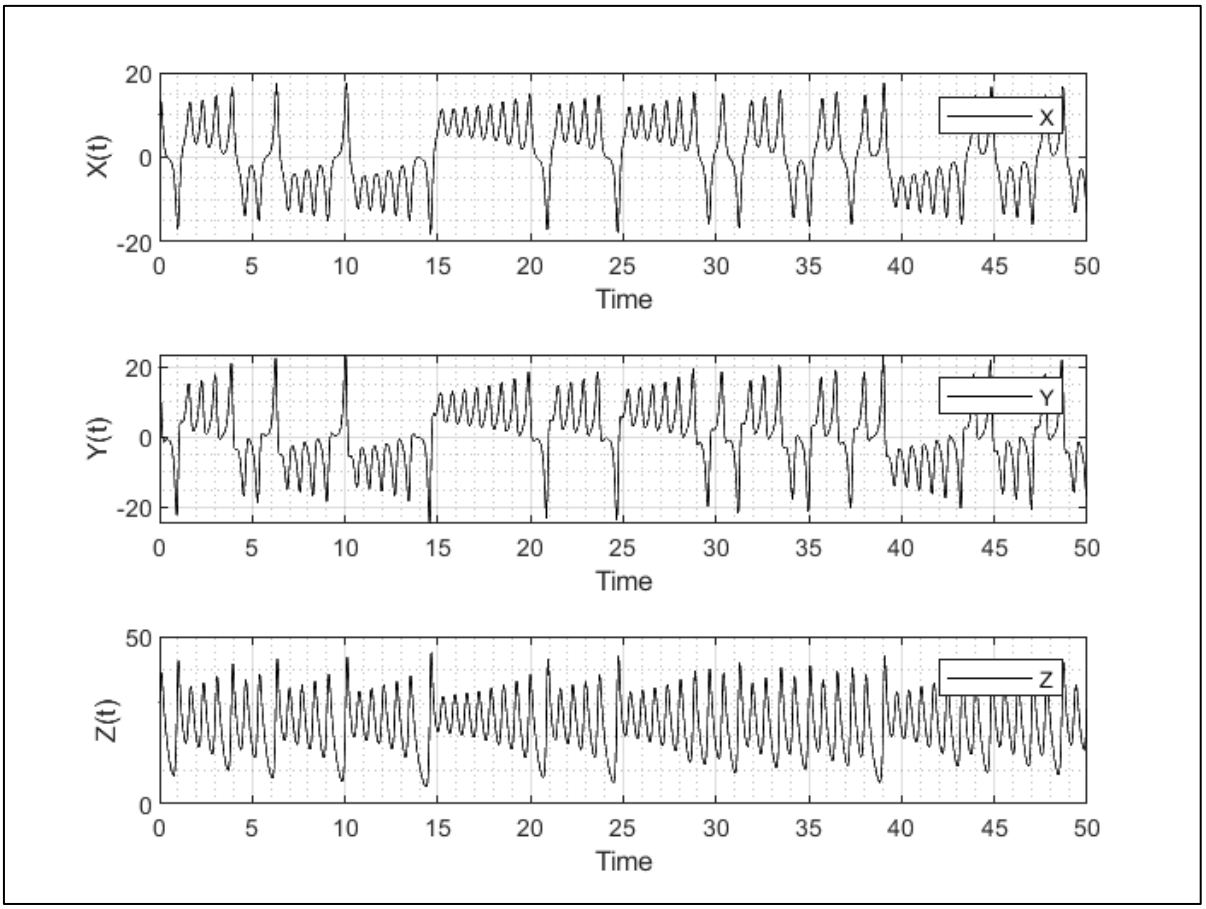

Figure 2-5 Lorenz Chaotic Time Series

The ordinary differential equations can be solved either analytically (which is very difficult and almost impossible) or numerically using one of the numerical solution methods (will described later in this chapter). The solution of this system of equations can be represented graphically either in time series as shown in figure 2-5, or by a strange attractor as shown in figure 2-6. Matlab codes that used to generate these figures are appended in the appendix A.

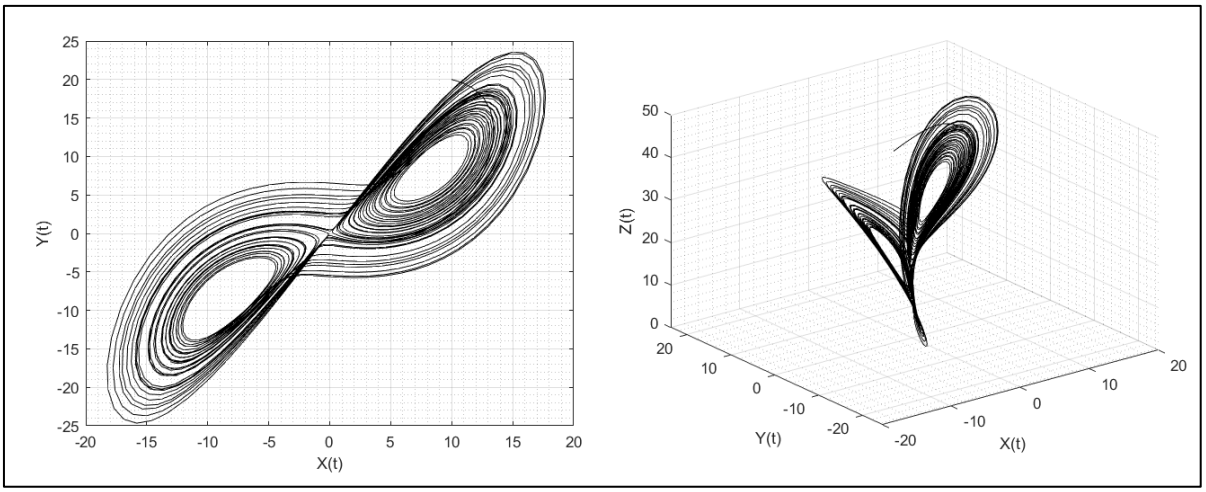

Figure 2-6 Lorenz Strange Attractor 2D and 3D

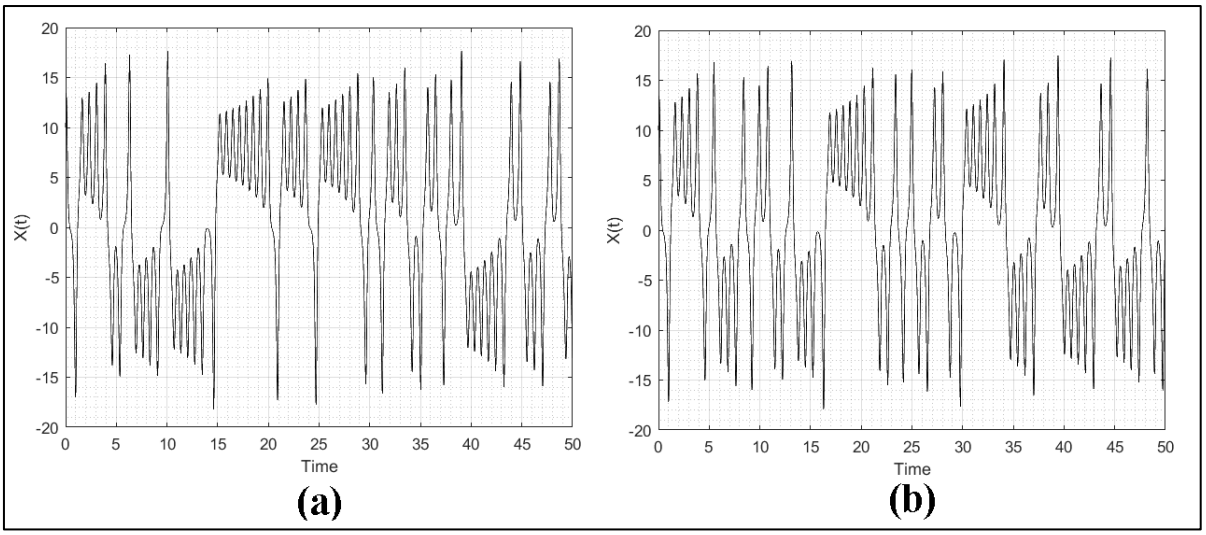

Figure 2-7 System Parameters Alteration (a: α=10,ρ=28, and β=8/3), (b: α=10,ρ=28, and β=8.1/3)

Figure 2-7 shows the x(t) time series of two Lorenz systems with modest changes in the system parameters, illustrating the remarkable sensitivity of chaotic systems to any tiny parameter perturbations. Figure 2-5 shows that a tiny change to a system parameter results in the system producing a completely new time series output.

2.10.1.2. Rössler System

Unlike Lorenz chaotic flow system, Otto Rössler proposed continuous time chaotic flow system in 1976 which contains only single second order nonlinear

term. The differential equations that describe the new system behavior is presented in equations 2.4 to 2.6 below.

$$dx/dt = -(y + z) \tag{2.4}$$

$$dy/dt = x + ay \tag{2.5}$$

$$dz/dt = b + z(x - c) \tag{2.6}$$

Where, a, b, and c are the system parameters. The Rössler system show chaotic flow system with a=0.2, b=0.2, and c=5.7. The system constant value (a) is represent the bifurcation factor, which means the parameter that makes the system reach the fix point or not (periodic system or chaotic system). [61]

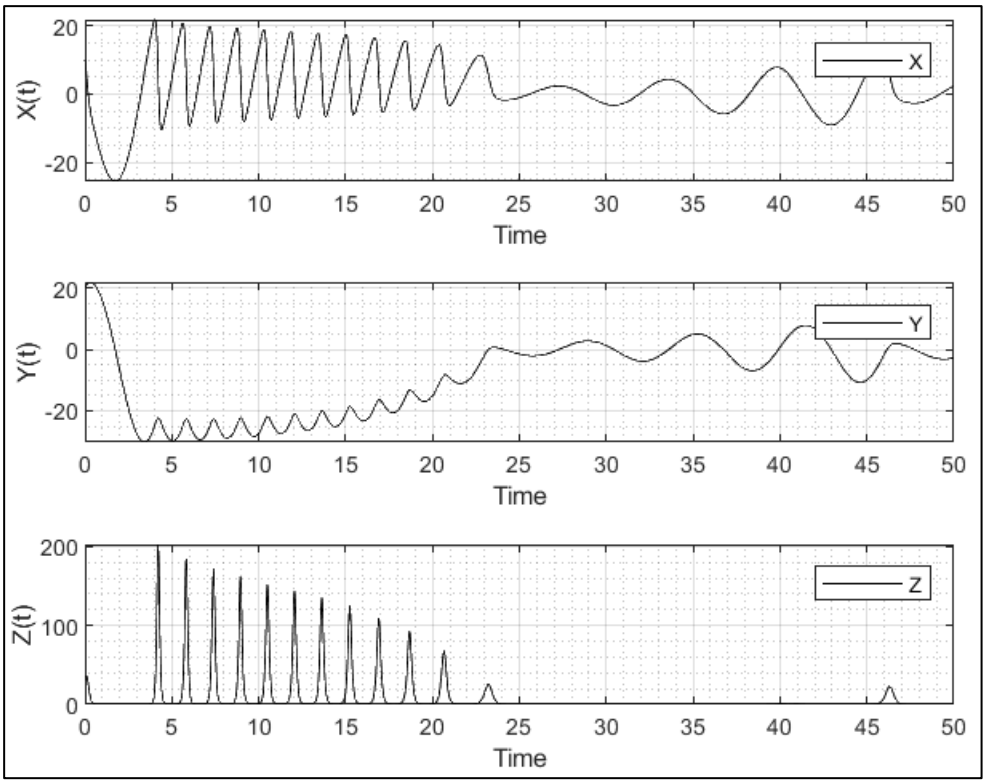

Figure 2-8 Rössler Chaotic System Time Series

Figure 2-8, show the time series of the new Rössler system. In the other hand the strange attractor of the system (2D and 3D) is presented in figure 2-9.

Rössler had several chaotic traits with other chaotic systems, including as sensitivity to initial conditions and system parameters. Figure 2-10 represent

the time series of x(t) component for two Rössler systems (that almost identical) with small variation in the system parameter as indicated in figure below this small variation leads make the system generate different time series signal.

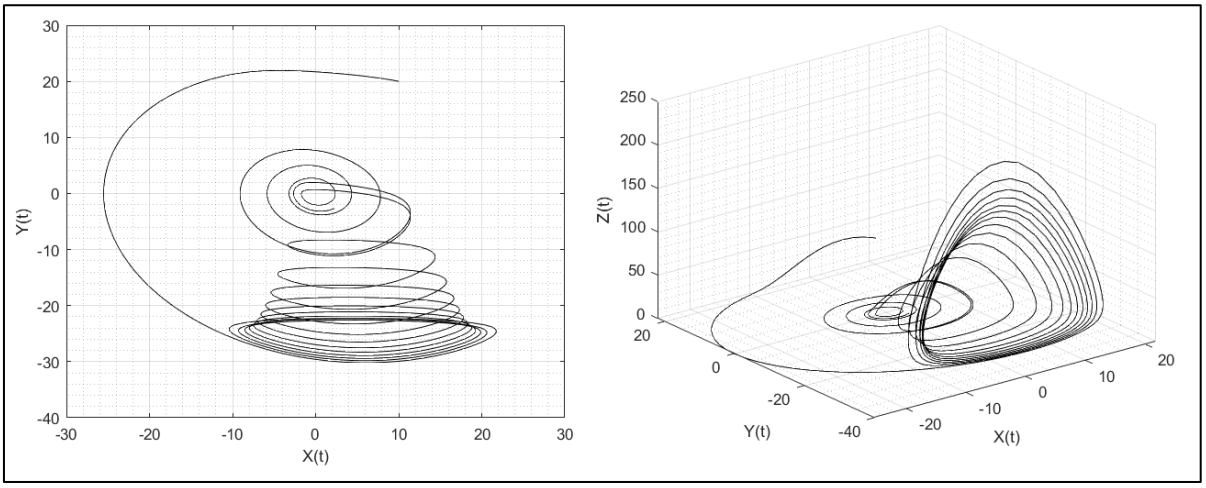

Figure 2-9 Rossler Strange Attractor 2D and 3D

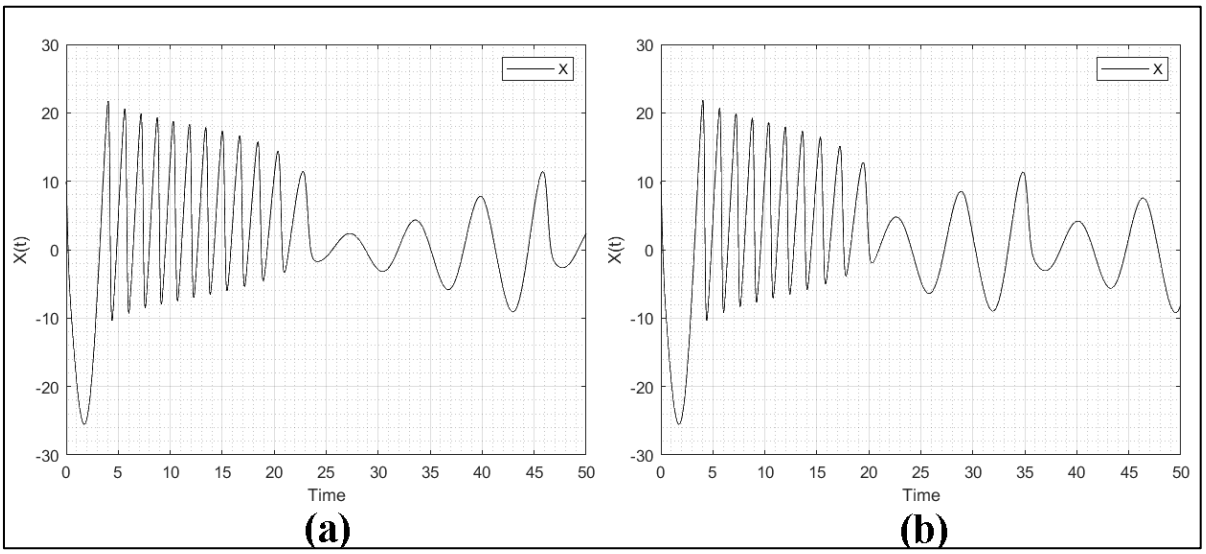

Figure 2-10 System Parameters Alteration (a: a=0.2,b=0.2, and c=5.7), (b: a=0.2,b=0.2, and c=5.8)

2.10.1.3. Chua System

Chua system is a simple electronic circuit that show a chaotic oscillation behavior (nonperiodic) in a certain circumstance. This circuit has been invented by Leon O. Chua in 1983. Chua system can be expressed using the differential equations shown in 2.7 to 2.10. [62]

$$dx/dt = a(y - x - \phi) \quad (2.7)$$

$$dy/dt = x - y + z \quad (2.8)$$

$$dz/dt = -bz - cy \quad (2.9)$$

$$\phi = m_1 x(1) + 0.5(m_0 - m_1)(|x + 1| - |x - 1|) \quad (2.10)$$

Where, ϕ represent the electrical response of the nonlinear resistor of the Chua electronic circuit, while in the other hand a, b, and c are system parameters with values 10, 14.78, and 0.0385 respectively. The parameters of nonlinear resistor have a value of m_0= -1.27 and m_1 = -0.68. The time series representation and strange attractors of the of the Chua system is described in figures 2-11 and 2-12, that shown below.

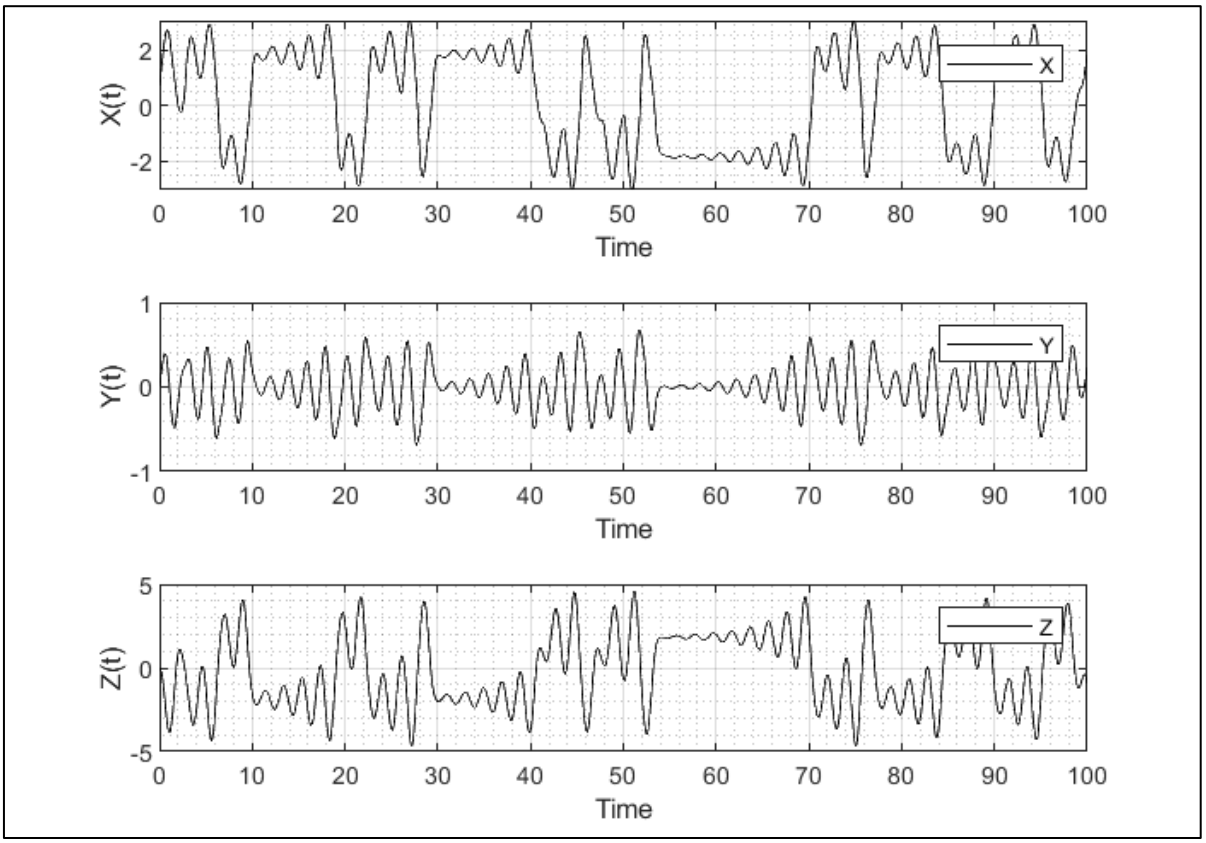

Figure 2-11 Chua Chaotic System Time Series

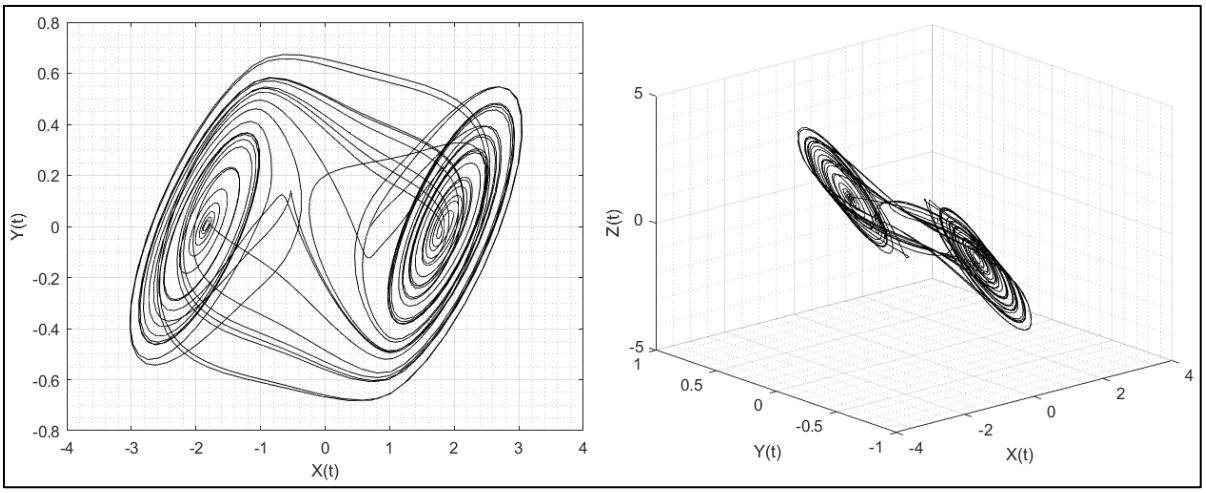

Figure 2-12 Chua System Strange Attractor

The system sensitivity to parameters perturbations is also investigated using the two identical Chua oscillators with tiny variation in the system parameters, this variation leads to make the system generate entirely different time series x component, as shown in figure 2-13.

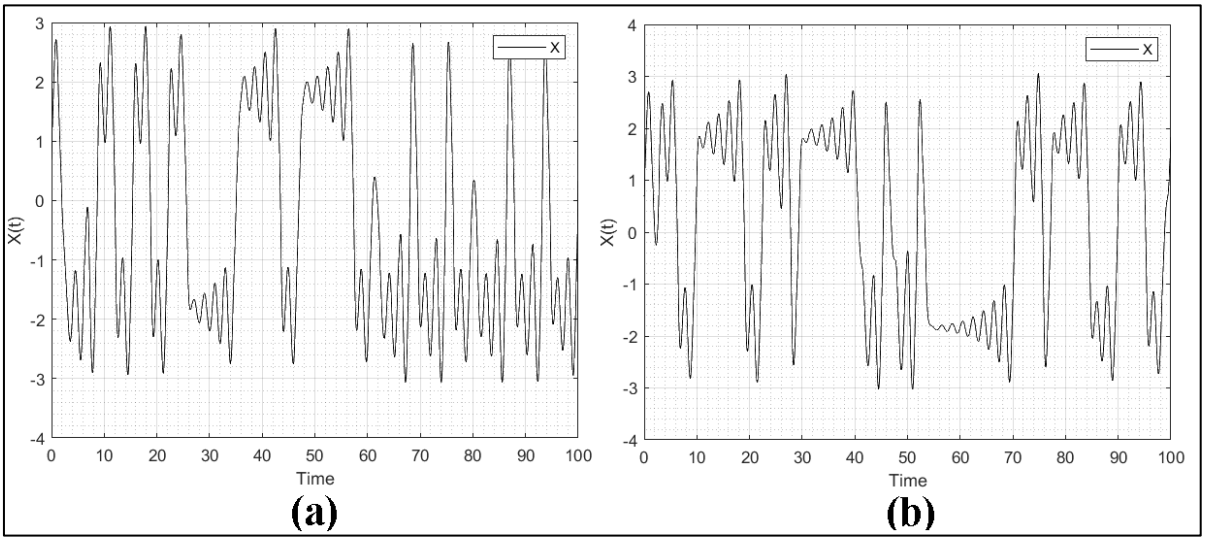

Figure 2-13 System Parameters Alteration (a: a=10, b=14.78, and c=0.0385), (b: a=10.1,b=14.78, and c=0.0385)

2.10.1.4. Rucklidge System

In 1992, Rucklidge proposed a new chaotic flow system with one nonlinear term (yz). The differential equations that describe the system behavior is illustrated in the equations 2.11 to 2.13 as shown below. [63]

$$dx/dt = -kx + ly - yz \quad (2.11)$$

$$dy/dt = x \quad (2.12)$$

$$dz/dt = -z + y^2 \quad (2.13)$$

The system parameters k equals to 2, while L equals to 6.7. the system at these values expresses chaotic behavior. System initial conditions are chosen to be x_0=1, y_0=0, and z_0=4.5. Time series response of the chaotic system as well as the strange attractor in 2D and 3D are depicted in figures 2-14 and 2-15 respectively.

In the other hand system sensitivity to parameters perturbations are also investigated as depicted in figure 2-16, where a completely different signal is generated when a small variation is taken place.

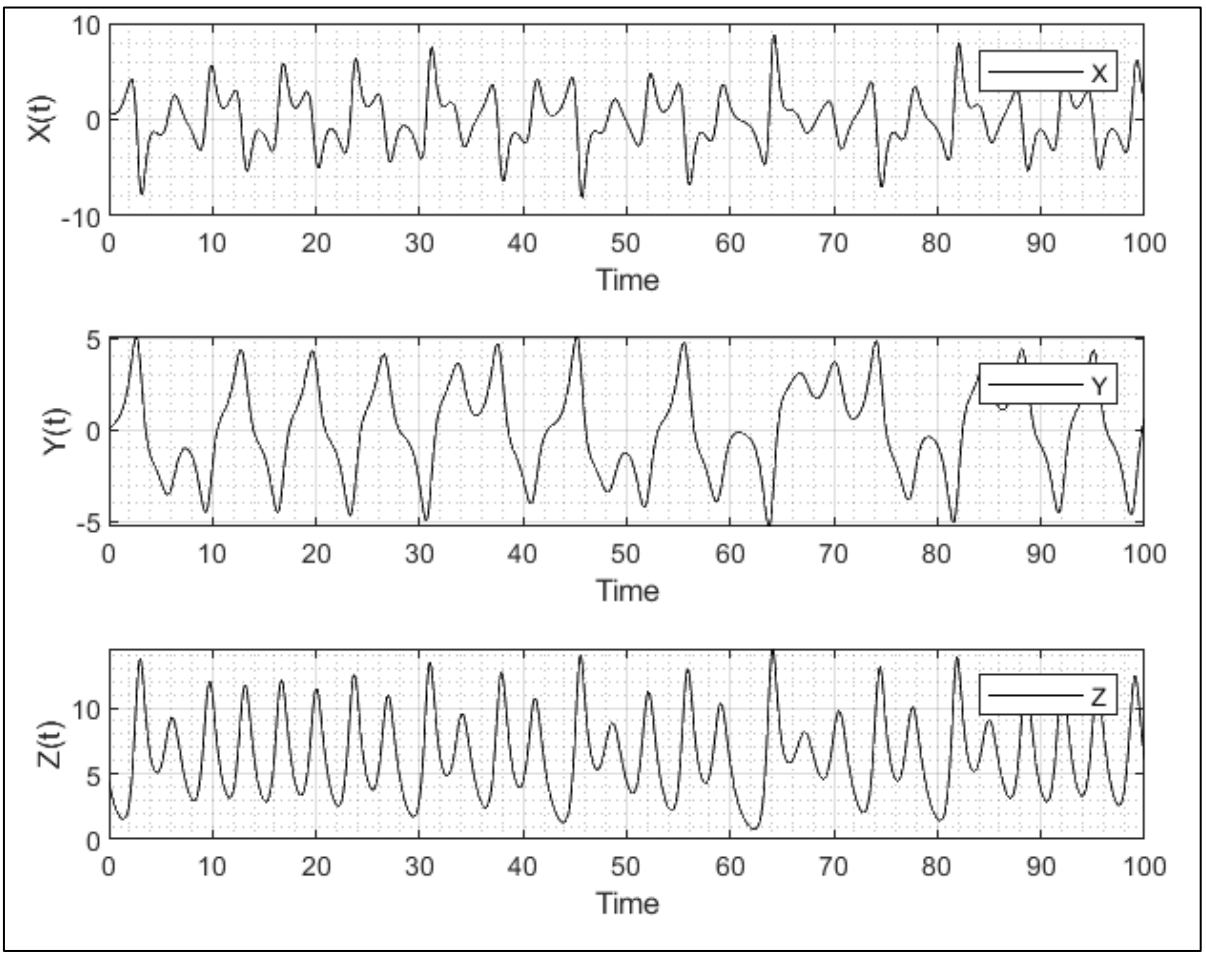

Figure 2-14 Rucklidge Chaotic System Time Series

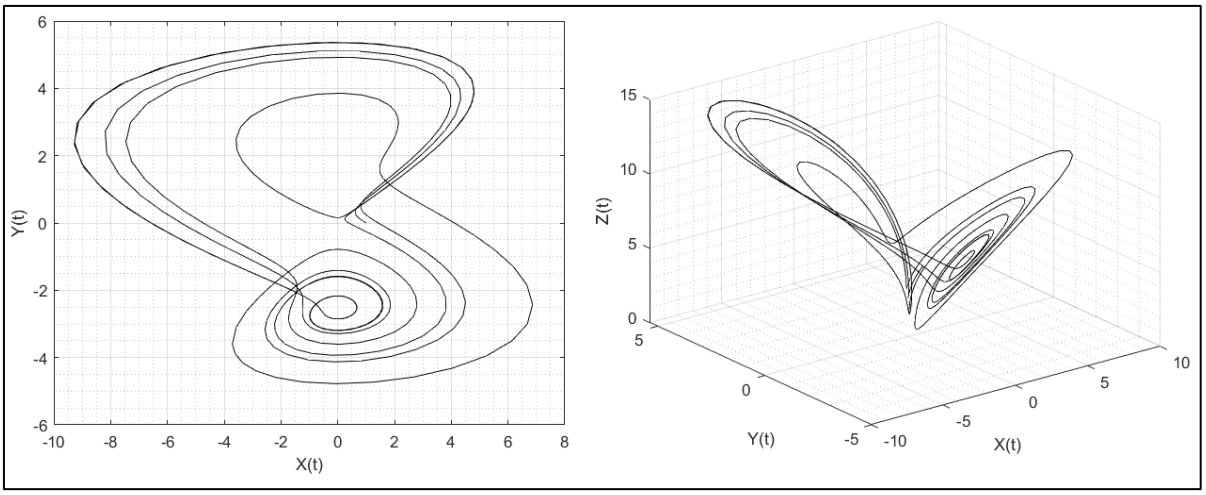

Figure 2-15 Rucklidge System Strange Attractor

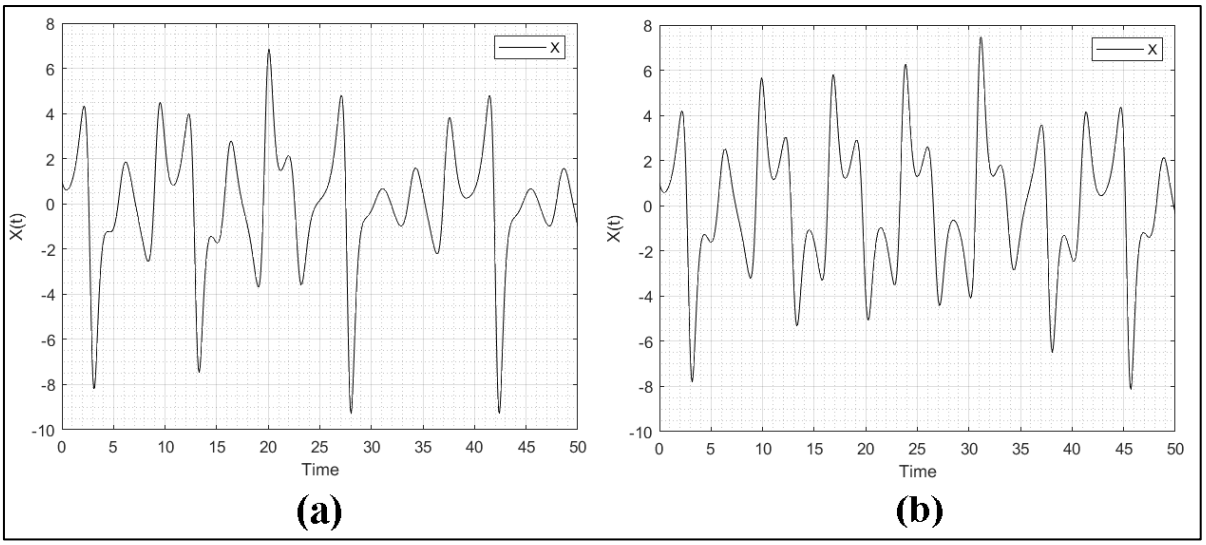

Figure 2-16 System Parameters Alteration (a: K=2, and L=6.7), (b: K=2.1, and L=6.7)

2.10.1.5. Nien System

Chua's differential equations from 1992 were revised by H.H. Nein in 2007 to create a new chaotic system that was proposed in 2007. [64]. The differential equations that describe the behavior of the new Nien chaotic system are expressed in equations 2.14 to 2.17 below.

$$dx/dt = -\alpha(x + y + h) \tag{2.14}$$

$$\frac{dy}{dt} = -\beta(x + y) - \gamma z \tag{2.15}$$

$$dz/dt = y \tag{2.16}$$

$$h = bx + 0.5(a - b)(|x + I_0| - |x - I_0|) \quad (2.17)$$

Where, α, β, γ, a, b, and I_0 are all system parameters with values α=6.3, β=0.7, γ=7, a=-1.143, b=-0.714, and I_0=3. Time series response of the chaotic system as well as the strange attractor in 2D and 3D are depicted in figures 2-17 and 2-18 respectively. In the other hand system sensitivity to parameters perturbations are also investigated as depicted in figure 2-19, where a completely different signal is generated when a small variation is taken place. [64]

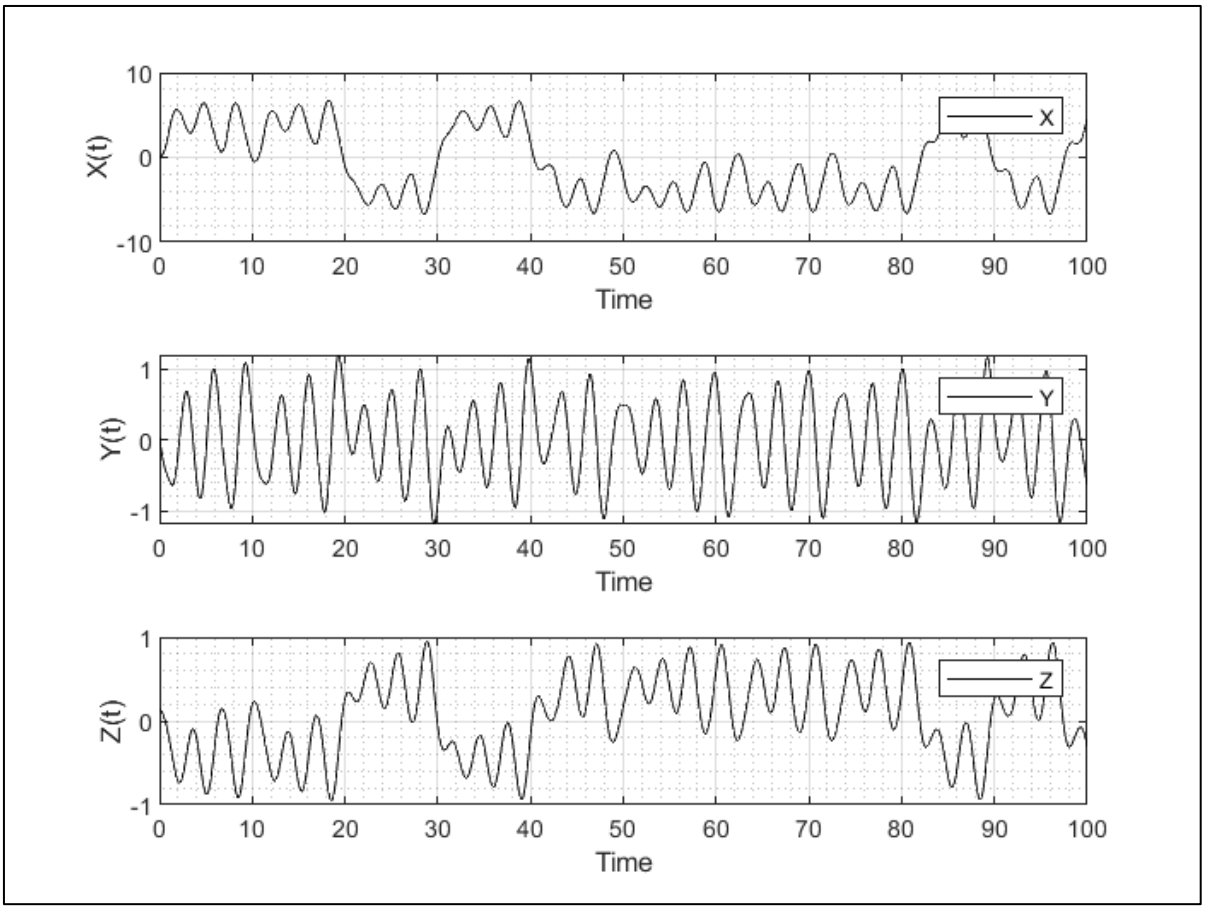

Figure 2-17 Nien Chaotic System Time Series

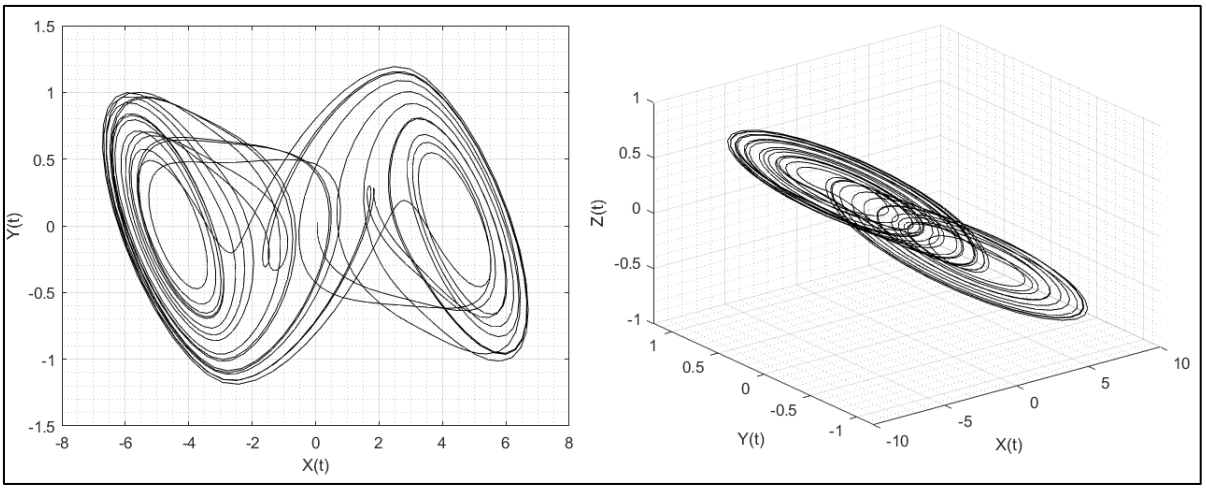

Figure 2-18 Nien System Strange Attractor

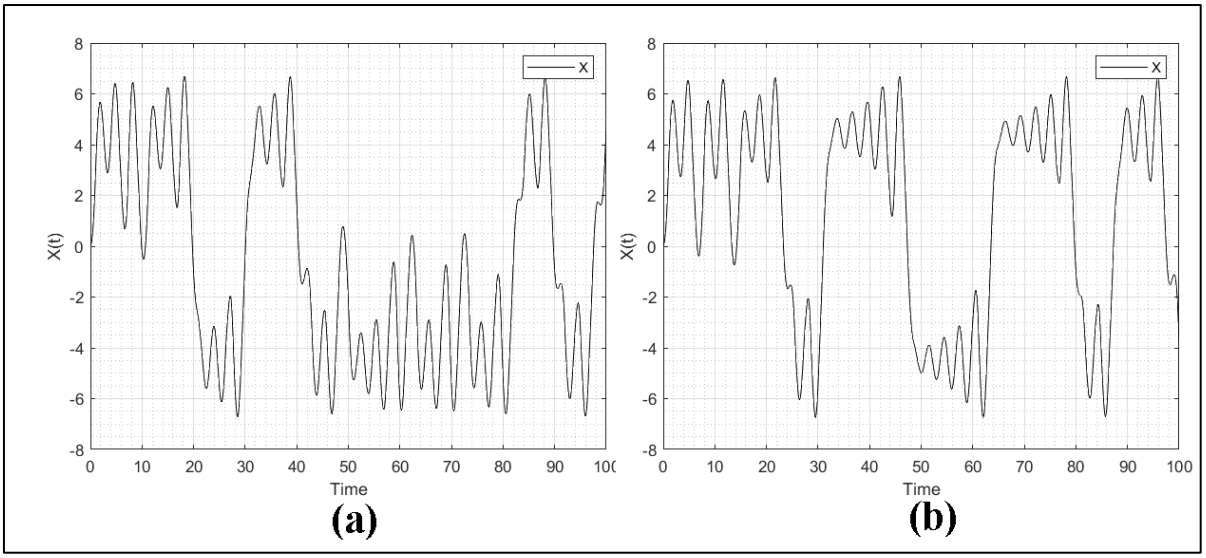

Figure 2-19 System Parameters Alteration (a: α=6.3, β=0.7, and γ=7), (b: α=6.4, β=0.7, and γ=7)

2.10.2. Chaotic Maps

Chaotic maps are the second type of chaotic systems, which can be defined as a nonlinear unpredictable system. Chaotic map is a discrete time system that can be constructed by a set of difference equations, and described by a strange attractor (trajectory). There are a wide range of chaotic maps such as, Logistic map,

Difference equations can be used to represent chaotic maps. Numerous well-known chaotic maps exist, including the Hénon, Logistic, Arnold Cat,

Chirikov, and Orbit maps. Hénon and Logistic maps will be selected to investigate this type of chaotic systems. [65] [66]

2.10.2.1. Logistic Chaotic Map

This is one of the most well-known 1-dimensional nonlinear unpredictable discrete time chaotic system that can be expressed by the following difference equation. [67]

$$x_{k+1} = \beta x_k (1 - x_k) \tag{2.18}$$

Where, β represent the bifurcation parameter (control parameter) and usually has the value between 0 and 4. The system in equation 2.18 exhibits chaotic behavior within the range $3.5 < \beta < 3.999$ otherwise the system has fixed stable point as shown in the attractor of the Logistic chaotic map shown in figure 2-20 below.

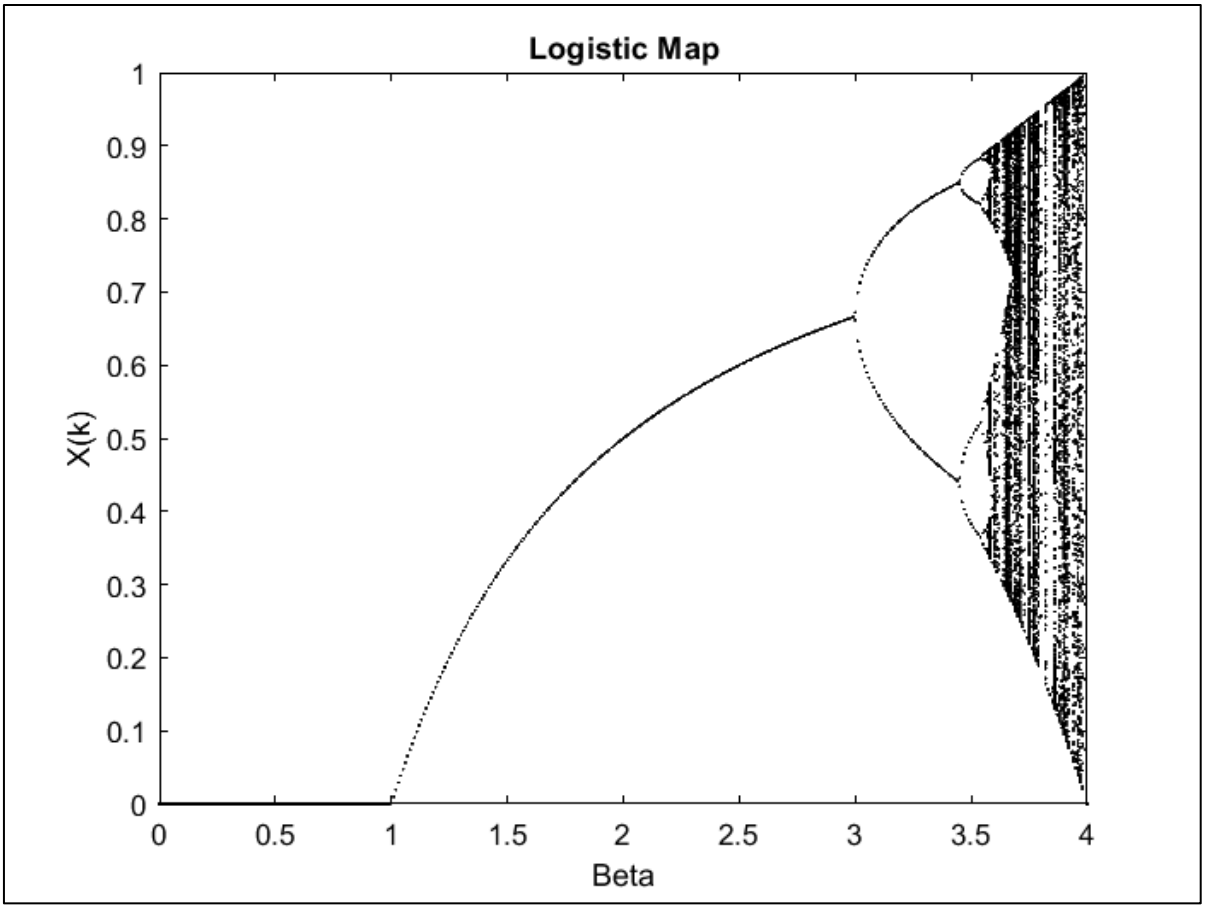

Figure 2-20 Logistic Map Strange Attractor (Bifurcation)

2.10.2.2. Hénon Chaotic Map

Hénon chaotic map is a 2-dimensional nonlinear discrete time system that maps any point in the cartesian coordinate to another point. It can be expressed be the difference equations shown in 2.19 and 2.20 below. [10] [68]

$$x_{k+1} = 1 - ax_k^2 + y_k \tag{2.19}$$

$$y_k = bx_k \tag{2.20}$$

Where, a, and b are the system parameters with values of 1.4 and 0.3 respectively. These parameter values make the Hénon system act as a chaotic map especially within the range of 1.07 < a < 1.4, outside this range Hénon system will acts as a regular periodic predictable system as shown in figure 2-21 that depicts the bifurcation of the proposed system. While in the other hand figure 2-22 depicts the x-y representation of the Hénon chaotic system.

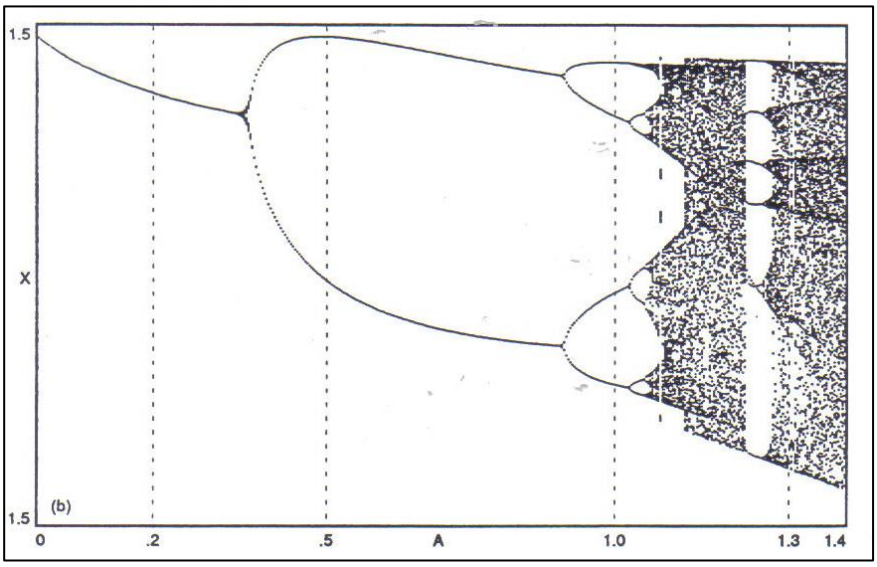

Figure 2-21 Hénon Chaotic Map Bifurcation

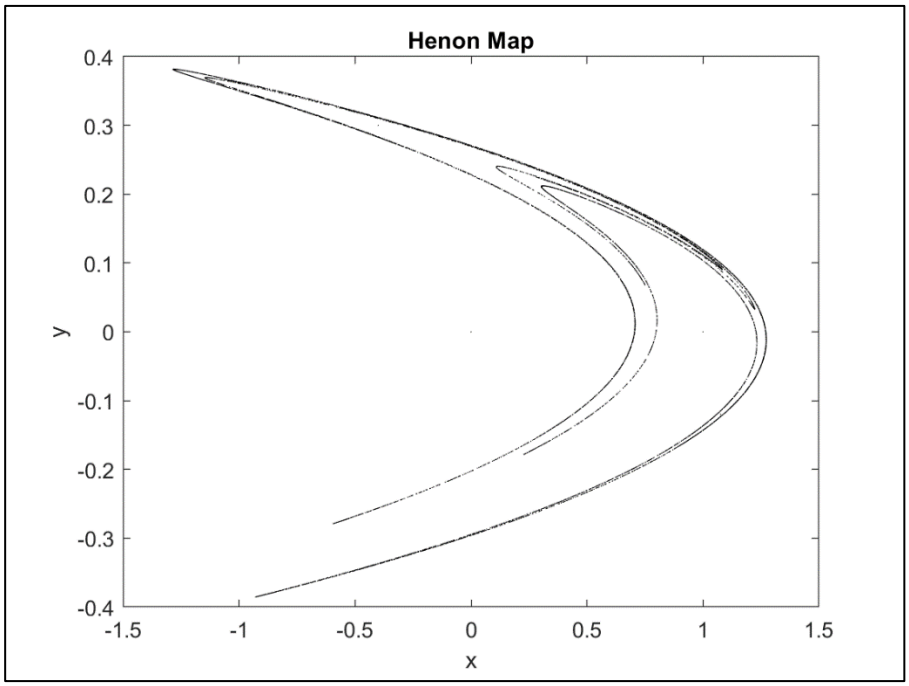

Figure 2-22 Hénon Chaotic Map Attractor

2.11. Lyapunov Exponents Definition

The Lyapunov exponent is a measurement that may be used to assess the degree of chaos in a system as well as if it is present at all. Given two orbits (trajectories) for the same chaotic system in space, but with a slight variance in their initial conditions, the lyapunov exponent measures the divergence rate between these two orbits.

