People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Yahia Fares University of Medea
Faculty of Technology
Department of Electrical Engineering

End of studies Dissertation for obtaining a Master's Degree in Electrical Engineering

Electronics for Embedded Systems

Theme

Design and FPGA Implementation of a New Chaotic System for the Generation of Robust S-Box

---

Presented by:

MEDANI Mohamed Akram

BOUMEDDIENE Mohammed Elghazali

Supervised by:

Dr. MAAZOUZ Mohamed

Promotion 2023/2024

Acknowledgments

First and foremost, We express our infinite gratitude to Allah, the All Mighty, the Merciful, who granted us the strength, courage, and wisdom, and who provided us with wonderful people who supported and encouraged us.

Our deepest gratitude to our parent for their long lasting support.

And we would like to extend our sincere appreciation to our supervisor, Dr. MAAZOUZ Mohamed for his for his advice, guidance and invaluable support during our study and the completion of this humble work.

Dedications

TO our family,

TO our friends,

TO mention all our teachers who contributed and participated in our formation, from primary to present,

TO all those whom we love, who love us,

AND to those who share good memories.

we dedicate this work.

ملخص

يركز مشروع البحث على تصميم نظام فوضوي جديد للاستعمال في توليد صناديق إستبدال (S-Box) قوية لتعزيز أمن أنظمة التشفير. أنظمة الفوضى، التي تتميز بحساسية للظروف الأولية، توفر إمكانات كبيرة لتوليد SBox، وتقديم طبقة من التعقيد التي يصعب فك تشفيرها. باستخدام بطاقات FPGA ، يهدف هذا البحث إلى استغلال ميزة التوازي في معالجة المعلومات والمرونة لهذه الأجهزة لتحسين كفاءة أنظمة تشفير متطورة.

كلمات مفتاحية: الأنظمة الفوضوية، صناديق الإستبدال، التشفير، فك التشفير، بطاقات FPGA، حماية المعلومات، شبكة فايستل.

Abstract

The research project focuses on the design and implementation of a new chaotic system specifically adapted to FPGAs to generate robust SBox to strengthen the security of cryptographic systems. Chaotic systems, characterized by their sensitivity to initial conditions, offer significant potential for SBox generation, introducing a layer of complexity that is difficult to decipher. Using FPGA, this research aims to harness the parallelism and flexibility of these devices to optimize the efficiency of the new chaotic system in SBox generation.

Keywords: Chaotic system, Cryptography, S-box, Encrytion, Decryption, FPGA, Information Security, Fiestel Network.

Résumé

Le projet de recherche se concentre sur la conception et l'implémentation d'un nouveau système chaotique spécifiquement adapté aux FPGA afin de générer des SBox robustes pour renforcer la sécurité des systèmes cryptographiques. Les systèmes chaotiques, caractérisés par leur sensibilité aux conditions initiales, offrent un potentiel significatif pour la génération de SBox, introduisant une couche de complexité difficile à décrypter. En utilisant les FPGA, cette recherche vise à exploiter le parallélisme et la flexibilité de ces dispositifs pour optimiser l'efficacité du nouveau système chaotique dans la génération de Sbox.

Mots clés : Système chaotique, Cryptographie, S-box, Encrytion, Decryption, FPGA, Securité des Information, Réseau de Fiestel.

Table of Contents

Acknowledgments....II
Dedications....III
Abstract....IV
Abbreviation list....X
List Of Indices & Parameters....XII
General Introduction....1
Chapter I Chaotic Systems....4
1.Introduction....5
2. Chaos Theory....5
3. Chaotic Systems....5
4. Key elements of chaotic systems....6
4.1. Sensitivity to initial conditions....6
4.2. Non-linearity....6
4.3. Deterministic Chaos....6
4.4. Aperiodic behavior....7
5. Chaotic maps and systems classification....7
6.Chaos Detection Techniques....8
6.1 Bifurcation Diagram....8
6.2 Lyapunov Exponent....8
6.3. Time Series Analysis....8
6.4. Phase Portrait....8
6.5. Poincare Section....9
6.6. Kolmogorov Entropy....9
7. Showcase of well-known Chaotic systems and maps....9
7.1 Chaotic Systems....10
7.1.1 Lorenz System....10
7.1.2 Chen System....11
7.2 Chaotic Maps....12
7.2.1 Logistic Map....12
7.2.2 Tent Map....12
8. Chaotic systems in cryptology....13
9. Conclusion....13
Chapter II Review of Cryptography....14
1.Introduction....15
2.Cryptography Termonology....15
3.Cryptography Branches....16
3.1 Asymmetric Key Cryptography....16
3.2 Symmetric Key Cryptography....17
3.2.1. Classic Cryptography....17
3.2.1.I. Substitution Cipher....17
3.2.1.II. Transposition Cipher....17
3.2.2. Modern Cryptography....18
3.2.2.I. Stream Cipher....18