The result of computing the lyapunov exponent gives three cases which are:

- Lyapunov exponent are less than zero, this result means that the system is not chaotic (dissipative system).
- Lyapunov exponent equal to zero, the system is not chaotic and stable with constant separation (conservative systems).
- Lyapunov exponent is greater than zero (positive), the system has chaotic behavior (chaotic system).

According to the above restrictions, a system of ordinary differential equations (with a certain phase space dimension) is called chaotic system if it

has one positive lyapunov exponent among all the lyapunov exponent values (usually one for each dimension of its phase space).

2.12. Application of Chaos to Cryptography and Communications

The main intrinsic properties of any chaotic system are aperiodic, unpredictable, deterministic (not random) and have extremely sensitivity to system parameters and initial conditions (start point). These properties make the chaotic systems a prominent candidate to construct a cryptographic algorithm based on chaos systems, since chaotic systems are unlike pseudorandom algorithms that generate limited numbers of binary streams and they are periodic. Chaotic signals are entirely aperiodic and can theoretically be produced in an unlimited number by chaotic systems. Based on these facts, chaotic systems can be used to securely communicate by concealing (encrypting) the messages.

The synchronization principle between the communicated nodes, which are sometimes referred to as master and slave systems, must be satisfied in order to build a secure communication system based on chaos theory. When two chaotic systems synchronize, it signifies that their behavior is converging toward the same trajectory value as that of the other system, and that they will continue to move in the same direction [53]. Figure 2-23 presents the chaotic synchronization principle, where usually the master chaotic system sends one or more of its dynamical signals to the slave system chaotic system (which may be identical or not to the master system).

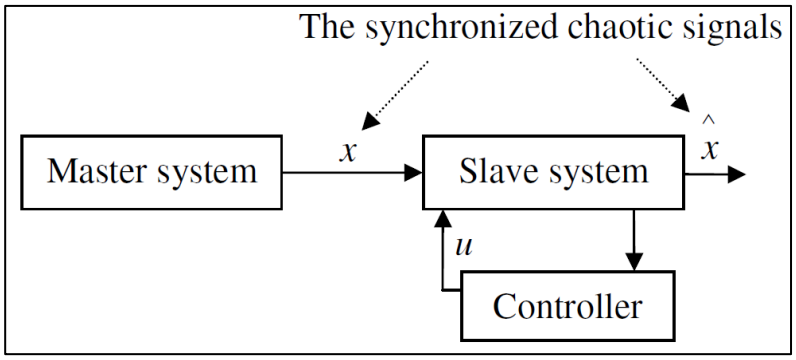

Figure 2-23 Chaos Synchronization (x denotes to master signal and $\hat{x}$ denotes to slave signal, u denotes to control signal) [58]

The slave system may or may not synchronize the master system depending on the master system signal that was sent to the slave system. It is required to develop a controller system that enforces the synchronization if it does not occur naturally. [58]

Figure 2-24 and 2-25 present the 2D and 3D synchronization graphs between the master and slave systems, where as shown below the two systems are initially asynchronized but after a while synchronization is achieved due to the control signals provided by the controller system.

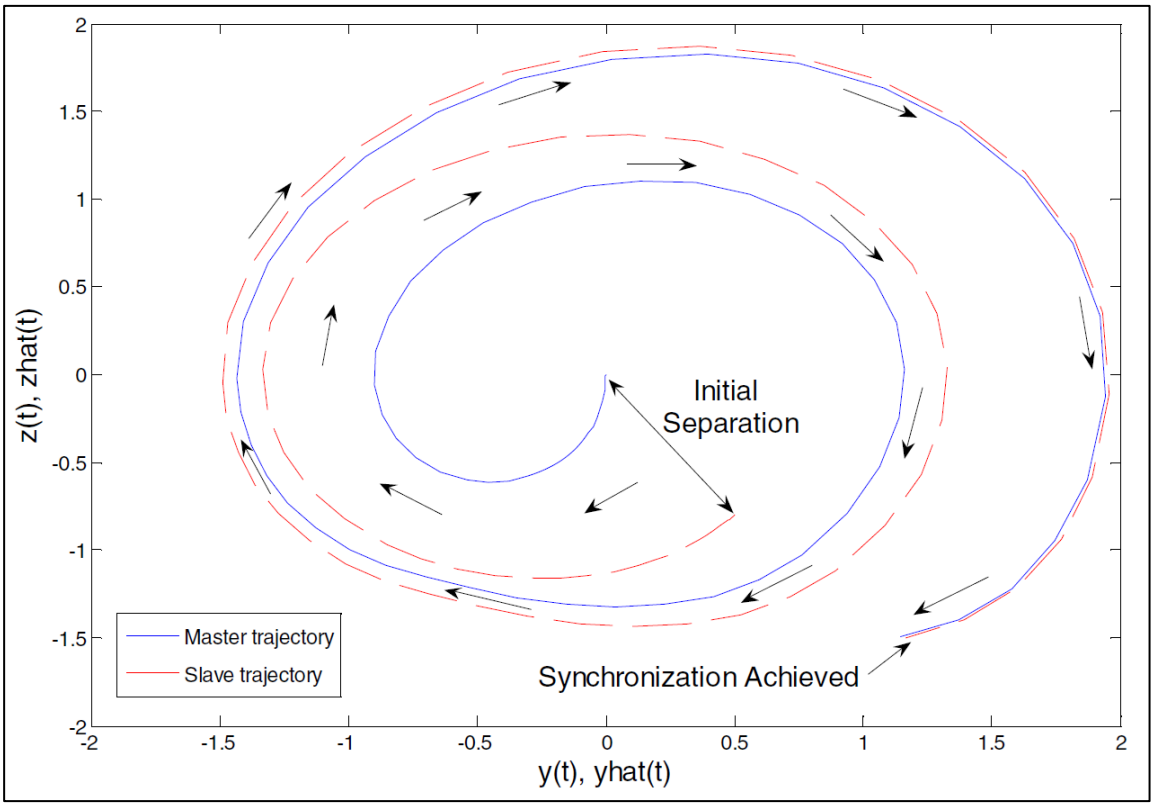

Figure 2-24 Signals Synchronization between Master and Slave Systems (2D) [58]

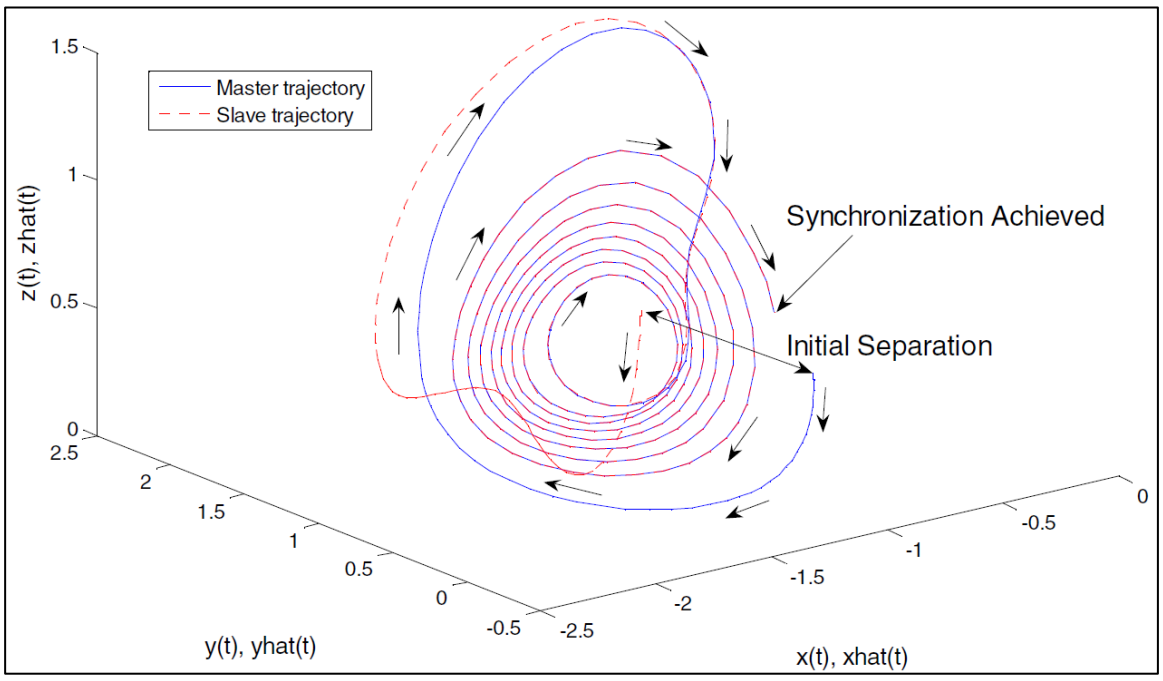

Figure 2-25 Signals Synchronization between Master and Slave Systems (3D) [58]

2.13. Problems of Chaos Encryption Algorithms

The application of chaos systems to the encryption algorithms make the encryption process higher and hardly attacking but in the other hand there are three main problems with using chaotic system in cryptography, which are listed below. [69]

- **Statistical tests:** Generally chaotic systems are not complex as much as required to generate random sequence to be used in the encryption process, where the generated chaotic sequence correlation can be obtained using different statistical tests and the system can be violated.
- **Confidentiality Inconsistency:** It is possible that two neighboring states for chaotic encryption systems to be on the same linear segment. In this case with high precision digital implementation, it is possible to use this characteristic for easily recovering large segments encryption key in the decryption process after identifying quantum of plaintext and ciphertext.
- **Precision effect:** Usually implementing chaotic systems on devices and circuits collide with a limited precision of that devices and circuits. So, it is worthwhile to know that the chaotic generator may exceeds the large number of outcome researches which are yielded by limited automation and Boolean algebra theory. Numerous chaotic-based applications cannot be used because the chaotic system's characteristic that is executed with finite precision completely differs from its theoretical outcomes. Because of this, some scholars concur that this impact becomes one of the challenges for industrial applications.

2.14. Hyperchaotic Systems

Hyperchaotic systems can be described as a nonlinear system with dynamical responses (x, y, z, w, etc) that expand in multi-direction, because they have more than one positive Lyapunov exponents (two or more). This

reality leads to dynamical behavior (trajectory) with more complexity than the ordinary chaotic systems.

One of the first observations of the hyperchaotic systems were in 1979 by Rössler in [70], where it consists of four dimensional ordinary differential equations that was derived from the simplest three dimensional chaotic system by adding extra linear variable. The differential equations that describe the new Rössler hyperchaotic system are shown in equations 2.21 to 2.24 below.

$$\frac{dx}{dt} = -(y + z) \tag{2.21}$$

$$dy/dt = x + 0.25y + w \tag{2.22}$$

$$dz/dt = 3 + xz \tag{2.23}$$

$$dw/dt = 0.5(0.1w - z) \tag{2.24}$$

As aforementioned above hyperchaotic system could generate multiple of positive lyapunov exponents so its dynamics expands in more complex manner and as result it is harder to predict their behavior compared to ordinary chaotic systems. So hyperchaotic systems provide higher security levels in wireless communications and got a great deal of attention.

2.15. Constructing New Hyperchaotic System

In order to construct a hyperchaotic system, the following significant requirements must be satisfied:

- The ordinary differential equations are at least four dimensions.
- Two nonlinear terms should be existed.
- Two or more positive lyapunov exponents should be generated from the proposed ODEs.

2.16. Numerical Integration Methods

Traditionally the nonlinear dynamical systems are described by using a set of ordinary differential equations ODEs. Solving the ODEs requires an integration-based method to perform the solution. This integration can be written as follows in equation (2.25). [71]

$$y(t) = y(i) + \int_{i}^{t} f(x)dx \qquad (2.25)$$

Where, i denotes to the initial state of the system while t, represents the solution at the required time. The integration formula above represents the analytical solution which is practically very difficult and, in some cases, unsolvable. The most effective methods used to perform this kind of integrations are numerical based methods. There are four widely used numerical methods that described below.

2.16.1. Forward Euler Integration Method

This is the simplest first order integration method. It has relatively low accuracy in its computations because of the roughly estimation of the next coordination point in the solution. The general formula for this method is presented below in equation (2.26).

$$y_{t+1} = y_t + hf(x_t, y_t) \qquad (2.26)$$

Where, y_{t+1} is the next time variable value, y_t is the current time variable, $f(x_t, y_t)$ is the derivative of the current time variable, and h represent the interval of the computational time. [71]

2.16.2. Heun Integration Method

Mathematically, Heun method is the improved Euler's method (modified one), where this method is used to fix the problem of roughly estimation solution (refining the solution). Heun Integration method can be formulated as in 2.27 shown below.

$$y_{t+1} = y_t + \frac{h}{2}(f(x_t, y_t) + f(x_{t+1}, y_t + hf(x_t, y_t)) \quad (2.27)$$

2.16.3. Adams - Bash forth - Moulton Integration Method

This method known as the multi-step linear integration method. This method uses the multistep principle to gain high efficiency by keeping and using the data from the previous steps rather than discarding them. Equation 2.28 illustrate the mathematical formula for this method.

$$y_{t+1} = y_t + \frac{h}{2}(3f(x_t, y_t) - f(x_{t-1}, y_{t-1})) \quad (2.28)$$

Where, y_{t+1} is the next time variable value, y_t is the current time variable, $f(x_t, y_t)$ is the derivative of the current time variable, $f(x_{t-1}, y_{t-1})$ is the derivative of the previous time variable, and h represent the interval of the computational time.

2.16.4. Runge Kutta Integration Method

Due to its high stability and accuracy Runge Kutta integration method is the most used method for solving the ordinary differential equations, but in the other hand this method requires higher computational resources and time. The formulas used to solve ODEs using the Runge-Kutta method are described in the equation set (2.29 to 2.33). [72], [73]

$$y_{t+1} = y_t + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad (2.29)$$

$$K_1 = hf(x_t, y_t) \quad (2.30)$$

$$K_2 = hf(x_t + \frac{h}{2}, y_t + \frac{k_1}{2}) \quad (2.31)$$

$$K_3 = hf(x_t + \frac{h}{2}, y_t + \frac{k_2}{2}) \quad (2.32)$$

$$K_4 = hf(x_t + h, y_t + k_3) \quad (2.33)$$

Where, y_{t+1} represent the approximation signal or the estimated one, while in the other hand k_i refer to:

K_1: refer to the slope at the beginning of the interval, using y (Euler's method).

K2: refer to the slope at the midpoint of the interval, using y and k1.

K3: refer to the slope at the midpoint, but now using y and k2.

K4: refer to the slope at the midpoint, using y and k3

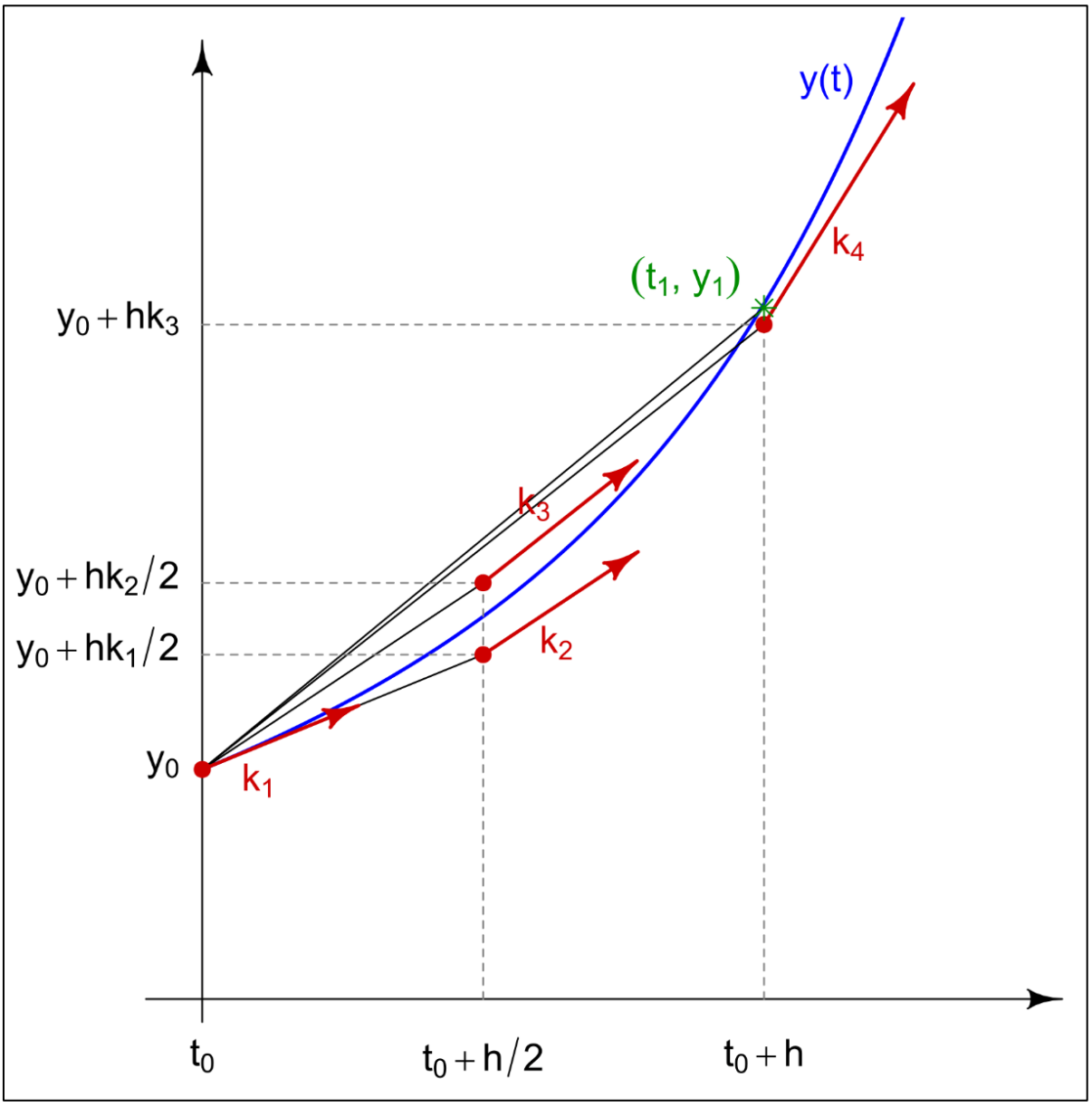

Figure 2-26 Runge Kutta Numerical Integration Method

2.17. Measuring Techniques for the Encryption Quality

In the field of cryptography and encryption systems, it is not important to design the encryption system as much as it is important to test the performance of the designed system. There are different measurement techniques and algorithms that can be used to test the encryption algorithm quality especially those algorithms that designed based on chaos and hyperchaos systems. The first group contains Lyapunov exponents and 0-1 test that used to inspect the chaotic/hyperchaotic behavior of the system. The tests performed to test and verify the system's randomness in the second group are based on the system key. These tests include the long run, run, poker, and monobit tests. The third category of tests, such as peak signal to noise ratio, mean square error, correlation coefficient, etc., examines the resilience of the encryption technique. The system statistical analysis, which consists of the unified average change intensity test and the number of pixels change rate test, is included in the fourth category. A brief description of each test is provided in the section after that.

2.17.1. Chaotic/Hyperchaotic System Behavior Tests

The first group of the measuring techniques is consisting of two widely used tests, which are lyapunov exponent test and 0-1 test, where these two tests are used to inspect and check the system behavior wither it is chaotic or not. Lyapunov test will be considered for testing the dynamical system behavior.

Lyapunov Exponent Test

Lyapunov exponent is the most important quantity to the dynamical systems, whereas the maximal positive lyapunov exponent is a strong signature to chaotic behavior. In fact, this test is based on calculating the natural algorithm of the divergence rate of two trajectories in the phase space for the same system but with a very tiny difference in the initial conditions or/and system parameter. In other word, this test measures the system sensitivity to

starting point and system parameter. A dynamical system with m-dimensions should have m lyapunov exponents named as λ_1, λ_2, λ_3, λ_4…… λ_m. Lyapunov exponent test can be applied to the chaotic flow systems (continuous time) and for the chaotic maps (discrete time). Equation 2-25 is used to calculate the lyapunov exponent for continuous time, while equation 2-26 is used to do the calculations in discrete time systems. [57] [74]

$$\lambda_i = \lim_{\sigma x_i(0)\to 0} \lim_{t\to\infty} \frac{1}{t} log \frac{\sigma x_i(t)}{\sigma x_i(0)} \quad (2.25)$$

$$\lambda_i = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} log \frac{\sigma x_i(n+1)}{\sigma x_i(n)} \quad (2.26)$$

Based on the result of computing the lyapunov exponent there are three possible cases which are:

- Lyapunov exponent are less than zero, this result means that the system is not chaotic (dissipative system).
- Lyapunov exponent equal to zero, the system is not chaotic and stable with constant separation (conservative systems).
- Lyapunov exponent is greater than zero (positive), the system has chaotic behavior (chaotic system).

The equation 2.25 has been used to write a Matlab code to find the lyapunov exponent of the Lorenz chaotic flow (previously expressed). As depicted in figure 2-27, Lorenz system has three lyapunov exponent dynamics, which are +0.937307, 0, and -14.593. The positive lyapunov exponent indicate the chaotic behavior of Lorenz system (Matlab code is appended in appendix A).

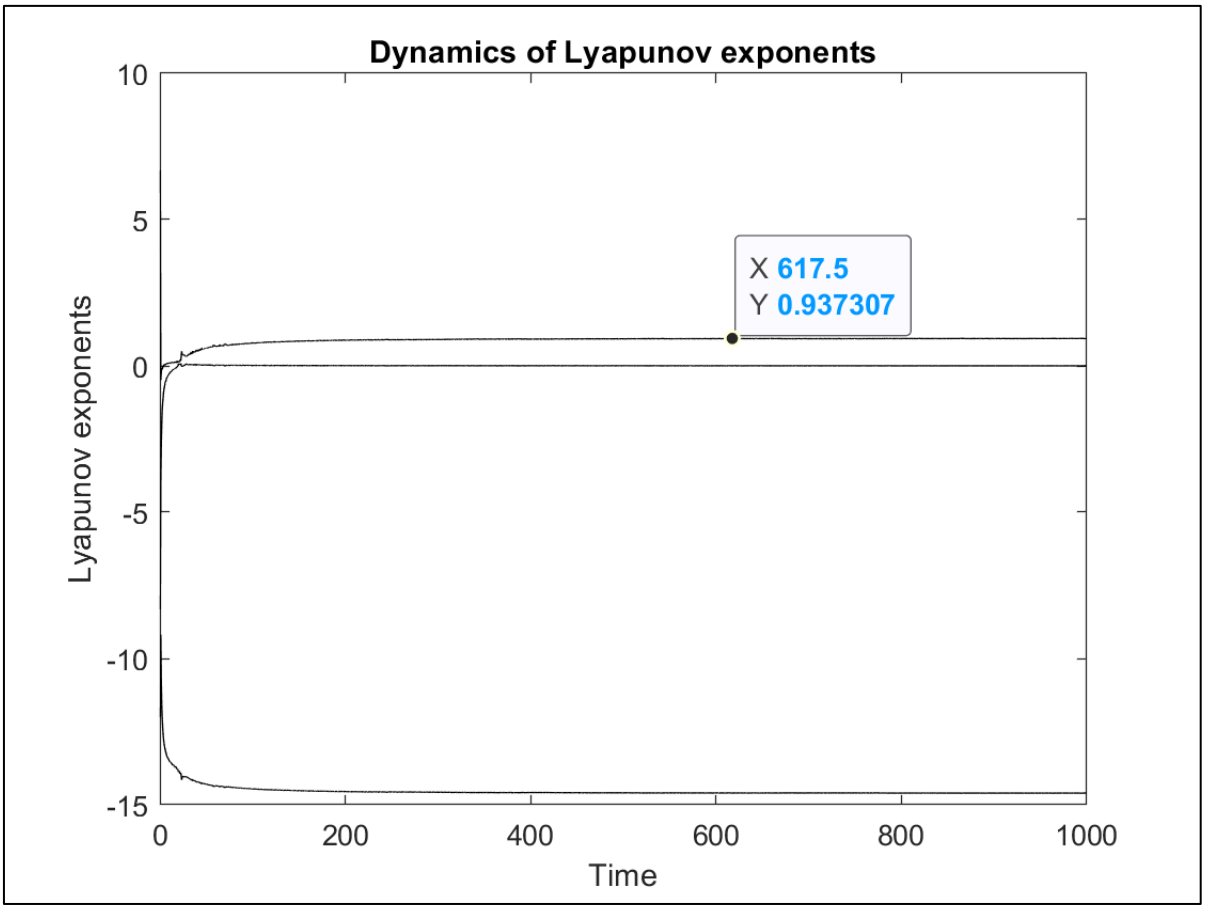

Figure 2-27 Lyapunov Exponent Dynamics of Lorenz System

2.17.2. System Randomness

This group of tests are used to check the generated data from the dynamical system, whether random or regular. There are mainly four well-known tests which are, monobit test, poker test, run test and the long run test. These tests will be investigated in brief manner below. The set of data should be converted into 20000 continuous bits before applying the randomness tests. [75]

2.17.2.1. Monobit Test

In this test, the number of ones or zeros are calculated within the 20000 bits that generated from the dynamical system. If the number of ones or zeros is between the range 9654 and 10346, the system passes this test and the generated data is considered to be random. [75]

2.17.2.2. Poker Test

In this test, the generated 20000 bits are divided into 5000 segments, each segment consist of 4 bits. Then the number of occurrences of each 16 possibilities for the 5000 segments. Equation 2.27 shown below has been used to calculate the Poker test. [75]

$$P = \frac{16}{5000}\sum_{i=0}^{15}(x(i))^2 - 5000 \qquad (2.27)$$

Where, x(i) represents the generated numbers, and i is converging between 0 and 15 (counting the 16 possibilities). If and only if the P calculated value is between the range 1.03 to 57.4, the system is passing this test and the data is considered to be random.

2.17.2.3. Run Test

This number of successive zeros and ones of the generated 20000 bits stream data are calculated in this test. The term run is referred to the sequence of data with same samples (sequence of zeros or ones). All of the run length of the size one or above in the generated 20000 bits should be counted and stored then compared to the run length table that shown below. If the calculated runs are within the range that determined in the table 2-1 shown below, the system is passing in this test. [76]

Table 2-1 Run Length Boundaries [76]

Length of Run (number of successive same samples)	Required Interval (repeat number within 20000-bit data)
1	2267 to 2733
2	1079 to 1421
3	502 to 748
4	223 to 402
5	90 to 223
6 or greater	90 to 223

2.17.2.4. Long Run Test

This test is focus on the longest run of successive ones and/or zeros within 20000-bits of the generated data. The run length (number of successive bits of the same sample) should be less than 34 bits to achieve the test passing, otherwise the system fails passing this test. [75]

2.17.3. Encryption Algorithm Strength Tests

Although one of the essential parts of encryption inspection in the field of data encryption (video, image, and audio) is feature skulking of the original data from visual monitoring, it is still insufficient for determining the effectiveness of the encryption algorithms. In order to test the strength of the encryption method, researchers and scientists offered some algorithms. To determine how effective the algorithm is, the data after encryption will be compared to the data before encryption. Higher and more erratic variations in the encrypted data indicate that the encryption process is working more effectively.

The encryption algorithm strength measurement uses a variety of methods and algorithms, which are briefly described in the list below.

2.17.3.1. Histogram

Histogram is one of the most widely used visual monitoring tests, that inspects the strength of the image-based encryption algorithms. Image histogram can be defined as a graphical representation that presents the pixel's intensity values distribution. Generally plain image histogram is fashioned in a specific oblique strip or line which is full with information about the image pixels distribution and this information is very useful for hacker or intruders. In fact, the hacker or intruders can use this distribution to deduce or reconstruct the original plain image. In order to cope with this type of threats, the adopted

encryption algorithm should be able to hide or conceal pixels intensity distribution. [77]

2.17.3.2. Information Entropy

The unpredictability and randomness of any information source can be determined by using information entropy metric. The mathematical description for the information entropy was firstly introduced by Shannon in 1949 [44], as shown in equation 2.28. The maximum value for the information entropy is 8. The information source that has 8 entropy value or close to it means that system has high randomness level and its prediction is too difficult. So, encryption algorithm that produce encrypted data with entropy close to or equal to 8 refers to a rigid and strong system that can stands against the different cyber-attacks.

$$Entropy = \sum_{i=0}^{2^n-1} p(i)\, log_2 \frac{1}{p(i)} \qquad (2.28)$$

Where, p(i) refers to the probability of the i th bit. As aforementioned above, the entropy of the encrypted data must be close to or equal to the maximum entropy value in order to withstand the entropy attack.. [78]

2.17.3.3. Mean Square Error / Peak Signal to Noise Ratio

The basic objective of any cryptographic algorithm is to make the difference between the ciphered and plain data is greater as maximum as possible to cope with differential and statistical attacks. The mean square error (**MSE**) and peak signal to noise ratio (**PSNR**) are adopted to calculate the difference between the plain and ciphered data. In the other hand MSE and PSNR between the recovered and ciphered data is also can be calculated to show the differences. The mathematical description of MSE and PSNR is given in equation 2.29 and 2.30 respectively. [29]

$$MSE = \frac{1}{mn} \sum_{i=0,j=0}^{n-1,m-1} (O(i,j) - E(i,j))^2 \quad (2.29)$$

$$PSNR = 10\, log_{10} \frac{255^2}{MSE} \quad (2.30)$$

Where, O (i, j) represent the original image matrix, and E (i, j) represent the encrypted image matrix, while i, j represents the dimensions of the matrices. Generally, greater MSE and PSNR value indicates that the two images are highly different and that means the cryptographic system can stands against the intruders and hackers and can cope with cyber-attacks, while zero MSE and infinity PSNR means that the images are identical.

2.17.3.4. Adjacent Pixels Correlation Coefficients

Any digital image is consisted of a number of pixels, and these pixels are organized or paved one beside the other. The pixels that sharing their boundaries with other pixels are known as adjacent pixels. The pixels can be vertically, horizontally or even diagonally adjacent, and these adjacent pixels are highly correlated with each other. The main objective of any image based cryptographic algorithms is to break this correlation. The correlation quantity can be calculated for these adjacent pixels by using the mathematical formula shown in 2.31 below. The correlation coefficients that close to zero indicate that there is no correlation between the image pixels, and hence the cryptographic algorithm copiously reduces the correlation level and breaks the correlation between the adjacent pixels. [79]

$$Corr = \frac{\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}((x_i - E(x)))^2}\sqrt{\sum_{i=1}^{N}((y_i - E(y)))^2}} \quad (2.31)$$

Where, x, y represent the pixel values of the original and encrypted images respectively.

2.17.3.5. Key Space

One of the most aspects related to data security and cryptosystem design is the cryptographic system key space. The cryptosystems with relatively large key space can provide more secure data, stronger and robust against the brute force attack. With respect to chaotic/ hyperchaotic systems, the system parameters and initial conditions (starting point) are all represent the encryption key, and this means the encryption key of the dynamical systems as huge space which increase the system robustness against the brute force attacks.

2.17.4. System Statistical Analysis

The intruders in some cases gain an access to the encryption machinery due to a general failure in the network (for example), the intruder makes a slight change in a single image pixel of the input plain image and get two ciphered images one without any changes and the other one with slight change. These two ciphered images have a potential relationship between them. This relationship can be discovered and exploited to break the cryptographic algorithm and to deduce the encryption keys and algorithm. In order to cope with this type of attacks two metrics are used, which are number of pixel change rate (NPCR) and unified average changed intensity (UACI), which are described below. [80]

2.17.4.1. Number of Pixel Change Rate

The NPCR is stands for number of pixels change rate, this metric is concentrated on the number of changed pixels and it can be calculated using the equation in 2.32 and 2.33.

$$D(i,j) = \begin{bmatrix} 1 & when\ C_1(i,j) \cong C_2(i,j) \\ 0 & when\ C_1(i,j) = C_2(i,j) \end{bmatrix} \quad (2.32)$$

$$NPCR = \frac{\sum_{i,j}^{N,M} D(i,j)}{M \times N} \times 100\ \% \quad (2.33)$$

Where, C_1 and C_2 represent the ciphertext images before and after one pixel change in a plaintext image respectively, while M and N are the width and height of the images. The NPCR metric should equal or close to 100% in order to ensure that the system can stands against the differential attacks.

2.17.4.2. Unified Averaged Changed Intensity

The UACI metric that stands for unified averaged changed intensity, it can be calculated using the equation in 2.34 to figure out the average difference between the two images (ciphertext images before and after one pixel change). [80]

$$UACI = \frac{\sum_{i,j}^{N,M} | C_1(i,j) - C_2(i,j)|}{M \times N \times 256} \times 100\% \qquad (2.34)$$

Where, C_1 and C_2 represent the ciphertext images before and after one pixel change in a plaintext image respectively, while M and N are the width and height of the images. The 256 denotes the largest supported pixel value compatible with the ciphertext image format. The UACI metric should equal or close to 33% in order to ensure that the system can stands against the differential attacks. [80]

All the above tests and statistical analysis systems are implemented using Matlab software, and the codes (M-files) are appended in the appendix A.

CHAPTER THREE

Proposed Encryption Algorithms

Chapter Three: The Proposed Encryption Algorithms (Simulation and Implementation)

3.1. Introduction

This chapter presents the mathematical description, simulation and implementation of the proposed five data encryption algorithms using Xilinx System Generator environment (**XSG**) and Vivado. The proposed encryption algorithms are based on hyperchaotic systems and stream cipher encryption principle, in which the data is **XOR-ed** with generated hyperchaotic random bit stream carrier as a prior for transmitting them. The data used to test the proposed algorithms performance are images (with different sizes including 64×64, 128×128, 176×144, 256×256, and 2500×1875). All the proposed algorithms are implemented with field programmable gate array (**FPGA**) board. All the simulations and implementations have been carried out using a PC of Core i7 CPU with 2.4 GHz, and RAM 16 GB, with Matlab 2020b and Vivado 2020.2. Table 3-1, summarizes the proposed image-based encryption algorithms using hyperchaotic systems. Figure 3-1 present a flow chart for the image encryption operation sequence.

Table 3-1 Proposed Encryption Algorithms

1st proposed algorithm	A Secure Communication System Based on Three Dimensional Lorenz Chaotic Attractor.
2nd proposed algorithm	Multidimensional Hyperchaotic System Based on XOR Mixture of Dynamical Systems
3rd proposed algorithm	Multidimensional Cascaded Hyperchaotic Systems Based on Chaos Switching Technique.
4th proposed algorithm	Robust Encryption System Based on Novel Hyperchaotic Flow System.
5th proposed algorithm	Novel Hyperchaotic Sequence Based on the combination of the 2nd, 3rd, and 4th, algorithms.

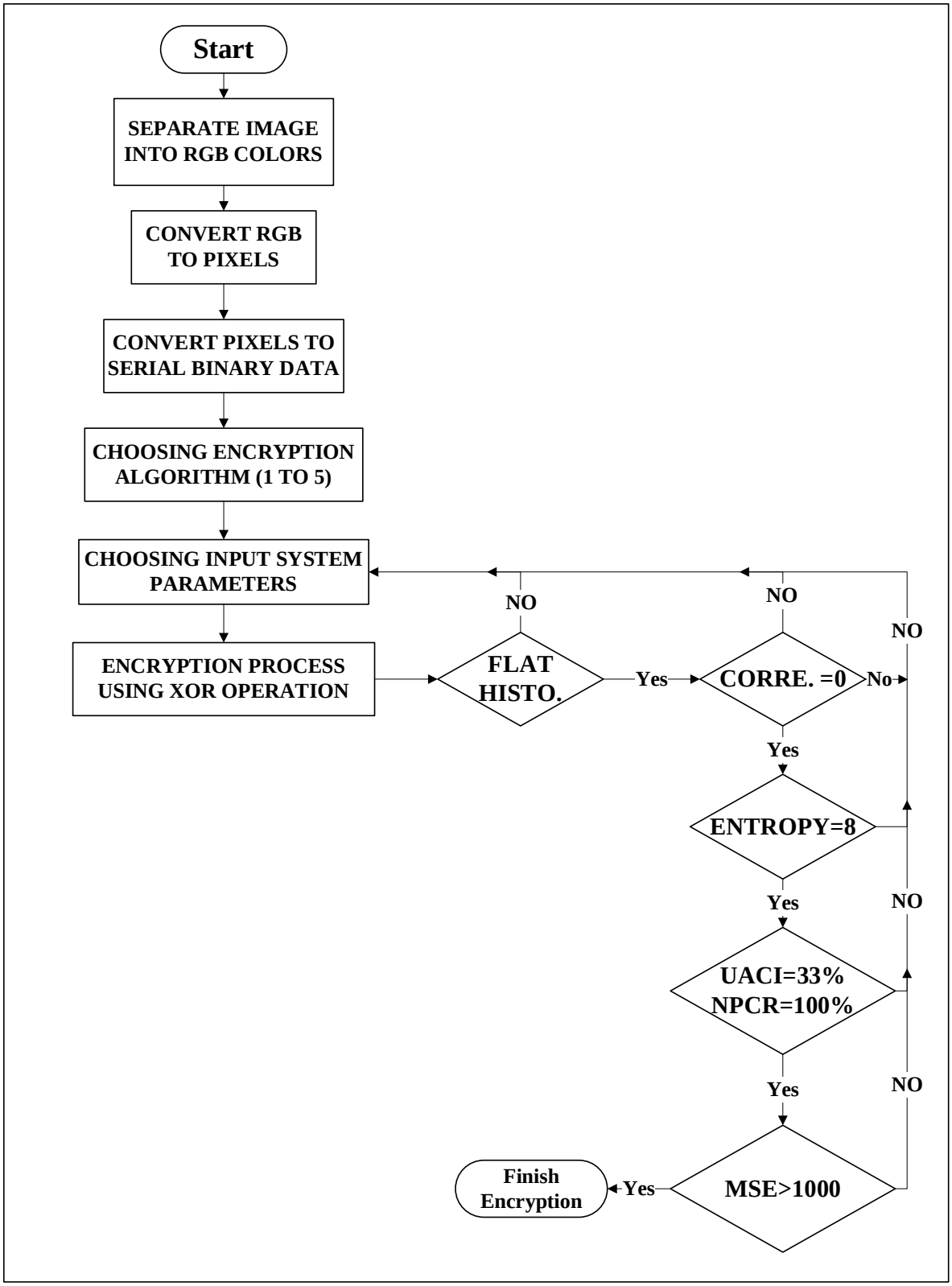

Figure 3-1 Designed Image Encryption System Flow Chart

3.2. Algorithm (1): Secure Communication System Based on Three Dimensional Lorenz Chaotic Attractor

The proposed image encryption system is presented in the block diagram in figure 3-2. The encryption system is based on Lorenz chaotic oscillator with a specific starting point (initial conditions) and system parameters.

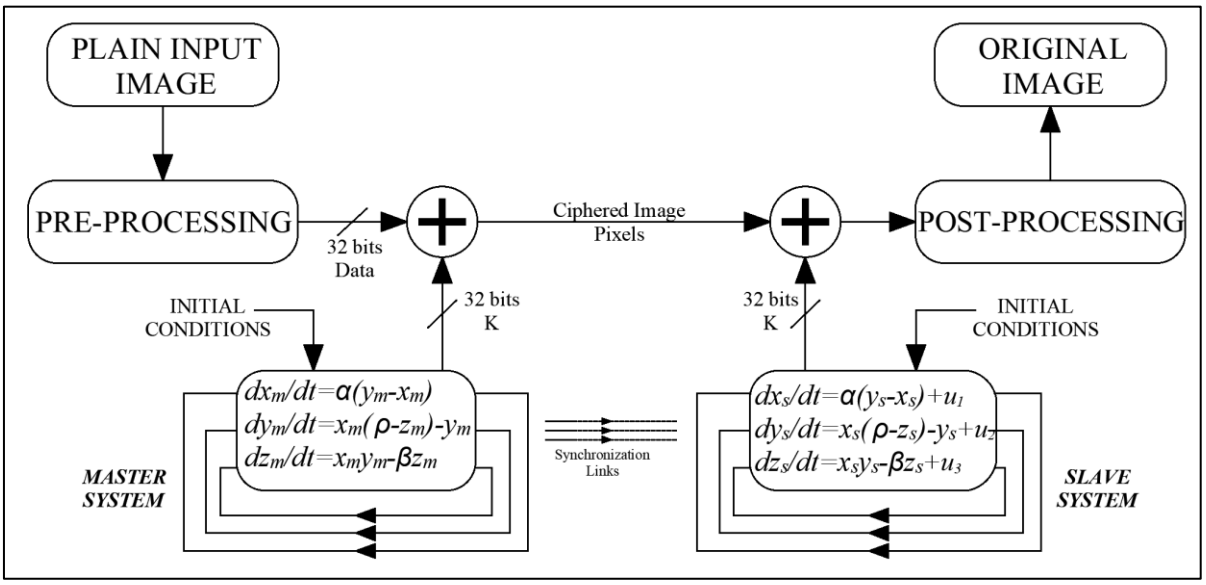

Figure 3-2 Block Diagram of Image Encryption System Using Proposed Algorithm 1

3.2.1. Algorithm Mathematical Description

The mathematical description of the transmitter (master) and receiver (slave) is depicted in equations 3.1 and 3.2 shown below. [81]

$$\begin{aligned} dx_m/dt &= \alpha(y_m - x_m) \\ dy_m/dt &= x_m(\rho - z_m) - y_m \\ dz_m/dt &= x_m y_m - \beta z_m \end{aligned} \quad (3.1) \qquad \begin{aligned} dx_s/dt &= \alpha(y_s - x_s) \\ dy_s/dt &= x_s(\rho - z_s) - y_s \\ dz_s/dt &= x_s y_s - \beta z_s \end{aligned} \quad (3.2)$$

Where the above ordinary differential equations represent Lorenz chaotic system, the subscript m denote to the master system (transmitter), while the s subscript denote to the slave system (receiver). These ODEs will be solved numerically two times in the transmitter and receiver to extract the x dynamical response signal. The x dynamical response is adopted to be used as a secret key

after converting it to binary stream. The generated random binary stream will be XOR-ed with input plain image pixels to generate the ciphered image.

3.2.2. Overall System Design Based XSG Model

Using Xilinx system generator XSG blocks that are set up with 32-bit fixed-point data representation, the transmitter, receiver, and adaptive synchronization systems are designed and implemented in this section using the Lorenz chaotic system. The 32-bit fixed point XSG model is created using the Fix32 18 data format, where 32 denotes the total number of bits divided as follows: The sign bit is one bit, there are 18 fractional bits, and there are 13 integer bits. The four subsystems that make up the overall system design are adaptive synchronization, transmitter/receiver, encryption/decryption process, and preprocessing/postprocessing systems.

3.2.2.1. Transmitter/ Receiver Synchronization

In order to ensure successful recovery of the encrypted data in the receiver side (slave system), the encryption key should be identically regenerated in the slave system to perform a successful decryption process. To ensure that the decryption key is completely identical to the encryption one, the synchronization between the two communicated nodes should be achieved. It is known that the chaotic systems are aperiodic, and have unpredictable behavior, where these systems are defying the synchronization, but synching two chaotic or hyperchaotic systems are still possible by using one of the known synchronization techniques. In this work, adaptive feedback controller is adopted and implemented to provide the necessary synchronization between the master and slave systems. If the control part combined to the slave system, then this synchronization kind is called as adaptive synchronization.

In adaptive feedback controller technique, the dynamical responses of the master system (x_m, y_m, z_m) are transmitted to the slave system (especially in

the beginning of systems boot) to calculate the error between the master and slave systems responses as shown in figure 3-3. The three difference signals are added to the slave ODEs to minimize the mismatch between them gradually. Equations 3.3 shown below depicts the mathematical expressions of error differences between the responses. Figure 3-4 shows the calculated dynamical error between the master and receiver systems with respect to time.

$$\begin{aligned} e_x &= \frac{dx_m}{dt} - \frac{dx_s}{dt} = \alpha(y_m - x_m) - \alpha(y_s - x_s) \\ e_y &= \frac{dy_m}{dt} - \frac{dy_s}{dt} = x_m(\rho - z_m) - y_m - (x_s(\rho - z_s) - y_s) \\ e_z &= \frac{dz_m}{dt} - \frac{dz_s}{dt} = x_m y_m - \beta z_m - (x_s y_s - \beta z_s) \end{aligned} \tag{3.3}$$

These error signals will be multiplied by a factor called signal gain (usually its value 10 to 20), the multiplication result is known as control signals (CS) as shown in equations 3.4 and figure 3-2.

$$\begin{aligned} CS_x &= G \times e_x = G \times [\alpha(y_m - x_m) - \alpha(y_s - x_s)] \\ CS_y &= G \times e_y = G \times [x_m(\rho - z_m) - y_m - (x_s(\rho - z_s) - y_s)] \\ CS_z &= G \times e_z = G \times [x_m y_m - \beta z_m - (x_s y_s - \beta z_s)] \end{aligned} \tag{3.4}$$

The above control signals are added to the ODEs of the slave system to force it to be converging to the same behavior of the master system. Slave system ODEs are adjusted to be as follows:

$$\begin{aligned} dx_s/dt &= \alpha(y_s - x_s) + CS_x \\ dy_s/dt &= x_s(\rho - z_s) - y_s + CS_y \\ dz_s/dt &= x_s y_s - \beta z_s + CS_z \end{aligned} \tag{3.5}$$

The slave system (with adaptive controller) is implemented using the updated ODEs that shown in equations 3.5. These ODEs represent the Lorenz

system that has the ability to synchronize the master system even if they start from different initial conditions (starting point) after a while of time. Figure 3-3, illustrates the error dynamical signals ($\mathbf{e_x}$, $\mathbf{e_y}$, and $\mathbf{e_z}$) between master system whose starting point is (x(0)=50, y(0)=60, and z(0)=40), and the slave system (with adaptive controller) that starts at (10, 20, and 30 for the dynamical responses x, y, and z), as shown in figure 3-3 after 0.25 second error signals goes to zero and the synchronization between master and slave has been achieved.

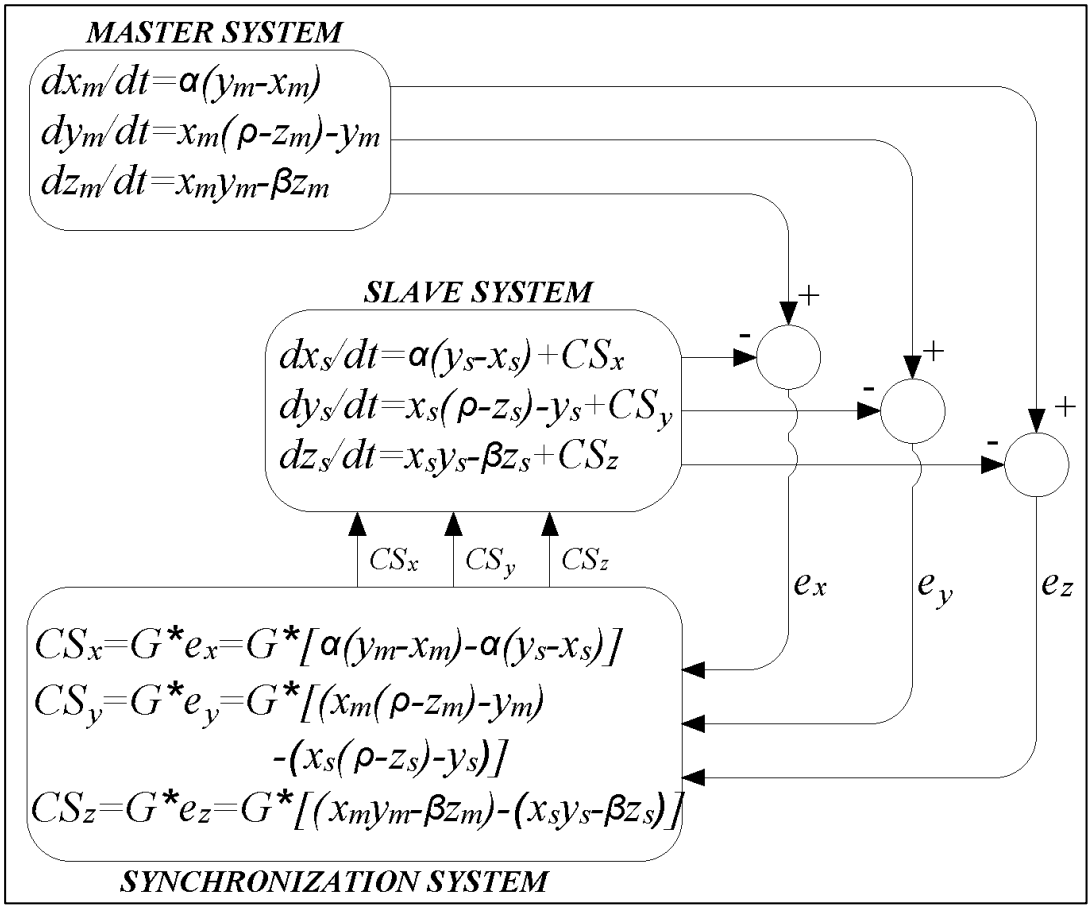

Figure 3-3 Block Diagram of Algorithm 1 Master/Slave Synchronization

Figure 3-5 depicts the **XSG** model of the adaptive controller to provide the necessary synchronization between the master and slave systems.

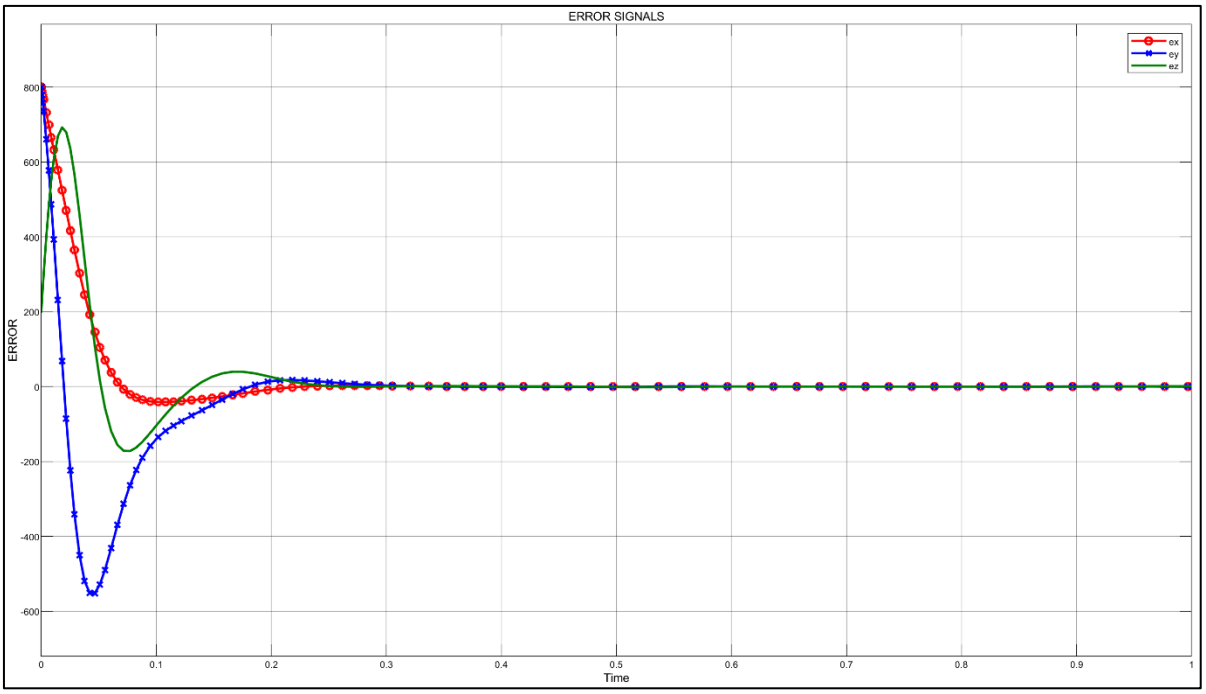

Figure 3-4 Master/ Slave Error Signals

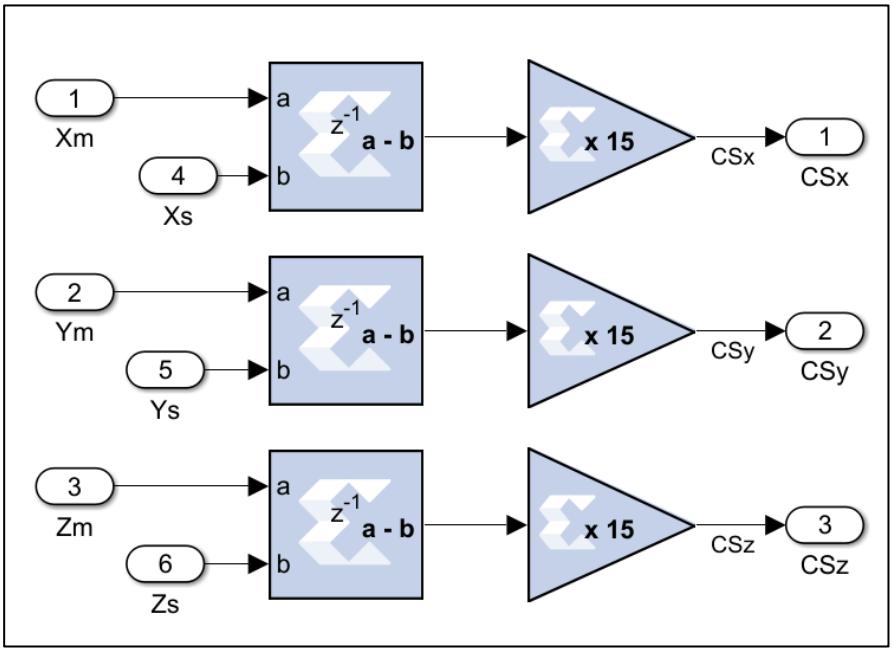

Figure 3-5 Adaptive Feedback Controller XSG Model

3.2.2.2. Encryption/Decryption Process Via XOR operation

Stream cipher encryption technique is the adopted type for encryption/decryption in this work, where the input plain image is converted into serial pixels and then to serial bits (via preprocessing system). At the same time, the x component of the dynamical chaotic system is also converted into

serial bits after solving the overall nonlinear system numerically. The generated bit streams are considered as the plaintext data and secret key, those bit streams are XOR-ed together to get the encrypted images.

3.2.2.3. Preprocessing and Postprocessing Systems

Figure 3-6, illustrates the preprocessing subsystem, where it consists of Matlab/Simulink blocks: Matrix Transpose, Reshape, to frame, and Unbuffer. The combination of these blocks used to convert the image matrix into serial samples each sample contain 8 bits, as a prior stage for encryption. Then these parallel bits (samples) are converted into serial bit stream by using the parallel to serial conversion, in order to encrypt theme using XOR operation with x-dynamic of the chaotic system.

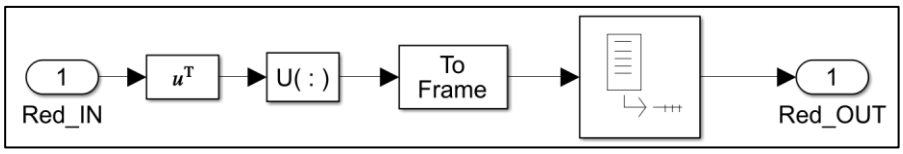

Figure 3-6 Preprocessing System Blocks

On the other hand, figure 3-7 presents the block combination that construct the post-processing subsystem that operate in reverse mode to the preprocessing subsystem, where the serial bits are combined together to form serial samples each one of 8 bits, by a means of serial to parallel conversion. These serial samples are then combined together to construct the image matrix again. The post-processing subsystem is consisted of Buffer, Reshape, Matrix Transpose, and Unit8 Matlab/Simulink blocks.

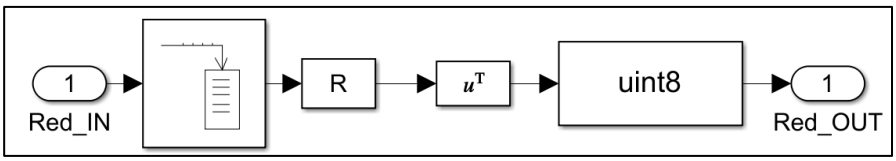

Figure 3-7 Postprocessing System Blocks

3.2.2.4. Transmitter/Receiver Systems Design

The set of equations in 3.1 and in 3.5 are solved two times to implement the systems using two different numerical integration methods to figure out the most effective method and to calibrate the design and simulation environment (XSG and Matlab). The first one is the Forward Euler integration method and the second one is the Runge-Kutta methods. The general formula of the Forward Euler method is presented in equation 3.6

$$y_{t+1} = y_t + hf(x_t, y_t) \tag{3.6}$$

The Euler solution (based on expression 3.6) is used to implement the transmitter (master) and receiver (slave) systems of the proposed cryptosystem using the XSG blocks that configured using **Fix32_18**. The XSG model is used then to generate the bit file that will be used later to configure the FPGA board.

Figures 3-8 and 3-9, illustrate the 32-bits fixed point XSG model for the transmitter and the receiver subsystems designed with forward Euler integration method. Figure 3-10 show the overall system based on Forward Euler method contains the transmitter/ receiver, adaptive feedback controller synchronization, and preprocessing/postprocessing systems.

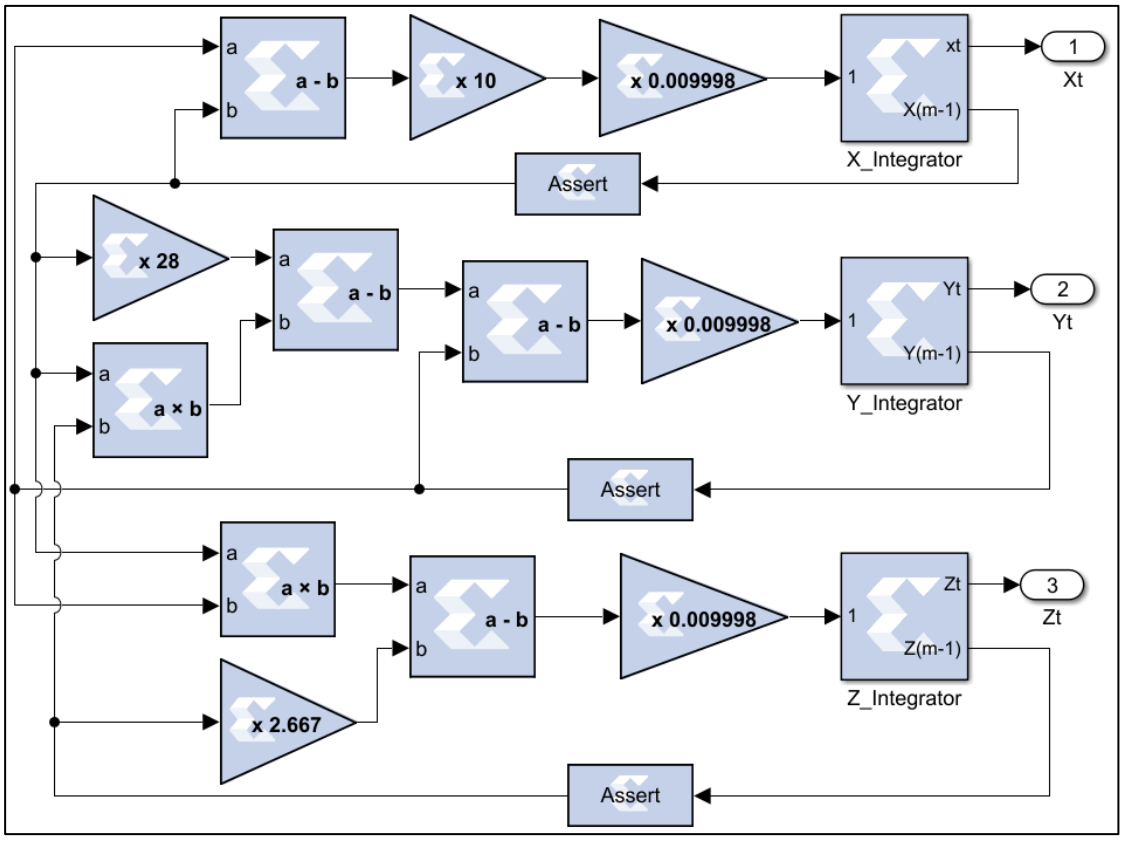

Figure 3-8 32-bit Fixed-Point Master Lorenz Chaotic System Based on Forward Euler

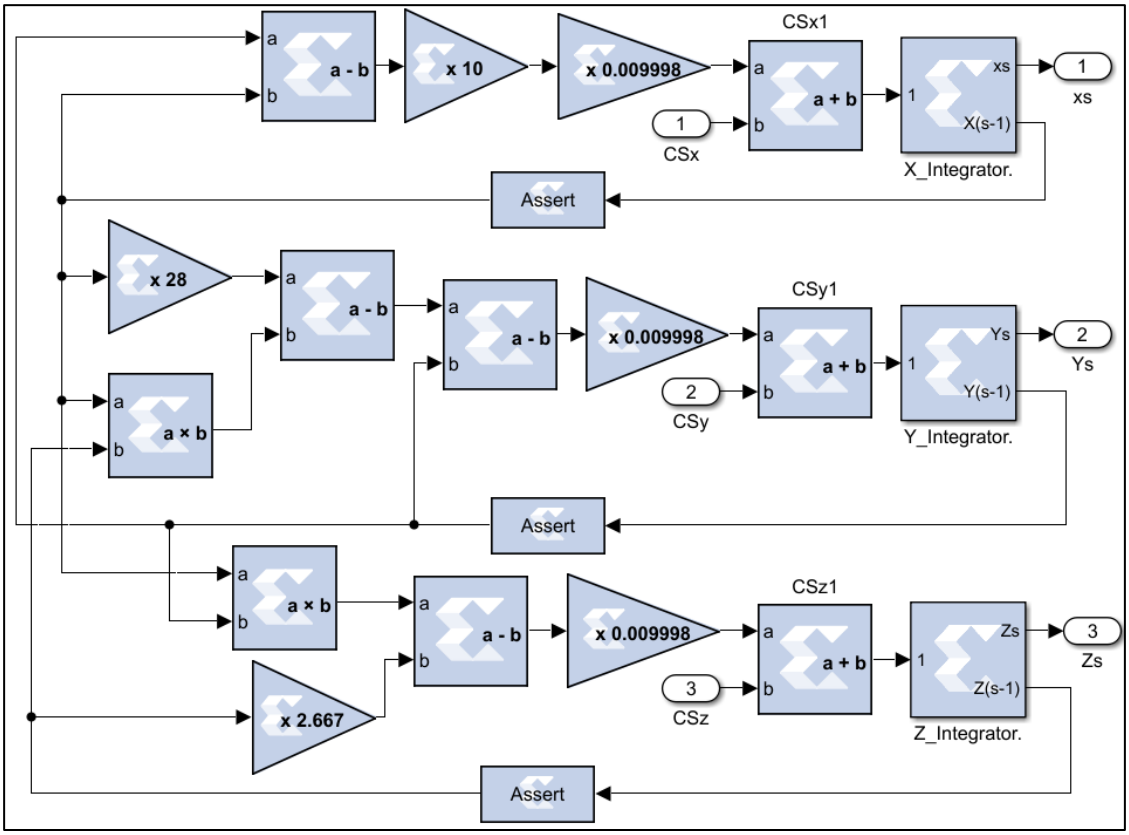

Figure 3-9 32-bit Fixed-Point Slave Lorenz Chaotic System Based on Forward Euler

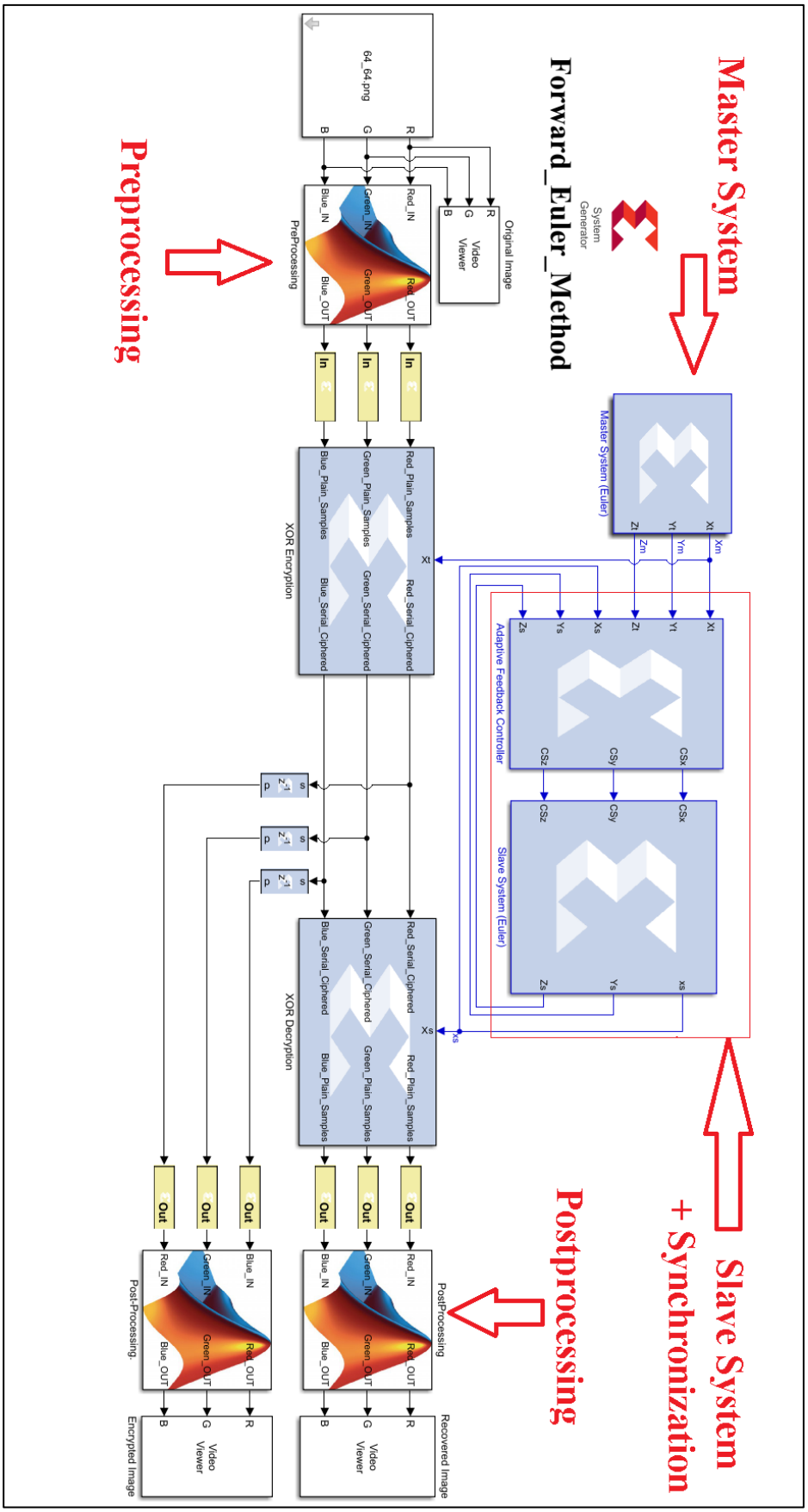

Figure 3-10 Overall Communication System Based on Euler Method

While on the other hand, the Runge-Kutta method is described by the mathematical expression presented in equations 3-7.

$$\begin{aligned} K_1 &= hf(x_t, y_t) \\ K_2 &= hf(x_t + \frac{h}{2}, y_t + \frac{k_1}{2}) \\ K_3 &= hf(x_t + \tfrac{h}{2}, y_t + \tfrac{k_2}{2}) \\ K_4 &= hf(x_t + h, y_t + k_3) \\ y_{t+1} &= y_t + \tfrac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{aligned} \tag{3.7}$$

The Runge-Kutta method solution (based on expressions 3.7) is used to implement the transmitter (master) and receiver (slave) systems of the proposed cryptosystem using the XSG blocks that configured using **Fix32_18**, where 32-bits fixed point data representation is also adopted in this XSG model for the transmitter and the receiver systems. The XSG model is used then to generate the bit file that will be used later to configure the FPGA board.

Figure 3-11 presents the K parameters XSG model of the Runge-Kutta method based of the expressions 3.7 (previously explained in chapter two).

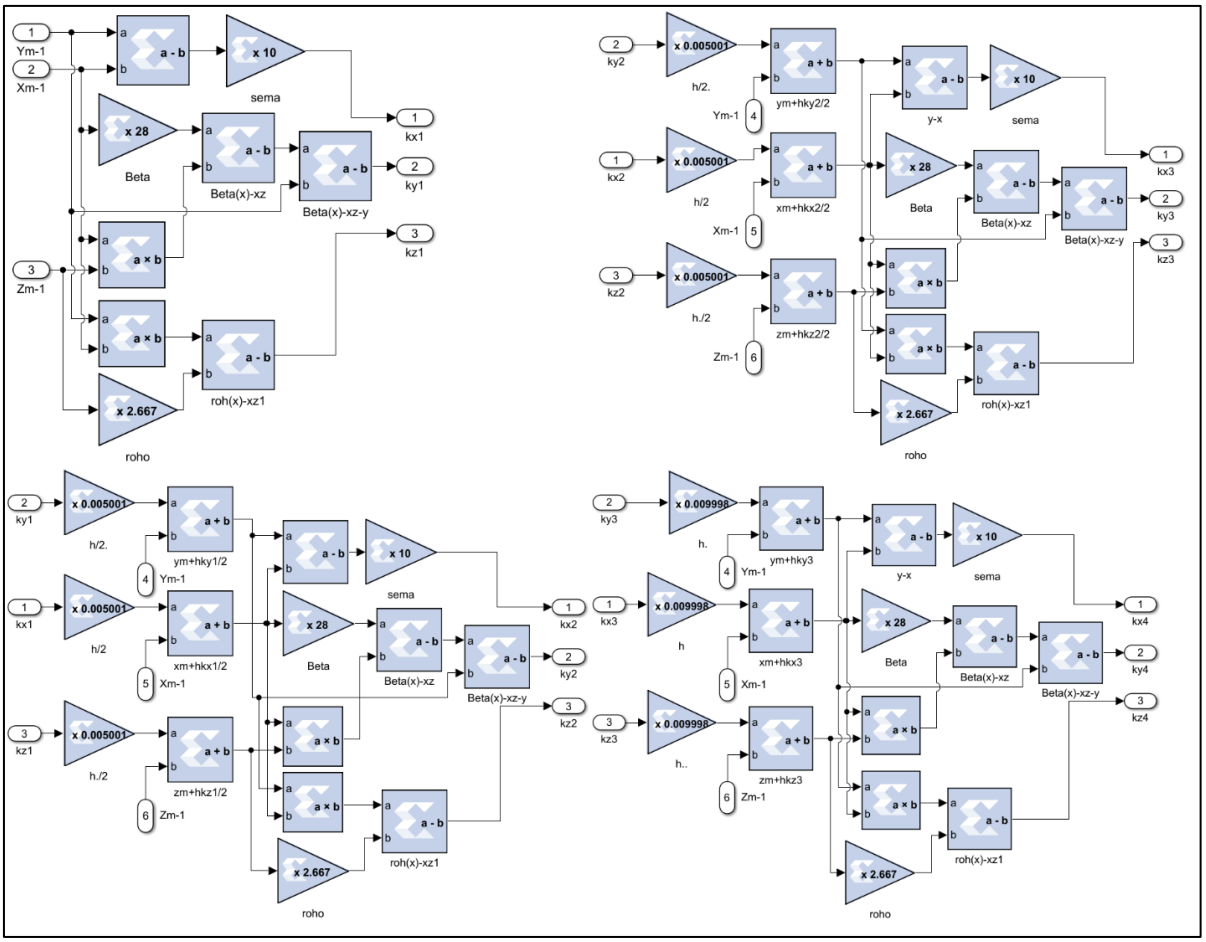

Figure 3-11 XSG Units Block Diagram of K1, K2, K3, and K4

The outputs of the Ki units that presented in figure 3-10 above are combined together to construct the estimated or approximated signal that is calculated based on expression 3.21. The XSG model combination for the estimated signal is presented in figure 3-12. Figure 3-13 shows the outside connection of the K parameters and the estimated signal, while figure 3-14 illustrates the overall system based on Runge-Kutta integration method that contains the transmitter/ receiver, adaptive feedback controller synchronization, and preprocessing/postprocessing systems.

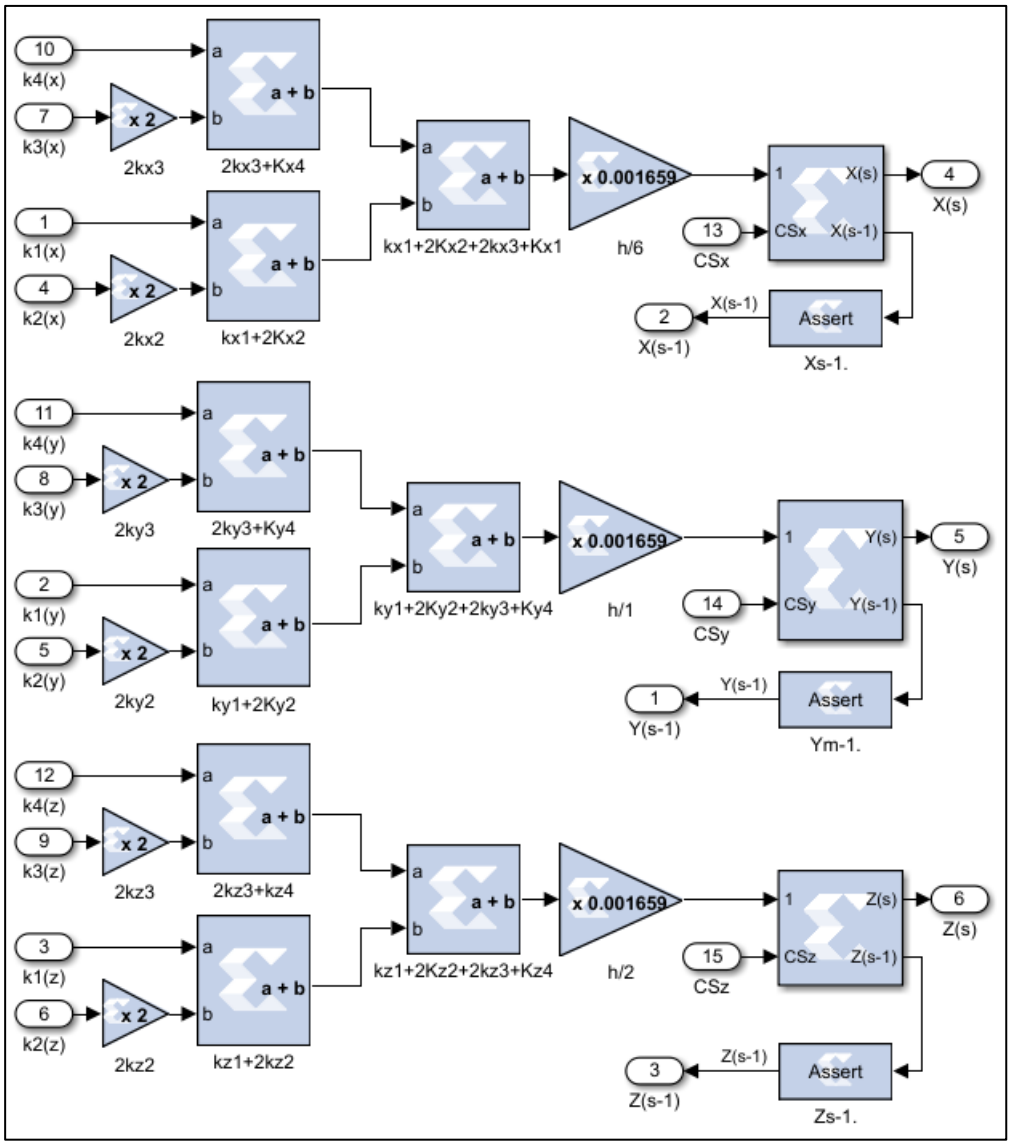

Figure 3-12 Runge-Kutta Based Estimated Signal

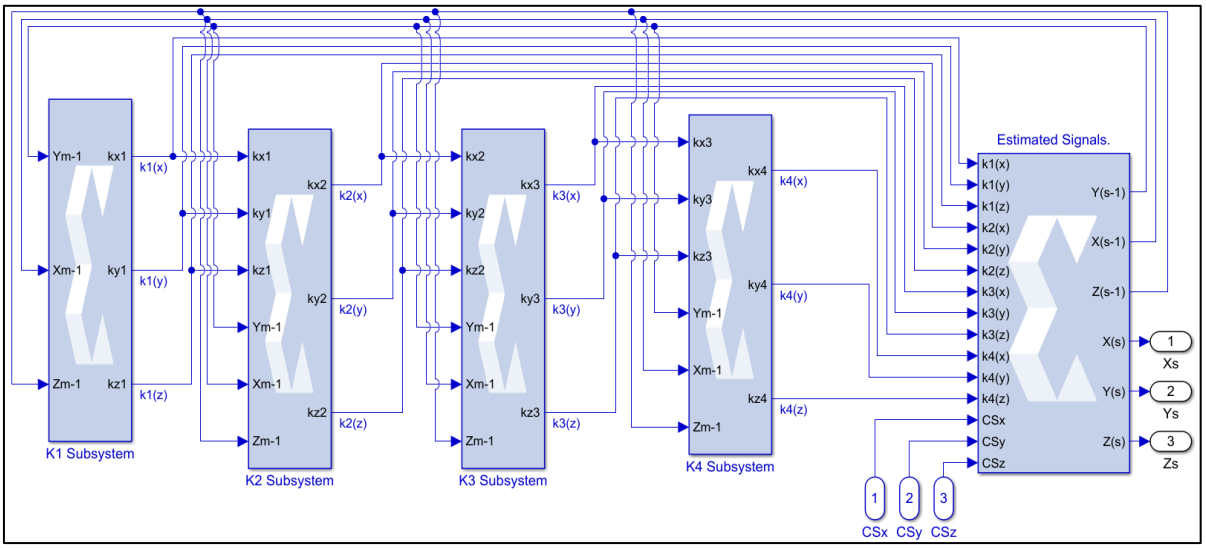

Figure 3-13 Overall Connection of Transmitter/Receiver System

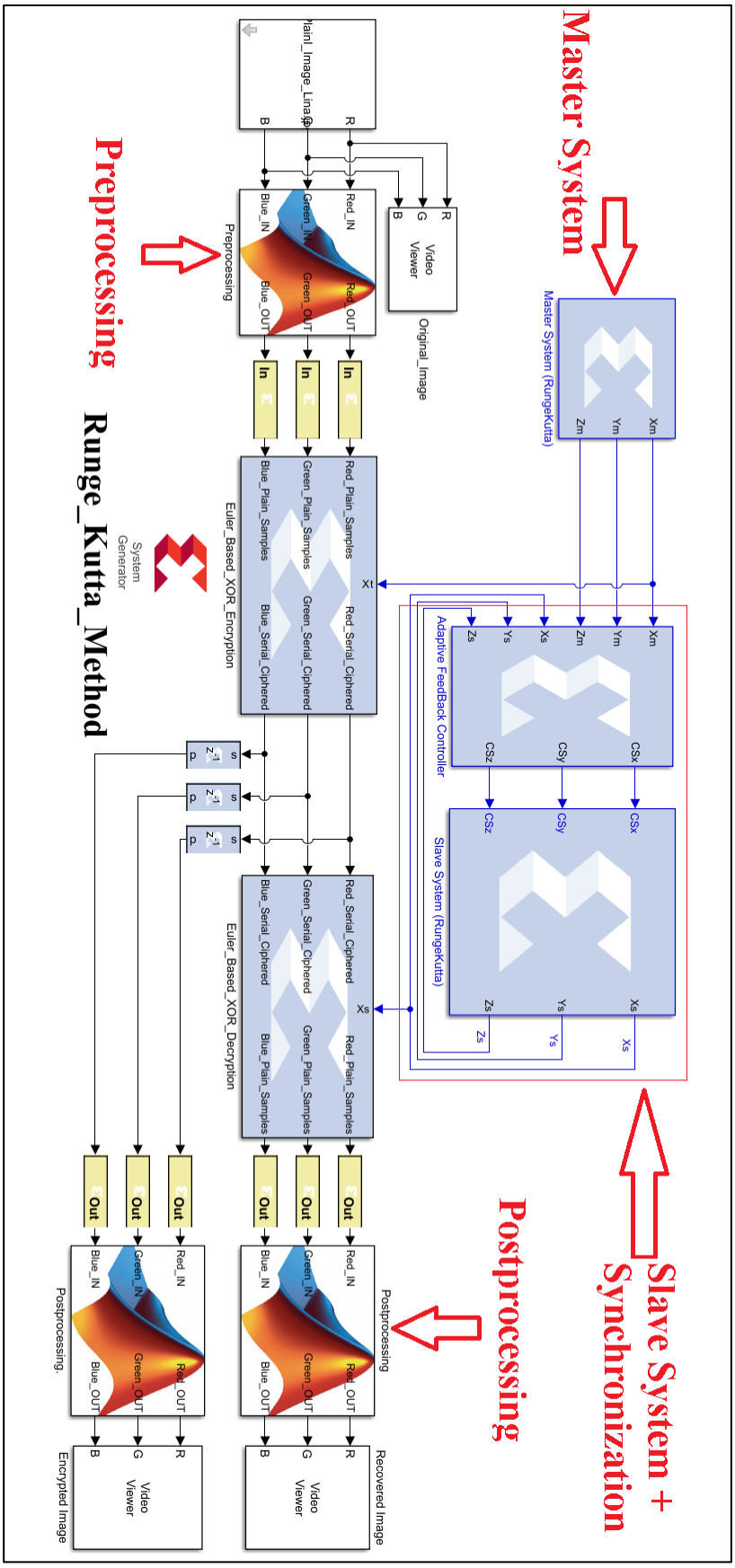

Figure 3-14 Overall Communication System Based on Runge-Kutta Method

3.2.3. FPGA Co-simulation and Implementation

The proposed communication system based on Lorenz chaotic oscillator has been implemented with FPGA PYNQ-Z1 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board. JTAG link (based on Ethernet cable connection) has been adopted for the communication between the PC and the board. The plain image is called (from its location in the PC) and sent to the board serially to encrypt them, after completing the encryption process the encrypted samples are sent back to the PC to display the ciphered image. The hardware platform used to implement the proposed algorithms is PYNQ-Z1 zynq xc7z020 board. Corresponding to available hardware characteristics, there are no limitations in FPGA implementation of the systems especially the maximum pixels per image, where it can accept any image size with any dimensions.

3.2.3.1. Forward Euler Based System Implementation

The proposed communication system in algorithm 1 (which solved numerically using forward Euler integration method) has been implemented using FPGA board as shown in figure 3-15. All the system components including master system, slave system, synchronization and the links which are described and connected in figure 3-9 has been implemented inside the FPGA board. The JTAG connection via point-to-point ethernet cable has been adopted to provide the necessary link with PC.

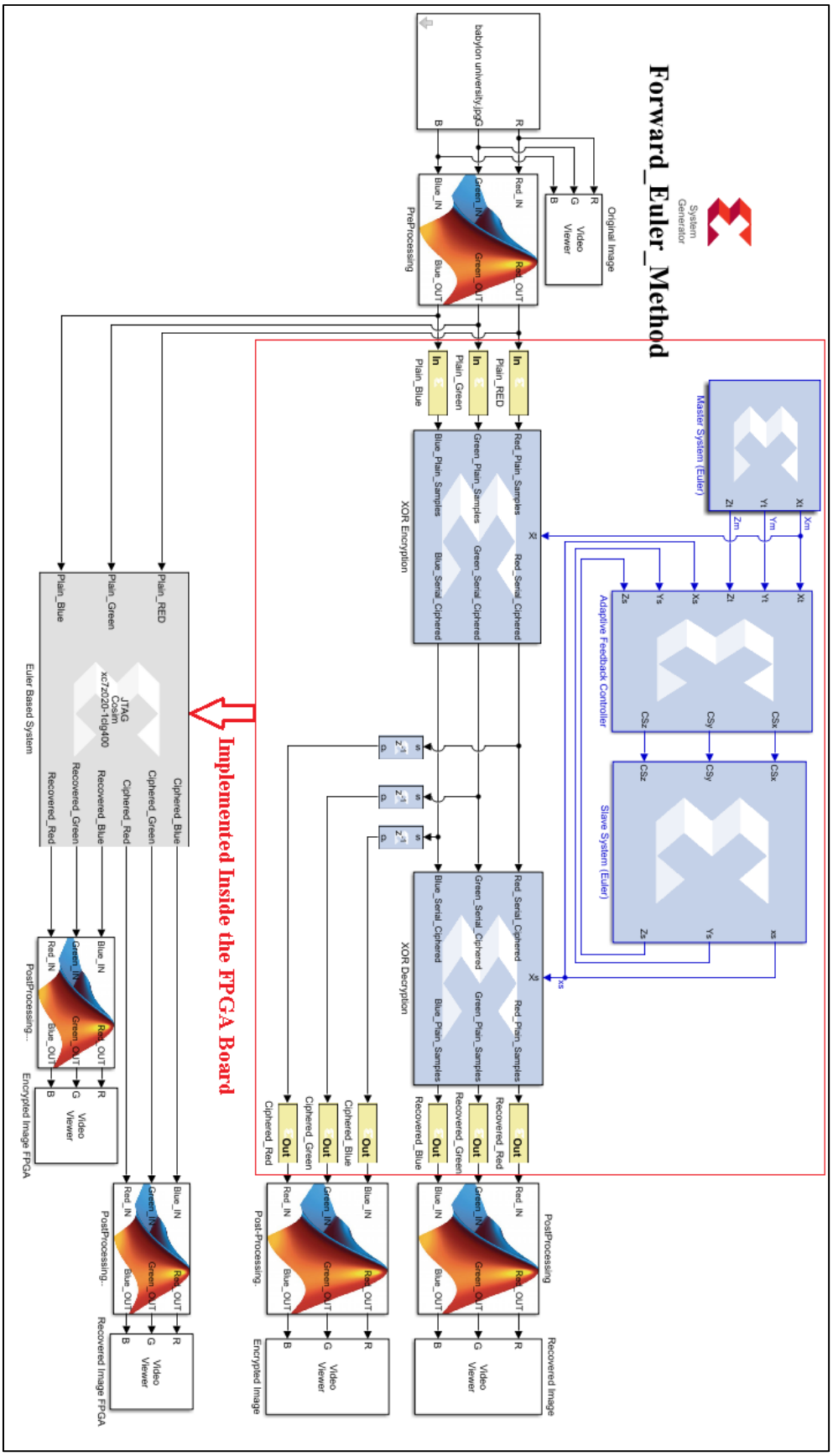

Figure 3-15 System Co-simulation of Algorithm 1 Based on Euler Method

Device utilization summary is presented in table 3-2 below, where it clears the available and utilized resources based on this design. The total on chip power consumption is 0.253 watt with junction temperature of 27.9 C.

Table 3-2 Board Utilization in Algorithm 1 (Forward Euler)

Resource	Utilization	Available	Utilization %
Look Up Table **(LUT)**	2002	53200	3.76
Look Up Table RAM **(LUTRAM)**	1	17400	0.01
Flip Flops **(FF)**	1334	106400	1.25
Block RAM **(BRAM)**	2	140	1.43
Digital Signal Processing **(DSP)**	36	220	16.36
Input Output **(IO)**	1	125	0.8
Global Buffer **(BUFG)**	4	32	12.5
Mixed Mode Clock Manager **(MMCM)**	1	4	25

3.2.3.2. Runge-Kutta Based System Implementation

As aforementioned, the proposed communication system in algorithm 1 is solved two times, the first with Forward Euler and the second with Runge-Kutta methods, in this section the FPGA implementation using Runge-Kutta method is presented. All the system components including master system, slave system, synchronization and the links which are described and connected in figure 3-13 has been implemented inside the FPGA board as presented in figure 3-16. The JTAG connection via point-to-point ethernet cable has been adopted to provide the necessary link with PC.

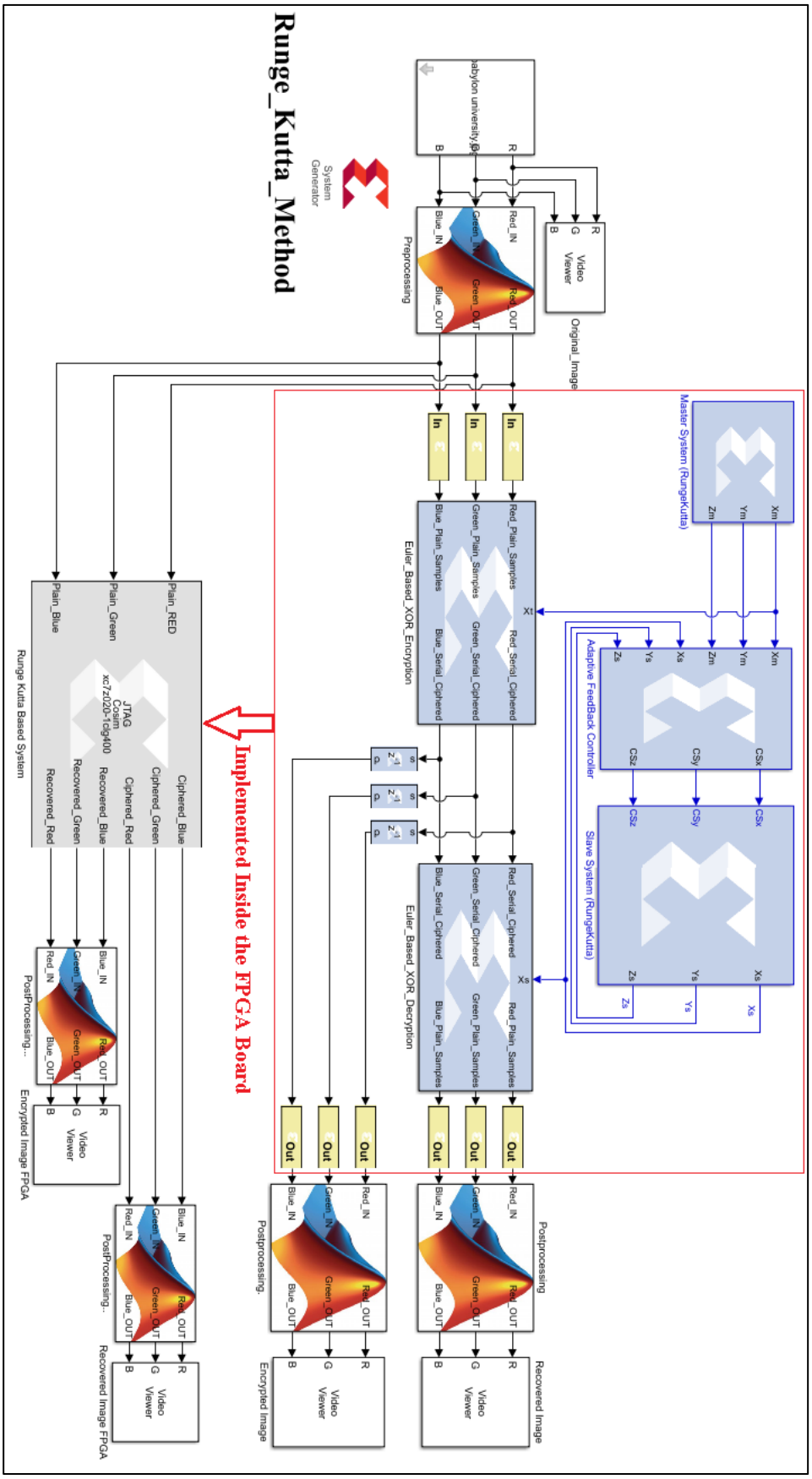

Figure 3-16 System Co-simulation of Algorithm 1 Based on Runge-Kutta Method

Device utilization summary is presented in table 3-3 below, where it clears the available and utilized resources based on this design. The total on chip power consumption is 0.395 watt with junction temperature of 29.6 C.

Table 3-3 Board Utilization in Algorithm 1 (Runge-Kutta)

Resource	Utilization	Available	Utilization %
Look Up Table **(LUT)**	5430	53200	10.21
Look Up Table RAM **(LUTRAM)**	1	17400	0.01
Flip Flops **(FF)**	1861	106400	1.75
Block RAM **(BRAM)**	2	140	1.43
Digital Signal Processing **(DSP)**	144	220	65.45
Input Output **(IO)**	1	125	0.8
Global Buffer **(BUFG)**	4	32	12.5
Mixed Mode Clock Manager **(MMCM)**	1	4	25

As shown in tables 3-2 and 3-3, it is clear that the Runge-Kutta method consumes resources more than Euler due to its repetition behavior during the solution of the dynamical nonlinear system and due to this fact, the reset of the proposed algorithms will be solved using Euler integration method.

3.3. Algorithm (2): Multidimensional Hyperchaotic System Based on XOR Mixture of Dynamical Systems

The proposed image encryption system in Algorithm 2 is presented in the block diagram in figure 3-17. The encryption system consists of three different nonlinear hyperchaotic systems with different dimensions. These nonlinear systems are numerically solved and the x components are converted to binary stream and XOR-ed together to generate highly random bit stream.

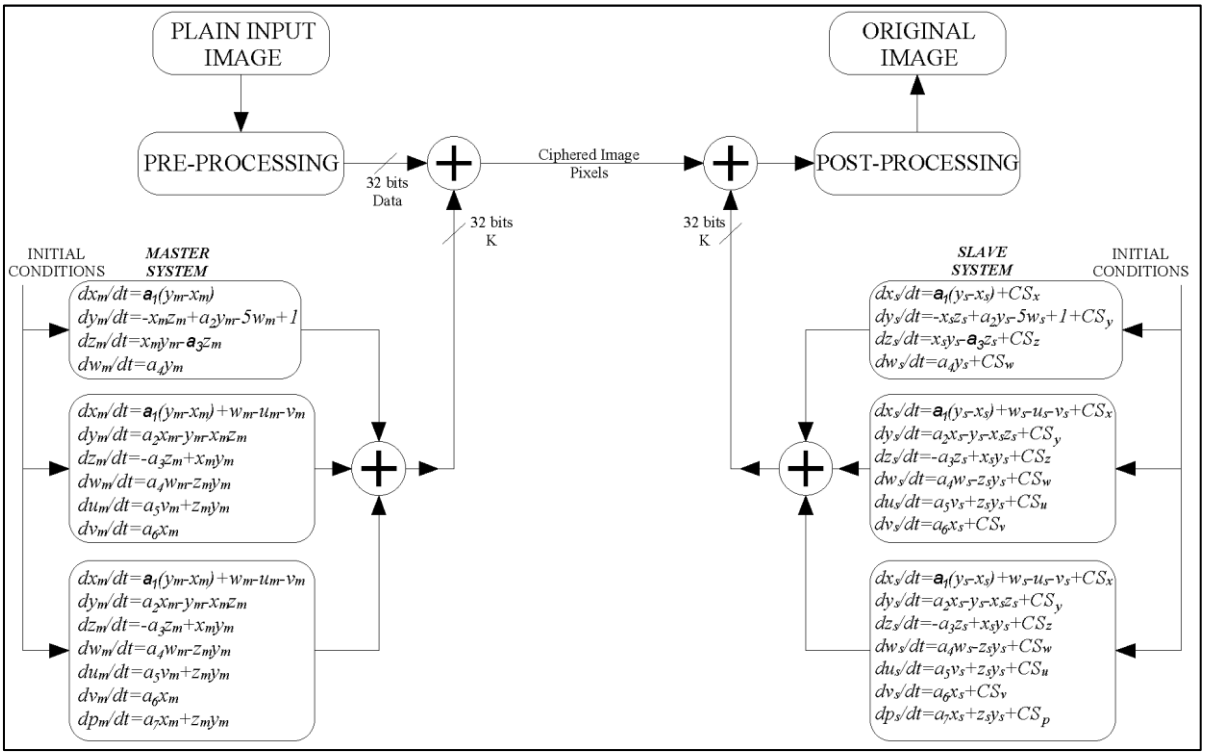

Figure 3-17 Image Encryption System Block Diagram Using Proposed Algorithm 2

3.3.1. Algorithm Mathematical Description

The proposed secure communication system is based on adopting three different nonlinear hyperchaotic systems. These nonlinear systems have different dimensions, different ordinary differential equations, and different behavior. The first hyperchaotic system has four-dimensional ordinary differential equations that presented in equation 3.8.

$$\begin{aligned}
\frac{dx_m}{dt} &= a_1(y_m - x_m) & \frac{dx_s}{dt} &= a_1(y_s - x_s) \\
\frac{dy_m}{dt} &= -x_m z_m + a_2 y_m - 5w_m + 1 & \frac{dy_s}{dt} &= -x_s z_s + a_2 y_s - 5w_s + 1 \\
\frac{dz_m}{dt} &= x_m y_m - a_3 z_m & \frac{dz_s}{dt} &= x_s y_s - a_3 z_s \\
\frac{dw_m}{dt} &= x_m y_m - a_4 z_m & \frac{dw_s}{dt} &= x_s y_s - a_4 z_s
\end{aligned} \tag{3.8}$$

The second hyperchaotic system consists of six-dimensional ordinary differential equations which presented in expression 3.9. The last nonlinear system is consisting of seven-dimensional ordinary differential equations that presented in expression 3.10. The subscripts m and s attached with ODEs dynamical states are denoted to master and slave systems respectively

$$\begin{aligned}
\frac{dx_m}{dt} &= a_1(y_m - x_m) + w_m - u_m - v_m & \frac{dx_s}{dt} &= a_1(y_s - x_s) + w_s - u_s - v_s \\
\frac{dy_m}{dt} &= a_2 x_m - y_m - x_m z_m & \frac{dy_s}{dt} &= a_2 x_s - y_s - x_s z_s \\
\frac{dz_m}{dt} &= -a_3 z_m + x_m y_m & \frac{dz_s}{dt} &= -a_3 z_s + x_s y_s \\
\frac{dw_m}{dt} &= a_4 w_m - y_m z_m & \frac{dw_s}{dt} &= a_4 w_s - y_s z_s \\
\frac{du_m}{dt} &= a_5 v_m + y_m z_m & \frac{du_s}{dt} &= a_5 v_s + y_s z_s \\
\frac{dv_m}{dt} &= a_6 x_m & \frac{dv_s}{dt} &= a_6 x_s
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
&\frac{dx_m}{dt} = a_1(y_m - x_m) + w_m - u_m - v_m && \frac{dx_s}{dt} = a_1(y_s - x_s) + w_s - u_s - v_s \\
&\frac{dy_m}{dt} = a_2 x_m - y_m - x_m z_m && \frac{dy_s}{dt} = a_2 x_s - y_s - x_s z_s \\
&\frac{dz_m}{dt} = -a_3 z_m + x_m y_m && \frac{dz_s}{dt} = -a_3 z_s + x_s y_s \\
&\frac{dw_m}{dt} = a_4 w_m - y_m z_m && \frac{dw_s}{dt} = a_4 w_s - y_s z_s \\
&\frac{du_m}{dt} = a_5 v_m + y_m z_m && \frac{du_s}{dt} = a_5 v_s + y_s z_s \\
&\frac{dv_m}{dt} = a_6 x_m && \frac{dv_s}{dt} = a_6 x_s \\
&\frac{dp_m}{dt} = a_7 x_m + z_m y_m && \frac{dp_s}{dt} = a_7 x_s + z_s y_s
\end{aligned} \tag{3.10}$$

The parameters and the initial conditions of the systems are selected as shown in table 3-4, according to those values the systems exhibit a hyperchaotic behavior.

Table 3-4 Characteristics of Hyperchaotic Systems [82][83][51]

Hyperchaotic System	Initial Conditions At t = 0	Constants	Lyapunov Exponents	Sum of Lyapunov Exponents
4-Dimensional	x=0.1	a_1=30	L_1=2.1726	-12.91147
	y =0.1	a_2=20	L_2=0.01493	
	z =0.1	a_3=3	L_3=0	
	w =0.1	a_4=0.1	L_4=-15.099	
6-Dimensional	X =1	a_1=10	L_1=0.8	-14.733355
	y =1	a_2=28	L_2=0.3	
	z =1	a_3=8/3	L_3=0	
	w =1	a_4=-1	L_4=-0.5	
	u =1	a_5=8	L_5=-0.7	
	v =1	a_6=3	L_6=-14.7	
7-Dimensional	x=1	a_1=10	L_1=0.6	-14.667183
	y =1	a_2=28	L_2=0.2	
	z =1	a_3=8/3	L_3=0	
	w =1	a_4=-1	L_4=-0.14	
	u =1	a_5=8	L_5=-0.404591	
	v =1	a_6=5	L_6=-0.8	
	p =1	a_7=1	L_7=-14	

On the other hand, the lyapunov exponents are also calculated and presented in table 3-4. The sum of all lyapunov exponents for each system is -12.91147, -14.733355, -14.667183 respectively. Obviously, the sums are less than zero which proof that all of the dynamical systems are in a hyperchaotic state. As shown in the table, each system has two positive lyapunov exponent which proof that the systems are in hyperchaotic state. These systems will be implemented two times for transmitter and receiver systems.

3.3.2. Overall System Design Based XSG Model

The design and implementation of the three different nonlinear hyperchaotic systems is performed in this section using Xilinx system generator XSG blocks which are configured with 32-bit fixed-point data representation (**Fix32_18** data format) to build up the transmitter, receiver systems and adaptive synchronization system. The design of the overall system consists of four subsystems which are: adaptive synchronization, transmitter/ receiver, encryption/decryption process and preprocessing /postprocessing systems.

3.3.2.1. Transmitter/ Receiver Synchronization

The mathematical description of the adaptive feedback controller that provide the necessary synchronization is expressed in the expressions 3.11, 3.12 and 3.13, where the adaptive feedback controller has been designed with three separate controllers circuits, in which each controller is attached with a specific nonlinear hyperchaotic systems in the slave system.