3.2.2.II. Block Cipher....18
4. Modes of Operations....18
4.1. The Electronic Codebook (ECB) Mode....19
4.2. The Cipher Block Chaining (CBC) Mode....20
4.3. The Counter (CTR) Mode....20
5. Substitution–Permutation Networks....21
6 Substitution Box....21
6.1. Bijectivity Property....22
6.2. Non-linearity Property....22
6.3. Strict Avalanche Criterion (SAC)....22
6.4. Bit Independence Criterion (BIC)....22
7. Feistel Networks....23
8.Conclusion....24
Chapter III Chaos Based Cryptosystem....25
1.Introduction....26
2.Background on constructing chaotic systems....26
2.Construction of a new chaotic system....27
3.Performance analysis of the proposed chaotic system....28
4.The proposed S-Box generation algorithm....30
5.Performance evaluation of the proposed S-box....31
6. Cipher Block Algorithm....32
7.Conclusion....34
Chapter IV FPGA Implementation Results and Analysis....35
1.Introduction....36
2.Overview of FPGA....36
3.Design Requirements....37
3.1.Hardware....37
3.2.Software....39
Vivado HLS....39
Vivado HLx....39
Vivado SDK....39
2.3.IP Core Generation....41
4.Block Design....42
5.Implementation....43
6.Results....44
6.1.Correlation Coefficient....44
6.2.Key Sensitivity....47
6.3.Histogram....48
6.4.Diffusion Analysis....50
6.5.Entropy....50
6.6.Encryption/decryption Runtime....51
7.Comparison to state-of-art works....51
8.Conclusion....52
General Conclusion....53
Bibliography....56

Table of Figures

Figure 1: Classification of chaotic non-linear dynamical systems....7
Figure 2: Sensitivity to Initial Conditions for Lorenz system....10
Figure 3: Lorenz System's....10
Figure 4: Lyapunov Exponent for the Lorenz system....10
Figure 5: Sensitivity to Initial Conditions for Chen system....11
Figure 6: Chen System's....11
Figure 7: Lyapunov Exponent for Chen system....11
Figure 8: Sensitivity to Initial Conditions for the Logistic Map....12
Figure 9: Lyapunov Exponent for the Logistic Map....12
Figure 10: Bifurcation Diagram....12
Figure 11: Sensitivity to Initial Conditions for Tent Map....12
Figure 12: Lyapunov Exponent for Tent Map....12
Figure 13: Bifurcation Diagram....12
Figure 14: Cryptography Branches....16
Figure 15: Asymmetric Encryption Scheme....16
Figure 16: Symmetric Encryption Scheme....17
Figure 17: Caeser cipher using a left shift of 3 positions....17
Figure 18: Stream Cipher Scheme....18
Figure 19: Block Cipher Scheme....18
Figure 20: The ECB Mode....19
Figure 21: Tux mascot....19
Figure 22: Tux mascot encrypted using ECB mode....19
Figure 23: The CBC Mode....20
Figure 24: The CTR Mode....20
Figure 25: Feistel encryption/decryption structure....23
Figure 26: Time Series Analysis....28
Figure 27: Sensetivity to Initial Conditions for the Proposed system....28
Figure 28: Phase Portrait of the Proposed system....29
Figure 29: Lyapunov Exponent for the Proposed system....29
Figure 30: Function used in Feistel Network....32
Figure 31: Encryption Scheme....32
Figure 32: Decryption Scheme....33
Figure 33: Flowchart of the Proposed Block Cipher....33
Figure 34: Architecture of an FPGA Device....37
Figure 35: The FPGA card Nexys A7 100T....38
Figure 36: Typical FPGA Design Flow....40
Figure 37: High Level Synthesis Flow....41
Figure 38: Block Design Overview....43
Figure 39: FPGA Implementation Process....43
Figure 40: Correlation test on Cameraman Image....45
Figure 41: Correlation test on Peppers Image....46
Figure 42: Correlation test on Clock Image....46
Figure 43: Correlation test on Tree Image....47