Expression 3.11 presents the mathematical description for the four-dimensional hyperchaotic system error signals. These signals are multiplied with a specific gain to generate the control signals (CS_x). The XSG model implementation is presented in figure 3-18 in which the **Fix32-18** data representation is also adopted.

$$e_x = \frac{dx_m}{dt} - \frac{dx_s}{dt} = a_1(y_m - x_m) - a_1(y_s - x_s)$$

$$e_y = \frac{dy_m}{dt} - \frac{dy_s}{dt} = (-x_m z_m + a_2 y_m - 5w_m + 1) - (-x_s z_s + a_2 y_s - 5w_s + 1) \quad (3.11)$$

$$e_z = \frac{dz_m}{dt} - \frac{dz_s}{dt} = x_m y_m - a_3 z_m - (x_s y_s - a_3 z_s)$$

$$e_w = \frac{dw_m}{dt} - \frac{dw_s}{dt} = x_m y_m - a_4 z_m - (x_s y_s - a_4 z_s)$$

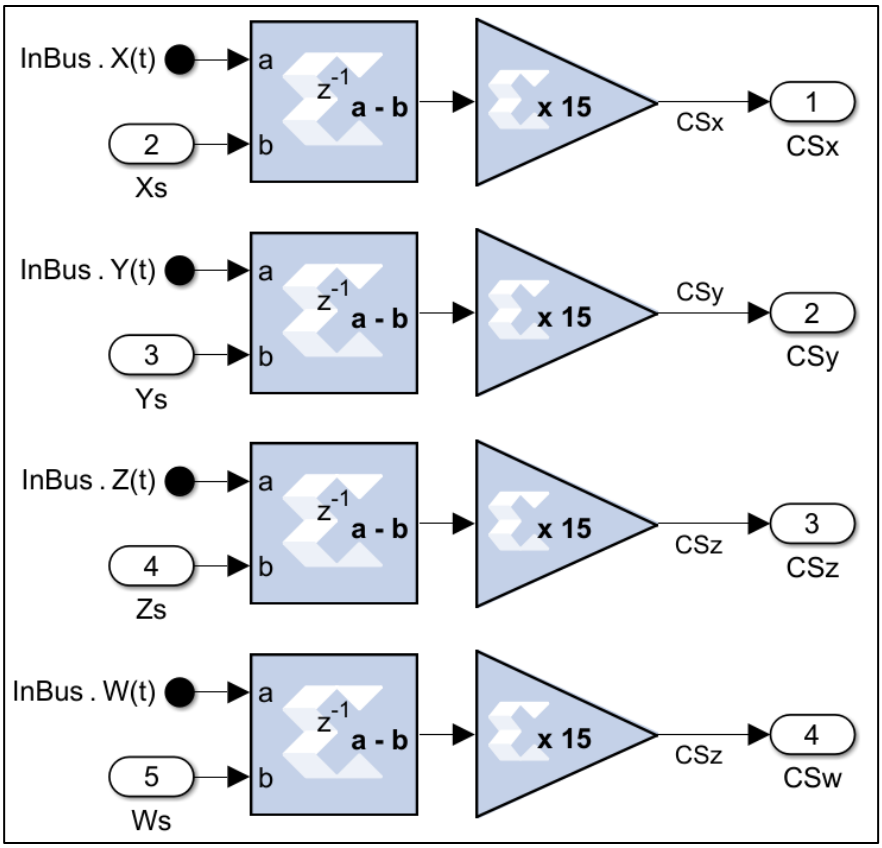

Figure 3-18 Adaptive Feedback Controller (Four-Dimensional System)

Expression 3.12 presents the mathematical description for the six-dimensional hyperchaotic system error signals. These signals are multiplied with a specific gain to generate the control signals (CS_x). The XSG model implementation is presented in figure 3-19.

$$
\begin{aligned}
e_x &= \frac{dx_m}{dt} - \frac{dx_s}{dt} = a_1(y_m - x_m) + w_m - u_m - v_m - (a_1(y_s - x_s) + \\
&\quad w_s - u_s - v_s) \\
e_y &= \frac{dy_m}{dt} - \frac{dy_s}{dt} = a_2 x_m - y_m - x_m z_m - (a_2 x_s - y_s - x_s z_s) \\
e_z &= \frac{dz_m}{dt} - \frac{dz_s}{dt} = -a_3 z_m + x_m y_m - (-a_3 z_s + x_s y_s) \\
e_w &= \frac{dw_m}{dt} - \frac{dw_s}{dt} = a_4 w_m - y_m z_m - (a_4 w_s - y_s z_s) \\
e_u &= \frac{du_m}{dt} - \frac{du_s}{dt} = \mathrm{a_5 v_m} + \mathrm{y_m z_m} - (a_5 v_s + y_s z_s) \\
e_v &= \frac{dv_m}{dt} - \frac{dv_s}{dt} = a_6 x_m - a_6 x_s
\end{aligned}
\tag{3.12}
$$

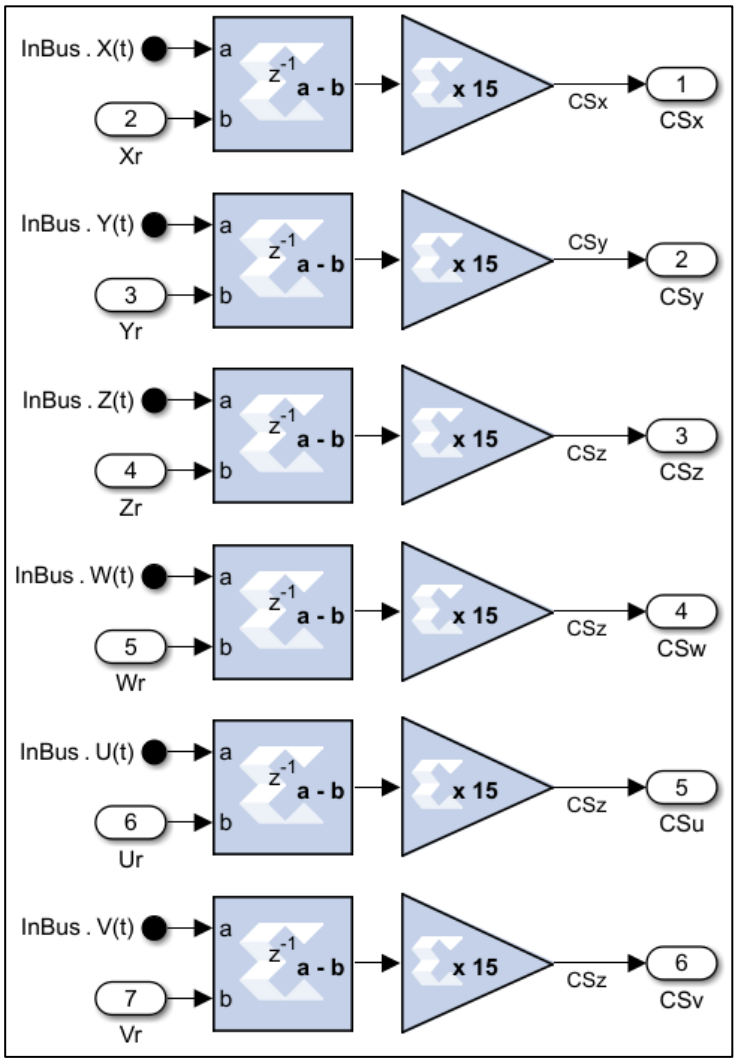

Figure 3-19 Adaptive Feedback Controller (Six-Dimensional System)

Expression 3.13 states the mathematical description for the seven-dimensional hyperchaotic system error signals. These signals are multiplied with a specific gain to generate the control signals (CS_x). The XSG model implementation is presented in figure 3-20.

$$
\begin{aligned}
e_x &= \frac{dx_m}{dt} - \frac{dx_s}{dt} = a_1(y_m - x_m) + w_m - u_m - v_m - (a_1(y_s - x_s) + \\
&\quad w_s - u_s - v_s) \\
e_y &= \frac{dy_m}{dt} - \frac{dy_s}{dt} = a_2 x_m - y_m - x_m z_m - (a_2 x_s - y_s - x_s z_s) \\
e_z &= \frac{dz_m}{dt} - \frac{dz_s}{dt} = -a_3 z_m + x_m y_m - (-a_3 z_s + x_s y_s) \\
e_w &= \frac{dw_m}{dt} - \frac{dw_s}{dt} = a_4 w_m - y_m z_m - (a_4 w_s - y_s z_s) \\
e_u &= \frac{du_m}{dt} - \frac{du_s}{dt} = \mathrm{a_5 v_m} + \mathrm{y_m z_m} - (a_5 v_s + y_s z_s) \\
e_v &= \frac{dv_m}{dt} - \frac{dv_s}{dt} = a_6 x_m - a_6 x_s \\
e_p &= \frac{dp_m}{dt} - \frac{dp_s}{dt} = a_7 x_m + z_m y_m - (a_7 x_s + z_s y_s)
\end{aligned}
\tag{3.13}
$$

The circuit outputs (CS_x, CS_y, CS_z, and CS_w……. To CS_p) are used to control the output of the slave system to make it converges to the same behavior of the master system and to minimize the error between master and slave systems.

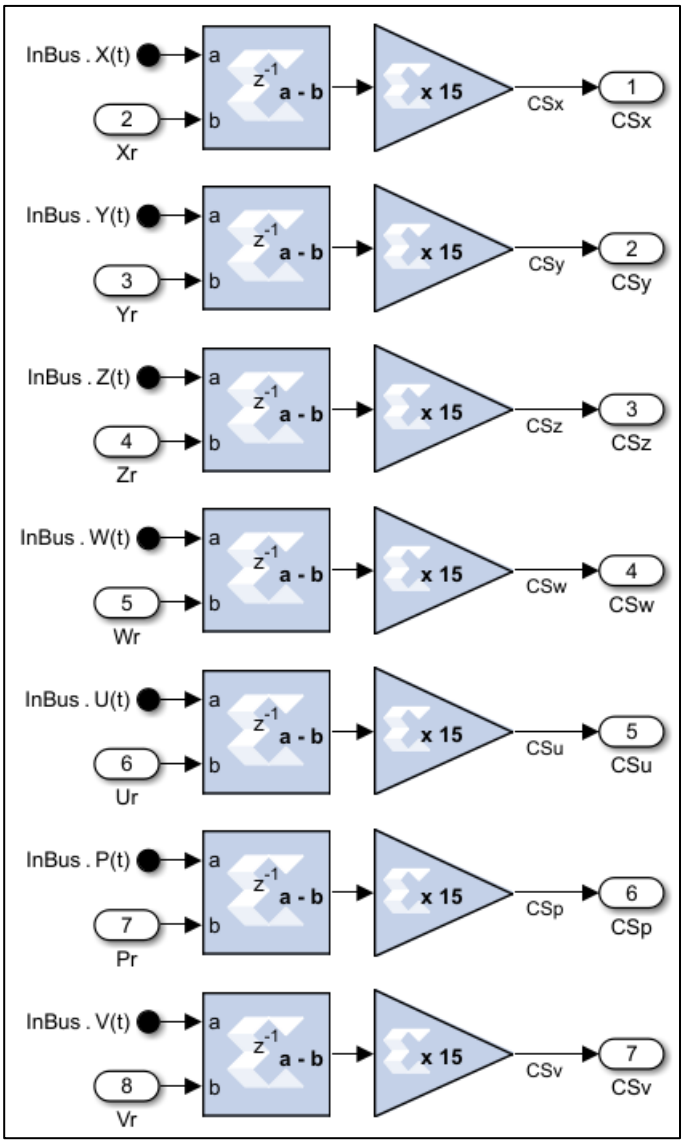

Figure 3-20 Adaptive Feedback Controller (Seven-Dimensional System)

3.3.2.2. Encryption/Decryption Process Via XOR operation

Stream cipher encryption technique is also adopted in this proposed system. The input plain image is converted into serial pixels and then to serial bits (via preprocessing system) and in the same time the x component of the dynamical chaotic system is also converted into serial bits after solving the overall nonlinear system numerically. The generated bit streams are considered as the plaintext data and secret key, those bit streams are XOR-ed to generate the encrypted images.

3.3.2.3. Preprocessing and Postprocessing Systems

Figure 3-21, shown below illustrates the preprocessing subsystem, where it consists of Matlab/Simulink blocks: Matrix Transpose, Reshape, to frame, and Unbuffer. The combination of these blocks used to convert the image matrix into serial samples each sample contain 8 bits, as a prior stage for encryption. Then these parallel bits (samples) are converted into serial bit stream by using the parallel to serial conversion, in order to encrypt theme using XOR operation with x-dynamic of the chaotic system.

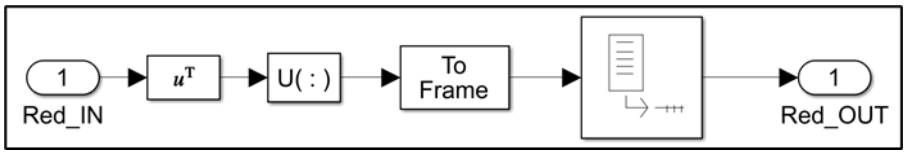

Figure 3-21 Preprocessing System Blocks

On the other hand, Figure 3-22 shows the block combination that construct the post-processing subsystem that operates in reverse mode to the preprocessing subsystem, where the serial bits are combined together to form serial samples each one of 8 bits, by a means of serial to parallel conversion. These serial samples are then combined together to construct the image matrix again. The post-processing subsystem is consisted of Buffer, Reshape, Matrix Transpose, and Unit8 Matlab/Simulink blocks.

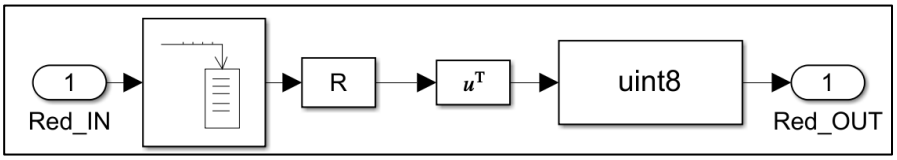

Figure 3-22 Postprocessing System Blocks

3.3.2.4. Transmitter/Receiver Systems Design

The proposed multi-dimensional hyperchaotic generator that used in this communication system is composed from the combination of three different hyperchaotic oscillators. The dimensions of the oscillators are four, six and seven as shown in figure 3-23 below. These oscillators are combined together using XOR logical operation. The output of the XOR operation represents the random bit stream that will be used to encrypt the input plain images.

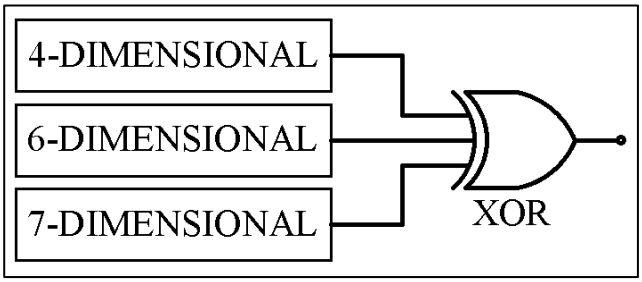

Figure 3-23 Proposed Multi-Dimensional Hyperchaotic System

Figure 3-24, illustrates the 32-bits fixed point XSG model for the proposed multi-dimensional hyperchaotic system generator that used to implement the encryption and decryption systems. Encryption (master) and decryption (slave) systems are designed based on the solution of the ordinary differential equations ODEs of the hyperchaotic systems that described in expressions 3.8, 3.9, and 3.10. The forward Euler integration method has been adopted to solve the ODEs equations numerically due to its simplicity and ease of practical implementation. The output binary random data stream of the encryption (master) and decryption (slave) systems will be used to encrypt and decrypt the plain images with any size.

Figure 3-25 presents the overall proposed communications system including master (transmitter), slave (receiver), feedback controller, preprocessing/postprocessing systems and XOR operations.

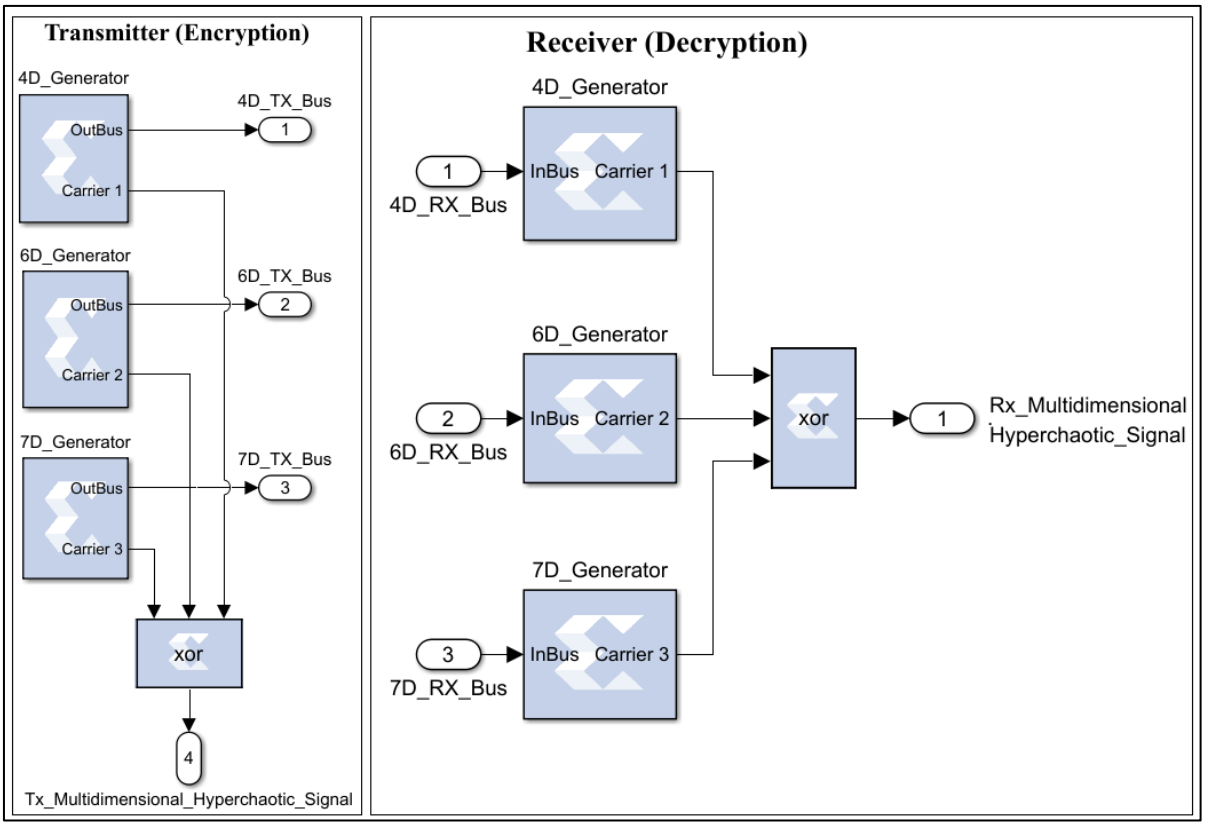

Figure 3-24 32 bits Fixed Point Representation of System Generator for Transmitter and Receiver Systems (Algorithm 2)

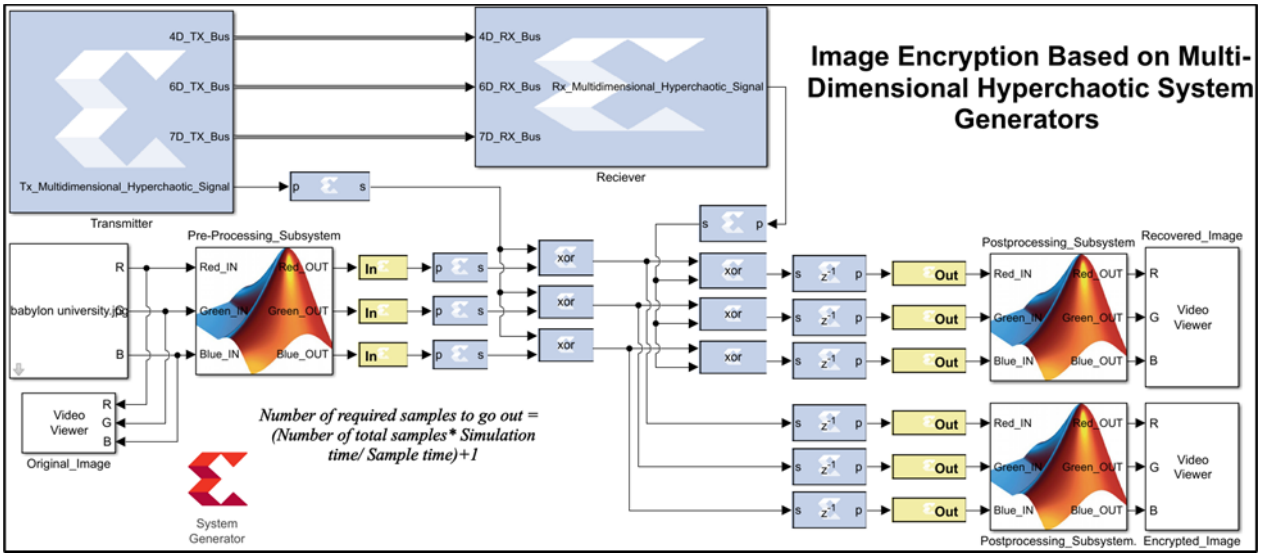

Figure 3-25 Overall Master/ Receiver System

3.3.3. FPGA Co-simulation and Implementation

The proposed multi-dimensional hyperchaotic system has been implemented with FPGA PYNQ-Z1 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board. Figure 3-26, depicted implementation of the proposed cryptographic algorithm using the FPGA board. JTAG link has been adopted for the communication between the PC and the board. The plain image is called from its location in the PC and sent to the board serially to encrypt them. After completing the encryption process the encrypted samples are sent back to the PC to display the ciphered image. On the other hand, in the receiver side, the same steps are carried out to recover the plain image, where the ciphered image is received and converted into serial bits (using the preprocessing system). In the same time, the serial bits are generated from the proposed hyperchaotic system. Finally, XOR operation is performed to produce the original plain bits.

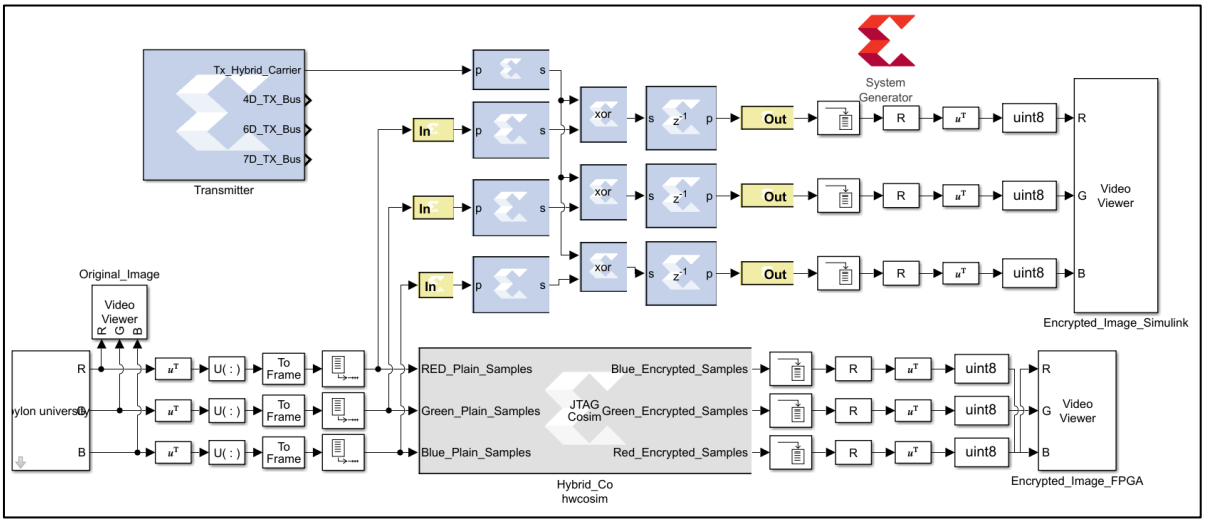

Figure 3-26 Hardware Co-simulation of the Proposed Cryptographic System (Algorithm 2)

Device utilization summary is presented in table 3-5 below for this proposed system, where it clears the available and utilized resources based on this design. The total on chip power consumption is 0.439 Watt with junction temperature of 30.1 C.

Table 3-5 Board Utilization in Algorithm 2

Resource	Utilization	Available	Utilization %
Look Up Table **(LUT)**	9174	53200	17.24
Look Up Table RAM **(LUTRAM)**	1	17400	0.01
Flip Flops **(FF)**	2514	106400	2.36
Block RAM **(BRAM)**	2	140	1.43
Digital Signal Processing **(DSP)**	204	220	92.73
Input Output **(IO)**	1	125	0.80
Global Buffer **(BUFG)**	4	32	12.50
Mixed Mode Clock Manager **(MMCM)**	1	4	25.00

3.4. Algorithm (3): Multidimensional Cascaded Hyperchaotic Systems Based on Chaos Switching Technique.

This communication system is constructed based on the combination of the proposed systems in algorithm 1 and algorithm 2, where this system contains three different dimensional hyperchaotic systems which are four-dimensional, six-dimensional, and seven-dimensional (presented in algorithm 2). These hyperchaotic systems are solved numerically using Forward-Euler integration method. The numerical solutions are combined together using high speed selector switch. The selector switch is electronically controlled by the three-dimensional chaotic system (presented in algorithm 1). The output of the selector switch will be considered as the random bit stream that will be adopted for image encryption purposes using XOR operation. The proposed image encryption block diagram is depicted in figure 3-27.

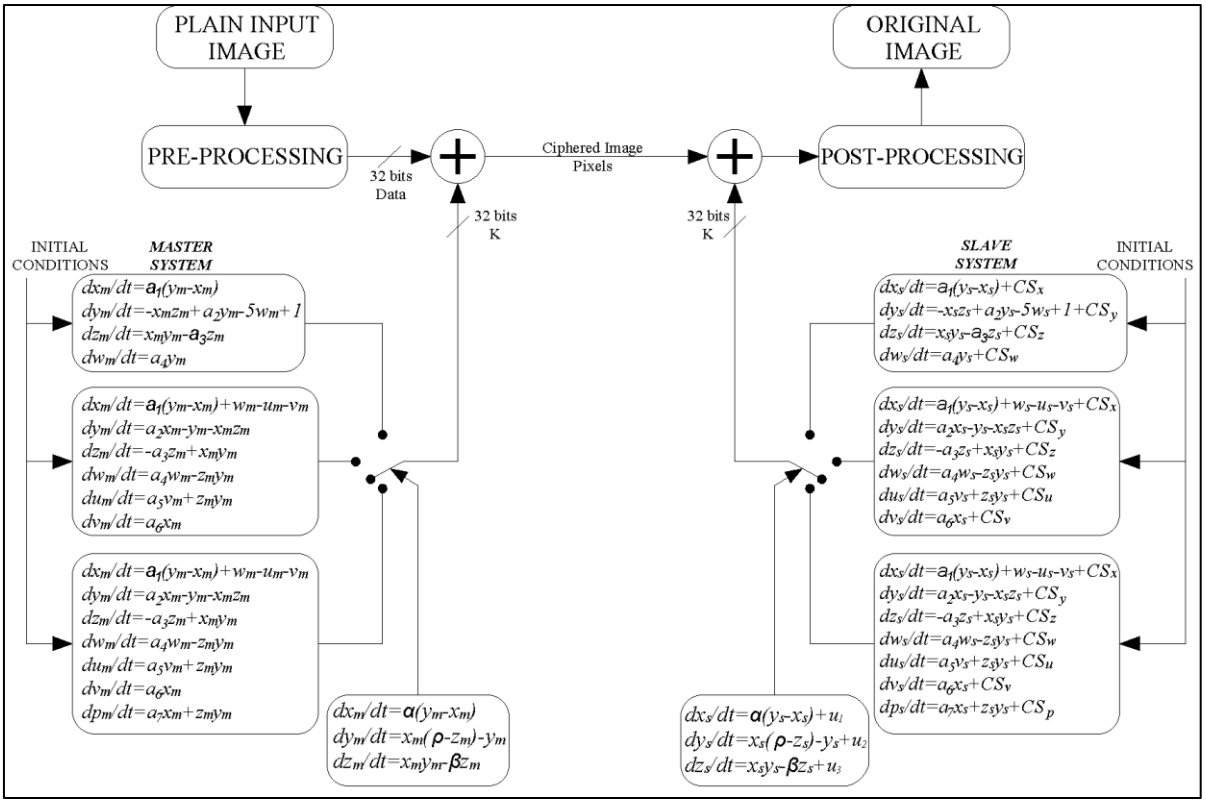

Figure 3-27 Image Encryption System Block Diagram Using Proposed Algorithm 3

3.4.1. Algorithm Mathematical Description

In this algorithm, there are four different nonlinear systems which adopted in the design. The first system is three-dimensional Lorenz chaotic oscillator which is employed as electronic high-speed switch. Where the system ODEs are solved numerically and the dynamical states are converted to binary stream and operate as a selector signal. The mathematical description for the chaos switch is presented in expression 3.14 shown below.

$$\begin{aligned} dx_m/dt &= \alpha(y_m - x_m) \\ dy_m/dt &= x_m(\rho - z_m) - y_m \\ dz_m/dt &= x_m y_m - \beta z_m \end{aligned} \tag{3.14}$$

The mathematical description of the rest nonlinear systems that are adopted in the design are presented in equations 3.15, 3.16, and 3.17 below. Where, equation 3.15 presents the four-dimensional hyperchaotic system.

$$\begin{aligned} \frac{dx_m}{dt} &= a_1(y_m - x_m) & \frac{dx_s}{dt} &= a_1(y_s - x_s) \\ \frac{dy_m}{dt} &= -x_m z_m + a_2 y_m - 5w_m + 1 & \frac{dy_s}{dt} &= -x_s z_s + a_2 y_s - 5w_s + 1 \\ \frac{dz_m}{dt} &= x_m y_m - a_3 z_m & \frac{dz_s}{dt} &= x_s y_s - a_3 z_s \\ \frac{dw_m}{dt} &= x_m y_m - a_4 z_m & \frac{dw_s}{dt} &= x_s y_s - a_4 z_s \end{aligned} \tag{3.15}$$

Six-dimensional hyperchaotic system is presented in expression 3.16. The last nonlinear system contains seven-dimensional ordinary differential equations that presented in expression 3.17. The subscripts m and s attached with ODEs dynamical states are denoted to master and slave systems respectively

$$\begin{aligned}
&\frac{dx_m}{dt} = a_1(y_m - x_m) + w_m - u_m - v_m && \frac{dx_s}{dt} = a_1(y_s - x_s) + w_s - u_s - v_s \\
&\frac{dy_m}{dt} = a_2 x_m - y_m - x_m z_m && \frac{dy_s}{dt} = a_2 x_s - y_s - x_s z_s \\
&\frac{dz_m}{dt} = -a_3 z_m + x_m y_m && \frac{dz_s}{dt} = -a_3 z_s + x_s y_s \\
&\frac{dw_m}{dt} = a_4 w_m - y_m z_m && \frac{dw_s}{dt} = a_4 w_s - y_s z_s \\
&\frac{du_m}{dt} = a_5 v_m + y_m z_m && \frac{du_s}{dt} = a_5 v_s + y_s z_s \\
&\frac{dv_m}{dt} = a_6 x_m && \frac{dv_s}{dt} = a_6 x_s
\end{aligned} \tag{3.16}$$

The parameters and the initial conditions of the systems are selected as shown in table 3-6, according to those values the systems exhibit a hyperchaotic behavior.

$$\begin{aligned}
&\frac{dx_m}{dt} = a_1(y_m - x_m) + w_m - u_m - v_m && \frac{dx_s}{dt} = a_1(y_s - x_s) + w_s - u_s - v_s \\
&\frac{dy_m}{dt} = a_2 x_m - y_m - x_m z_m && \frac{dy_s}{dt} = a_2 x_s - y_s - x_s z_s \\
&\frac{dz_m}{dt} = -a_3 z_m + x_m y_m && \frac{dz_s}{dt} = -a_3 z_s + x_s y_s \\
&\frac{dw_m}{dt} = a_4 w_m - y_m z_m && \frac{dw_s}{dt} = a_4 w_s - y_s z_s \\
&\frac{du_m}{dt} = a_5 v_m + y_m z_m && \frac{du_s}{dt} = a_5 v_s + y_s z_s \\
&\frac{dv_m}{dt} = a_6 x_m && \frac{dv_s}{dt} = a_6 x_s \\
&\frac{dp_m}{dt} = a_7 x_m + z_m y_m && \frac{dp_s}{dt} = a_7 x_s + z_s y_s
\end{aligned} \tag{3.17}$$

Table 3-6 Hyperchaotic Systems Parameters and Initial Conditions [82][83][51]

Hyperchaotic System	Initial Conditions At t = 0	Constants	Lyapunov Exponents	Sum of Lyapunov Exponents
4-Dimensional	x=0.1	a_1=30	L_1=2.1726	-12.91147
	y =0.1	a_2=20	L_2=0.01493	
	z =0.1	a_3=3	L_3=0	
	w =0.1	a_4=0.1	L_4=-15.099	
6-Dimensional	X =1	a_1=10	L_1=0.8	-14.733355
	y =1	a_2=28	L_2=0.3	
	z =1	a_3=8/3	L_3=0	
	w =1	a_4=-1	L_4=-0.5	
	u =1	a_5=8	L_5=-0.7	
	v =1	a_6=3	L_6=-14.7	
7-Dimensional	x=1	a_1=10	L_1=0.6	-14.667183
	y =1	a_2=28	L_2=0.2	
	z =1	a_3=8/3	L_3=0	
	w =1	a_4=-1	L_4=-0.14	
	u =1	a_5=8	L_5=-0.404591	
	v =1	a_6=5	L_6=-0.8	
	p =1	a_7=1	L_7=-14	

On the other hand, the lyapunov exponents are also calculated and presented in table 3-6. The sum of all lyapunov exponents for each system is -12.91147, -14.733355, -14.667183 respectively. Obviously, the sums are less than zero which proof that all of the dynamical systems are in a hyperchaotic state. As shown in the table, each system has two positive lyapunov exponent which proof that the systems are in hyperchaotic state. These systems will be implemented two times for transmitter and receiver systems.

3.4.2. Overall System Design Based XSG Model

The design and implementation of the four different nonlinear systems is performed in this section using Xilinx system generator XSG blocks which are configured with 32-bit fixed-point data representation (Fix32_18 data format) to build up the transmitter, receiver systems and adaptive synchronization system. The design of the overall system consists of five subsystems which are: chaos switching, adaptive synchronization, transmitter/receiver, encryption/ decryption process and preprocessing /postprocessing systems.

3.4.2.1. Transmitter/ Receiver Synchronization

The synchronization between master and slave nonlinear systems is presented in this section. Where, four separate adaptive feedback controllers are designed and implemented to provide the necessary synchronization. The mathematical description for the controlling signals is listed below (which are derived in a previous section in this chapter) in equations 3.18, 3.19, 3.20, and 3.21.

Three-dimensional chaotic system:

$$\begin{aligned} CS_x &= G \times e_x = G \times [\alpha(y_m - x_m) - \alpha(y_s - x_s)] \\ CS_y &= G \times e_y = G \times [x_m(\rho - z_m) - y_m - (x_s(\rho - z_s) - y_s)] \\ CS_z &= G \times e_z = G \times [x_m y_m - \beta z_m - (x_s y_s - \beta z_s)] \end{aligned} \tag{3.18}$$

Four-Dimensional Hyperchaotic System Controlling Signals:

$$
\begin{aligned}
CS_x &= G \times e_x = G \times \left(\frac{dx_m}{dt} - \frac{dx_s}{dt}\right) \\
&= G \times (a_1(y_m - x_m) - a_1(y_s - x_s)) \\
CS_y &= G \times e_y = G \times \left(\frac{dy_m}{dt} - \frac{dy_s}{dt}\right) \\
&= G \times ((-x_m z_m + a_2 y_m - 5w_m + 1) - (-x_s z_s + a_2 y_s - 5w_s + 1)\,) \\
CS_z &= G \times e_z = G \times \left(\frac{dz_m}{dt} - \frac{dz_s}{dt}\right) \\
&= G \times (x_m y_m - a_3 z_m - (x_s y_s - a_3 z_s)) \\
CS_w &= G \times e_w = G \times \left(\frac{dw_m}{dt} - \frac{dw_s}{dt}\right) \\
&= G \times (x_m y_m - a_4 z_m - (x_s y_s - \ a_4 z_s))
\end{aligned}
\tag{3.19}
$$

Six-Dimensional Hyperchaotic System Controlling Signals:

$$
\begin{aligned}
CS_x &= G \times e_x = \frac{dx_m}{dt} - \frac{dx_s}{dt} \\
&= G \times (a_1(y_m - x_m) + w_m - u_m - v_m - (a_1(y_s - x_s) + w_s - u_s - \\
&\quad v_s)) \\
CS_y &= G \times e_y = G \times \left(\frac{dy_m}{dt} - \frac{dy_s}{dt}\right) \\
&= G \times (a_2 x_m - y_m - x_m z_m - (a_2 x_s - y_s - x_s z_s)) \\
CS_z &= G \times e_z = G \times \left(\frac{dz_m}{dt} - \frac{dz_s}{dt}\right) \\
&= G \times (-a_3 z_m + x_m y_m - (-a_3 z_s + x_s y_s)) \\
CS_w &= G \times e_w = G \times \left(\frac{dw_m}{dt} - \frac{dw_s}{dt}\right) \\
&= G \times (a_4 w_m - y_m z_m - (a_4 w_s - y_s z_s)) \\
CS_{\mathrm{u}} &= G \times e_u = G \times \left(\frac{du_m}{dt} - \frac{du_s}{dt}\right) \\
&= G \times (\mathrm{a_5 v_m} + \mathrm{y_m z_m} - (a_5 v_s + y_s z_s)) \\
CS_{\mathrm{v}} &= G \times e_v = G \times \left(\frac{dv_m}{dt} - \frac{dv_s}{dt}\right) = G \times (a_6 x_m - a_6 x_s)
\end{aligned}
\tag{3.20}
$$

Seven-Dimensional Hyperchaotic System Controlling Signals:

$$
\begin{aligned}
CS_x &= G \times e_x = G \times \left(\frac{dx_m}{dt} - \frac{dx_s}{dt}\right) \\
&= G \times (a_1(y_m - x_m) + w_m - u_m - v_m - (a_1(y_s - x_s) + w_s - u_s - v_s)) \\
CS_y &= G \times e_y = G \times \left(\frac{dy_m}{dt} - \frac{dy_s}{dt}\right) \\
&= G \times (a_2 x_m - y_m - x_m z_m - (a_2 x_s - y_s - x_s z_s)) \\
CS_z &= G \times e_z = G \times \left(\frac{dz_m}{dt} - \frac{dz_s}{dt}\right) \\
&= G \times (-a_3 z_m + x_m y_m - (-a_3 z_s + x_s y_s)) \\
CS_w &= G \times e_w = G \times \left(\frac{dw_m}{dt} - \frac{dw_s}{dt}\right) \\
&= G \times (a_4 w_m - y_m z_m - (a_4 w_s - y_s z_s)) \\
CS_{\mathrm{u}} &= G \times e_u = G \times \left(\frac{du_m}{dt} - \frac{du_s}{dt}\right) \\
&= G \times (\mathrm{a_5 v_m} + \mathrm{y_m z_m} - (a_5 v_s + y_s z_s)) \\
CS_{\mathrm{v}} &= G \times e_v = G \times \left(\frac{dv_m}{dt} - \frac{dv_s}{dt}\right) = G \times (a_6 x_m - a_6 x_s) \\
CS_{\mathrm{p}} &= G \times e_p = G \times \left(\frac{dp_m}{dt} - \frac{dp_s}{dt}\right) \\
&= G \times (a_7 x_m + z_m y_m - (a_7 x_s + z_s y_s))
\end{aligned}
\tag{3.21}
$$

The XSG model implementation of the adaptive feedback controllers are presented in figure 3-28 for the four implemented nonlinear systems.

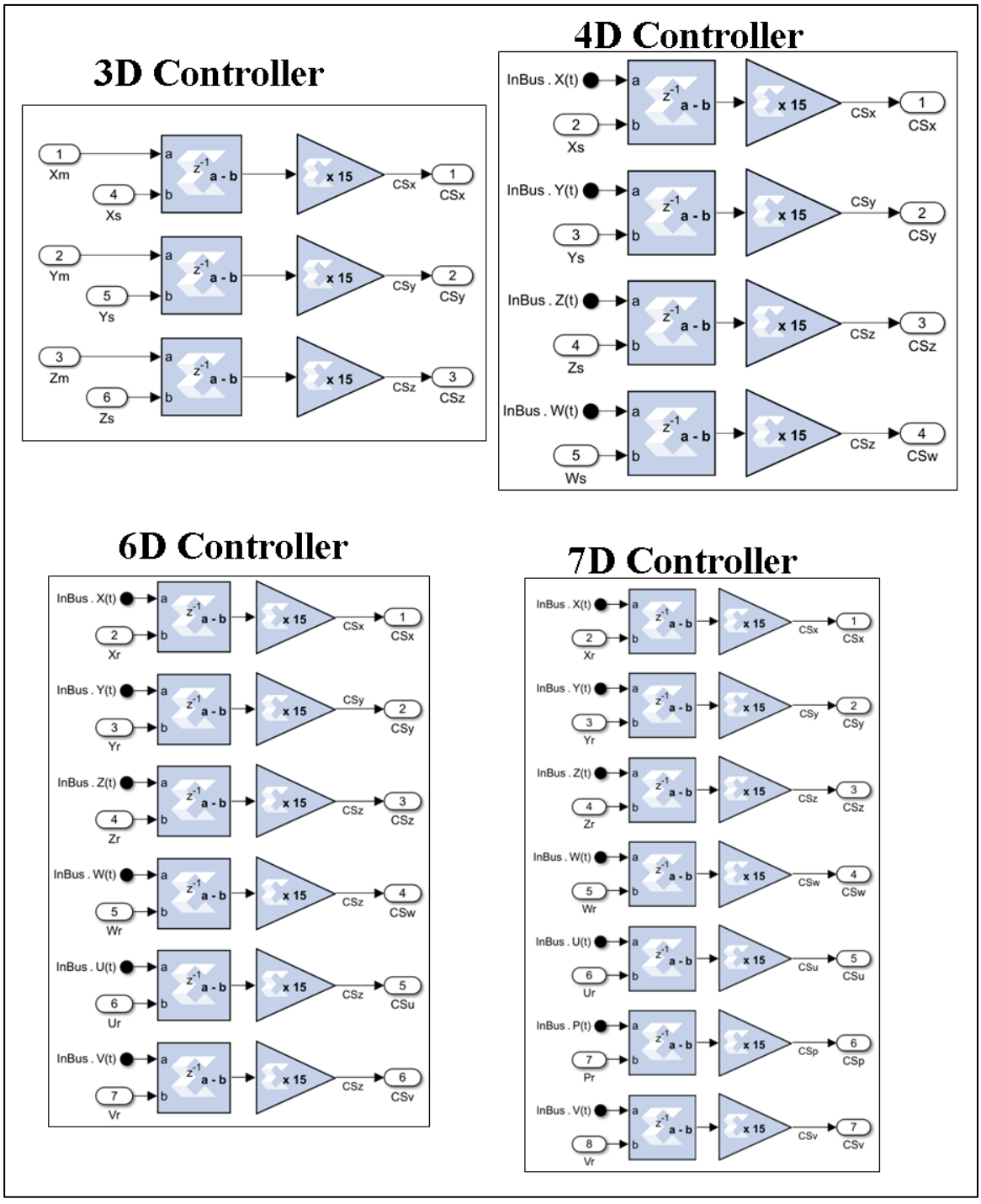

Figure 3-28 Adaptive Feedback Controllers for Nonlinear Systems (Algorithm 3)

3.4.2.2. Encryption/Decryption Process Via XOR Operation

Stream cipher encryption technique is also adopted in this proposed system. The input plain image is converted into serial pixels and then to serial bits (via preprocessing system) and at the same time, the x component of the dynamical chaotic system is also converted into serial bits after solving the overall nonlinear system numerically. The generated bit streams are considered as the plaintext data and secret key, those bit streams are XOR-ed together to produce the encrypted images.

3.4.2.3. Preprocessing and Postprocessing Systems

Figure 3-29, illustrates the preprocessing subsystem, where it consists of Matlab/Simulink blocks: Matrix Transpose, Reshape, to frame, and Unbuffer. The combination of these blocks used to convert the image matrix into serial samples each sample contain 8 bits, as a prior stage for encryption. Then these parallel bits (samples) are converted into serial bit stream by using the parallel to serial conversion, in order to encrypt theme using XOR operation with x-dynamic of the chaotic system.

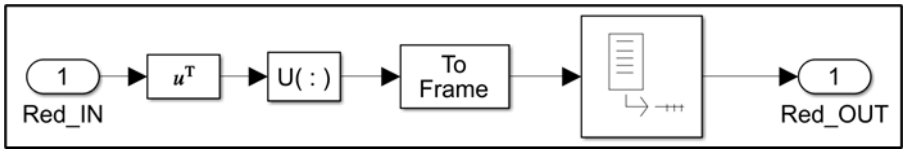

Figure 3-29 Preprocessing Matlab Blocks

On the other hand, Figure 3-30 shows the block combination that construct the post-processing subsystem that operate in reverse mode to the preprocessing subsystem, where the serial bits are combined together to form serial samples each one of 8 bits, by a means of serial to parallel conversion. These serial samples are then combined together to construct the image matrix again. The post-processing subsystem is consisted of Buffer, Reshape, Matrix Transpose, and Unit8 Matlab/Simulink blocks.

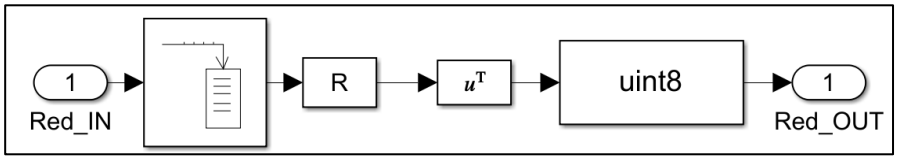

Figure 3-30 Postprocessing Matlab Blocks

3.4.2.4. Transmitter/Receiver Systems Design

The proposed multidimensional cascaded hyperchaotic system oscillator based on chaos switching technique is presented in this communication system. Figure 3-31 shown below depicts the main concept of this cryptosystem, where three different hyperchaotic system are solved numerically and their output bit stream is selected for encryption purposes in cascaded manner by a means of chaos switching technique and three-way selector switch. The final output of the selector switch represents the random bit stream that will be used to encrypt the input plain images. Figure 3-32 presents the transmitter and receiver XSG model with feedback controller signals, while figure 3-33 illustrates the overall proposed communications system including master (transmitter), slave (receiver), feedback controller, chaos switching wit selector unit, preprocessing/postprocessing systems and XOR operations.

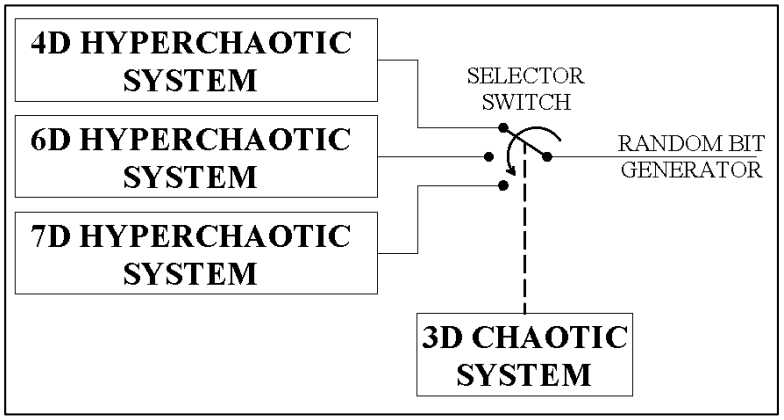

Figure 3-31 Block Diagram of Transmitter/ Receiver in Proposed Algorithm 3

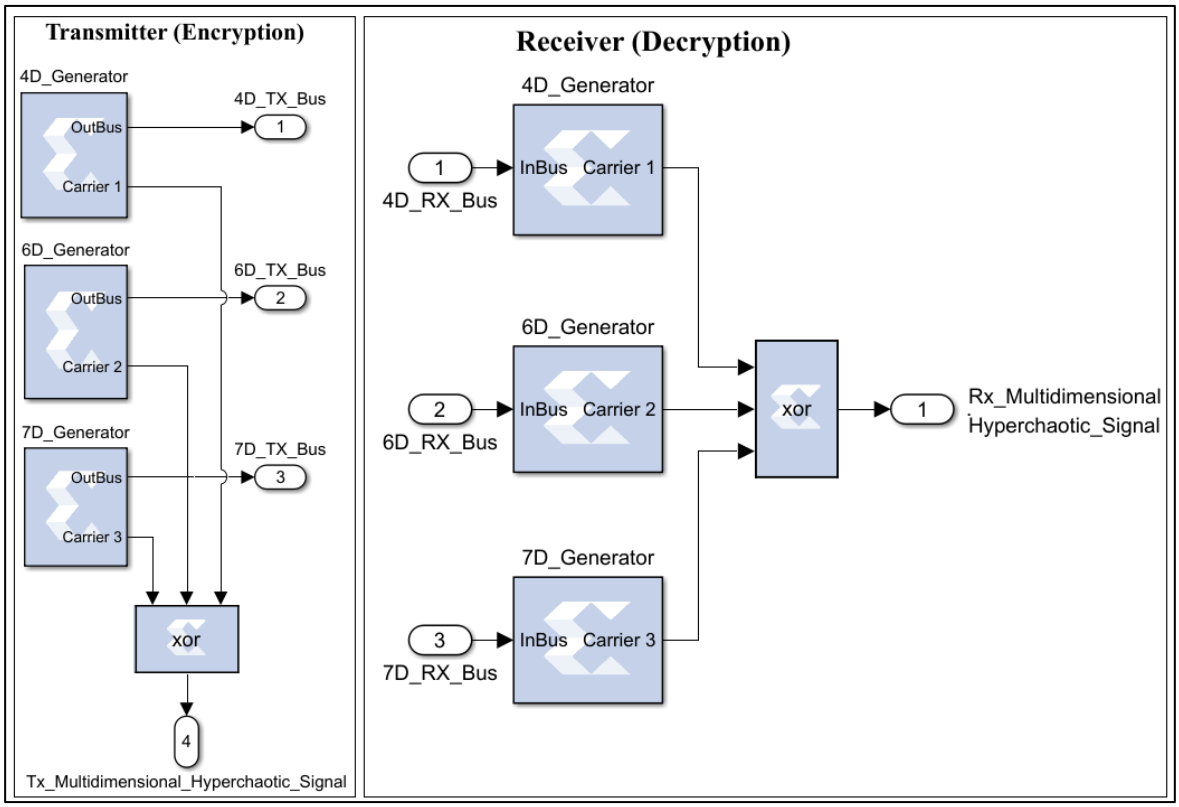

Figure 3-32 32 bits Fixed Point Representation of System Generator for Transmitter and Receiver Systems (Algorithm 3)

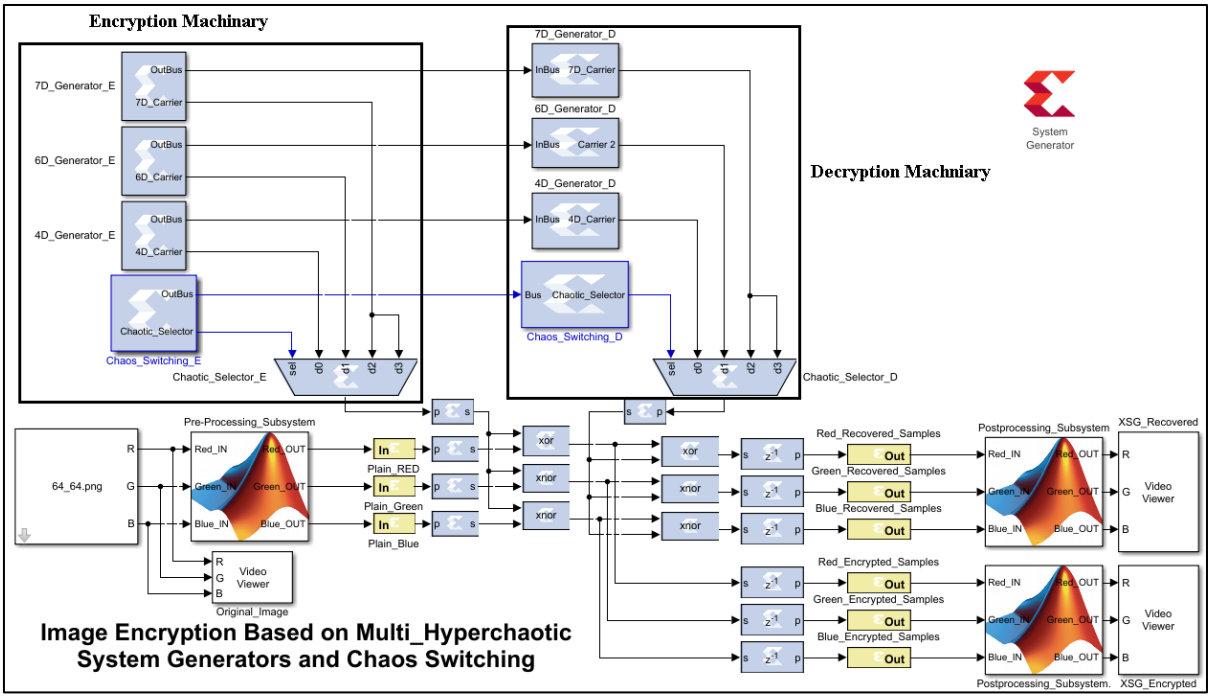

Figure 3-33 Overall Master/ Receiver System (Algorithm 3)

3.4.3. FPGA Co-simulations and Implementation

The proposed multi-dimensional hyperchaotic system has been implemented with FPGA PYNQ-Z1 zynq xc7z020 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board. Figure 3-34, depicts implementation of the proposed cryptographic algorithm using the FPGA board. JTAG link has been utilized through this design and adopted for the communication between the PC and the board. The plain image is called from its location in the PC and sent to the board serially to encrypt them. After completing the encryption process the encrypted samples are sent back to the PC to display the ciphered image. On the other hand, in the receiver side, the same steps are carried out to recover the plain image, where the ciphered image is received and converted into serial bits (using the preprocessing system) concurrently the serial bits are generated from the proposed hyperchaotic system. Finally performing the XOR operation to produce the original plain bits.

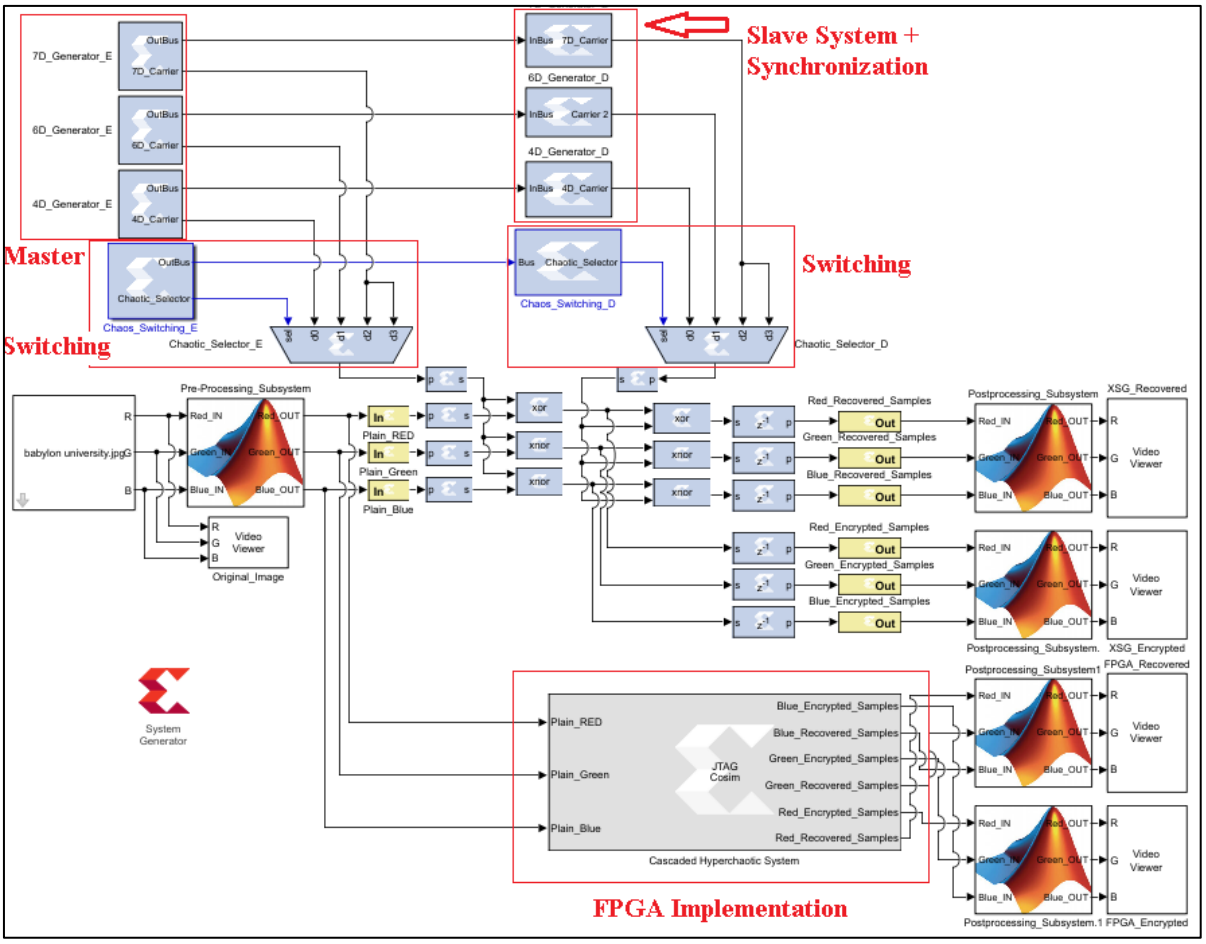

Figure 3-34 Hardware Co-simulation of the Proposed Cryptographic System (Algorithm 3)

Device utilization summary is presented in table 3-7 below for this proposed system, where the available and utilized resources based on this design is cleared. The total on chip power consumption is 0.436 watt with junction temperature of 30 C.

Table 3-7 Board Utilization in Algorithm 3

Resource	Utilization	Available	Utilization %
Look Up Table **(LUT)**	9174	53200	17.24
Look Up Table RAM **(LUTRAM)**	1	17400	0.01
Flip Flops **(FF)**	2514	106400	2.36
Block RAM **(BRAM)**	2	140	1.43
Digital Signal Processing **(DSP)**	204	220	92.73
Input Output **(IO)**	1	125	0.80
Global Buffer **(BUFG)**	4	32	12.50
Mixed Mode Clock Manager **(MMCM)**	1	4	25.00

3.5. Algorithm (4): Robust Encryption System Based on Novel Hyperchaotic Flow System.

Many hyperchaotic chaotic systems are proposed for the use in encryption algorithms like Rössler and Lorenz hyperchaotic systems. In this proposed algorithm a novel five dimensional hyperchaotic flow system is invented with maximum number of positive lyapunov exponents. The block diagram in figure 3-35 shows the suggested encryption algorithm.

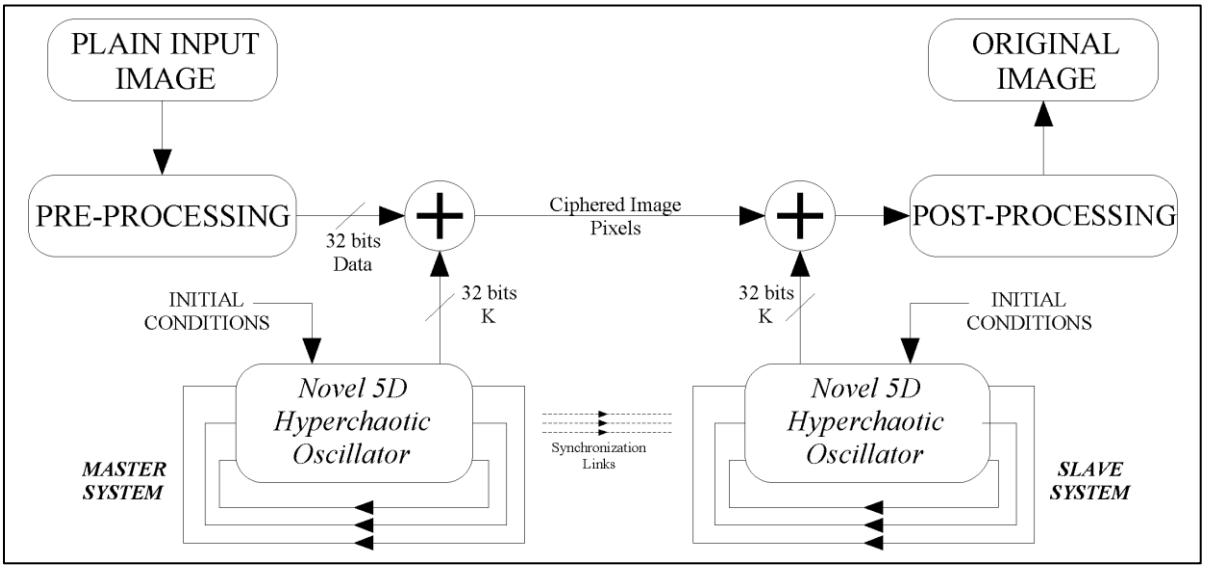

Figure 3-35 Image Encryption System Block Diagram `Using Proposed Algorithm 4

3.5.1. Algorithm Mathematical Description

The mathematical description of the proposed novel hyperchaotic system with maximum number of positive lyapunov exponents is presented in this section. Where a five-dimensional hyperchaotic system based on ordinary Lorenz chaotic system is presented. Two feedback dynamical states are fed to the ordinary three-dimensional Lorenz chaotic system as well as the system parameters are modified to accept a sine wave generator as a parameter for the new hyperchaotic system. The mathematical description of the new hyperchaotic system is presented in equation 3.22.

$$\frac{dx}{dt} = \alpha(y - x) + \beta p + w$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z \qquad (3.22)$$
$$\frac{dw}{dt} = -kx + Ly + zy$$
$$\frac{dp}{dt} = y^2 - z$$

Where, the system parameters are α=10, ρ=30, k=2, L=6.7, and β=sin(t). Initial conditions are selected to be 10, 10, 10, 40, and 50 for the dynamical responses x, y, z, w, and p respectively. The above proposed mathematical system meets the necessary requirements for designing a new hyperchaotic system which are:

- Mathematical system has at least four ordinary differential equations and in this system, there are five ODEs.
- Two nonlinear terms should be existed where in this system, there are four nonlinear terms (xz, xy, yz, y^2).
- Two or more positive lyapunov exponents should be generated from the proposed mathematical system and in this system, there are three positive exponents.

The nonlinear system presented in 3.22 will be used to design a transmitter and receiver systems for secure communication purposes. The mathematical description of the master and slave are depicted in equations 3.23 and 3.24. The m and s subscripts are referred to master and slave respectively.

$$\frac{dx_m}{dt} = \alpha(y_m - x_m) + \beta P_m + w_m \qquad \frac{dx_s}{dt} = \alpha(y_s - x_s) + \beta P_s + w_s$$

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

$$\frac{dz_m}{dt} = x_m y_m - \beta z_m \quad (3.33) \qquad \frac{dz_s}{dt} = x_s y_s - \beta z_s \quad (3.44)$$

$$\frac{dw_m}{dt} = -kx_m + Ly_m + z_m y_m \qquad \frac{dw_s}{dt} = -kx_s + Ly_s + z_s y_s$$

$$\frac{dp_m}{dt} = {y_m}^2 - z_m \qquad \frac{dp_s}{dt} = {y_s}^2 - z_s$$

3.5.2. Overall System Design Based XSG Model

This section designs and implements the transmitter, receiver, and adaptive synchronization systems of a novel five-dimensional hyperchaotic system using Xilinx system generator XSG blocks with 32-bit fixed-point data representation. The 32-bit fixed point XSG model is created using the Fix32 18 data format, where 32 denotes the total number of bits divided as follows: The sign bit is one bit, there are 18 fractional bits, and there are 13 integer bits. The four subsystems that make up the overall system design are adaptive synchronization, transmitter/receiver, encryption/decryption process, and preprocessing/postprocessing systems.

3.5.2.1. Transmitter/ Receiver Synchronization

The synchronization between master and slave nonlinear systems is presented in this section. Where, the same situation that performed in the previous algorithms will be applied in this cryptosystem. An adaptive feedback controller is designed and implemented to provide the necessary synchronization between the five-dimensional master (transmitter) and slave (receiver). The mathematical description for the controlling signals is presented equation 3.25.

$$CS_x = G \times e_x = G \times \left(\frac{dx_m}{dt} - \frac{dx_s}{dt}\right) = G \times (\alpha(y_m - x_m) + \beta P_m + w_m - (\alpha(y_s - x_s) + \beta P_s + w_s))$$

$$CS_y = G \times e_y = G \times \left(\frac{dy_m}{dt} - \frac{dy_s}{dt}\right) = G \times (x_m(\rho - z_m) - y_m - (x_s(\rho - z_s) - y_s))$$

$$CS_z = G \times e_z = G \times \left(\frac{dz_m}{dt} - \frac{dz_s}{dt}\right) = G \times (x_m y_m - \beta z_m - (x_s y_s - \beta z_s)) \quad (3.25)$$

$$CS_w = G \times e_w = G \times \left(\frac{dw_m}{dt} - \frac{dw_s}{dt}\right) = G \times (-kx_m + Ly_m + z_m y_m - (-kx_s + Ly_s + z_s y_s))$$

$$CS_p = G \times e_p = G \times \left(\frac{dp_m}{dt} - \frac{dp_s}{dt}\right) = G \times ({y_m}^2 - z_m - ({y_s}^2 - z_s))$$

The XSG model implementation of the adaptive feedback controller is presented in figure 3-36.

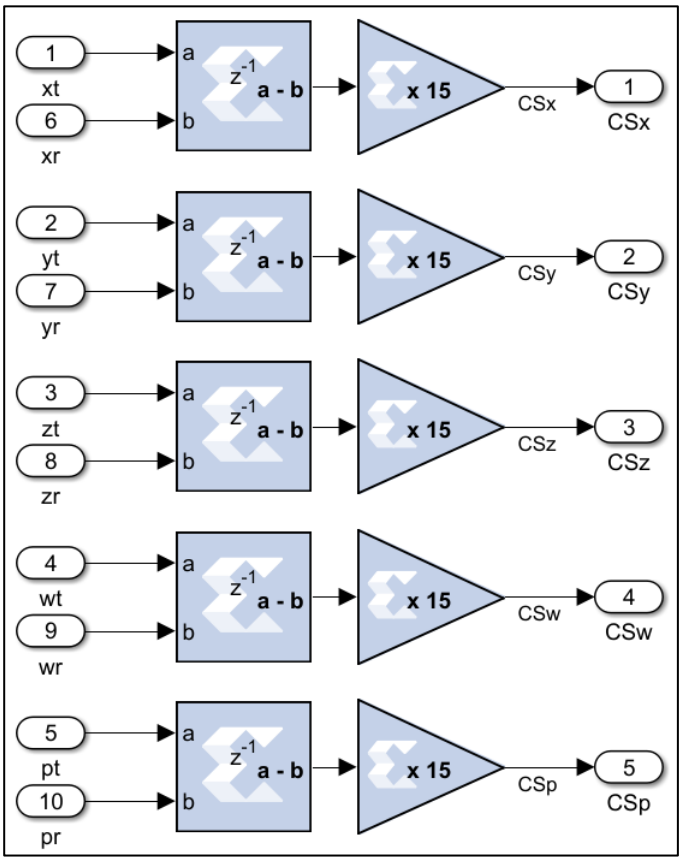

Figure 3-36 Adaptive Feedback Controller for Novel Five Dimensional Nonlinear Hyperchaotic System (Algorithm 4)

3.5.2.2. Encryption/Decryption Process Via XOR operation

Stream cipher encryption technique is the adopted type for encryption/ decryption in this work, where the input plain image is converted into serial pixels and then to serial bits (via preprocessing system) and in the same time the x component of the dynamical chaotic system is also converted into serial bits after solving the overall nonlinear system numerically. The generated bit streams are considered as the plaintext data and secret key, those bit streams are XOR-ed together to generate the encrypted images.

3.5.2.3. Preprocessing and Postprocessing Systems

Figure 3-37, illustrates the preprocessing subsystem, where it consists of Matlab/Simulink blocks: Matrix Transpose, Reshape, to frame, and Unbuffer. The combination of these blocks used to convert the image matrix into serial samples each sample contain 8 bits, as a prior stage for encryption. Then these parallel bits (samples) are converted into serial bit stream by using the parallel to serial conversion, in order to encrypt theme using XOR operation with x-dynamic of the chaotic system.

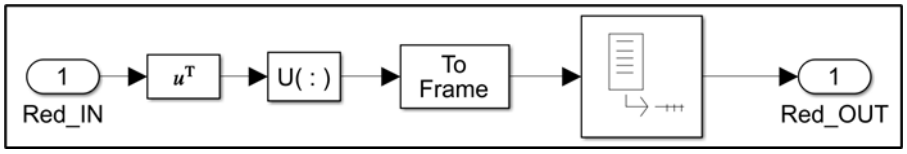

Figure 3-37 Preprocessing Matlab Blocks

On the other hand, Figure 3-38 shows the block combination that construct the post-processing subsystem that operate in reverse mode to the preprocessing subsystem, where the serial bits are combined together to form serial samples each one of 8 bits, by a means of serial to parallel conversion. These serial samples are then combined together to construct the image matrix again. The post-processing subsystem is consisted of Buffer, Reshape, Matrix Transpose, and Unit8 Matlab/Simulink blocks.

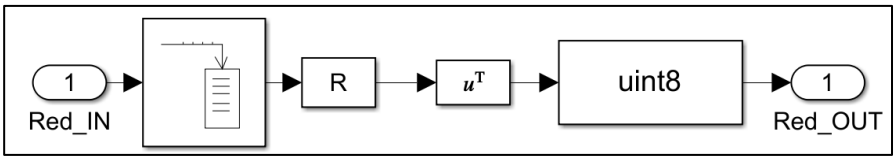

Figure 3-38 Postprocessing Matlab Blocks

3.5.2.4. Transmitter/ Receiver System Design

The proposed cryptography system based on novel five dimensional hyperchaotic system is presented. The equations that describe the system design is presented in 3.23, 3.24, and 3.25. The five-dimensional hyperchaotic system is solved numerically (using Forward Euler method) its x output is converted into binary bit stream that will be used to encrypt the input plain images. Figure 3-39 presents the transmitter and receiver XSG model with feedback controller signals, while on the other hand figure 3-40 depicts the overall proposed communications system including master (transmitter), slave (receiver), feedback controller, chaos switching with selector unit, preprocessing/postprocessing systems and XOR operations.

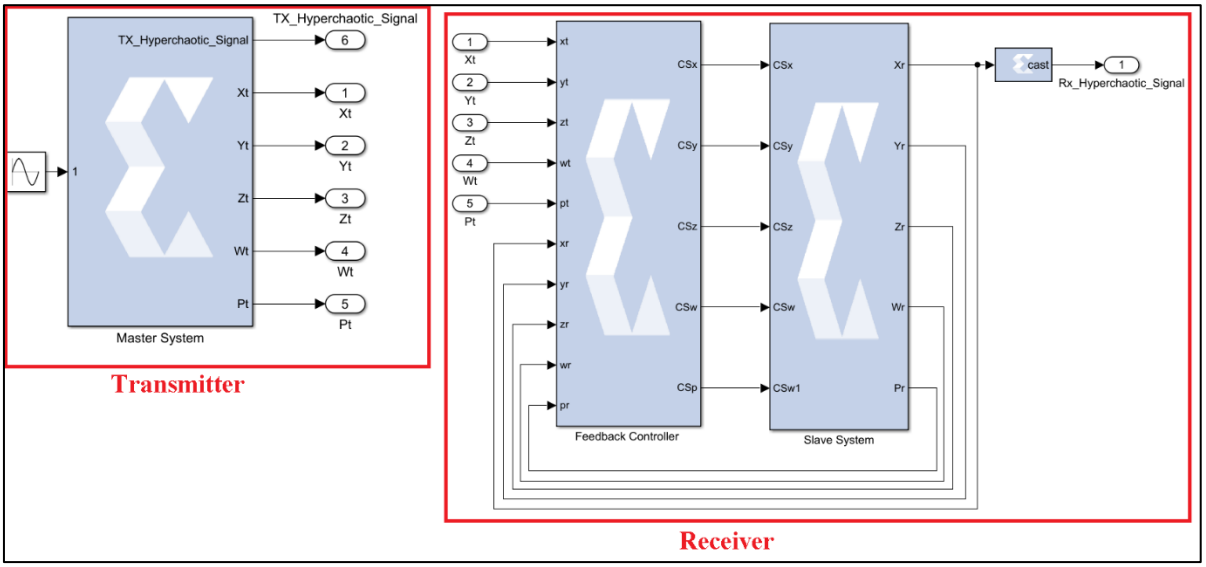

Figure 3-39 32 bits Fixed Point Representation of System Generator for Transmitter and Receiver Systems (Algorithm 4)

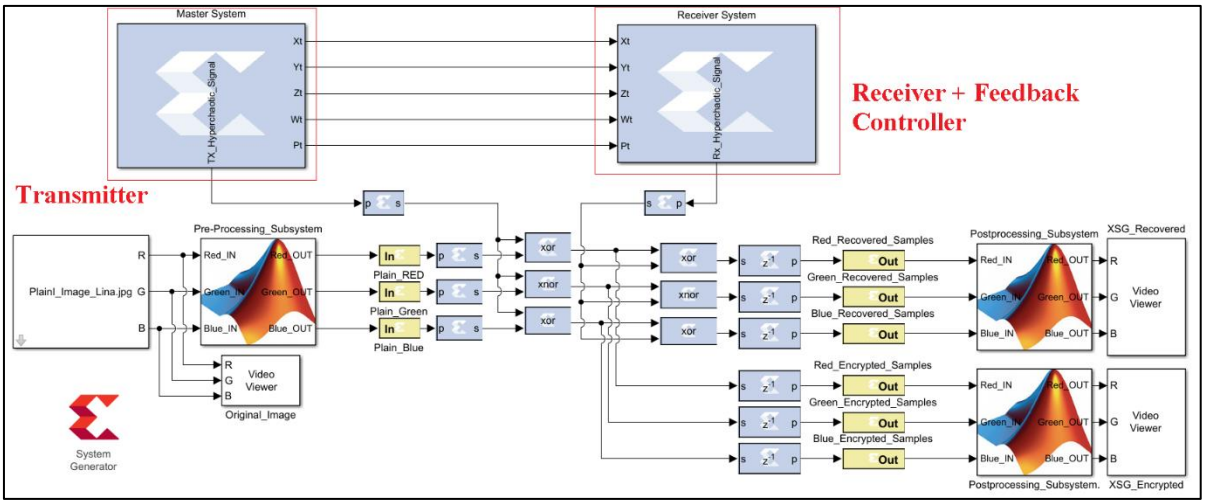

Figure 3-40 Overall Master/ Receiver System (Algorithm 4)

3.5.3. FPGA Co-simulation and Implementation

The suggested innovative five-dimensional hyperchaotic system was constructed utilizing the Xilinx System Generator XSG and FPGA PYNQ-Z1 zynq xc7z020 evolution board. The VHDL codes needed to configure and program the board are obtained via the XSG. The proposed cryptographic algorithm was illustrated using the FPGA board in Figure 3-41. In this design, JTAG link has been chosen for communication between the PC and the board. The co-simulation feature of the Xilinx technology is used to implement the entire system (Master, Slave, and synchronization unit) inside the FPGA board.

The board serially sends the encrypted samples back to the PC to show the ciphered image after the plain image is summoned from its place in the PC and transferred there to be encrypted. On the other hand, the receiver side follows the identical procedures to recover the plain image, where the ciphered image is received and transformed into serial bits at the same time the serial bits are generated from the suggested hyperchaotic system (using the preprocessing system). XORing the bits back together to their original state is the final step.

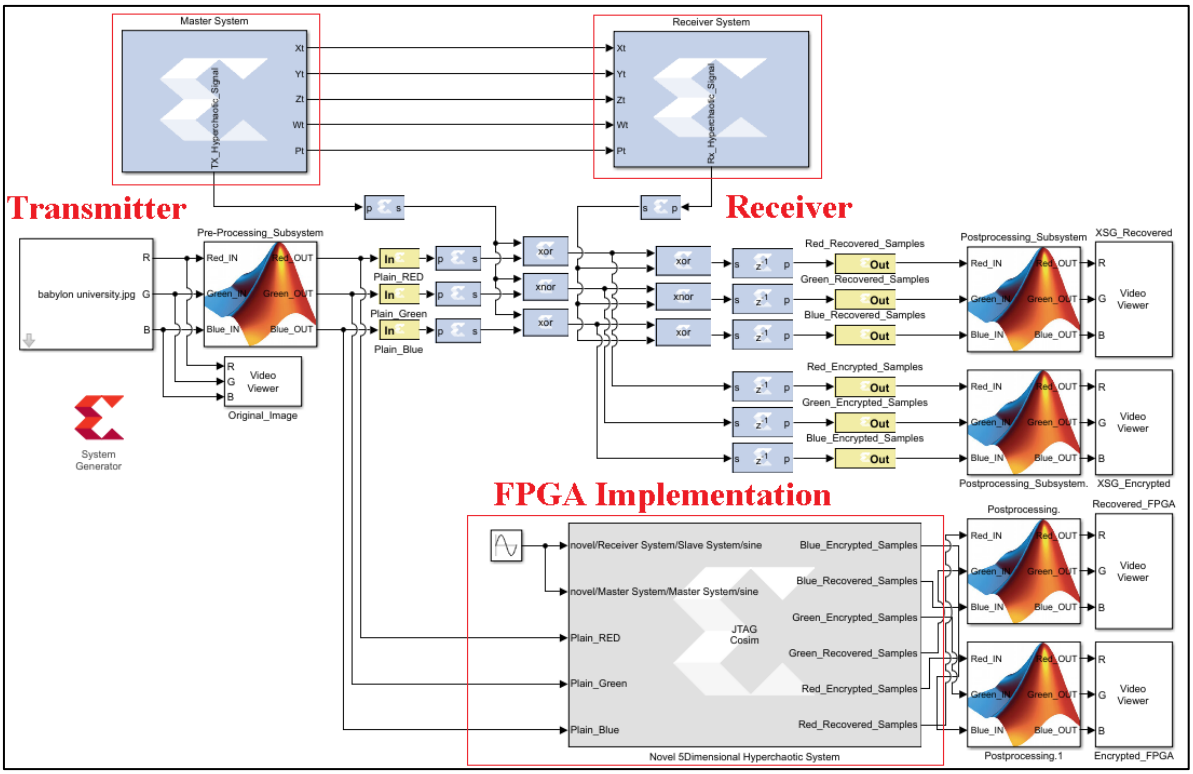

Figure 3-41 Hardware Co-simulation of the Proposed Cryptographic System (Algorithm 4)

For this suggested system, a summary of device use is shown in tables 3-8 below. Based on this architecture, it is clear which resources are available and which ones are used. With a junction temperature of 28.4 C, the whole chip power consumption is 0.298 watts.

Table 3-8 Board Utilization in Algorithm 4

Resource	Utilization	Available	Utilization %
Look Up Table **(LUT)**	2850	53200	5.36
Look Up Table RAM **(LUTRAM)**	1	17400	0.01
Flip Flops **(FF)**	1598	106400	1.50
Block RAM **(BRAM)**	2	140	1.43
Digital Signal Processing **(DSP)**	96	220	43.64
Input Output **(IO)**	1	125	0.80
Global Buffer **(BUFG)**	4	32	12.50
Mixed Mode Clock Manager **(MMCM)**	1	4	25.00

3.6. Algorithm (5): New Hyperchaotic Sequence Based on the Combination of the 2nd, 3rd, and 4th, Pre-Designed Algorithms.

The combination of the four previous designed algorithms is collected in one robust, high immunity against cyber-attacks, unpredictable, and very complex cryptosystem. Figure 3-42 depicts the block diagram of the proposed generator for random bit stream generation.

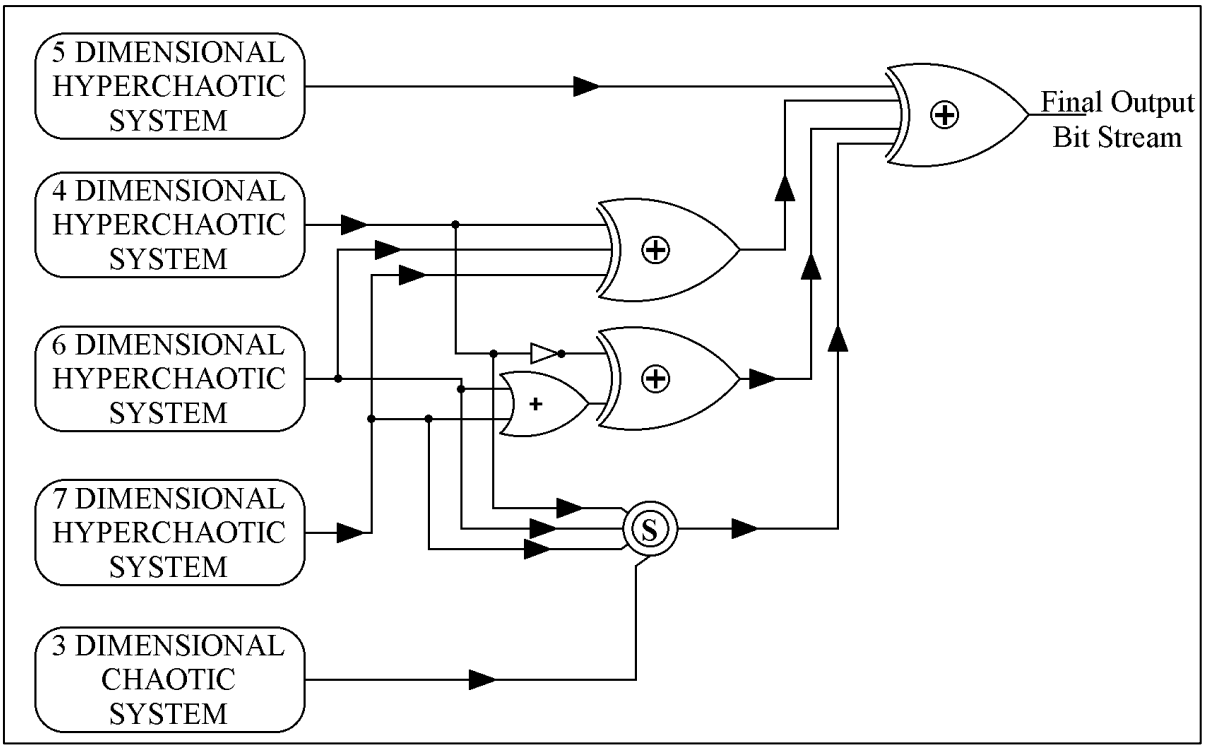

Figure 3-42 Block Diagram of the Proposed Algorithm 5 for Random Bit Stream Generation

3.6.1. Transmitter and Receiver Systems Design

The design and implementation of a new cryptosystem based on different hyperchaotic systems presented in this section using Xilinx system generator XSG blocks which are configured with 32-bit fixed-point data representation to build up the transmitter, receiver systems and adaptive synchronization system. The design of the overall system contains four subsystems which are: adaptive synchronization, transmitter/ receiver, encryption/decryption process and preprocessing /postprocessing systems.

The transmitter consists of four different dimensional hyperchaotic nonlinear systems (4D, 5D, 6D, 7D) as depicted in figure 3-43 that shown below. The mathematical description of these nonlinear systems is introduced in the previous four algorithms. The Lorenz chaotic system is used as a chaos switching for the branch three as shown in figure 3-43 below and does not contribute in the process of random bit stream generation. Different logical operations have been carried out to ensure the randomness of the output bit stream as well as to ensure that the proposed system can cope with differential and cyber-attacks.

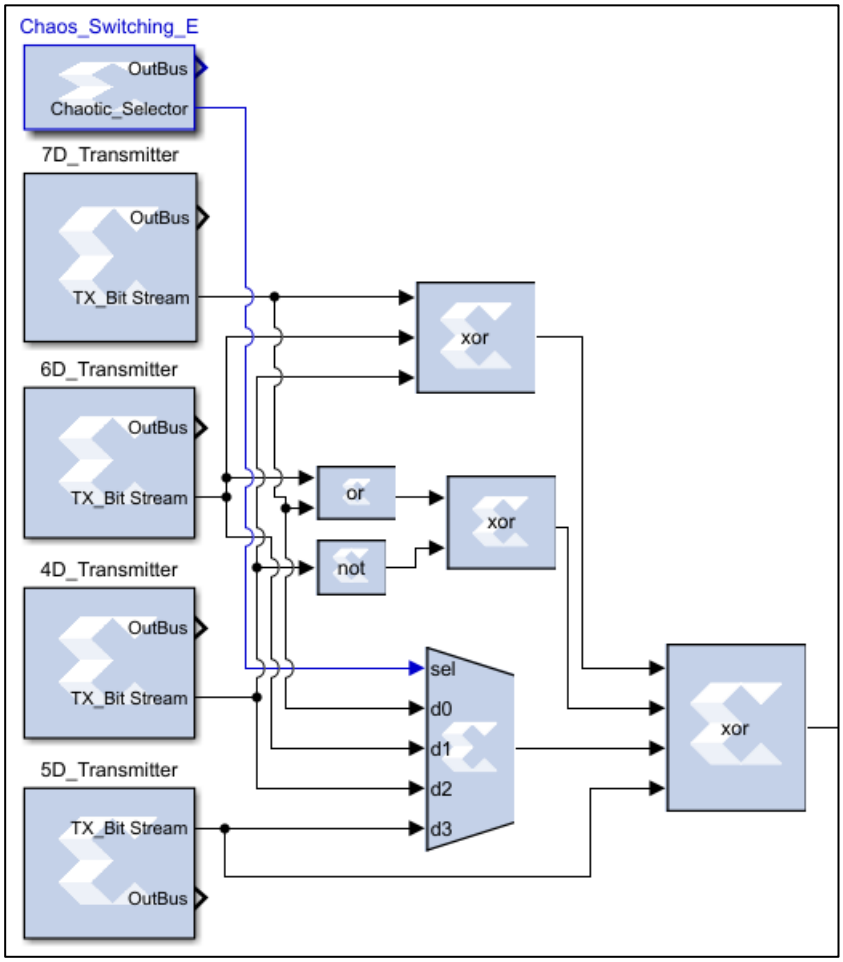

Figure 3-43 32 bits Fixed Point Representation of System Generator for Transmitter Systems (Algorithm 5)

in contrast the receiver system uses the same nonlinear systems with feedback controller to provide the necessary synchronization between the communicated nodes. Figure 3-44 presents the XSG model of the receiver system. While on the other hand figure 3-45 depicts the overall proposed communications system including master (transmitter), slave (receiver), feedback controller, preprocessing/postprocessing systems and XOR operations.

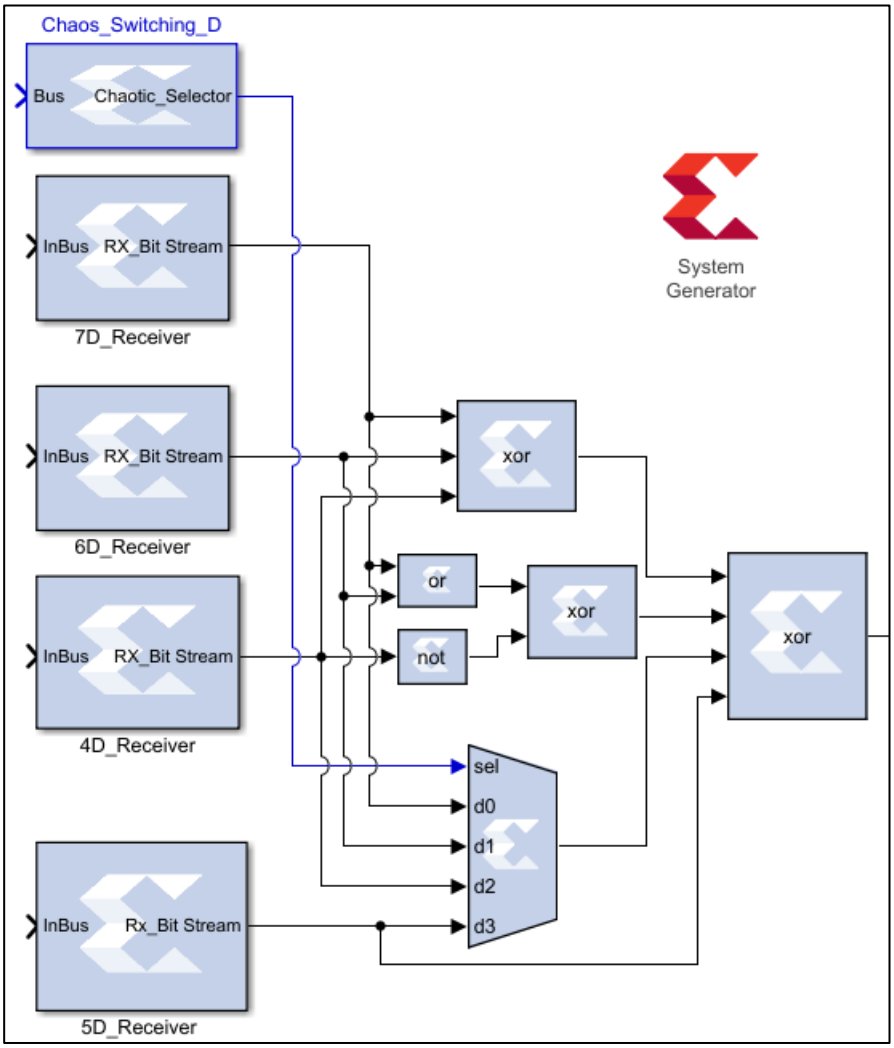

Figure 3-44 32 bits Fixed Point Representation of System Generator for Receiver Systems (Algorithm 5)

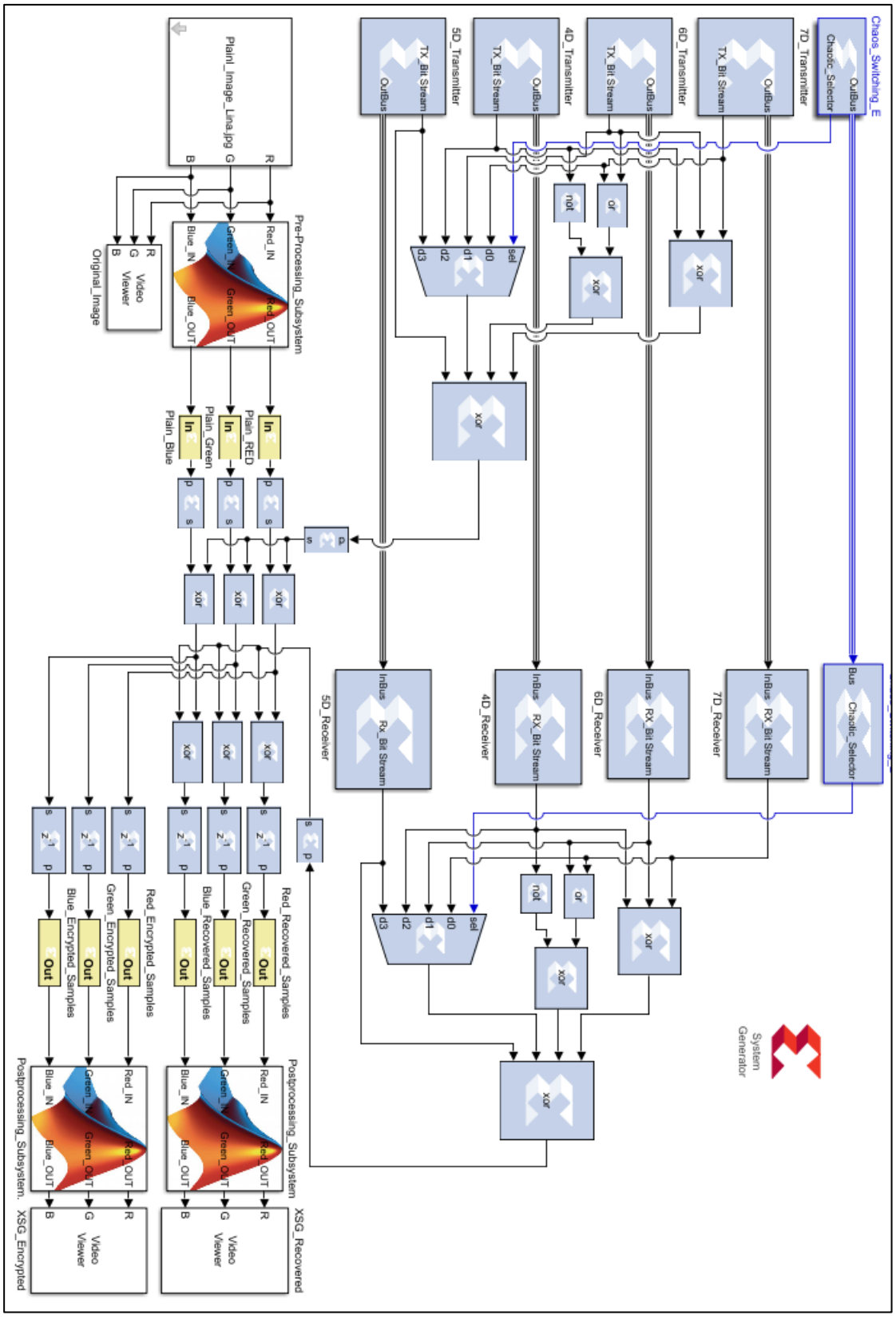

Figure 3-45 Overall Master/ Receiver System (Algorithm 5)

3.6.2. FPGA Co-simulation and Implementation

The proposed overall hyperchaotic system has been implemented with FPGA PYNQ-Z1 zynq xc7z020 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board. Due to the limitation of the board resources, the transmitter and receiver systems are implemented within the FPGA board separately. Figure 3-46 depicts implementation of the proposed transmitter using the FPGA board. JTAG link has been utilized through this design and adopted for the communication between the PC and the board. The overall system (Master, Slave and synchronization unit) is implemented inside the FPGA board using the co-simulation property of the Xilinx technology.

The plain image is called from its location in the PC and sent to the board serially to encrypt them. After completing the encryption process the encrypted samples are sent back to the PC to display the ciphered image. In contrast, in the receiver side, the same steps are carried out to recover the plain image, where the ciphered image is received and converted into serial bits (using the preprocessing system) concurrently the serial bits are generated from the proposed hyperchaotic system. Finally performing the XOR operation to regenerate the original plain bits.

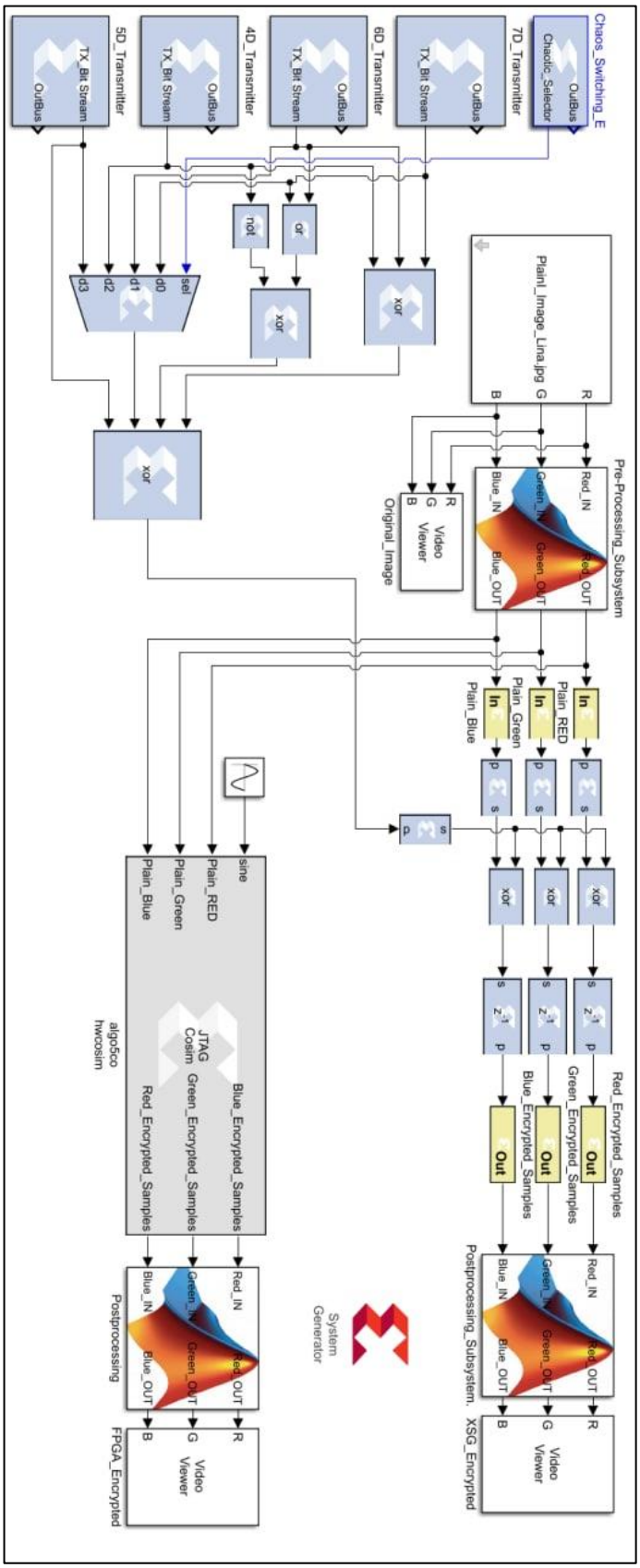

Figure 3-46 Hardware Co-simulation of the Proposed Cryptographic System (Algorithm 5)

Device utilization summary is presented in table 3-9 below for the transmitter system only, where the available and utilized resources based on this design is cleared. The total on chip power consumption is 0.343 Watt with junction temperature of 29 C.

Table 3-9 Board Utilization in Algorithm 5

Resource	Utilization	Available	Utilization %
Look Up Table **(LUT)**	5411	53200	10.17
Look Up Table RAM **(LUTRAM)**	1	17400	0.01
Flip Flops **(FF)**	2015	106400	1.89
Block RAM **(BRAM)**	2	140	1.43
Digital Signal Processing **(DSP)**	126	220	57.27
Input Output **(IO)**	1	125	0.80
Global Buffer **(BUFG)**	4	32	12.50
Mixed Mode Clock Manager **(MMCM)**	1	4	25.00

CHAPTER FOUR

Experimental Results, Statistical Analysis and Discussion

Chapter Four: Experimental Results, Statistical Analysis and Discussion

4.1. Introduction

The simulation and implementation results as well as the time response of the proposed algorithms that are explained in the previous chapter (chapter three) are presented in this chapter with the same order in chapter three. Each of the proposed algorithms is implemented and tested via testing techniques presented in chapter two. Different images are used with different dimensions to deeply evaluate the performance of the proposed algorithms. Some of the images are squared (have equal dimensions), and some of them are not, the image sizes 64×64, 128×128, 176×144, 256×256, and 2500×1875 are used in the proposed algorithms implementation through this chapter.

4.2. Algorithm (1): Secure Communication System Based on Three Dimensional Lorenz Chaotic Attractor

Dynamical time response (x, y, z) of the transmitter and receiver, image encryption/decryption results, encryption strength measurements, as well as the statistical analysis and FPGA implementation and board utilization summary for proposed algorithm 1 is presented in this section.

4.2.1. System Dynamical Response

The time response of system state variables including the x, y and z components is presented in figure 4-1 using forward Euler method and Runge-Kutta method. The two solutions are completely identical (so only one graph has been added). Figure 4-2 presents a X dynamical response comparison between the transmitter and receiver systems, where it is clear that the two-time responses are completely identical which proofs that the synchronization between the communicated nodes is achieved and the two systems can be used for the purpose of data encryption purposes over the public channels.

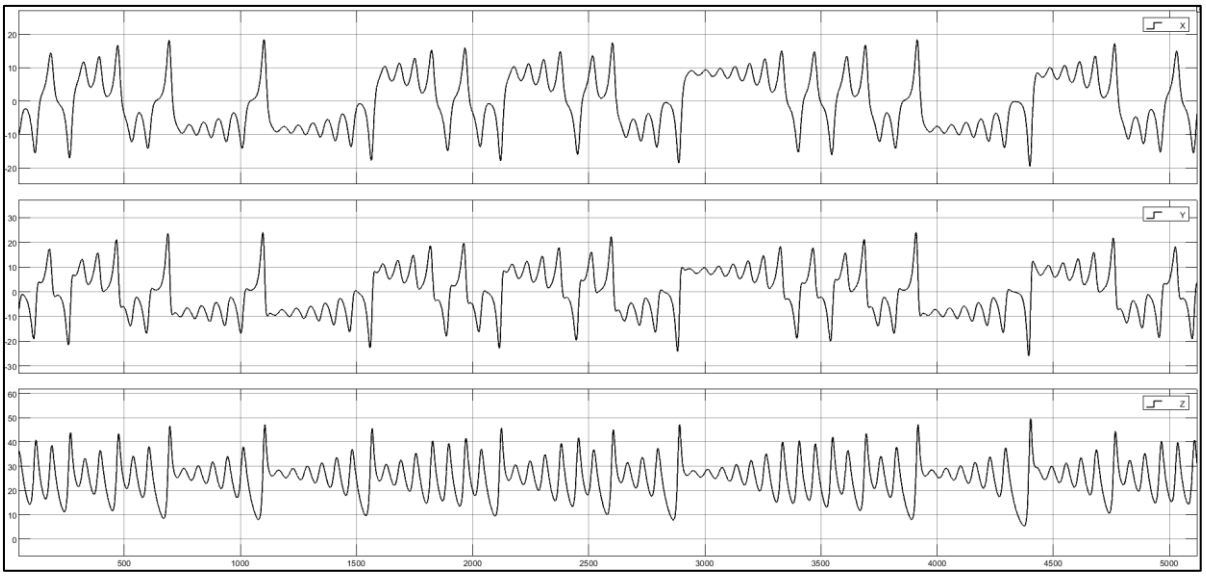

Figure 4-1 State Variables (x, y, z) Dynamical Response of Transmitter/Receiver System (with Forward Euler and Runge-Kutta Methods)

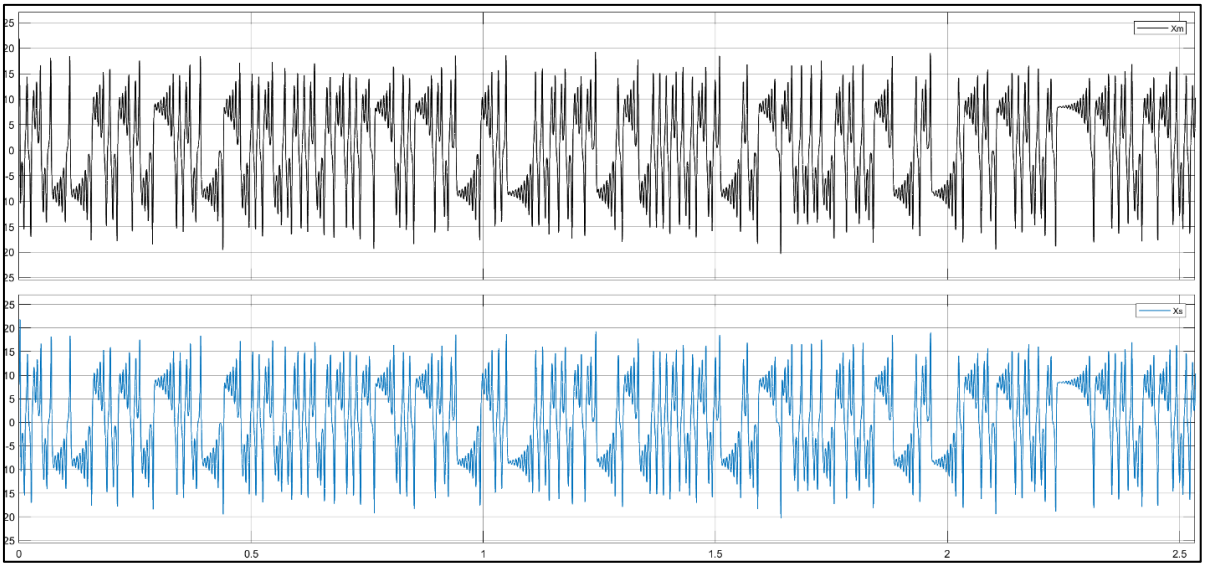

Figure 4-2 X-Dynamics of Transmitter and Receiver (with Forward Euler and Runge-Kutta Methods)

The three-dimensional strange attractor of the proposed system is presented in figure 4-3, using both Forward Euler and Runge-Kutta integration methods. The strange attractor graphs show identical solution for the proposed system which proof that the two numerical integration methods are both suitable for solving chaotic/hyperchaotic ordinary differential equations.