Figure 44: Cameraman Image decrypted with small change in Initial Key.....47
Figure 45: Histogram Analysis of Cameraman Image.....48
Figure 46: Histogram Analysis of Peppers Image.....48
Figure 47: Histogram Analysis of Clock Image.....49
Figure 48: Histogram Analysis of Tree Image.....49

Index of Tables

Table 1: S-Box Generated using Algorithm 1.....30
Table 2: Performance test of the SBox.....31
Table 3: Detailed SAC test result.....31
Table 4: Detailed result of the BIC-Non-linearity test.....31
Table 5: Characteristics of the FPGA Nexys A7 100T.....38
Table 6: Utilization Estimates Summary.....41
Table 7: Correlation Coefficient analysis.....45
Table 8: Diffusion analysis.....50
Table 9: Entropy analysis.....51
Table 10: Performance Comparison using Peppers image as refrence.....51
Table 11: Encryption/decryption time comparison.....51

Abbreviation list

AES: Advanced Encryption Standard

ASIC: Application-Specific Integrated Circuit

AXI: Advanced eXtensible Interface

BIC: Bit Independence Criteria

CBC: Cipher Block Chaining

CLB: Configurable Logic Block

CPU: Central Processing Unit

CTR: Counter

DMA: Direct Memory Access

DSP: Digital Signal Processing

ECB: Electronic Code Book

FF: Flip-Flop

FIFO: First in First Out

FPGA: Field-Programmable Gate Array

HDL: Hardware Description Language

HLS: High Level Synthesis

IBM: International Business Machines

IV: Initial Value

IOB: Input/Output Blocks

LUT: Look-Up-Table

NPCR: Number of Pixels Change Rate

OS: Operating System

RTL: Register Transfer Level

RAM: Random Access Memory

SAC: Strict Avalanche Criteria

S-Box: Substitution-Box

SDK: Software Development Kit

SRAM: Static Random Access Memory

UACI: Unified Average Changing Intensity

UART: Universal Asynchronous Receiver Transmitter

VHDL: VHSIC Hardware Description Language

XOR: Exclusive OR

List Of Indices & Parameters

$\dot{x} = \frac{dx}{dt}$: Derivative of the variable x with respect to time.

$\dot{y} = \frac{dy}{dt}$: Derivative of the variable y with respect to time.

$\dot{z} = \frac{dx}{dt}$: Derivative of the variable z with respect to time.

Rn: Set of real integers.

$R+$: Set of positive real numbers.

x_0: Initial state of variable x.

x_n: State of x at time t=nT.

J: Jacobian of a matrix.

λi: Lyapunov exponent of order i.

H: Entropy.

General Introduction

General Introduction

Data protection is regarded as one of the biggest issues for sensitive information whether it was transmitted through different networks or stored to be safeguarded. With the internet having reached a level that merges with our lives, growing explosively during the last several decades, cyber threats have been plaguing the internet, many experiments and researchs are garnered in cryptosystem, and continuous progress is made to shield the multimedia data from the ever continuous cyber threats and attacks, as to ensures that our data is only accessible by the intended receiver and prevents any modification or alteration of data. In order to achieve this level of security, numerous encryption techniques have been used and various algorithms and methods have been developed to address these issues and protect data from any malicious attempts and assaults. Cryptography can be defined as techniques that cipher data, depending on specific algorithms that make the data unreadable to the human eye unless decrypted by algorithms that are predefined by the sender.

Nowadays, most encryption algorithms relies on Substitution-Boxes, it is the single non-linear component in the cipher block thus providing confusion in data. An S-box that possesses high nonlinearity and low linear and differential probability is considered cryptographically secure.

For generating robust S-box, chaotic systems are the foundation, they serve as a vital element in a more robust encryption scheme due to their amplification effect on the deviation of the initial value.

In this dissertation, a new 3D chaotic system is introduced. The proposed chaotic system is applied to the process of building the S-box, and through experimental analysis, it can be concluded that the S-box constructed has higher encryption advantage and has strong resistance to various security attacks. Then a cipher block algorithm is designed based on Feistel Network and elements of Advanced Encrytion Standanrd. Analysis of experimental data indicates that the proposed encryption scheme is capable of withstanding common security threats.