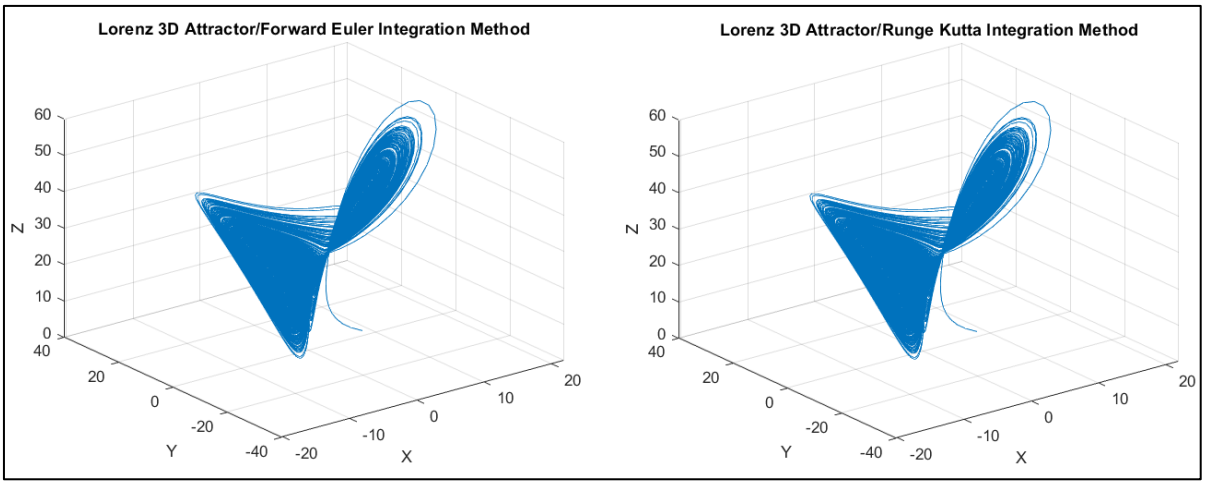

Figure 4-3 3D Lorenz Trajectories using Forward Euler and Runge-Kutta Methods

4.2.2. Image Encryption Strength and Statistical Tests

Figure 4-4, which is a depiction of the proposed cryptographic system, shows the histogram level of the plain and encrypted images. The plain and encrypted image histograms differ significantly, as shown by the distribution levels, but the encrypted image histogram is not completely flat, and this fact suggests that the attacker will be able to produce an important information about the color distributions over the pixels in the plain image.

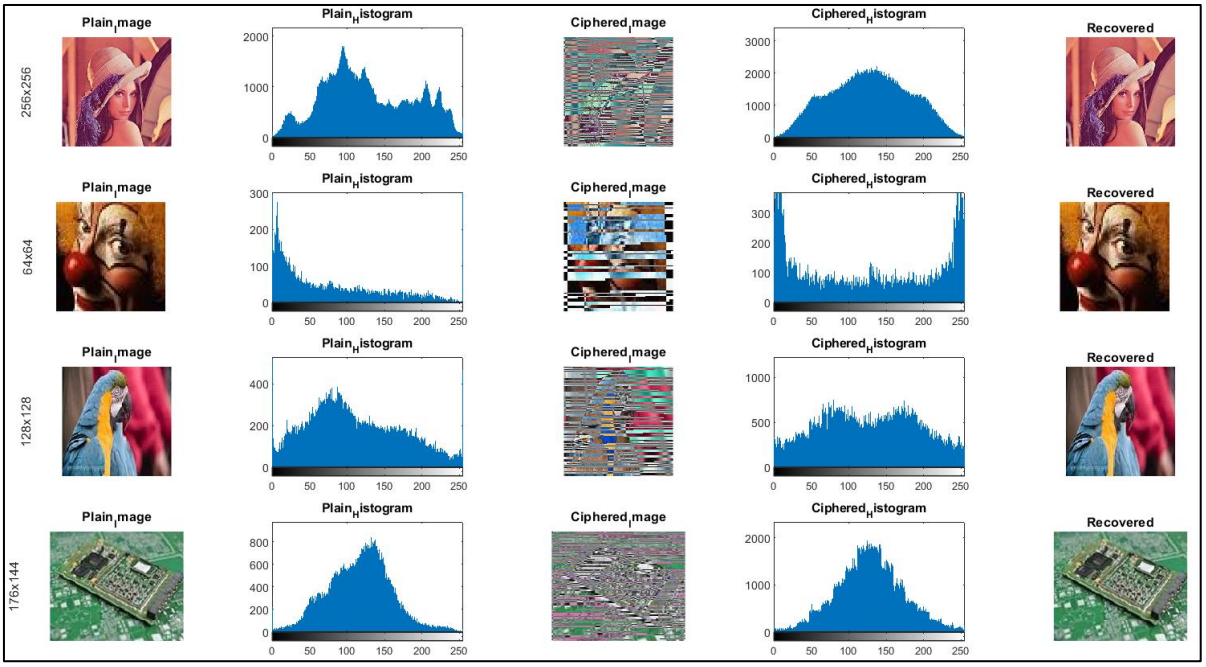

Figure 4-4 Plain Images, Encrypted Images, and Histogram analysis (Algorithm 1)

Table 4-1 compares the PSNR and MSE values for the original, ciphered, and recovered images at various image sizes. The PSNR and MSE values of the recovered images and the original photos demonstrate that the two images are the same (because MSE=0). The suggested cryptosystem can significantly recover the original images from the encrypted ones, but it is vulnerable to statistical threats because the MSE and PSNR values between the ciphered and the original images show some considerable disparities (although not adequate) between them.

Table 4-1 PSNR and MSE of the Proposed Cryptosystem

Image Size	Ciphered Image & Plain Image		Recovered Image & Plain 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

The correlation coefficients between the plain image and the ciphered images (with different sizes) are calculated and presented in table 4-2. From the correlation values it is clear that the adjacent pixel correlations have been broken (correlation values close to zero) and the attacker cannot obtain information to make an attack based on the pixel correlation.

Entropy attack is another type of attacks that the attacker can obtain information about the data source. Close to eight entropy level indicates that the system can cope with entropy attacks. The image entropy of the proposed cryptosystem is calculated and presented in table 4-2 for different image sizes and for the three-color layers.

On order to cope with differential attacks, the NPCR and UACI metrics should equal or very close to 100% and 33% respectively as indicated in chapter 2. Table 4-2 presents the NPCR and UACI metrics values of the

proposed system. Since that the UACI and NPCR values are insufficient, as described above, it is obvious that the proposed cryptosystem is weak, especially to differential attacks.

Table 4-2 Correlation Coefficients, Entropy, NPCR, and UACI Results

Image Size	Color Layer	Correlation	Entropy	NPCR	UACI
256x256	Red	-0.006	7.8138	99.52	31.5
	Green	-0.006	7.7266	99.19	17.19
	Blue	-0.006	7.5924	99.15	13.81
64x64	Red	0.006	7.5898	98.71	11.2
	Green	0.006	7.3368	98.46	18.2
	Blue	0.006	6.7945	96.39	24.4
128x128	Red	-0.0447	7.7498	98.46	27.4
	Green	-0.0447	7.7584	98.85	29.7
	Blue	-0.0447	7.8446	99.14	24.2
176x144	Red	0.0536	7.6641	98.99	55.7
	Green	0.0536	7.3966	98.73	25
	Blue	0.0536	7.4762	98.79	26.1

One of the most aspects related to data security and cryptosystem design is the cryptographic system key space. The cryptosystems with relatively large key space can provide more secure data, stronger and robust against the brute force attack. The proposed image encryption system has six secret keys represented by the initial value of the chaos system **(x_o, y_o, z_o)** as well as the system parameters **(β, ρ, σ)**. The chaotic parameters and the initial states required 32 bits to represent them therefore the suggested system's key space will be $(2^{32})^6=2^{192}$. Therefore, this key space is suitable for image encryption and effective against the brute force attack since it greater than 2^{100}. Table 4-3 shows a comparison between the proposed system and a traditional cryptosystem that used widely regarding to systems key space.

Table 4-3 Key Space Comparison

Encryption Algorithm	Key space
Proposed System	2^{192}
Reference [84]	2^{45}
Reference [85]	2^{199}
Reference [18]	2^{203}
Reference [17]	2^{149}
Reference [86]	2^{200}

4.2.3. FPGA Implementation Results and Analysis

Both solutions (Forward Euler and Runge-Kutta) are implemented on the FPGA board. The Xilinx system generator XSG is used through this work to obtain the VHDL code for proposed chaotic based cryptographic system. The obtained code is used and then to configure the FPGA PYNQ-Z1 evolution board. Figure 4-5, depicts the FPGA implementation using Forward Euler while figure 4-6 illustrate the Runge-Kutta method-based solution. As clear from the two figures (4-5 and 4-6), the encryption/decryption results are identical for the two solution methods which proof that both of them are suitable for solving chaotic/hyperchaotic ODEs, but more accuracy and computational resources are required in the case of Runge-Kutta method.

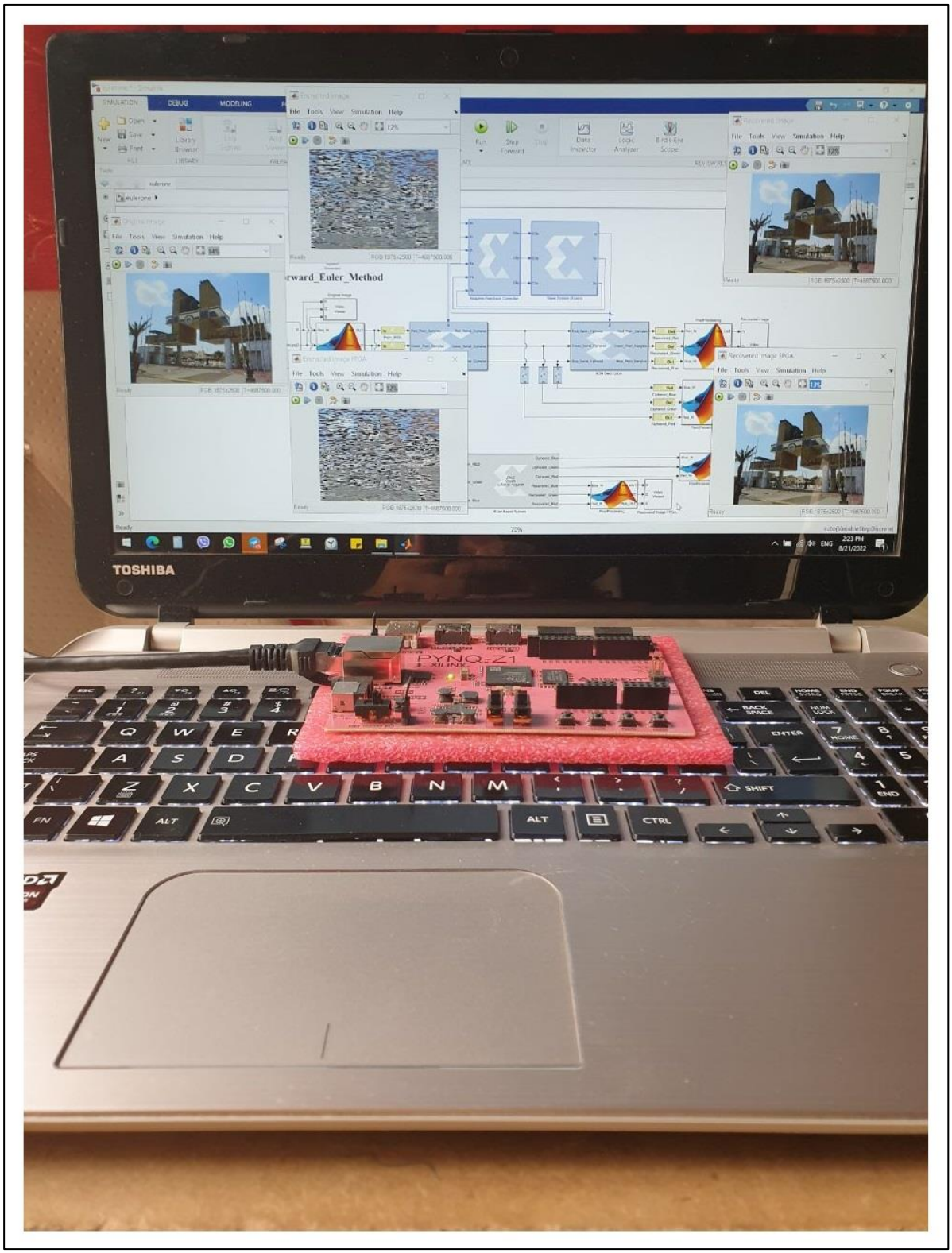

Figure 4-5 Image Encryption Cryptography Hardware Co-Simulation in a Real-Time Environment (Forward Euler method)

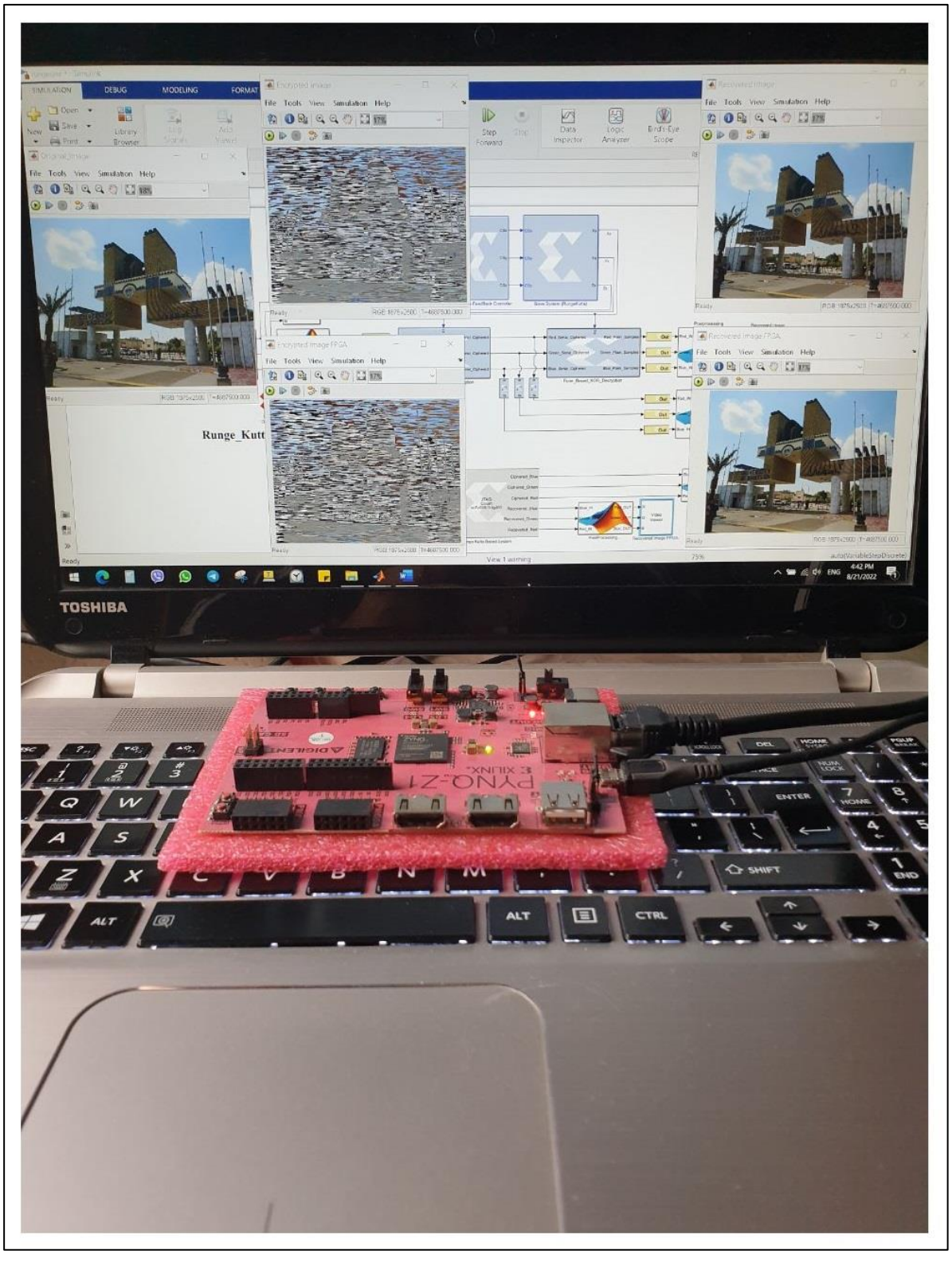

Figure 4-6 Image Encryption Cryptography Hardware Co-Simulation in a Real-Time Environment (Runge-Kutta method)

4.3. Algorithm (2): Multidimensional Hyperchaotic System Based on XOR Mixture of Dynamical Systems

Dynamical time response of the system state variables (x, y, z, w, u, v) of the transmitter and receiver, image encryption/decryption results, encryption strength measurements, as well as the statistical analysis and FPGA implementation for proposed algorithm 2 are presented in this section.

4.3.1. System Dynamical Response

The time response of the three nonlinear hyperchaotic systems (4D, 6D, and 7D) including all the state variable components is presented, where figure 4-7 presents the state variables of the four-dimensional hyperchaotic system including the variables (x, y, z, and w). Figure 4-8 shows the state variable of the six-dimensional hyperchaotic system and the seven-dimensional hyperchaotic system state variables are presented in figure 4-9. The Forward Euler integration method has been used to solve them. The X components of the systems are combined together using XOR logical operation to generate the random bit stream that will be used to encrypt the plain images.

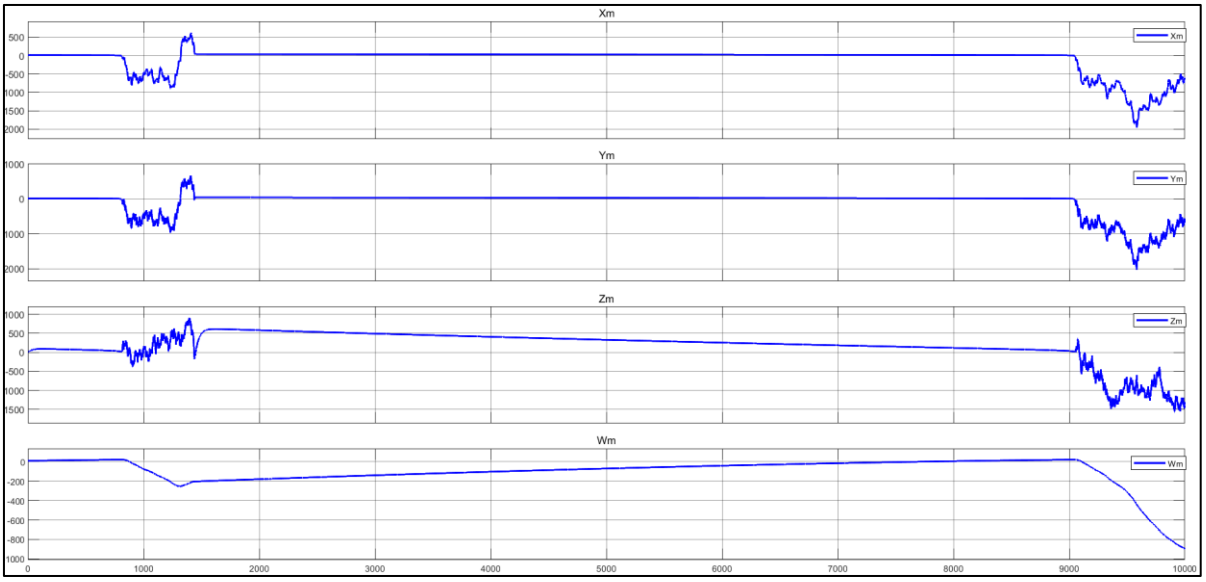

Figure 4-7 State Variables (x, y, z, w) Dynamical Response of 4Dimensional hyperchaotic System (Algorithm 2)

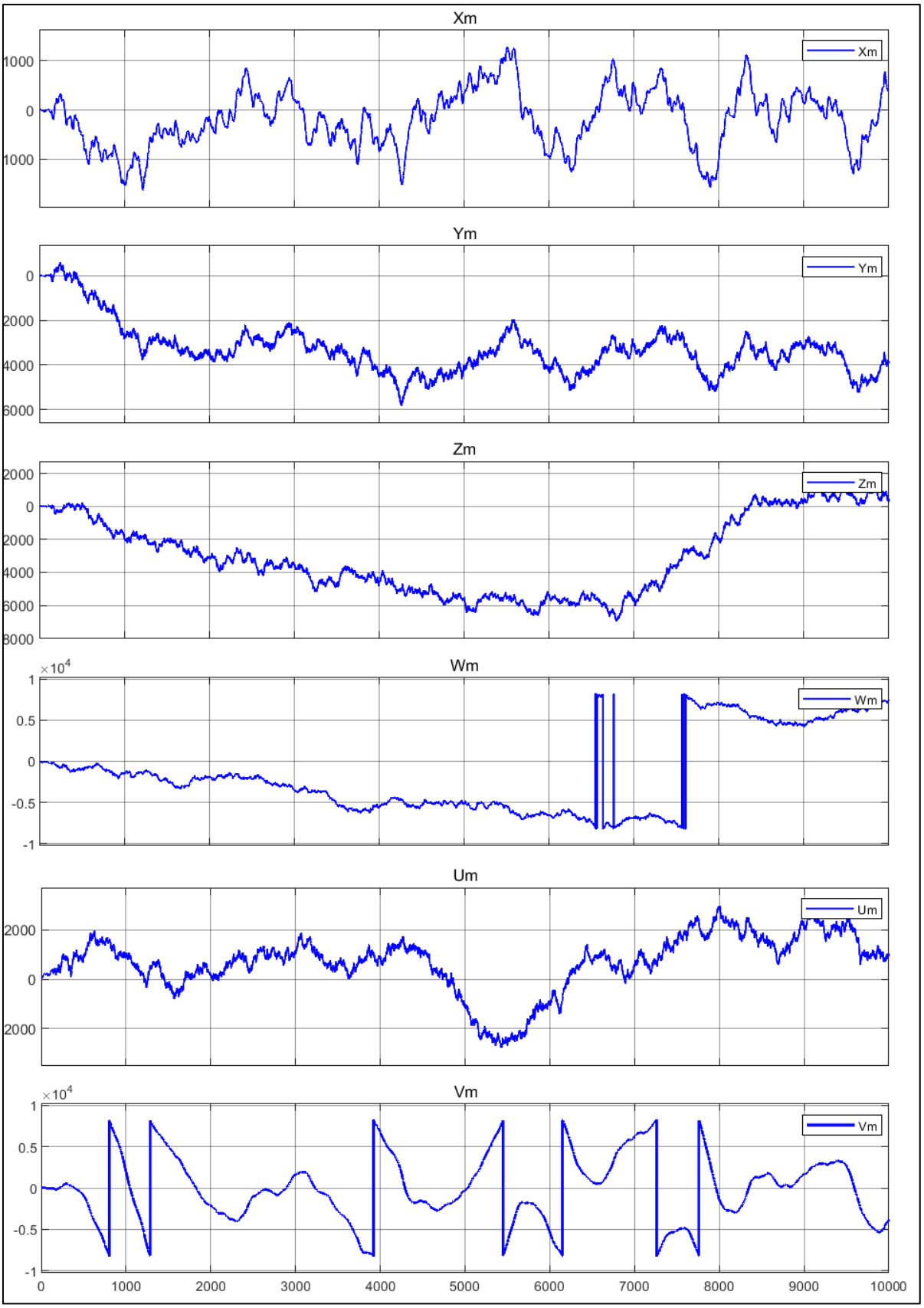

Figure 4-8 State Variables (x, y, z, w, u, v) Dynamical Response of 6 Dimensional hyperchaotic System (Algorithm 2)

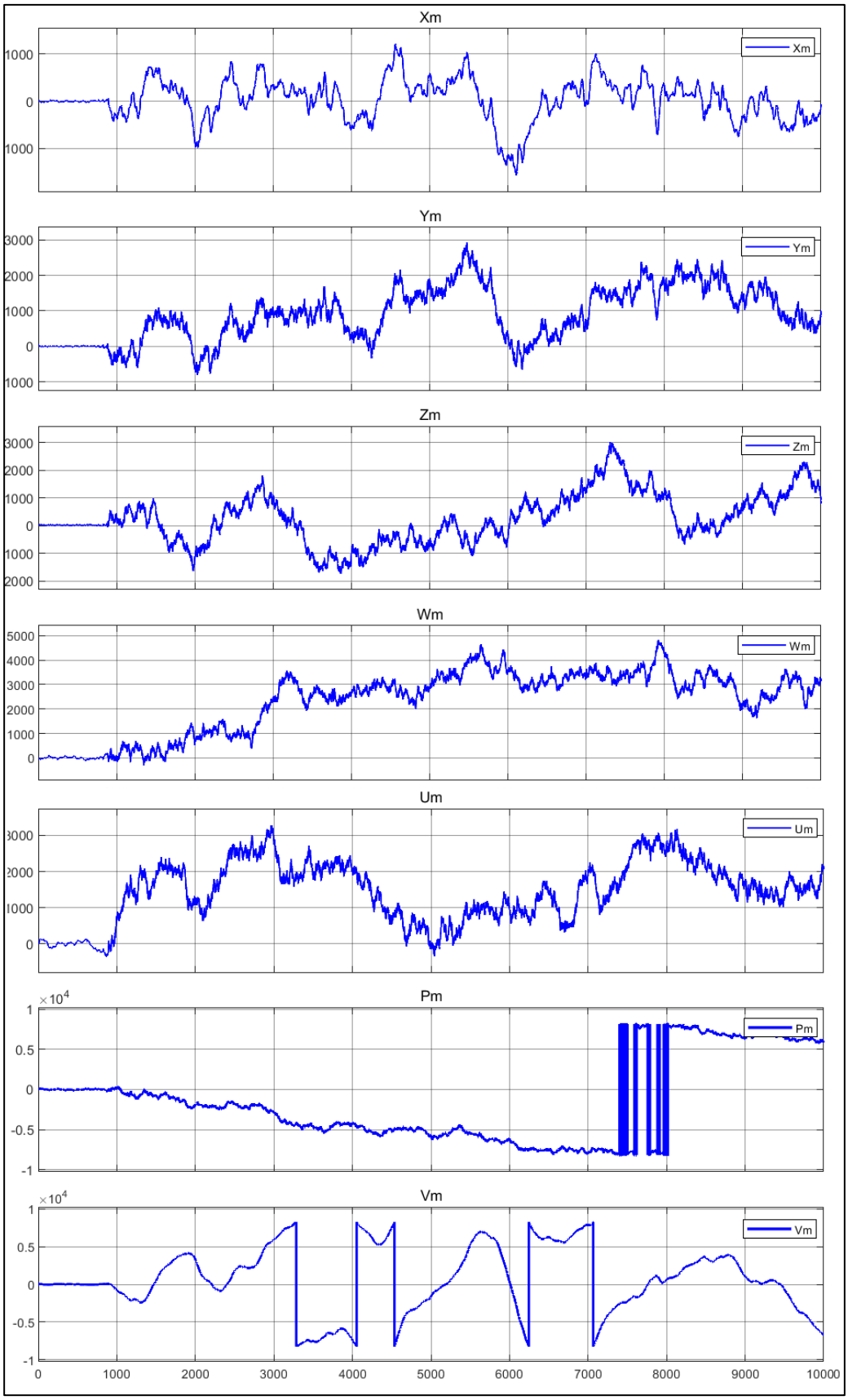

Figure 4-9 State Variables (x, y, z, w, u, v, p) Dynamical Response of 7 Dimensional hyperchaotic System (Algorithm 2)

4.3.2. Image Encryption Strength and Statistical Tests

Figure 4-10, shows the histogram level of the plain and encrypted photos for the suggested cryptographic system. The plain and encrypted image histograms differ significantly, as seen by the distribution levels, but the encrypted image histogram is completely flat, which means that the attacker won't be able to discern anything about the color distributions across the pixels in the plain image.

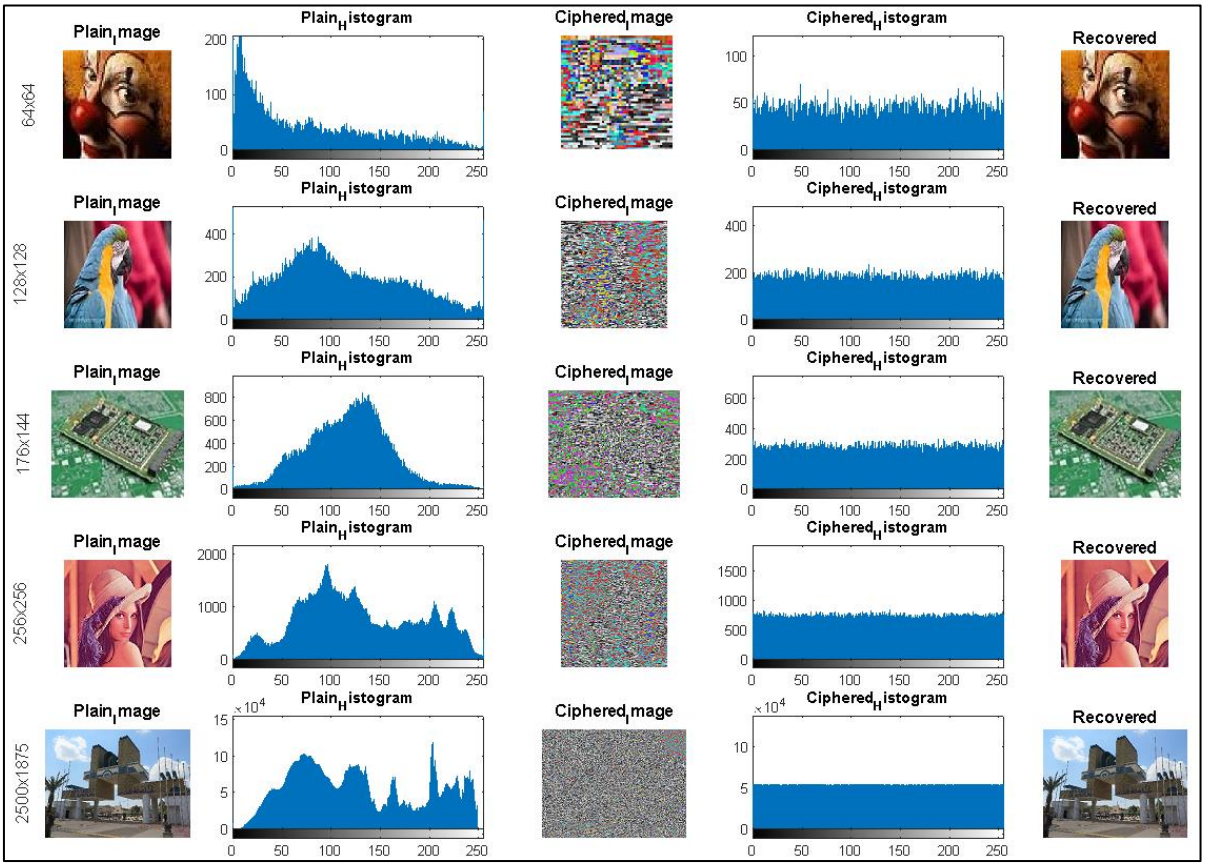

Figure 4-10 Plain Images, Encrypted Images, and Histogram analysis (Algorithm 2)

The PSNR and MSE values for the original, ciphered, and recovered images at different image sizes are contrasted in Table 4-4. Since MSE=0, and PSNR=inf, the recovered images and the original images are identical. However, the comparison of MSE and PSNR values between the ciphered and the original images reveals that there are significant differences between them. The proposed cryptosystem is significantly capable of recovering the original

images from the encrypted one and is strong enough to withstand statistical attacks.

Table 4-4 PSNR and MSE of the Proposed Cryptosystem (Algorithm 2)

Image Size	Ciphered Image & Plain Image		Recovered Image & Plain Image	
	PSNR	MSE	PSNR	MSE
64 x 64	6.9019	1.3271e+04	Inf	0
128 x 128	8.3780	9.4467e+03	Inf	0
176 x 144	9.4193	7.4328e+03	Inf	0
256 x 256	8.5495	9.0810e+03	Inf	0
2500 x 1875	8.2779	9.6671e+03	Inf	0

The correlation coefficients between the plain image and the ciphered images (with different sizes) are calculated and presented in table 4-5. From the correlation values, it is clear that the adjacent pixel correlations have been broken (correlation values are very close to zero) and the attacker cannot obtain any information to vulnerable the cryptosystem.

Entropy attack is another type of attacks that the attacker can obtain information about the data source. Close to eight entropy level indicates that the system can cope with entropy attacks. The image entropy of the proposed cryptosystem is calculated and presented in table 4-5 for different image sizes and for the three-color layers, where all the entropy calculations are very close to eight which means that the cryptosystem has high immunity to the entropy-based attacks.

As shown in table 4-5, the proposed system's NPCR and UACI metrics are very close to 100% and 33%, respectively, and these values indicate that the system is robust enough to withstand differential attacks.

Table 4-5 Correlation, Entropy, NPCR, and UACI Results

Image Size	Color Layer	Correlation	Entropy	NPCR	UACI
64x64	Red	0.0263	7.9457	99.34	33.4
	Green	0.0099	7.9561	99.68	33.6
	Blue	0.0042	7.9582	99.49	33.5
128x128	Red	0.0245	7.9873	99.62	33.8
	Green	0.0264	7.9892	99.62	33.1
	Blue	0.0015	7.9881	99.58	33.2
176x144	Red	0.0024	7.9938	99.59	33.1
	Green	-0.0020	7.9913	99.66	33.8
	Blue	-0.0034	7.9921	99.62	33.7
256x256	Red	-0.0080	7.9971	99.61	33.2
	Green	-0.0106	7.9973	99.57	33.5
	Blue	-0.0072	7.9970	99.58	33.1
2500x1875	Red	-0.0013	8.0000	99.61	33.2
	Green	-0.0011	8.0000	99.61	33.7
	Blue	-0.0014	8.0000	99.61	33.2

A comparison between the key space of the proposed cryptosystem and well-known classical algorithms are shown in Tables 4-6. It is clear that the proposed system has a significant increase in the key space which makes it immune against the brute force attack.

Table 4-6 Key Space Comparison

Encryption Algorithm	**Key space**
Proposed System	$\mathbf{2^{1088}}$
Reference [84]	$\mathbf{2^{45}}$
Reference [85]	$\mathbf{2^{199}}$
Reference [18]	$\mathbf{2^{203}}$
Reference [17]	$\mathbf{2^{149}}$
Reference [86]	$\mathbf{2^{200}}$

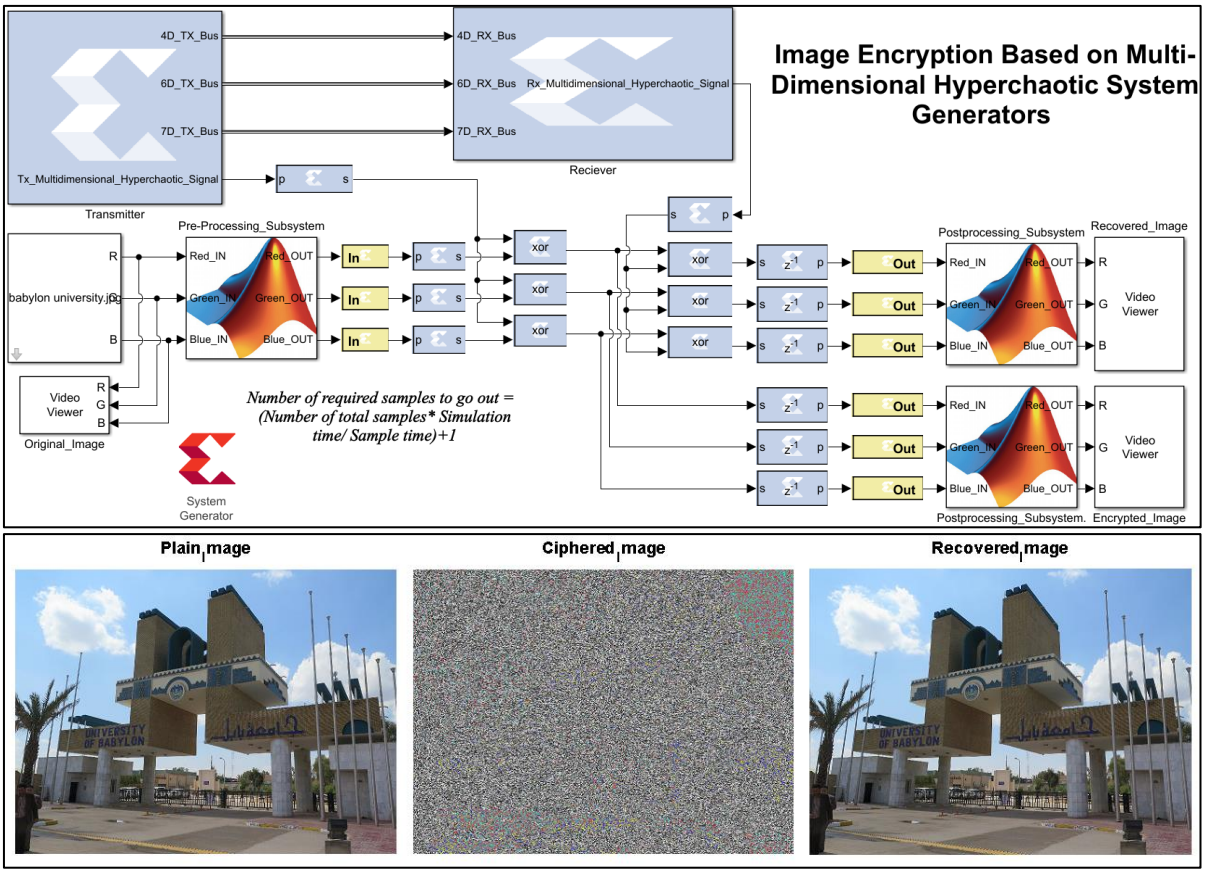

Figure 4-11 Plain, Encrypted, and Recovered Image of the Proposed Algorithm 2 (XSG Environment)

Figure 4-11 presents the overall communication system based on algorithm 2, with plain, ciphered and recovered images of a size (2500×1875). As shown in the figure, the plain image is highly secured and effectively encrypted. Consequently, the system has the ability to recover exactly the same image from the encrypted one which indicates the robustness of the proposed system.

4.3.3. FPGA Implementation Results and Analysis

The proposed multi-dimensional hyperchaotic system has been implemented with FPGA PYNQ-Z1 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board. Figure 4-12, depicts implementation of the proposed cryptographic algorithm using the FPGA board. The plain image is called from its location in the PC and sent to the board serially to encrypt them,

after completing the encryption process the encrypted samples are sent back to the PC to display the ciphered image as shown in figure 4-12 below.

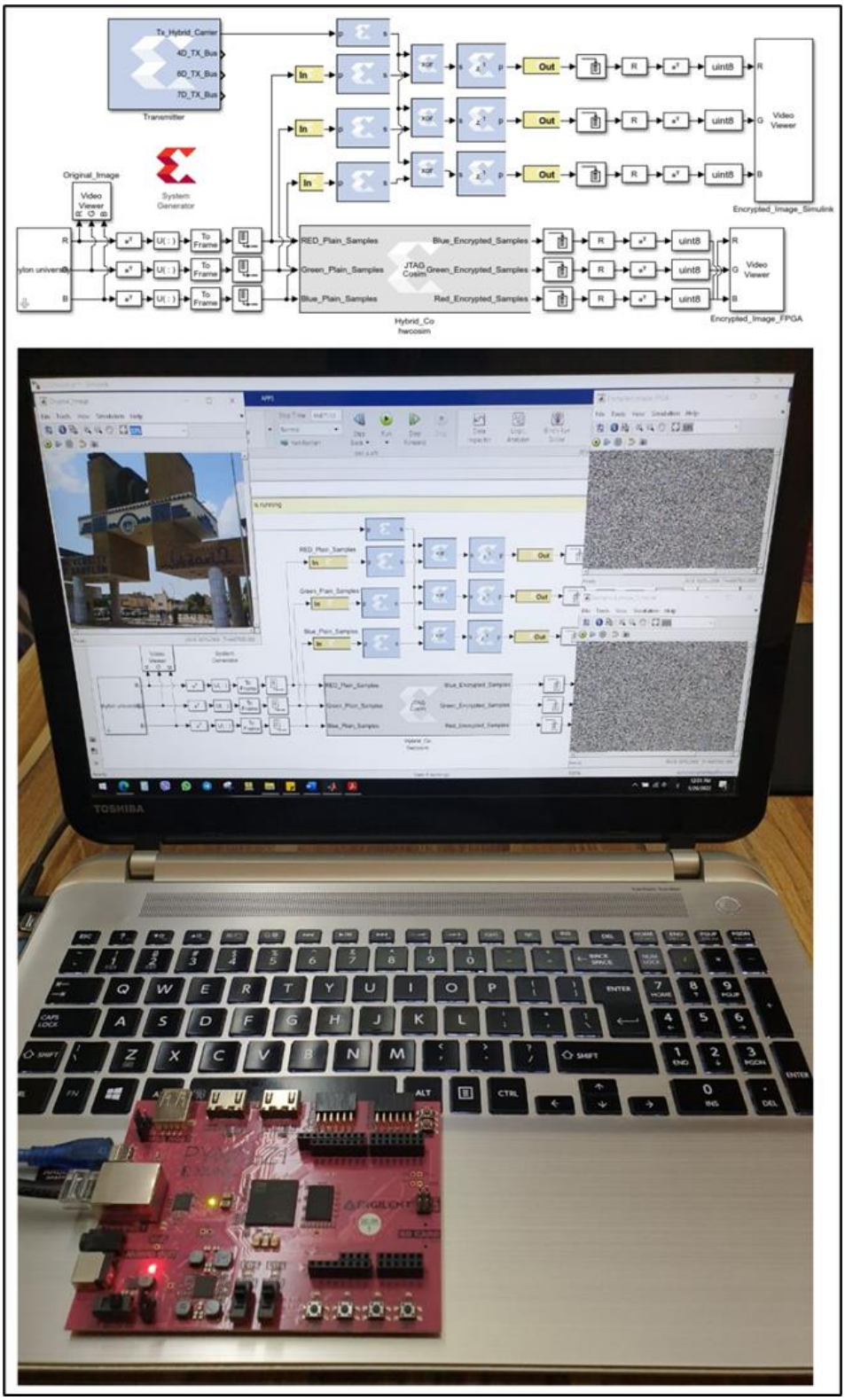

Figure 4-12 Image Encryption Cryptography Hardware Co-Simulation in a Real-Time Environment (Algorithm 2)

4.4. Algorithm (3): Multidimensional Cascaded Hyperchaotic Systems Based on Chaos Switching Technique

Dynamical time response of the system state variables (x, y, z, w, u, v) of the transmitter and receiver, image encryption/decryption results, encryption strength measurements, as well as the statistical analysis and FPGA implementation for proposed algorithm 3 are presented in this section.

4.4.1. System Dynamical Response

The time response of the four nonlinear systems (3D, 4D, 6D, and 7D) including all the state variable components are presented in this section. Figure 4-13 presents the state variables of the three-dimensional chaotic system including the variables (x, y, z, and w) that will be act as a high-speed switch, where the x, y, z dynamics are converted into binary stream and they are combined together by using XOR operation. The XOR operation stream will be used as a chaos switching that choose one of the three generated streams.

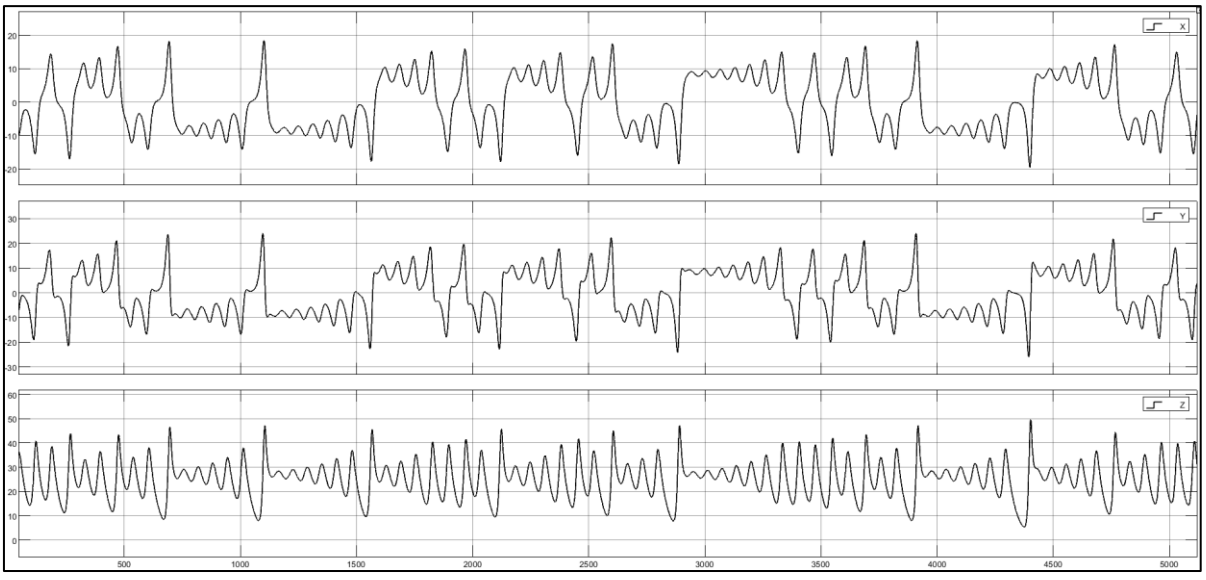

Figure 4-13 State Variables (x, y, z) Dynamical Response of 3-Dimensional Chaotic System (Algorithm 3)

Figure 4-14 presents the state variable of the four-dimensional hyperchaotic system, six-dimensional hyperchaotic system state variables are presented in figure 4-15. The last seven-dimensional hyperchaotic system

dynamics are illustrated in figure 4-16. The Forward Euler integration method has been used to solve them.

The X components of the three hyperchaotic systems (4D, 6D, and 7D) are combined together using high speed selector switch that controlled by the output of the binary stream generated from the three-dimensional chaotic system. The output of the selector switch represents the random bit stream that will be used to encrypt the plain images.

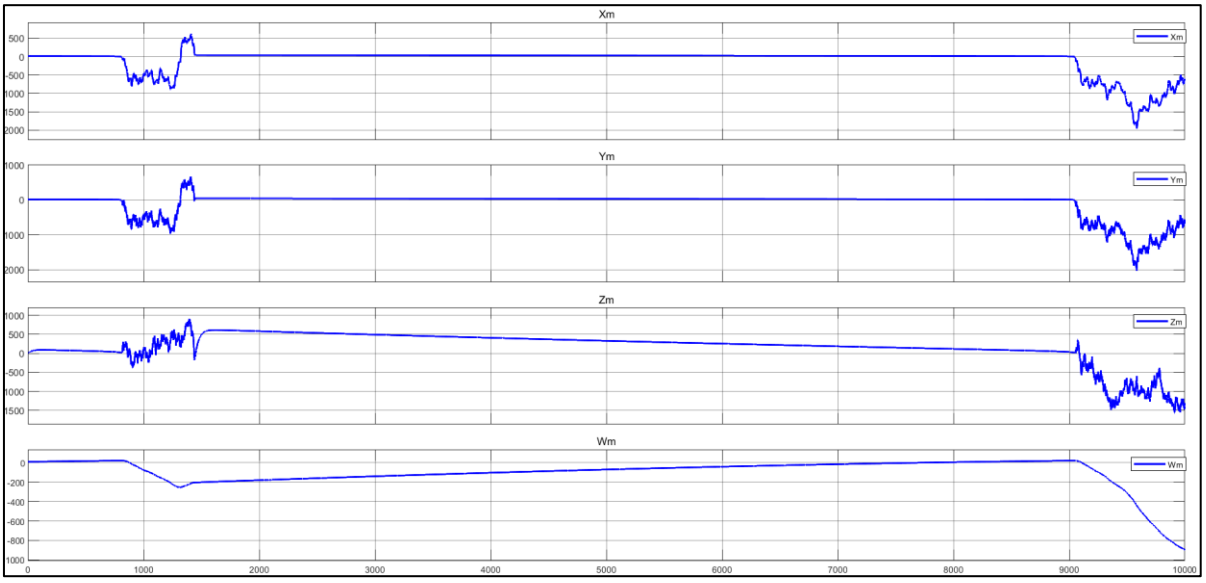

Figure 4-14 State Variables (x, y, z, w) Dynamical Response of 4Dimensional hyperchaotic System (Algorithm 3)

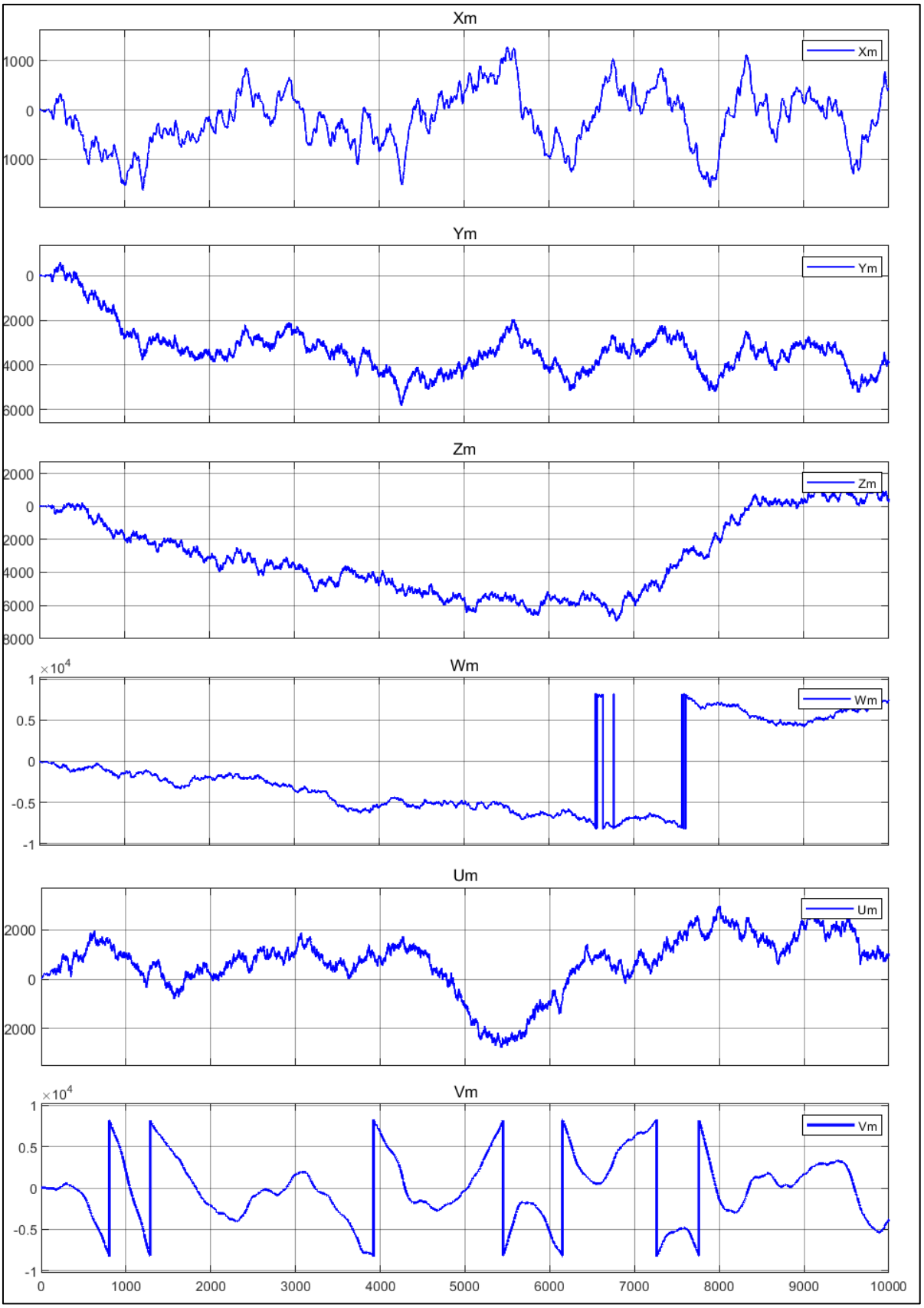

Figure 4-15 State Variables (x, y, z, w, u, v) Dynamical Response of 6 Dimensional hyperchaotic System (Algorithm 3)

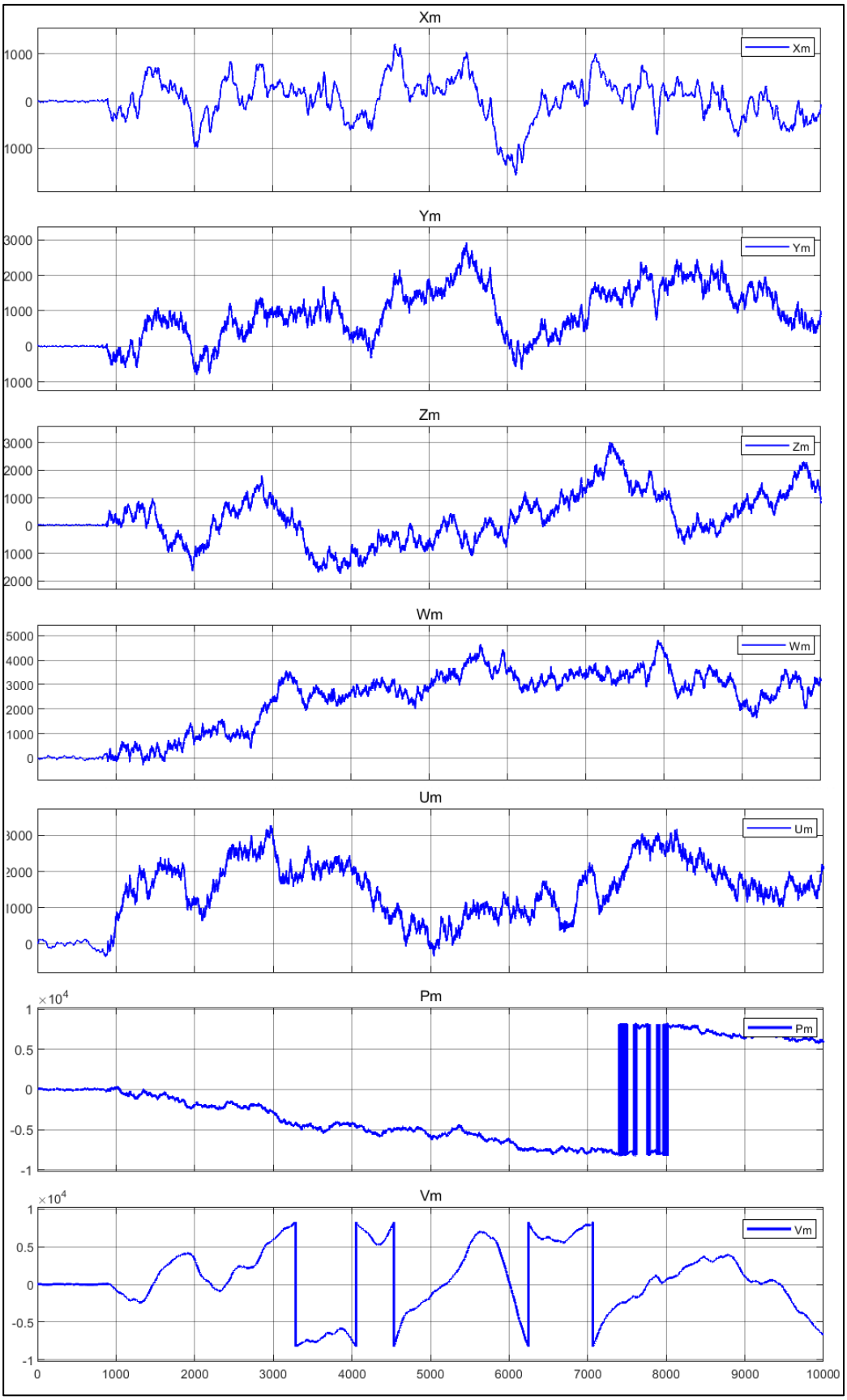

Figure 4-16 State Variables (x, y, z, w, u, v, p) Dynamical Response of 7 Dimensional hyperchaotic System (Algorithm 3)

4.4.2. Image Encryption Strength and Statistical Tests

The histogram analysis of the cryptographic system that proposed in algorithm 3 is presented in figure 4-17 shown below and for different image sizes. Where pixels values in the plain images are irregular (uneven) and some pixels have higher frequencies than the others where this can be considered as a threat. In order to resist this kind of attacks, ciphered image histogram should be uniformly distributed as much as possible to resist this kind of attacks and the proposed algorithm histogram achieves this purpose as shown in figure below.

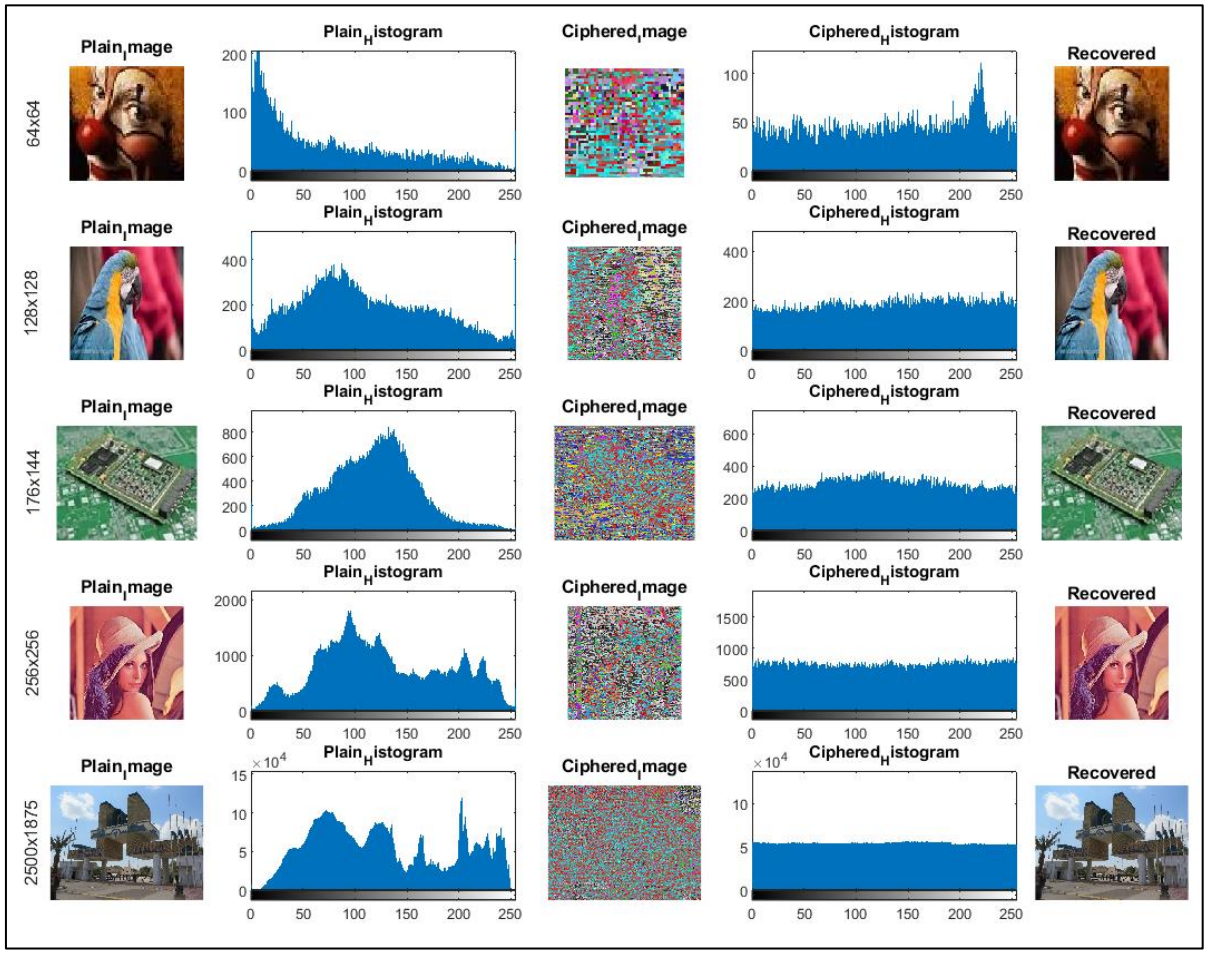

Figure 4-17 Plain Images, Encrypted Images, and Histogram analysis (Algorithm 3)

Again, using the MSE and PSNR, the difference level between the plain image, the ciphered image, and the recovered image is also evaluated for this method. Higher MSE values, as noted, show a significant difference between

the plain and ciphered images. Table 4-7, depicts a concise summary of the MSE and PSNR calculation values. It is obvious that ciphered and plain images are very distinct from one another and can withstand cyber-attacks.

Table 4-7 PSNR and MSE of the Proposed Cryptosystem (Algorithm 3)

Image Size	Ciphered Image & Plain Image		Recovered Image & Plain Image	
	PSNR	MSE	PSNR	MSE
64 x 64	6.4527	1.4717e+04	Inf	0
128 x 128	8.2493	9.7308e+03	Inf	0
176 x 144	9.6122	7.1099e+03	Inf	0
256 x 256	8.5478	9.0846e+03	Inf	0
2500 x 1875	8.1696	9.9110e+03	Inf	0

The adjacent pixels in any digital image are closely related (connected or corelated) to one another in the vertical, horizontal, and even diagonal axes. Table 4-8 displays the correlation coefficients for the neighboring pixels. Based on the correlation values, the system could be compromised. If it conceals the association between adjacent pixels, a good cryptography algorithm. The correlation calculations are shown in table 4-8, which demonstrates how well the system hides and reduces the correlation data that outsiders might utilize.

The suggested technique also calculates the information entropy analysis, which is frequently used to describe the randomness and unpredictability of the information source. Table 4-8 also demonstrates that the entropy is extremely close to eight, indicating that the system's behavior is very random and unpredictable and that it can withstand a variety of electronic attacks.

Table 4-8 Correlation, Entropy, NPCR, and UACI Results (Algorithm 3)

Image Size	Color Layer	Correlation	Entropy	NPCR%	UACI%
64x64	Red	0.1516	7.9454	99.41	33.4
	Green	-0.1345	7.9076	99.54	33.6
	Blue	-0.0841	7.8692	99.56	33.5
128x128	Red	0.1178	7.9832	99.51	33.8
	Green	-0.0760	7.9857	99.63	33.1
	Blue	-0.0609	7.9825	99.66	33.2
176x144	Red	0.0031	7.9880	99.55	33.1
	Green	-0.0207	7.9833	99.55	33.8
	Blue	-0.0100	7.9870	99.61	33.7
256x256	Red	-0.0127	7.9949	99.62	33.2
	Green	0.0091	7.9963	99.61	33.5
	Blue	0.0056	7.9938	99.62	33.1
2500x1875	Red	0.0422	7.9995	99.62	33.2
	Green	-0.0545	7.9997	99.63	33.7
	Blue	-0.0619	7.9994	99.63	33.2

As shown in table 4-8, the proposed system's NPCR and UACI metrics are very close to 100% and 33%, respectively, and these values indicate that the system is robust enough to withstand differential attacks.

The cryptographic algorithms that possess large key space offer more security level and these systems can cope with brute force attacks. The proposed hyperchaotic algorithm has 34 keys represented by the initial states of the hyperchaotic systems and the systems parameters. The hyperchaotic system parameters and initial states required 32 bits to represent them and this means that the key space of the cascaded hyperchaotic system-based chaos switching will be $(2^{32})^{34}$. Table 4-9 shows a comparison in the key space between the proposed system and the traditional cryptographic systems where it can be noted that the proposed system key space is larger than the others which consequently provide higher security and can stands against different types of attacks.

Table 4-9 Key Space Comparison (Algorithm 3)

Encryption Algorithm	Key space
Proposed System	2^{1088}
Reference [84]	2^{45}
Reference [85]	2^{199}
Reference [18]	2^{203}
Reference [17]	2^{149}
Reference [86]	2^{200}

4.4.3. FPGA Implementation Results and Analysis

The proposed chaos-based switching cascaded hyperchaotic system has been implemented with FPGA PYNQ-Z1 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board. Figure 4-18, depicts implementation of the proposed cryptographic algorithm using the FPGA board. The plain image is called from its location in the PC and sent to the board serially to encrypt them. After completing the encryption process the encrypted samples are sent back to the PC to display the ciphered image as shown in figure 4-18. Due to the limitation in the FPGA board resources, the transmitter system is implemented only and a comparison is presented in figure 4-18 between the board encryption and XSG model encryption which show identical results.

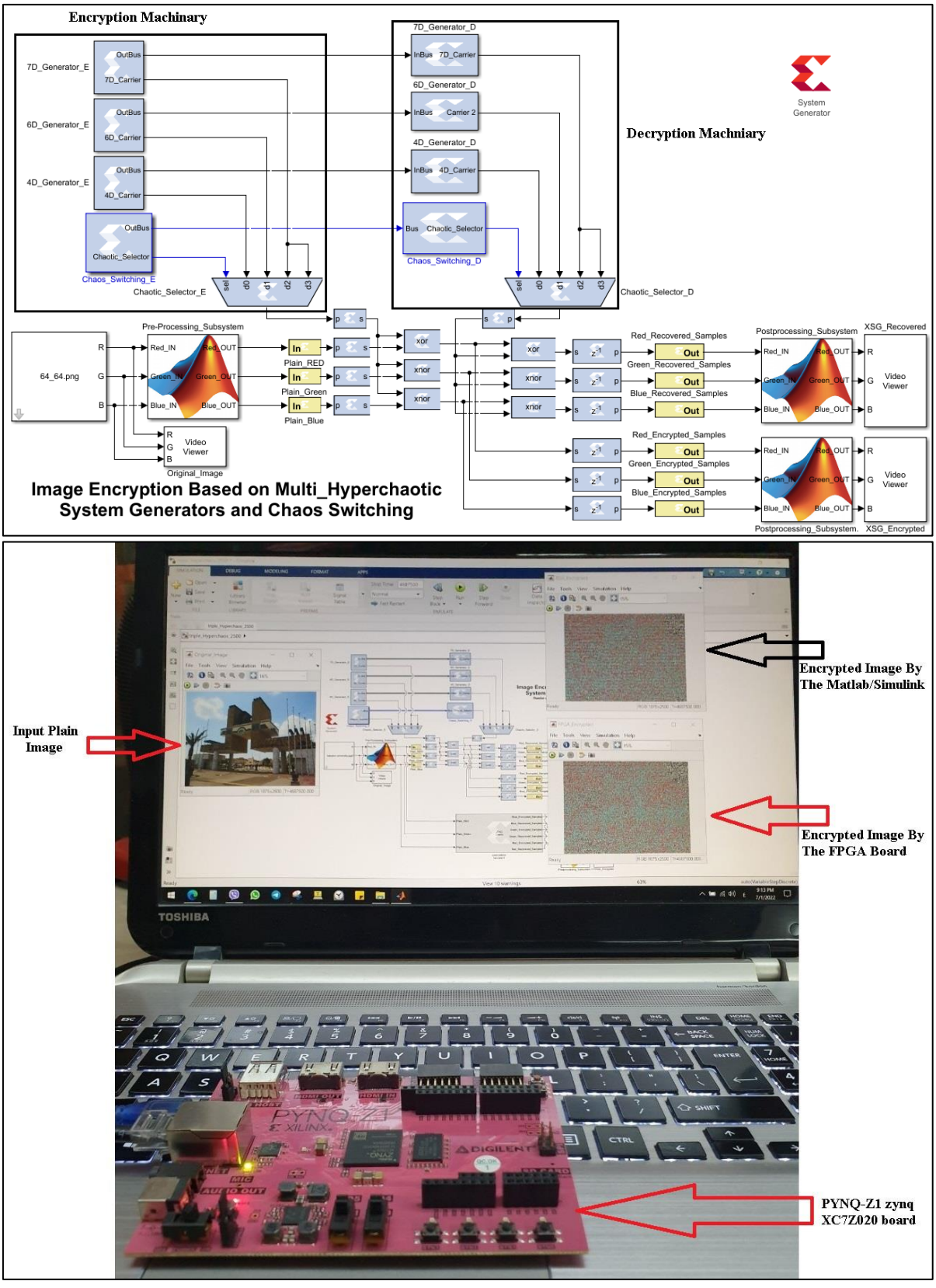

Figure 4-18 Image Encryption Cryptography Hardware Co-Simulation in a Real-Time Environment (Algorithm 3)

4.5. Algorithm (4): Robust Encryption System Based on Novel Hyperchaotic Flow System.

Dynamical time response of the system state variables (x, y, z, w, p) of the transmitter and receiver, nonlinear system strange attractor, image encryption/decryption results, encryption strength measurements, as well as the statistical analysis and FPGA implementation and board utilization summary for proposed algorithm 4 are presented in this section.

4.5.1. System Dynamical Response

The time response of the novel five-dimensional hyperchaotic systems including all the state variable components as well as the strange attractors in 2D and 3D is presented in this section. Figure 4-19 presents the state variables of the five-dimensional hyperchaotic system including the variables (x, y, z, w, and p). Figure 4-20 shows the system strange attractor in 3D view, while the figure 4-21 presents the 2D view attractor. The x dynamics of the new attractor is converted into binary data to be used for the process of image encryption.

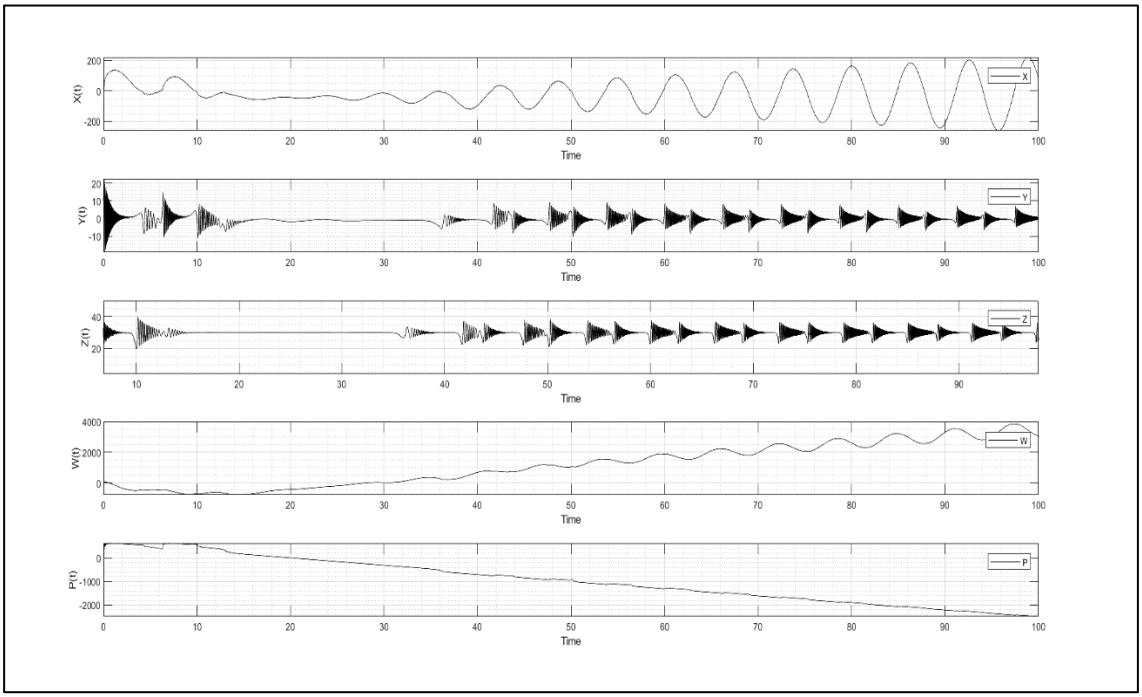

Figure 4-19 State Variables (x, y, z, w, p) Dynamical Response of 5-Dimensional Hyperchaotic System (Algorithm 4)

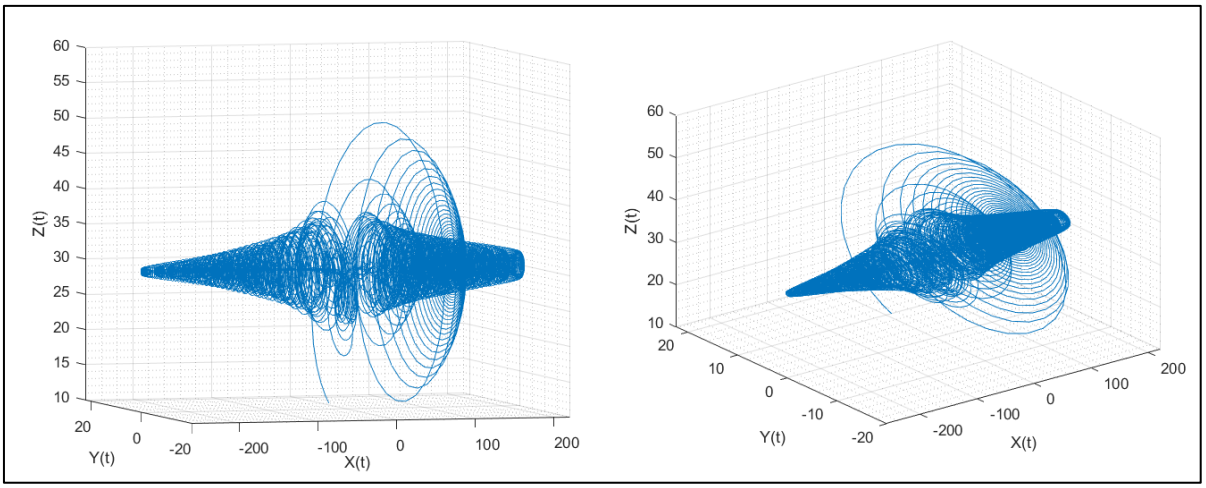

Figure 4-20 3D New Hyperchaotic System Strange Attractor (Algorithm 4)

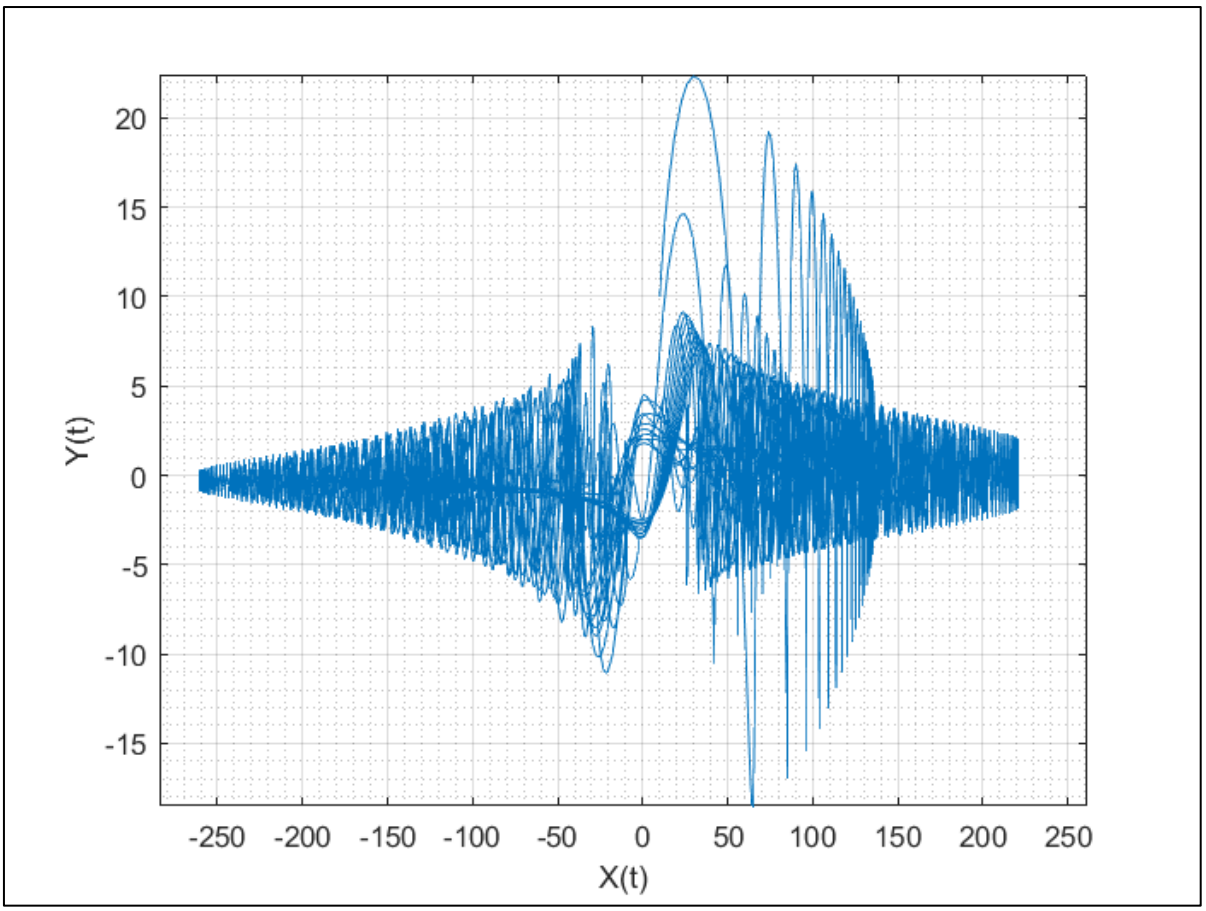

Figure 4-21 2D New Hyperchaotic System Strange Attractor (Algorithm 4)

4.5.2. Image Encryption Strength and Statistical Tests

Histogram analysis of five different plain and ciphered images is presented in figure 4-22. It is clear that the system can resist the cyber-attacks that carried out based on the color distribution, where the distribution of the colors is totally disappear and hide in the flat and uniform histogram that directed in figure 4-22.

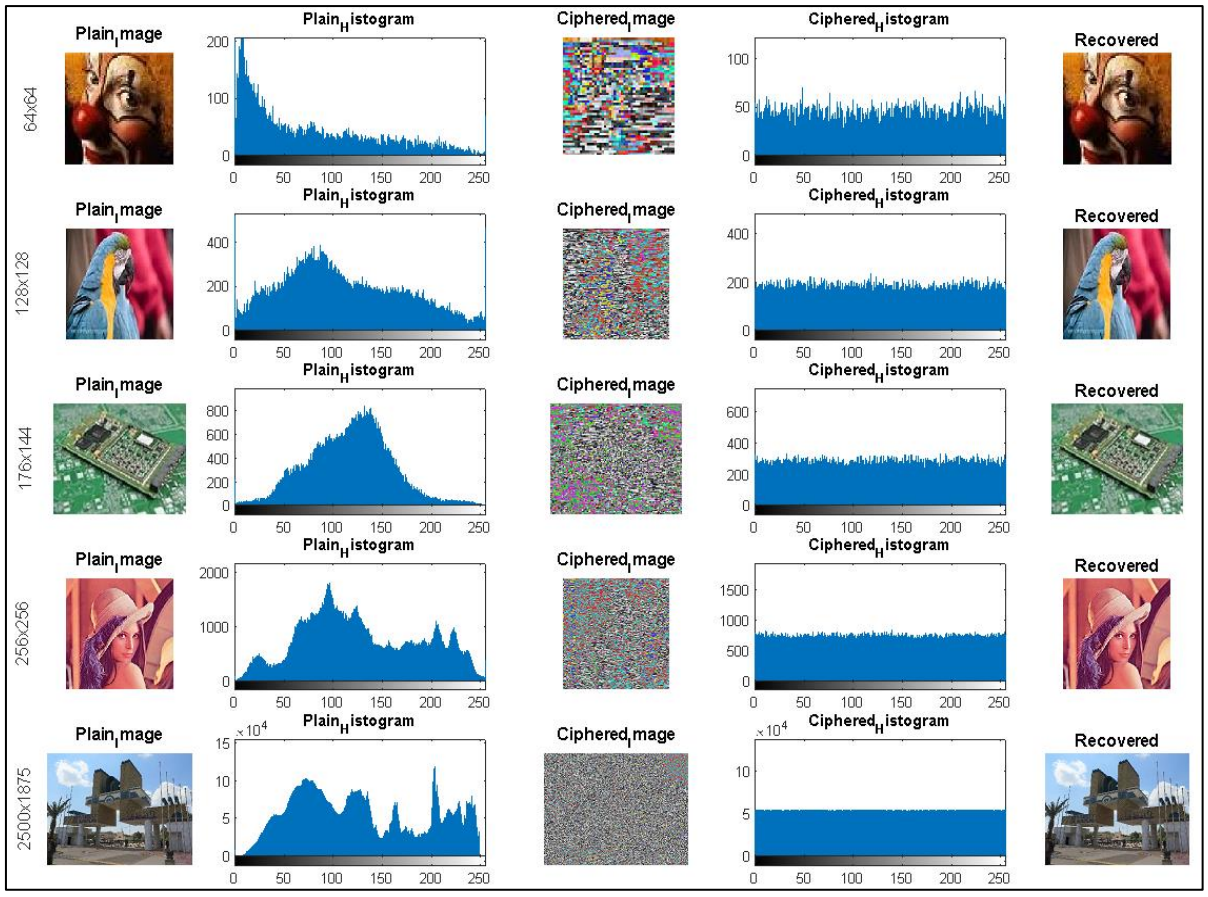

Figure 4-22 Plain Images, Encrypted Images, and Histogram analysis (Algorithm 4)

MSE and PSNR are also employed in this method to assess how much the recovered image differs/similar from the plain image, ciphered image. Table 4-10's findings demonstrate how the two images—plain and ciphered—are distinctly different from one another and how this makes the system resilient to cyberattacks.

Table 4-10 PSNR and MSE of the Proposed Cryptosystem (Algorithm 4)

Image Size	Ciphered Image & Plain Image		Recovered Image & Plain Image	
	PSNR	MSE	PSNR	MSE
64 x 64	6.2527	1.5717e+04	Inf	0
128 x 128	9.4493	9.8308e+03	Inf	0
176 x 144	8.6122	7.4099e+03	Inf	0
256 x 256	9.2478	9.3846e+03	Inf	0
2500 x 1875	8.9696	9.8610e+03	Inf	0

Some statistical tests that evaluate the proposed cryptography system's encryption strength are shown in Table 4-11. The correlation coefficients are among the most crucial factors. Table 4-11 calculates and displays the correlation between the plain picture and the ciphered images (with various sizes). Since the correlation values are so near to zero, it is obvious that the neighboring pixel correlations have been broken, making it impossible for the attacker to get any information that would make the cryptosystem susceptible.

Entropy attacks are another way for the attacker to learn more about the data source. The system can withstand entropy attacks if its entropy level is close to eight. For various image sizes and for the three-color layers, the image entropy of the proposed cryptosystem is computed and shown in table 4-11. All entropy estimates are extremely close to eight, indicating that the cryptosystem has a high resistance to entropy-based assaults.

As shown in table 4-11, the proposed system's NPCR and UACI metrics are very close to 100% and 33%, respectively, and these values indicate that the system is robust enough to withstand differential attacks.

Table 4-11 Correlation, Entropy, NPCR, and UACI Results (Algorithm 4)

Image Size	Color Layer	Correlation	Entropy	NPCR %	UACI %
64x64	Red	0.0063	7.8957	99.44	33.3
	Green	0.0199	7.9761	99.78	33.2
	Blue	0.0032	7.9982	99.39	33.4
128x128	Red	0.0175	7.9973	99.52	33.7
	Green	0.0194	7.9592	99.32	33.1
	Blue	0.0021	7.9381	99.78	33.2
176x144	Red	0.0024	7.8938	99.29	33.1
	Green	-0.0019	7.8913	99.96	33.6
	Blue	-0.0024	7.9421	99.52	33.7
256x256	Red	-0.0079	7.9371	99.31	33.2
	Green	-0.0216	7.9273	99.17	33.5
	Blue	-0.0182	7.9270	99.28	33.1
2500x1875	Red	-0.0213	8.0000	99.78	33.2
	Green	-0.0023	8.0000	99.81	33.7
	Blue	-0.0052	8.0000	99.51	33.2

4.5.3. FPGA Implementation Results and Analysis

The proposed new five-dimensional hyperchaotic system has been implemented with FPGA PYNQ-Z1 evolution board using Xilinx System Generator XSG. The XSG is used to obtain the VHDL codes that used to configure and program the board.

The proposed cryptographic algorithm was illustrated using the FPGA board in Figure 4-23. As illustrated in figure 4-23, the plain image is summoned from its location in the PC and transferred serially to the board to be encrypted. Once the encryption process is complete, the encrypted samples are sent back to the PC to display the ciphered image. The transmitter system is the

only one that has been implemented due to the FPGA board's resource limitations. Figure 4-23 compares the results of board encryption with XSG model encryption, which are identical.

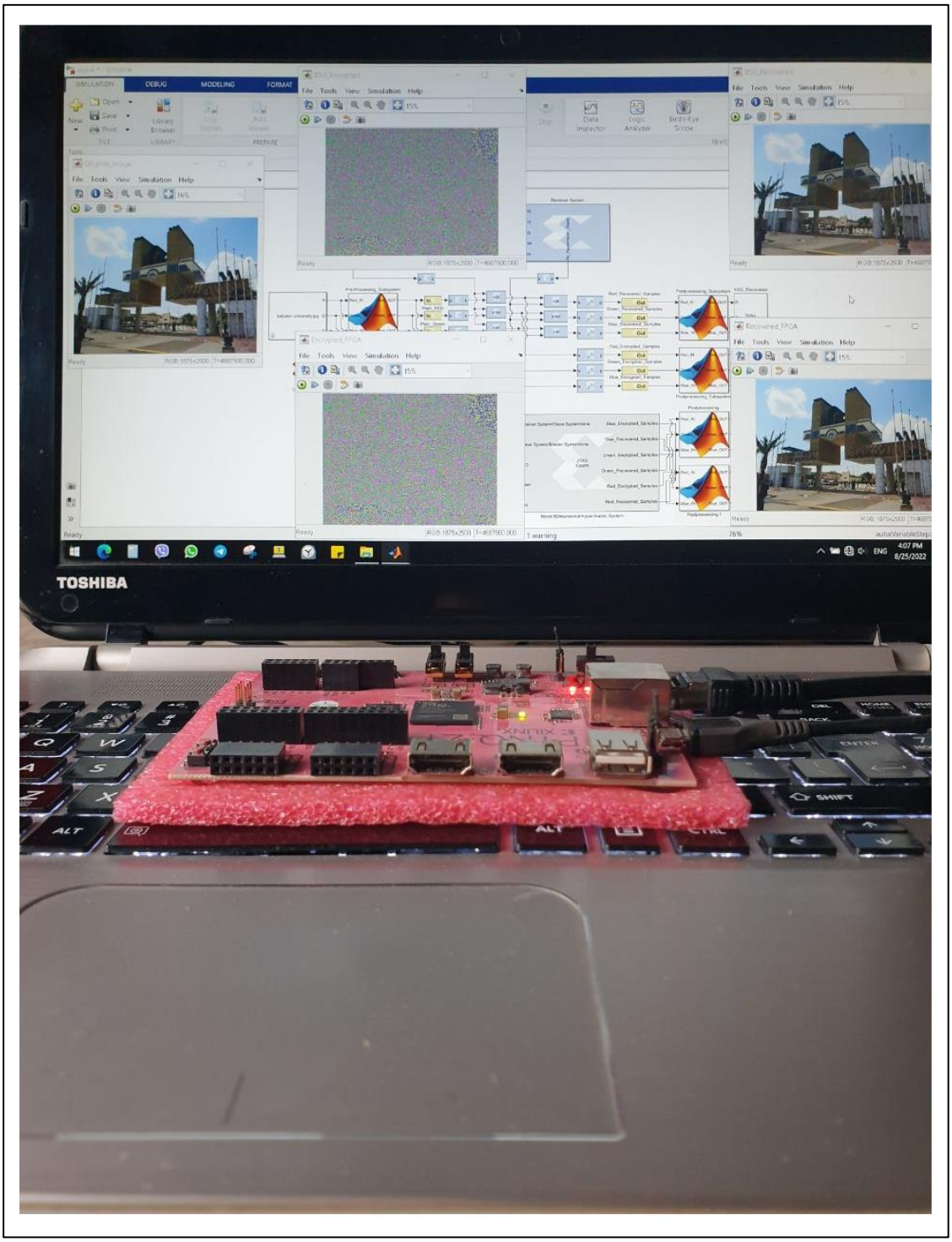

Figure 4-23 Image Encryption Cryptography Hardware Co-Simulation in a Real-Time Environment (Algorithm 4)

4.6. Algorithm (5): New Hyperchaotic Sequence Based on the combination of the 2nd, 3rd, and 4th, Pre-Designed Algorithms.

Figure 4-24 presents the final suggested image encryption scheme, which combines all the other suggested algorithms. This section presents the results of the image encryption and decryption, measurements of the encryption strength, statistical analysis, and FPGA implementation.

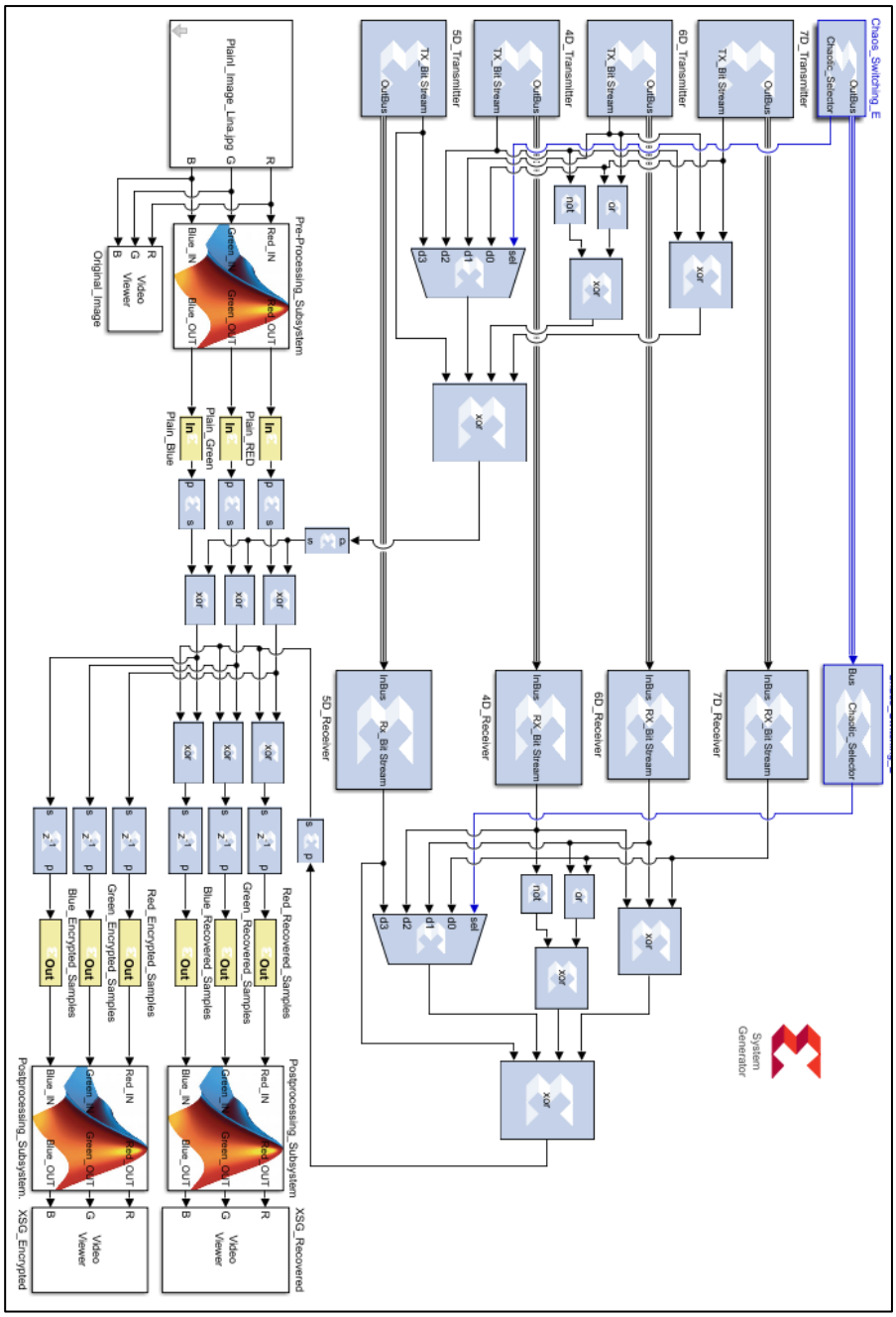

Figure 4-24 Overall Image Encryption System (Algorithm 5)

4.6.1. Image Encryption Strength and Statistical Tests

The final proposed cryptography system's histogram analysis is computed and shown in figure 4-25. Because the cryptosystem can conceal the color distribution in a flat and uniform distribution, as illustrated in the image below, it is obvious that the system can fend off cyberattacks based on color distribution.

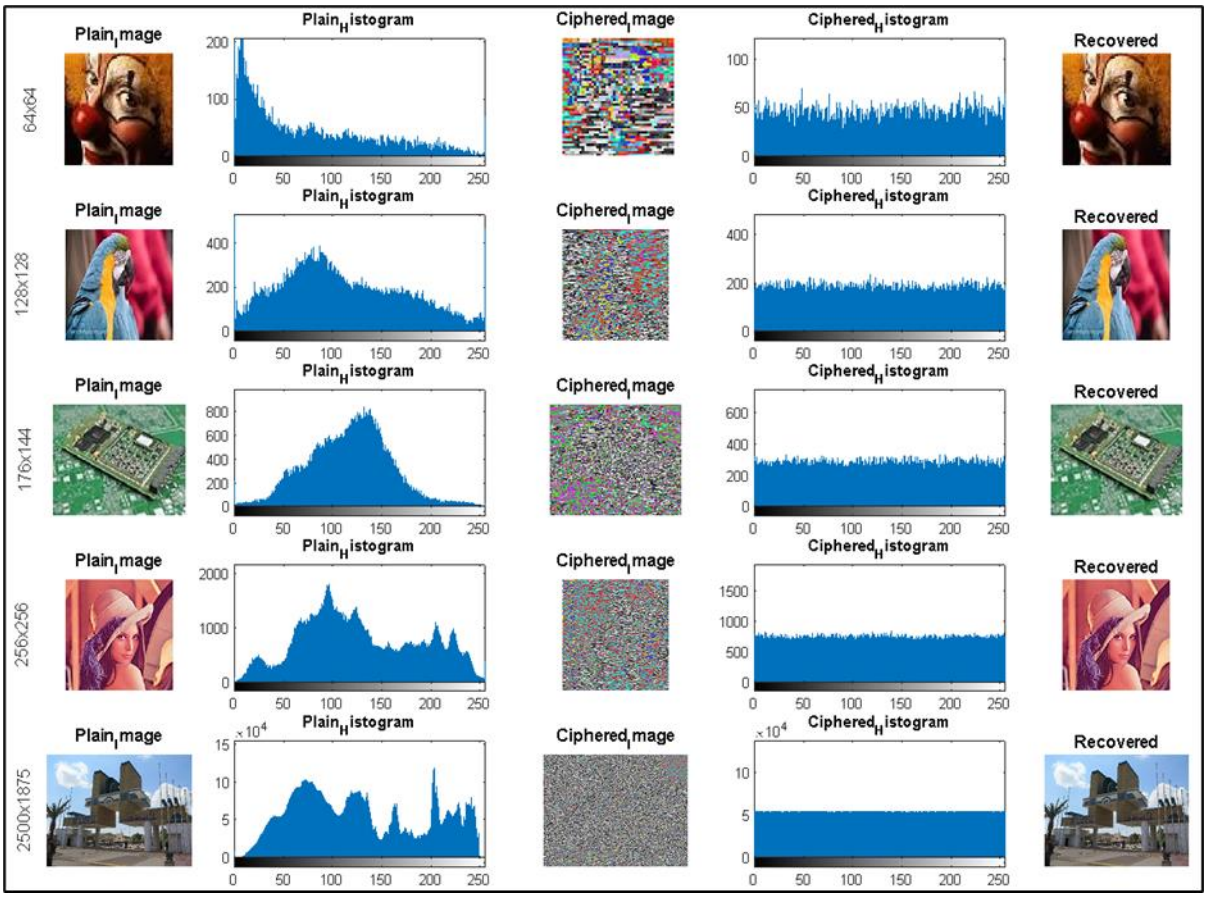

Figure 4-25 Plain Images, Encrypted Images, and Histogram analysis (Algorithm 5)

The mean square error and peak signal to noise ratio are also employed in this method to assess how much the recovered image differs/similar from the plain image, ciphered image. Table 4-12's findings demonstrate how the two images—plain and ciphered—are distinctly different from one another and how this makes the system resilient to cyberattacks.

Table 4-12 PSNR and MSE of the Proposed Cryptosystem (Algorithm 5)

Image Size	Ciphered Image & Plain Image		Recovered Image & Plain Image	
	PSNR	MSE	PSNR	MSE
64 x 64	6.5527	1.6717e+04	Inf	0
128 x 128	9.3393	9.6608e+03	Inf	0
176 x 144	8.7122	7.4659e+03	Inf	0
256 x 256	9.4478	9.5646e+03	Inf	0
2500 x 1875	8.8696	9.7610e+03	Inf	0

The correlation coefficients are among the metrics listed in Table 4-13 that are typically used to evaluate the strength of encryption algorithms. In table 4-13, the estimated correlation between the plain image and the ciphered images (with various sizes) is shown. The cryptography system ought to be able to overcome the strong connection between nearby image pixels. Since the correlation values in table 4-13 are very near to zero, it is obvious that the adjacent pixel correlations have been destroyed and that the attacker is not able to obtain any information that would make the cryptosystem susceptible.

Another type of attack that allows the attacker to learn more about the data source is the entropy attack. The system's entropy level being close to eight means it has a high degree of randomness and exhibits unexpected behavior, but it can also withstand entropy attacks. The proposed cryptosystem's picture entropy is computed and shown in table 4-13 for various image sizes and for three-color layers. All entropy estimates are extremely close to eight, indicating that the cryptosystem has a high resistance to entropy-based assaults.

As shown in table 4-13, the proposed system's NPCR and UACI metrics are very close to 100% and 33%, respectively, and these values indicate that the system is robust enough to withstand differential attacks.

Table 4-13 Correlation, Entropy, NPCR, and UACI Results (Algorithm 5)

Image Size	Color Layer	Correlation	Entropy	NPCR %	UACI %
64x64	Red	0.0053	7.9957	99.44	33.1
	Green	0.0099	7.8761	99.78	33.3
	Blue	0.0022	7.8882	99.39	33.4
128x128	Red	0.0075	7.9533	99.52	33.6
	Green	0.0094	7.8594	99.32	33.4
	Blue	0.0011	7.8481	99.78	33.1
176x144	Red	0.0034	7.9938	99.29	33.5
	Green	-0.0089	7.9323	99.96	33.7
	Blue	-0.0014	7.9621	99.52	33.9
256x256	Red	-0.0069	7.9681	99.31	33.4
	Green	-0.0016	7.9833	99.17	33.7
	Blue	-0.0082	7.9350	99.28	33.3
2500x1875	Red	-0.0113	8.0000	99.78	33.5
	Green	-0.0013	8.0000	99.81	33.9
	Blue	-0.0042	8.0000	99.51	33.4

4.6.2. FPGA Implementation Results and Analysis

The Xilinx System Generator XSG was used to implement the novel combination cryptography system that has been developed utilizing the FPGA PYNQ-Z1 evolution board. The VHDL codes needed to configure and program the board are obtained via the XSG. Due to board resource constraints, Figure 4-26 mainly shows the proposed transmitter's implementation utilizing the FPGA board. As illustrated in figure 4-26, the plain image is summoned from its location in the PC and transferred serially to the board to be encrypted. Once the encryption process is complete, the encrypted samples are sent back to the PC to display the ciphered image. Due to the limitation in the FPGA board resources, the transmitter system is implemented only and a comparison is

presented in figure 4-26 between the board encryption and XSG model encryption which show identical results.

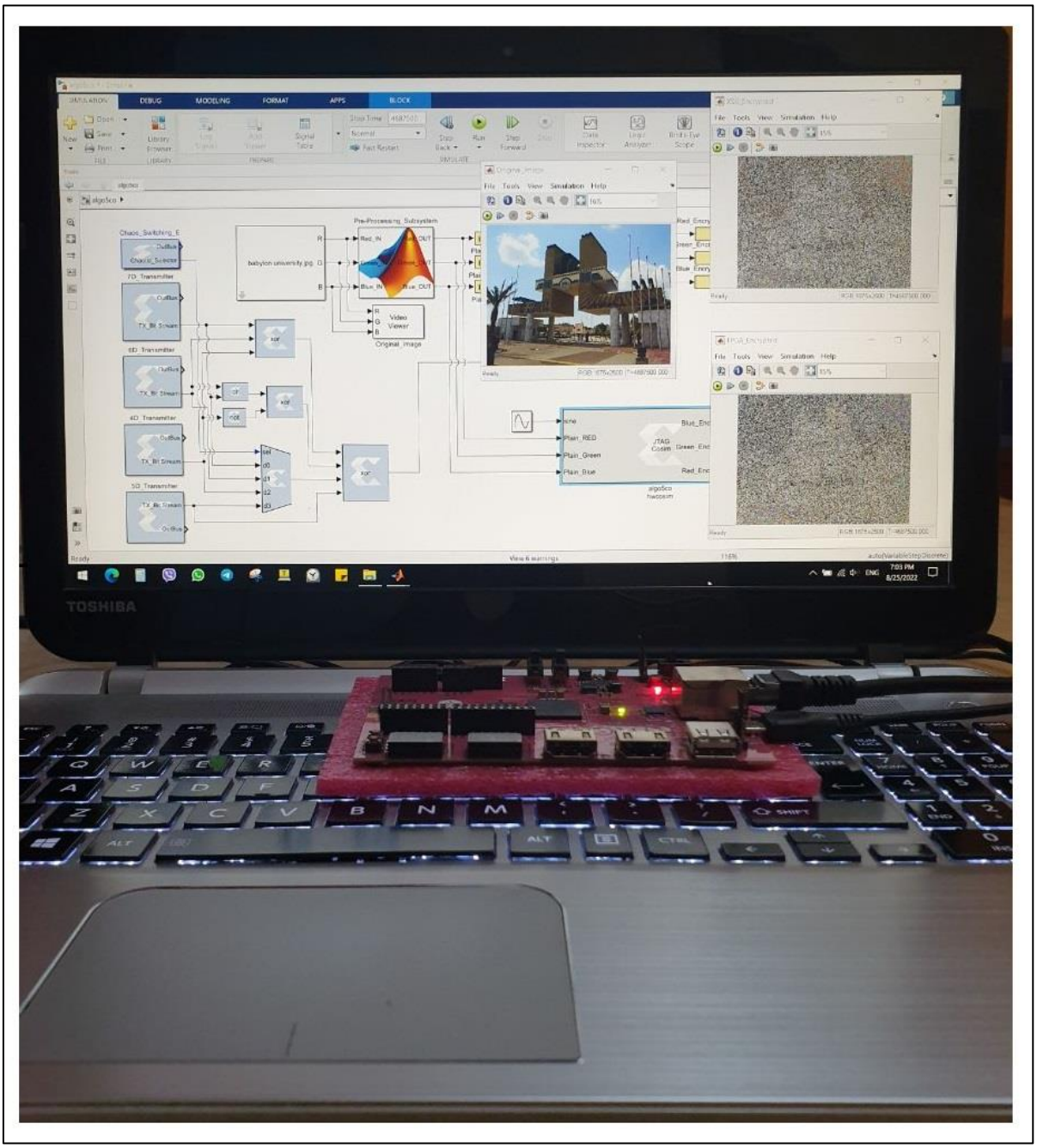

Figure 4-26 Real Time Hardware Co-Simulation of the Image Encryption Transmitter (Algorithm 5)

CHAPTER FIVE

Conclusions and Suggested Future Work

Chapter Five: Conclusions and Suggested Future Work

5.1. Conclusions:

1. The proposed multi-dimensional hyperchaotic system generator by mixture of different hyperchaotic systems together by a means of XOR logical operation could generate a new random binary bit stream to be used for data encryption purposes.
2. The proposed cascaded hyperchaotic system generator that constructed based on chaos switching and three different hyperchaotic systems generate binary sequences that could be employed for image encryption /decryption effectively.
3. The designed five-dimensional hyperchaotic system with a sine wave input as input parameter to the ordinary differential equations has a random and unpredictable behavior which could be utilized to generate a random binary sequence that are used for data encryption. The proposed system shows a significant enhancement in the system performance.
4. The proposed new random binary bit stream generator that designed in the transmitters and receivers of the communication systems by a means of mixing the previous three designed binary generators using XOR operation shows a significant enhancement in the system performance. The designed system provides highly randomness and unpredictable behavior and it is suitable for image encryption/decryption purposes in the communication systems.
5. Hyperchaotic systems are an appropriate candidate for the next generation of the data security field due to their superior characteristics which are suitable for designing robust modern cryptography systems.
6. Higher dimensional hyperchaotic system order provides larger key space, and thus more security level is achieved.

7. Any slight change or perturbations in the hyperchaotic system parameters, or initial conditions (starting point) leads to completely different system behavior.
8. Stream cipher encryption techniques are faster than block cipher which make it suitable for real time processing systems.
9. Cascading the hyperchaotic systems make the encryption results stronger, and makes key space very large to be equal to the multiplication of the two nonlinear hyperchaotic keys and this make the system immune against the brute force attacks.
10. Adaptive feedback controller can effectively provide and ensure the synchronization between any chaotic/ hyperchaotic systems even if they are nonidentical.
11. Using two or more stages of encryption/decryption is better than using one, in terms of encryption strength, security level and key space.

5.2. Suggestions for Future Works

1. Design a hyperchaotic communication system based on nonidentical nonlinear systems.
2. Design a hyperchaotic system with higher dimensional and maximum number of positive lyapunov exponents.
3. Elimination the amount of data by a means of data compression in order to reduce the required FPGA resources.
4. Study the feasibility of implementing the proposed algorithms to a real wireless channel.
5. Study the feasibility of implementing the proposed algorithms to the video and audio data types in terms of strength of encryption and time and others.
6. An efficient tool for chaotic encryption and chaotic secret communications has been developed through the study of hardware realization of hyperchaotic encryption technology.

References

References

[1] K. Rajagopal, L. Guessas, S. Vaidyanathan, A. Karthikeyan, and A. Srinivasan, “Dynamical analysis and FPGA implementation of a novel hyperchaotic system and its synchronization using adaptive sliding mode control and genetically optimized PID control,” *Math. Probl. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/7307452.

[2] R. B. Gandara, G. Wang, and D. N. Utama, “Hybrid Cryptography on Wireless Sensor Network: A Systematic Literature Review,” *Proc. 2018 Int. Conf. Inf. Manag. Technol. ICIMTech 2018*, no. September, pp. 241–245, 2018, doi: 10.1109/ICIMTech.2018.8528147.

[3] P. Lu, S. Sun, and L. Pei, “Improved color image encryption algorithm based on chaotic system,” *ACM Int. Conf. Proceeding Ser.*, vol. 4, no. 11, pp. 168–172, 2009, doi: 10.1145/3034950.3034953.

[4] Y. Tan and W. Zhou, “Image scrambling degree evaluation algorithm based on grey relation analysis,” *Proc. - 2010 Int. Conf. Comput. Inf. Sci. ICCIS 2010*, pp. 511–514, 2010, doi: 10.1109/ICCIS.2010.131.

[5] A. Pande and J. Zambreno, “A chaotic encryption scheme for real-time embedded systems: Design and implementation,” *Telecommun. Syst.*, vol. 52, no. 2, pp. 551–561, 2010, doi: 10.1007/s11235-011-9460-1.

[6] M. Prasad and K. L. Sudha, “Chaos Image Encryption using Pixel shuffling,” pp. 169–179, 2011, doi: 10.5121/csit.2011.1217.

[7] Z. Yong, “Image encryption with logistic map and cheat image,” *ICCRD2011 - 2011 3rd Int. Conf. Comput. Res. Dev.*, vol. 1, pp. 97–101, 2011, doi: 10.1109/ICCRD.2011.5763981.

[8] Z. Tang and X. Zhang, “Secure image encryption without size limitation using Arnold transform and random strategies,” *J. Multimed.*, vol. 6, no. 2, pp. 202–206, 2011, doi: 10.4304/jmm.6.2.202-206.

[9] I. S. I. Abuhaiba, A. Y. AlSallut, H. H. Hejazi, and H. A. AbuGhali, “Cryptography Using Multiple Two-Dimensional Chaotic Maps,” *Int. J. Comput. Netw. Inf. Secur.*, vol. 4, no. 8, pp. 1–7, 2012, doi: 10.5815/ijcnis.2012.08.01.

[10] O. M.AbuZaid, N. A. El-Fishawy, E. M. Nigm, and O. S. Faragallah, “A Proposed Encryption Scheme based on Henon Chaotic System (PESH) for Image Security,” *Int. J. Comput. Appl.*, vol. 61, no. 5, pp. 29–39, 2013, doi: 10.5120/9925-4549.

[11] S. Sadoudi, C. Tanougast, M. S. Azzaz, and A. Dandache, “Design and FPGA implementation of a wireless hyperchaotic communication system for secure real-time image transmission,” *Eurasip J. Image Video Process.*, vol. 2013, pp. 1–18, 2013, doi: 10.1186/1687-5281-2013-43.

[12] G. Hanchinamani and L. Kulakarni, “Image Encryption Based on 2-D Zaslavskii Chaotic Map and Pseudo Hadmard Transform,” *Int. J. Hybrid Inf. Technol.*, vol. 7, no. 4, pp. 185–200, 2014, doi: 10.14257/ijhit.2014.7.4.16.

[13] J. Zhang, “An image encryption scheme based on cat map and hyperchaotic lorenz system,” *Proc. - 2015 IEEE Int. Conf. Comput. Intell. Commun. Technol. CICT 2015*, no. 2, pp. 78–82, 2015, doi: 10.1109/CICT.2015.134.

[14] A. Prof., “Proposed Hyperchaotic System for Image Encryption,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, 2016, doi: 10.14569/ijacsa.2016.070105.

[15] 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.

[16] H. Natiq, N. M. G. Al-Saidi, M. R. M. Said, and A. Kilicman, “A new hyperchaotic map and its application for image encryption,” *Eur. Phys. J.*

Plus, vol. 133, no. 1, 2018, doi: 10.1140/epjp/i2018-11834-2.

[17] 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.

[18] 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.

[19] M. A. Al-Khasawneh, S. M. Shamsuddin, S. Hasan, and A. A. Bakar, "An Improved Chaotic Image Encryption Algorithm," *2018 Int. Conf. Smart Comput. Electron. Enterp. ICSCEE 2018*, pp. 1–8, 2018, doi: 10.1109/ICSCEE.2018.8538373.

[20] G. Cheng, C. Wang, and H. Chen, "A Novel Color Image Encryption Algorithm Based on Hyperchaotic System and Permutation-Diffusion Architecture," *Int. J. Bifurc. Chaos*, vol. 29, no. 9, 2019, doi: 10.1142/S0218127419501153.

[21] X. Zhang, L. Wang, Z. Zhou, and Y. Niu, "A Chaos-Based Image Encryption Technique Utilizing Hilbert Curves and H-Fractals," *IEEE Access*, vol. 7, pp. 74734–74746, 2019, doi: 10.1109/ACCESS.2019.2921309.

[22] H. A. Abdullah and H. N. Abdullah, "FPGA implementation of color image encryption using a new chaotic map," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 13, no. 1, pp. 129–137, 2019, doi: 10.11591/ijeecs.v13.i1.pp129-137.

[23] T. M. Hoang, "A Chaos-based Image Cryptosystem Using Nonstationary Dynamics of Logistic Map," *ICTC 2019 - 10th Int. Conf. ICT Converg. ICT Converg. Lead. Auton. Futur.*, pp. 591–596, 2019, doi:

10.1109/ICTC46691.2019.8939826.

[24] “FPGA Hardware Co-Simulation of Image Encryption Using Stream Cipher Based on Chaotic Map.pdf.” .

[25] J. Wu, J. Shi, and T. Li, “A Novel Image Encryption Approach Based on a Hyperchaotic System, Pixel-Level Filtering with Variable Kernels, and DNA-Level Diffusion,” *Entropy*, vol. 22, no. 1, p. 5, 2019, doi: 10.3390/e22010005.

[26] H. Liu, Y. Zhang, A. Kadir, and Y. Xu, “Image encryption using complex hyper chaotic system by injecting impulse into parameters,” *Appl. Math. Comput.*, vol. 360, pp. 83–93, 2019, doi: 10.1016/j.amc.2019.04.078.

[27] S. Zhu and C. Zhu, “Plaintext-Related Image Encryption Algorithm Based on Block Structure and Five-Dimensional Chaotic Map,” *IEEE Access*, vol. 7, pp. 147106–147118, 2019, doi: 10.1109/ACCESS.2019.2946208.

[28] M. Hanif, R. A. Naqvi, S. Abbas, M. A. Khan, and N. Iqbal, “A Novel and Efficient 3D Multiple Images Encryption Scheme Based on Chaotic Systems and Swapping Operations,” *IEEE Access*, vol. 8, pp. 123536–123555, 2020, doi: 10.1109/ACCESS.2020.3004536.

[29] I. Yasser, A. T. Khalil, M. A. Mohamed, A. S. Samra, and F. Khalifa, “A Robust Chaos-Based Technique for Medical Image Encryption,” *IEEE Access*, vol. 10, pp. 244–257, 2022, doi: 10.1109/ACCESS.2021.3138718.

[30] J. Arif *et al.*, “A Novel Chaotic Permutation-Substitution Image Encryption Scheme Based on Logistic Map and Random Substitution,” *IEEE Access*, vol. 10, pp. 12966–12982, 2022, doi: 10.1109/ACCESS.2022.3146792.

[31] C. Qiuqiong, D. Yao, and N. Zhiyong, “An Image Encryption Algorithm Based on Combination of Chaos and DNA Encoding,” *Proc. - 2020 Int.*

Conf. Comput. Vision, Image Deep Learn. CVIDL 2020, vol. 10, no. 3, pp. 182–185, 2020, doi: 10.1109/CVIDL51233.2020.00043.

[32] Y. Wang, X. Li, X. Li, Y. Guang, Y. Wu, and Q. Ding, "FPGA-Based Implementation and Synchronization Design of a New Five-Dimensional Hyperchaotic System," 2022.

[33] W. Stallings, *Cryptography and Network Security: Principles and Practices*. UK, 2014.

[34] A. Kahate, "Cryptography and Network Security." p. 535, 2008.

[35] A. Mostafa, N. F. Soliman, M. Abdalluh, and F. E. Abd El-Samie, "Speech encryption using two dimensional chaotic maps," *2015 11th Int. Comput. Eng. Conf. Today Inf. Soc. What's Next?, ICENCO 2015*, no. April 2016, pp. 235–240, 2016, doi: 10.1109/ICENCO.2015.7416354.

[36] S. Chandra, S. Bhattacharyya, S. Paira, and S. S. Alam, "A study and analysis on symmetric cryptography," *2014 Int. Conf. Sci. Eng. Manag. Res. ICSEMR 2014*, 2014, doi: 10.1109/ICSEMR.2014.7043664.

[37] 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.

[38] E. Swathi, G. Vivek, and G. S. Rani, "Role of Hash Function in Cryptography," vol. 6495, pp. 10–13, 2016, doi: 10.22161/ijaers/si.3.

[39] L. Huang, Z. Zhang, J. Xiang, and S. Wang, "A New 4D Chaotic System with Two-Wing, Four-Wing, and Coexisting Attractors and Its Circuit Simulation," *Complexity*, vol. 2019, no. i, 2019, doi: 10.1155/2019/5803506.

[40] A. M. Suhail, A. Vyas, M. Gudivada, P. T. Venkat, and N. Rao, "NEW CRYPTOGRAPHIC TECHNIQUE FOR," vol. 6, no. 4, pp. 449–455, 2015.

[41] *Cryptography and Security in Computing*, no. November 2016. 2012.

[42] P. Rakheja, R. Vig, and P. Singh, “An asymmetric watermarking scheme based on random decomposition in hybrid multi-resolution wavelet domain using 3D Lorenz chaotic system,” *Optik (Stuttg).*, vol. 198, 2019, doi: 10.1016/j.ijleo.2019.163289.

[43] *Number Theory and Modern*. 2010.

[44] C. E. Shannon, “Communication Theory of Secrecy Systems,” *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.

[45] W. Abdul, O. Nafea, and S. Ghouzali, “Combining watermarking and hyper-chaotic map to enhance the security of stored biometric templates,” *Comput. J.*, vol. 63, no. 3, pp. 479–493, 2020, doi: 10.1093/comjnl/bxz047.

[46] 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.

[47] 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.

[48] 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.

[49] . B. M., “a Multilevel Security Scheme Using Chaos Based Encryption and Steganography for Secure Audio Communication,” *Int. J. Res. Eng. Technol.*, vol. 02, no. 10, pp. 399–403, 2013, doi: 10.15623/ijret.2013.0210061.

[50] F. Mo, Y. C. Hsu, H. H. Chang, S. C. Pan, J. J. Yan, and T. L. Liao,

"Design of an improved RSA cryptosystem based on synchronization of discrete chaotic systems," *Proc. - 2016 Int. Conf. Inf. Syst. Artif. Intell. ISAI 2016*, no. 1, pp. 9–13, 2017, doi: 10.1109/ISAI.2016.0012.

[51] 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.

[52] D. Rontani and D. Rontani, "Nonlinear dynamics of photonic components . Chaos cryptography and multiplexing To cite this version : HAL Id : tel-00783267 par Dynamique Non-Linéaire de Composants Photoniques Cryptographie par Chaos et Multiplexage," 2013.

[53] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," vol. 64, no. 8, pp. 821–825, 1990, [Online]. Available: http://scitation.aip.org/docserver/fulltext/aip/journal/chaos/7/4/1.166278.pdf?expires=1405959266&id=id&accname=2101255&checksum=1B971CA8B24F6968F1A524031940752D.

[54] S. H. Strogatz and A. V. Oppenheim, "Synchronization of Lorenz-Based Chaotic Circuits with Applications to Communications," *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 40, no. 10, pp. 626–633, 1993, doi: 10.1109/82.246163.

[55] P. Prakash *et al.*, "A Novel Simple 4-D Hyperchaotic System with a Saddle-Point Index-2 Equilibrium Point and Multistability : Design and FPGA-Based Applications," *Circuits, Syst. Signal Process.*, 2020, doi: 10.1007/s00034-020-01367-0.

[56] M. P. Kennedy and M. Hasler, "Chaos Shift Keying: Modulation and Demodulation of a Chaotic Carrier Using Self-Synchronizing Chua's Circuits," *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*,

vol. 40, no. 10, pp. 634–642, 1993, doi: 10.1109/82.246164.

[57] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, “Determining Lyapunov exponents from a time series,” *Phys. D Nonlinear Phenom.*, vol. 16, no. 3, pp. 285–317, 1985, doi: 10.1016/0167-2789(85)90011-9.

[58] L. Vinet and A. Zhedanov, *Synchronization Techniques for Chaotic Communication Systems*, vol. 44, no. 8. 2011.

[59] Lorenz E.N., “Deterministic nonperiodic flow,” *J. Atmos. Sci.*, vol. 20, no. 1963, pp. 130–141, 2006, [Online]. Available: https://doi.org/10.1175/1520-0469(1963)020%3C0130:DNF%3E2.0.CO;2.

[60] O. A. Gonzales, G. Han, J. P. De Gyvez, and E. Sânchez-Sinencio, “Lorenz-based chaotic cryptosystem: a monolithic implementation,” *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 47, no. 8, pp. 1243–1247, 2000, doi: 10.1109/81.873879.

[61] O. E. Rössler, “An equation for continuous chaos,” *Phys. Lett. A*, vol. 57, no. 5, pp. 397–398, 1976, doi: 10.1016/0375-9601(76)90101-8.

[62] L. O. Chua, “Canonical Realization of Chua’s Circuit.” pp. 885–882, 1990.

[63] A. M. Rucklidge, “Chaos in models of double convection,” *J. Fluid Mech.*, vol. 237, no. 209, pp. 209–229, 1992, doi: 10.1017/S0022112092003392.

[64] H. H. Nien, C. K. Huang, S. K. Changchien, H. W. Shieh, C. T. Chen, and Y. Y. Tuan, “Digital color image encoding and decoding using a novel chaotic random generator,” *Chaos, Solitons and Fractals*, vol. 32, no. 3, pp. 1070–1080, 2007, doi: 10.1016/j.chaos.2005.11.057.

[65] A. M. Elshamy, A. I. Hussein, and A. Q. Alhamad, “Secure Implementation for Video Streams Based on Fully and Permutation Encryption Techniques.”

[66] 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.

[67] B. Balakrishnan and D. M. N. Mubarak, “An improved image encryption using 2D logistic adjusted sine chaotic map with shuffled index matrix,” *Proc. 2021 1st Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. ICAECT 2021*, 2021, doi: 10.1109/ICAECT49130.2021.9392392.

[68] 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.

[69] J. Sen Teh, M. Alawida, and Y. C. Sii, “Implementation and practical problems of chaos-based cryptography revisited,” *J. Inf. Secur. Appl.*, vol. 50, no. August 2019, 2020, doi: 10.1016/j.jisa.2019.102421.

[70] O. E. Rossler, “An equation for hyperchaos,” *Phys. Lett. A*, vol. 71, no. 2–3, pp. 155–157, 1979, doi: 10.1016/0375-9601(79)90150-6.

[71] 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.

[72] Z. Q. Chen, G. Q. Liang, and J. G. Tong, “The FPGA implementation of hyperchaotic system based upon VHDL design,” *Proc. - 4th Int. Work. Chaos-Fractals Theor. Appl. IWCFTA 2011*, no. 3, pp. 47–51, 2011, doi: 10.1109/IWCFTA.2011.82.

[73] 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.

[74] Z. Liu, “Chaotic time series analysis,” *Math. Probl. Eng.*, vol. 2010, 2010, doi: 10.1155/2010/720190.

[75] A. Rukhin, J. Soto, and J. Nechvatal, “Nistspecialpublication800-22R1a.Pdf,” *Nist Spec. Publ.*, vol. 22, no. April, pp. 1-1-G-1, 2010, [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf.

[76] T. Bonny, “Chaotic or Hyper-chaotic Oscillator? Numerical Solution, Circuit Design, MATLAB HDL-Coder Implementation, VHDL Code, Security Analysis, and FPGA Realization,” *Circuits, Syst. Signal Process.*, vol. 40, no. 3, pp. 1061–1088, 2021, doi: 10.1007/s00034-020-01521-8.

[77] H. G. Bhaskar, S. Rohith, and M. R. Mahesh, “Two level image encryption scheme using arnold map and combined key sequence of logistic map and Tent map,” *IFIP Int. Conf. Wirel. Opt. Commun. Networks, WOCN*, pp. 0–4, 2017, doi: 10.1109/WOCN.2015.8064518.

[78] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, “A novel image encryption scheme using the composite discrete chaotic system,” *Entropy*, vol. 18, no. 8, pp. 1–27, 2016, doi: 10.3390/e18080276.

[79] C. Wang, Z. Chen, and T. Li, “Blind Evaluation of Image Scrambling Degree based on the Correlation of Adjacent Pixels,” *TELKOMNIKA Indones. J. Electr. Eng.*, vol. 11, no. 11, pp. 6556–6562, 2013, doi: 10.11591/telkomnika.v11i11.3496.

[80] Y. Wu, J. P. Noonan, and S. Agaian, “NPCR and UACI Randomness Tests for Image Encryption,” *Cyberjournals.Com*, 2011, [Online]. Available: http://www.cyberjournals.com/Papers/Apr2011/05.pdf.

[81] T. Matsumoto, “A Chaotic Attractor from Chua’s Circuit,” *IEEE Trans. Circuits Syst.*, vol. 31, no. 12, pp. 1055–1058, 1984, doi: 10.1109/TCS.1984.1085459.

[82] Q. Yang, D. Zhu, and L. Yang, “A New 7D Hyperchaotic System with Five Positive Lyapunov Exponents Coined,” *Int. J. Bifurc. Chaos*, vol. 28, no. 5, pp. 1–20, 2018, doi: 10.1142/S0218127418500578.

[83] W. Yu *et al.*, “Design of a new seven-dimensional hyperchaotic circuit and its application in secure communication,” *IEEE Access*, vol. 7, pp. 125586–125608, 2017, doi: 10.1109/ACCESS.2019.2935751.

[84] 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.

[85] 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.

[86] 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.

Appendix A / Matlab Codes

Lorenz Chaotic Flow System

```
function Lorenz_System ()
    x0 = [10 20 30]; tspan = [0 50];
    [t,x] = ode45(@lorenz, tspan, x0);
    Tiledlayout (3,1)
    ax1 = nexttile;             plot(ax1, t,x(:,1),'k'), hold on
    legend('X')
    xlabel('Time')
    ylabel('X(t)')
    grid on
    grid minor
    ax2 = nexttile;             plot(ax2,t,x(:,2),'k'), hold on
    legend('Y')
    xlabel('Time')
    ylabel('Y(t)')
    grid on
    grid minor
    ax3 = nexttile;             plot(ax3,t,x(:,3),'k'), hold on
    legend('Z')
    xlabel('Time')
    ylabel('Z(t)')
    grid on
    grid minor
    figure
    plot(x(:,1),x(:,2),'k'), hold on
    xlabel('X(t)')
    ylabel('Y(t)')
    grid on
    grid minor
    figure
    plot3(x(:,1), x(:,2), x(:,3),'k'), hold off
    xlabel('X(t)')
    ylabel('Y(t)')
    zlabel('Z(t)')
    grid on
    grid minor
end
function xprime = lorenz(t,x)
```

```
        sig    = 10;          beta   = 8/3; rho    = 28;
        xprime = [-sig*x(1) + sig*x(2);          rho*x(1) - x(2) - x(1)*x(3);
        -beta*x(3) + x(1)*x(2)];
end
```

Rossler Chaotic Flow System

```
function Rossler_System ()
    clc; clear;
    x0 = [10 20 30]; tspan = [0 50];
    [t,x] = ode45(@rossler, tspan, x0);
    tiledlayout(3,1)
    ax1 = nexttile;          plot(ax1, t,x(:,1),'k'), hold on
    legend('X')
    xlabel('Time')
    ylabel('X(t)')
    grid on
    grid minor
    ax2 = nexttile;          plot(ax2,t,x(:,2),'k'), hold on
    legend('Y')
    xlabel('Time')
    ylabel('Y(t)')
    grid on
    grid minor
    ax3 = nexttile;          plot(ax3,t,x(:,3),'k'), hold on
    legend('Z')
    xlabel('Time')
    ylabel('Z(t)')
    grid on
    grid minor
    figure
    plot(x(:,1),x(:,2),'k'), hold on
    xlabel('X(t)')
    ylabel('Y(t)')
    grid on
    grid minor
    figure
    plot3(x(:,1), x(:,2), x(:,3),'k'), hold off
    xlabel('X(t)')
    ylabel('Y(t)')
    zlabel('Z(t)')
    grid on
```

```
    grid minor
end
function xprime = rossler(t,x)
                a    = 0.2;     b   = 0.2;              c    = 5.7;
                xprime = [-(x(2) + x(3)); x(1) + a*x(2);       b+(x(1)-c)*x(3)];
end
```

Chua Chaotic Flow System

```
function Chua_System()
    clc; clear;
    x0 = [0.7 0 0];             tspan = [0 100];
    [t,x] = ode45(@chua, tspan, x0);
    tiledlayout(3,1)
    ax1 = nexttile;             plot(ax1, t,x(:,1),'k'), hold on
    legend('X')
    xlabel('Time')
    ylabel('X(t)')
    grid on
    grid minor
    ax2 = nexttile;             plot(ax2,t,x(:,2),'k'), hold on
    legend('Y')
    xlabel('Time')
    ylabel('Y(t)')
    grid on
    grid minor
    ax3 = nexttile;             plot(ax3,t,x(:,3),'k'), hold on
    legend('Z')
    xlabel('Time')
    ylabel('Z(t)')
    grid on
    grid minor
    figure
    plot(x(:,1),x(:,2),'k'), hold on
    xlabel('X(t)')
    ylabel('Y(t)')
    grid on
    grid minor
    figure
    plot3(x(:,1), x(:,2), x(:,3),'k'), hold off
    xlabel('X(t)')
    ylabel('Y(t)')
```

```
    zlabel('Z(t)')
    grid on
    grid minor
end
function xprime = chua(t,x)
    a =10;      b= 14.78;     c= 0.0385;          m0=-1.27;          m1=-0.68;
    h = m1*x(1)+0.5*(m0-m1)*(abs(x(1)+1)-abs(x(1)-1));
    xprime = [a*(x(2)-x(1)-h);     x(1) - x(2)+ x(3);    -b*x(2)];
end
```

Rucklidge Chaotic Flow System

```
function Rucklidge_System()
    clc; clear;
    x0 = [1 0 4.5];             tspan = [0 100];
    [t,x] = ode45(@Rucklidge, tspan, x0);
    tiledlayout(3,1)
    ax1 = nexttile;             plot(ax1, t,x(:,1),'k'), hold on
    legend('X')
    xlabel('Time')
    ylabel('X(t)')
    grid on
    grid minor
    ax2 = nexttile;             plot(ax2,t,x(:,2),'k'), hold on
    legend('Y')
    xlabel('Time')
    ylabel('Y(t)')
    grid on
    grid minor
    ax3 = nexttile;             plot(ax3,t,x(:,3),'k'), hold on
    legend('Z')
    xlabel('Time')
    ylabel('Z(t)')
    grid on
    grid minor
    figure
    plot(x(:,1),x(:,2),'k'), hold on
    xlabel('X(t)')
    ylabel('Y(t)')
    grid on
    grid minor
    figure
```

```
    plot3(x(:,1), x(:,2), x(:,3),'k'), hold off
    xlabel('X(t)')
    ylabel('Y(t)')
    zlabel('Z(t)')
    grid on
    grid minor
end
function xprime = Rucklidge(t,x)
        k    = 2.1;     L   =6.7;
        xprime = [-k*x(1)+L*x(2)-x(2)*x(3);          x(1);   -x(3)+x(2)*x(2)];
end
```

Nien Chaotic Flow System

```
function Nien_System()
    clc; clear;
    x0 = [0.05 0.03 0.13];          tspan = [0 100];
    [t,x] = ode45(@Nien, tspan, x0);
    tiledlayout(3,1)
    ax1 = nexttile;          plot(ax1, t,x(:,1),'k'), hold on
    legend('X')
    xlabel('Time')
    ylabel('X(t)')
    grid on
    grid minor
    ax2 = nexttile;          plot(ax2,t,x(:,2),'k'), hold on
    legend('Y')
    xlabel('Time')
    ylabel('Y(t)')
    grid on
    grid minor
    ax3 = nexttile;          plot(ax3,t,x(:,3),'k'), hold on
    legend('Z')
    xlabel('Time')
    ylabel('Z(t)')
    grid on
    grid minor
    figure
    plot(x(:,1),x(:,2),'k'), hold on
    xlabel('X(t)')
    ylabel('Y(t)')
    grid on
```

```
    grid minor
    figure
    plot3(x(:,1), x(:,2), x(:,3),'k'), hold off
    xlabel('X(t)')
    ylabel('Y(t)')
    zlabel('Z(t)')
    grid on
    grid minor
end
function xprime = Nien(t,x)
    a =-1.143;b= -0.714;    I0=3; alpha=6.3;    beta=0.7;      yao=7;
    h = b*x(1)+0.5*(a-b)*(abs(x(1)+I0)-abs(x(1)-I0));
    xprime = [-alpha*(x(1)+x(2)+h);     -beta*(x(1)+x(2))-yao*x(3);       x(2)];
end
```

Hénon Chaotic Map

```
clc; clear all; close all
x(1)=0; y(1)=0; a=1.4; b=0.3;
for i=2:10000
    x(i)=1-1.4*(x(i-1)^2)+y(i-1); y(i)=b*x(i-1);
end
figure
plot(x,y,'.','MarkerSize',1.5, 'Color',[0 0 0])
xlabel ('x')
ylabel ('y')
title('Henon Map')
```

Logistic Chaotic Map

```
clc ; clear all; close all
 xvals=[];
   for beta = 0:0.01:4
    beta;
    xold = 0.5;
        for i=1:2000
       xnew=((xold-xold^2)*beta);                xold=xnew;
       end
xss=xnew;
       for i=1:1000
          xnew=((xold-xold^2)*beta);    xold=xnew;
          xvals(1,length(xvals)+1)=beta;xvals(2,length(xvals))=xnew;
          if(abs(xnew-xss)<.001)
          break
          end
```

```
      end
   end
plot(xvals(1,:), xvals(2,:), '.', 'LineWidth', .1, 'MarkerSize',1.2,...
'Color',[0 0 0])
set(gca, 'color', 'w', 'xcolor', 'k', 'ycolor', 'k')
set(gcf, 'color', 'w')
xlabel('Beta')
ylabel('X(k)')
title('Logistic Map')
```

Chaotic/Hyperchaotic System Behavior Tests:

Lyapunov Exponents Test

```
[T,Lyap]=lyapunov(3,@lorenz_ext,@ode45,0,0.5,1000,[0 1 0],10);
plot(T, Lyap);
title('Dynamics of Lyapunov exponents');
xlabel('Time'); ylabel('Lyapunov exponents');

function [Texp,Lexp]= lyapunov
(n,rhs_ext_fcn,fcn_integrator,tstart,stept,tend,ystart,ioutp);
n1=n; n2=n1*(n1+1);
nit = round((tend-tstart)/stept);
y=zeros(n2,1); cum=zeros(n1,1); y0=y;
gsc=cum; znorm=cum;
y(1:n)=ystart(:);
for i=1:n1 y((n1+1)*i)=1.0; end;
t=tstart;
for ITERLYAP=1:nit
  [T,Y] = feval(fcn_integrator,rhs_ext_fcn,[t t+stept],y);
   t=t+stept;
  y=Y(size(Y,1),:);
  for i=1:n1
     for j=1:n1 y0(n1*i+j)=y(n1*j+i); end;
  end;
  znorm(1)=0.0;
  for j=1:n1 znorm(1)=znorm(1)+y0(n1*j+1)^2; end;
  znorm(1)=sqrt(znorm(1));
  for j=1:n1 y0(n1*j+1)=y0(n1*j+1)/znorm(1); end;
  for j=2:n1
     for k=1:(j-1)
        gsc(k)=0.0;
        for l=1:n1 gsc(k)=gsc(k)+y0(n1*l+j)*y0(n1*l+k); end;
```

```
    end;
     for k=1:n1
       for l=1:(j-1)
          y0(n1*k+j)=y0(n1*k+j)-gsc(l)*y0(n1*k+l);
       end;
    end;
    znorm(j)=0.0;
    for k=1:n1 znorm(j)=znorm(j)+y0(n1*k+j)^2; end;
    znorm(j)=sqrt(znorm(j));
    for k=1:n1 y0(n1*k+j)=y0(n1*k+j)/znorm(j); end;
  end;
  for k=1:n1 cum(k)=cum(k)+log(znorm(k)); end;
  for k=1:n1
     lp(k)=cum(k)/(t-tstart);
  end;
  if ITERLYAP==1
    Lexp=lp;
    Texp=t;
  else
    Lexp=[Lexp; lp];
    Texp=[Texp; t];
  end;
  if (mod(ITERLYAP,ioutp)==0)
    fprintf('t=%6.4f',t);
    for k=1:n1 fprintf(' %10.6f',lp(k)); end;
    fprintf('\n');
  end;
  for i=1:n1
     for j=1:n1
        y(n1*j+i)=y0(n1*i+j);
     end;
  end;
end;
function f=lorenz_ext(t,X)
 SIGMA = 10; R = 30; BETA = 8/3;
x=X(1); y=X(2); z=X(3);
Y= [X(4), X(7), X(10);
   X(5), X(8), X(11);
   X(6), X(9), X(12)];
f=zeros(9,1);
f(1)=SIGMA*(y-x);
f(2)=-x*z+R*x-y;
```

```
f(3)=x*y-BETA*z;
 Jac=[-SIGMA, SIGMA,     0;
        R-z,    -1,    -x;
          y,     x, -BETA];
f(4:12)=Jac*Y;
```

Encryption Algorithm Strength Tests (Histogram, Entropy, MSE, PSNR, and Correlation) & Statistical Analysis Tests (NPCR & UACI)

```
clc; clear all;

        %%%%%% Calling the First Image 64 x 64%%%%%%

P1 = imread ('Plain text image path in PC/Plaintext image name.jpg');
C1 = imread ('ciphertext image path in PC/Ciphertext image name.jpg');
P1r=imread (' Recovered image path in PC/Recovered image name.jpg');
subplot(5,5,1),imshow(p_64)
ylabel('64x64')
title('Plain_Image')
subplot(5,5,2),imhist(p1) %%% Compute image histogram %%%
title('Plain_Histogram')
subplot(4,5,3),imshow(c1)
title('Ciphered_Image')
subplot(5,5,4),imhist(c1) %%% Compute image histogram %%%
title('Ciphered_Histogram')
subplot(5,5,5),imshow(p1)
title('Recovered')

        %%%%%% Calling the Second Image 128 x 128 %%%%%%

P2 = imread ('Plain text image path in PC/Plaintext image name.jpg');
C2 = imread ('ciphertext image path in PC/Ciphertext image name.jpg');
P2r=imread (' Recovered image path in PC/Recovered image name.jpg');
subplot(5,5,6),imshow(p2)
ylabel('128x128')
title('Plain_Image')
subplot(5,5,7),imhist(p2) %%% Compute image histogram %%%
title('Plain_Histogram')
subplot(5,5,8),imshow(c2)
title('Ciphered_Image')
subplot(5,5,9),imhist(c2) %%% Compute image histogram %%%
```

```
title('Ciphered_Histogram')
subplot(5,5,10),imshow(p2r)
title('Recovered')

        %%%%%% Calling the Third Image 176 x 144 %%%%%%

P3 = imread ('Plain text image path in PC/Plaintext image name.jpg');
C3 = imread ('ciphertext image path in PC/Ciphertext image name.jpg');
P3r=imread (' Recovered image path in PC/Recovered image name.jpg');
subplot(5,5,11),imshow(p3)
ylabel('176x144')
title('Plain_Image')
subplot(5,5,12),imhist(p3)        %%% Compute image histogram %%%
title('Plain_Histogram')
subplot(5,5,13),imshow(c3)
title('Ciphered_Image')
subplot(5,5,14),imhist(c3)        %%% Compute image histogram %%%
title('Ciphered_Histogram')
subplot(5,5,15),imshow(P3r)
title('Recovered')

        %%%%%% Calling the Fourth Image 256 x 256 %%%%%%

P4 = imread ('Plain text image path in PC/Plaintext image name.jpg');
C4 = imread ('ciphertext image path in PC/Ciphertext image name.jpg');
P4r=imread (' Recovered image path in PC/Recovered image name.jpg');
subplot(5,5,16),imshow(p4)
ylabel('256x256')
title('Plain_Image')
subplot(5,5,17),imhist(p4)        %%% Compute image histogram %%%
title('Plain_Histogram')
subplot(5,5,18),imshow(c4)
title('Ciphered_Image')
subplot(5,5,19),imhist(c4)        %%% Compute image histogram %%%
title('Ciphered_Histogram')
subplot(5,5,20),imshow(p4r)
title('Recovered')

        %%%%%% Calling the Fifth Image 2500 x 1875 %%%%%%

P5 = imread ('Plain text image path in PC/Plaintext image name.jpg');
C5 = imread ('ciphertext image path in PC/Ciphertext image name.jpg');
```

```
P5r=imread (' Recovered image path in PC/Recovered image name.jpg');

subplot(5,5,21),imshow(p5)
ylabel('2500x1875')
title('Plain_Image')
subplot(5,5,22),imhist(p5)        %%% Compute image histogram %%%
title('Plain_Histogram')
subplot(5,5,23),imshow(c5)
title('Ciphered_Image')
subplot(5,5,24),imhist(c5)        %%% Compute image histogram %%%
title('Ciphered_Histogram')
subplot(5,5,25),imshow(p5r)
title('Recovered')

            %%%%%Image 1 PSNR and MSE Calculations %%%%%

psnr_c1 = psnr(c1,p1)
mse_c1 = immse(c1,p1)
psnr_r1 = psnr(p1,p1r)
mse_r1 = immse(p1,p1r)

            %%%%%Image 2 PSNR and MSE Calculations %%%%%

psnr_c2 = psnr(c2,p2)
mse_c2 = immse(c2,p2)
psnr_r2 = psnr(p1,p1r)
mse_r2 = immse(p2,p2r)

            %%%%%Image 3 PSNR and MSE Calculations %%%%%

psnr_c3 = psnr(c3,p3)
mse_c3 = immse(c3,p3)
psnr_r3 = psnr(p3,p3r)
mse_r3 = immse(p3,p3r)

            %%%%%Image 4 PSNR and MSE Calculations %%%%%

psnr_c4 = psnr(c4,p4)
mse_c4 = immse(c4,p4)
psnr_r4 = psnr(p4,p4)
mse_r4 = immse(p4,p4)
```

```
%%%%%Image 5 PSNR and MSE Calculations %%%%%

psnr_c5 = psnr(c5,p5)
mse_c5 = immse(c5,p5)
psnr_r5 = psnr(p5,p5)
mse_r5 = immse(p4,p5)

%%%%%Image 1 Correlation Calculations %%%%%

P1_red = p1(:,:,1);
P1_green = p1(:,:,2);
P1_blue = p1(:,:,3);
C1_red= c1(:,:,1);
C1_green= c1(:,:,2);
C1_blue = c1(:,:,3);
R64_red = corr2(c1_red,p1_red)
R64_green = corr2(c1_green,p1_green)
R64_blue = corr2(c1_blue,p1_blue)

%%%%%Image 2 Correlation Calculations %%%%%

P2_red = p2(:,:,1);
P2_green = p2(:,:,2);
P2_blue = p2(:,:,3);
C2_red= c2(:,:,1);
C2_green= c2(:,:,2);
C2_blue = c2(:,:,3);
R64_red = corr2(c2_red,p2_red)
R64_green = corr2(c2_green,p2_green)
R64_blue = corr2(c2_blue,p2_blue)

%%%%%Image 3 Correlation Calculations %%%%%

P3_red = p3(:,:,1);
P3_green = p3(:,:,2);
P3_blue = p3(:,:,3);
C3_red= c3(:,:,1);
C3_green= c3(:,:,2);
C3_blue = c3(:,:,3);
R64_red = corr2(c3_red,p3_red)
R64_green = corr2(c3_green,p3_green)
R64_blue = corr2(c3_blue,p3_blue)
```

```
%%%%%Image 4 Correlation Calculations %%%%%

P4_red = p4(:,:,1);
P4_green = p4(:,:,2);
P4_blue = p4(:,:,3);
C4_red= c4(:,:,1);
C4_green= c4(:,:,2);
C4_blue = c4(:,:,3);
R64_red = corr2(c4_red,p4_red)
R64_green = corr2(c4_green,p4_green)
R64_blue = corr2(c4_blue,p4_blue)

%%%%%Image 5 Correlation Calculations %%%%%

P5_red = p5(:,:,1);
P5_green = p5(:,:,2);
P5_blue = p5(:,:,3);
C5_red= c5(:,:,1);
C5_green= c5(:,:,2);
C5_blue = c5(:,:,3);
R64_red = corr2(c5_red,p5_red)
R64_green = corr2(c5_green,p5_green)
R64_blue = corr2(c5_blue,p5_blue)

%%%%%Image Entropy Calculations %%%%%

I_64_red=entropy(c1_red)
I_64_green=entropy(c1_green)
I_64_blue=entropy(c1_blue)
I_128_red=entropy(c2_red)
I_128_green=entropy(c2_green)
I_128_blue=entropy(c2_blue)
I_176_red=entropy(c3_red)
I_176_green=entropy(c3_green)
I_176_blue=entropy(c3_blue)
I_256_red=entropy(c4_red)
I_256_green=entropy(c4_green)
I_256_blue=entropy(c4_blue)
I_2500_red=entropy(c5_red)
I_2500_green=entropy(c5_green)
I_2500_blue=entropy(c5_blue)
%%%%%NPCR Calculations %%%%%
```

```
npcr_red_64 = (sum( double( c1_red(:) ~= c11_red(:) ) ) /4096)
npcr_green_64 = (sum( double( c1_green(:) ~= c11_green(:) ) ) /4096)
npcr_blue_64 = (sum( double( c1_blue(:) ~= c11_blue(:) ) ) /4096)
npcr_red_128 = (sum( double( c2_red(:) ~= c12_red(:) ) ) /16384)
npcr_green_128 = (sum( double( c2_green(:) ~= c12_green(:) ) ) /16384)
npcr_blue_128 = (sum( double( c2_blue(:) ~= c12_blue(:) ) ) /16384)
npcr_red_176 = (sum( double( c3_red(:) ~= c13_red(:) ) ) /25344)
npcr_green_176 = (sum( double( c3_green(:) ~= c13_green(:) ) ) /25344)
npcr_blue_176= (sum( double( c3_blue(:) ~= c13_blue(:) ) ) /25344)
npcr_red_256 = (sum( double( c4_red(:) ~= c14_red(:) ) ) /65536)
npcr_green_256 = (sum( double( c4_green(:) ~= c14_green(:) ) ) /65536)
npcr_blue_256 =(sum( double( c4_blue(:) ~= c14_blue(:) ) ) /65536)
npcr_red_2500 =(sum( double( c5_red(:) ~= c15_red(:) ) ) /4687500)
npcr_green_2500=(sum( double( c5_green(:)~=c15_green(:)))/4687500)
npcr_blue_2500=(sum( double( c5_blue(:)~=c15_2500_blue(:)))/4687500)
```

%%%%%NPCR Calculations %%%%%

```
uaci_red_64 = sum(abs( c1_red_64(:) – c11_red_64(:) )  )/65536/255
uaci_green_64 = sum(abs( c1_green_64(:) – c11_green_64(:)))/65536/255
uaci_blue_64 = sum(abs( c1_blue_64(:) – c11_blue_64(:)))/65536/255
uaci_red_128 = sum(abs( c2_red_128(:) – c12_red_128(:) )  )/65536/255
uaci_green_128=sum(abs( c2_green_128(:)-c12_green_128(:)))/65536/255
uaci_blue_128 = sum(abs( c2_blue_128(:) – c12_blue_128(:)))/65536/255
uaci_red_176 = sum(abs( c3_red_176(:) – c13_red_176(:) )  )/65536/255
uaci_green_176=sum(abs( c3_green_176(:)-c13_green_176(:)))/65536/255
uaci_blue_176= sum(abs( c3_blue_176(:) -c13_blue_176(:)))/65536/255
uaci_red_256 = sum(abs( c4_red_256(:)–c14_red_256(:) )  )/65536/255
uaci_green_256=sum(abs( c4_green_256(:)-c14_green_256(:)))/65536/255
uaci_blue_256 = sum(abs( c4_blue_256(:) -c14_blue_256(:)))/65536/255
uaci_red_2500=sum(abs(c5_red_2500(:)–c15_red_2500(:)))/ 4687500/255
uaci_green_2500=sum(abs(c5_green_2500(:)-c15_green_2500(:)))
/4687500/255
uaci_blue_2500=sum(abs( c5_blue_2500(:)-c15_blue_2500(:)))/ 4687500/255
```

Startup Code (Environment Initializing)

```
clc;
clear;
addpath([getenv('XILINX_VIVADO') '/scripts/sysgen/matlab']);
```

```
xilinx.environment.setBoardFileRepos({'C:\Xilinx\Vivado\2020.2\data\boards\
board_files\arty-a7-35', 'C:\Xilinx\Vivado \2020.2\data\ boards\
board_files\pynq-z1'});  % enter the path for the required board from your PC
```

New Five Dimensional Hyperchaotic System

```
function Lorenz_System()
    clc; clear;           x0 = [10 10 10 40 50];
    tspan = [0 100];      [t,x] = ode45(@lorenz, tspan, x0);
    tiledlayout(5,1)
    ax1 = nexttile;      plot(ax1, t,x(:,1),'k'), hold on
    legend('X')
    xlabel('Time')
    ylabel('X(t)')
    grid on
    grid minor
    ax2 = nexttile;      plot(ax2,t,x(:,2),'k'), hold on
    legend('Y')
    xlabel('Time')
    ylabel('Y(t)')
    grid on
    grid minor
    ax3 = nexttile;       plot(ax3,t,x(:,3),'k'), hold on
    legend('Z')
    xlabel('Time')
    ylabel('Z(t)')
    grid on
    grid minor
    ax4 = nexttile;      plot(ax4,t,x(:,4),'k'), hold on
    legend('W')
    xlabel('Time')
    ylabel('W(t)')
    grid on
    grid minor
    ax5 = nexttile;       plot(ax5,t,x(:,5),'k'), hold on
    legend('P')
    xlabel('Time')
    ylabel('P(t)')
    grid on
    grid minor
    figure
    plot(x(:,1), x(:,2)), hold on
    xlabel('X(t)')
```

```
    ylabel('Y(t)')
    grid on
    grid minor
    figure
    plot3(x(:,1), x(:,2), x(:,3)), hold off
    xlabel('X(t)')
    ylabel('Y(t)')
    zlabel('Z(t)')
    grid on
    grid minor
end
function xprime = lorenz(t,x)
    sig    = 10;            s      = 1.5;            beta   = sin(t)+s;    rho    = 30;
    k      = 2;             l      =6.7;
       xprime = [-sig*x(1)  +  sig*x(2)+  beta*x(5)+x(4);
             rho*x(1)  -  x(2)    -  x(1)*x(3);
            -beta*x(3) +  x(1)*x(2);
            -k*x(1)    +  l*x(2)  +  x(2)*x(3);
            -x(3)      +  (x(2))^3];
     end
```

الخلاصة

ان تزايد استخدام الانظمة الفوضوية و الانضمة مفرطة الفوضى في خوارزميات تشفير الصورة بشكل كبير في السنوات الاخيرة كان بسبب الخصائص الرائعة لهذه الانضمة اللاخطية الا و هي العشوائية و الاعتماد على القيم البدائية و السلوك الذي لا يمكن التنبئ به. يقسم تشفير الصور إلى نوعين:النوع الاول يتم من خلاله تتغير قيمة عنصر الصورة (البكسل) والنوع الثاني تغيير موقع عنصر (بكسل) الصورة، وفي بعض الأحيان يتم تطبيق كلا النوعين على الصورة للحصول على درجة عالية من الأمان.
إن بناء نظام اتصال آمن وجدير بالثقة (تشفير وفك تشفير ومزامنة) خاص بنقل الصور وإنشاء مولد نظام فريد من نوعه يشكل المسعى الرئيسي في هذه الأطروحة. تقترح هذه الأطروحة خمس خوارزميات لتشفير الصور بالاعتماد على نظم الفوضى المفرطة.
أول خوارزمية مقترحة هي تصميم مولد نظام لورنز غير خطي ثلاثي الأبعاد على جانبي المرسل والمستقبل لاستخدامه في التحكم في المفاتيح السريعة التبديل لاحقا خلال الخورازميات المقترحة التالية.

الطريقة الثانية المقترحة تجمع بين أنظمة تشعبية مختلفة باستخدام عملية XOR المنطقية لبناء نظام تشعبي جديد متعدد الأبعاد يمكن استخدامه في أنظمة الاتصالات لأغراض تشفير الصور وفك تشفيرها.

اما في الخوارزمية الثالثة المقترحة فهي مولد نظام فرط التشتت المتتالي على أساس تبديل الفوضى اعتمادا على النظام المصمم في الخوارزمية الاولى، حيث يتم استخدام أنظمة فرط تشوه ثلاثية الأبعاد مختلفة ويتم دمجها باستخدام مفتاح عالي السرعة. يتم التحكم في إخراج المفتاح السريع من خلال نظام لورنز الفوضوي (الخوارزمية المقترحة الأولى). تولد الأنماط المختلطة للأنظمة فائقة تدفقات بتات ثنائية جديدة تمامًا تستخدم للاتصالات الآمنة القائمة على الصور.

تتضمن الخوارزمية الرابعة المقترحة تصميم نظام فرط فوضوي جديد خماسي الأبعاد مع موجة جيبية كمعامل إدخال إلى المعادلات التفاضلية العادية لزيادة العشوائية وعدم القدرة على التنبؤ بالنظام المطور. تم اعتماد النظام الجديد المقترح لبناء نظام اتصال آمن لأغراض تشفير الصور التي تظهر أداءً فائقًا.

اما الخوارزمية المقترحة الأخيرة فيتم إنشاء مولد بيانات ثنائية عشوائي جديد لاستخدامه في أجهزة إرسال ومستقبلات أنظمة الاتصالات بناءً على الخوارزميات المقترحة الأولى والثانية والثالثة والرابعة. يهدف النظام إلى إظهار سلوك غير متوقع للغاية وعشوائية ، مما يجعله مثاليًا لتشفير / فك تشفير الصور بشكل جدا آمن.

يتم استخدام واختبار جميع الخوارزميات المصممة من أجل صور الملونة RGB ورمادي بأحجام مختلفة (أبعاد متساوية وغير متساوية). تظهر نتائج المحاكاة أن الخوارزميات المصممة آمنة للغاية ويمكن استخدامها لأي حجم صورة بأي قدر من كمية البكسل. تظهر المراقبة المرئية للصور المشفرة المنتجة أن الخوارزميات المقترحة 2 ، 3 ، 4 ، 5 لها أداء متفوق ، حيث يكون الرسم البياني لهذه الصور المشفرة مسطحًا تمامًا ، حيث تم إخفاء توزيع الألوان للصور الأصلية.

متوسط الخطأ التربيعي (MSE) له قيمة قريبة جدًا من ثمانية ، مما يشير إلى أن الأنظمة لا يمكن التنبؤ بها إلى حد كبير. يتم أيضًا حساب NPCR و UACI من خلال هذا العمل ، حيث تبلغ قيمة NPCR قريبة من 100٪ ، بينما تقترب UACI من 33٪ ، مما يشير إلى أن الخوارزميات تتمتع بمناعة عالية ضد الهجمات التفاضلية.

تم تنفيذ جميع الخوارزميات المصممة بنجاح باستخدام لوحة FPGA PYNQ-Z1 zynq xc7z020 وقد كانت كمية الموارد المستخدمة في اللوحة مقبولة.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل / كلية الهندسة
قسم الهندسة الكهربائية

تصميم وتنفيذ نظام اتصالات لاسلكي أمن ذو عشوائية مفرطة بأستخدام مجموعة بوابات قابلة للبرمجة

اطروحة
مقدمة إلى كلية الهندسة في جامعة بابل
وهي جزء من متطلبات الحصول على درجة الدكتوراه
فلسفة في الهندسة / الهندسة الكهربائية/ الالكترونيك والاتصالات

من قبل

حيدر مازن مكي الابراهيمي

بأشراف

الأستاذ الدكتور أيهاب عبد الرزاق حسين
الأستاذ الدكتور قيس كريم عمران

A.D 2022
A.H 1444