The first chapter would cover some insight over the meaning of chaos providing key element of it in mathematical modeling and how some simple dynamical systems can have chaotic behavior in certain conditions. The second chapter dedicates its content to laying the foundation to cryptography, the meaning, the different types and how they are implemented illustrated with examples. The third chapter would go through the process of constructing the building blocks of the cryptosystem, from constructing a new 3D chaotic system, to generating an S-Box then designing a cipher block scheme. The last chapter is about practical implentation of the cryptosystem on an FPGA card, and conducting different tests and denoting their results illustrating the effectiveness of the cryptosystem.

Chapter I

Chaotic Systems

1.Introduction

In this chapter, we explore chaotic systems, the essential elements that define chaotic systems, and their characteristics, and categorize chaotic maps and systems based on their dynamic properties. In the end, we discover the significant advantages of chaotic systems in cryptology, specifically in generating robust and secure cryptographic algorithms.

2. Chaos Theory

Chaos, from the Greek khaos, "abyss, that which gapes wide open, that which is vast and empty", is a relatively young scientific discipline, which however has old and important roots, dating back (at least) to James Clerk Maxwell in 1860 and Henry Poincare′ in 1890. The birth of the modern age of Chaos is commonly referred to the work of Edward Lorenz in 1963 . Chaos is typically understood as a mathematical property of a ***dynamical system***. A dynamical system is a deterministic mathematical model, where time can be either a continuous or a discrete variable. Such models may be studied as mathematical objects or may be used to describe a target system [1].

3. Chaotic Systems

A chaotic system is defined as a physical system that is highly sensitive to initial conditions, according to physical science. But the word is associated with absolute confusion and disorder in the dictionary. Since the consequence of high sensitivity to initial conditions appears to be disordered, it is commonly associated with the dictionary meaning of chaos. However, in physical science, chaos is not random or disorder. The behavior of the system is deterministic and can be mathematically modeled. This governs the behavior of the individual step in the chaotic process. But the outcome of the combined effect of several iterations is unpredictable [2].

4. Key elements of chaotic systems

Chaos is characterized by many principal properties, which makes it desirable for building a strong cryptosystem. It is identified with sensitive dependence on initial conditions, nonlinearity, deterministic, ergodicity, non-periodicity and unpredictable behavior [3].

4.1. Sensitivity to initial conditions

The sensitivity to the initial state is one of the main features of chaotic dynamics which is sometimes called the "butterfly effect". It means that the system has sensitive dependence on initial conditions in which a small change in one state of a deterministic nonlinear system can result in large differences in later states. Chaotic systems depends largely on their sensitivity to initial conditions, as they can still be computable due to this trait [4]. Also, it is difficult to predict chaotic system when there is noise because stochastic fluctuations may cause trajectories with same initial conditions to diverge [5]. Transients, stability and sensitivity of parameter changes should be calculated when dealing with these mathematically modeled systems by differential equations so as to understand how they work.

4.2. Non-linearity

Non-linear systems give complexities such as chaos and unpredictability, chaotic systems prevalent in economics, sociology, and the natural sciences have dynamics that cannot be predicted due to swelling non-linear influences, notably when chaos reigns [6]. Several techniques shed light on action of non-linear systems. Such systems have trajectories that continue to stretch and fold, creating bifurcations and chaotic attractors.

4.3. Deterministic Chaos

Deterministic chaos is how these systems follow specific rules, their behavior to unpredictable and randomness, the process can be modeled and governed quantitatively by mathematical equations and to some extent approached quantitatively [7].

4.4. Aperiodic behavior

Aperiodic behavior in chaotic systems is the manner of irregular dynamic that cannot be repeated in a predictable style [8] and the model produces random values.

5. Chaotic maps and systems classification

Dynamical systems with discrete time are labeled as chaotic maps. These maps are further classified as one-dimensional or multi-dimensional. Another class of chaotic maps are developed to overcome the limited and discontinuous range of chaotic maps, namely the coupled map lattice (CML). It attributes spatiotemporal chaos, which is nonlinear and complex in spatially extended systems. A large number of positive Lyapunov exponents of the spatiotemporal system guarantee very high randomness and ergodicity. In contrast to the discrete chaotic maps, dynamical systems with continuous time are called the chaotic system, which is characterized of three or higher dimensions. The systems are classified as chaotic if they possess at least one positive Lyapunov exponent, and hyper-chaotic if more than one positive Lyapunov exponent is exhibited.

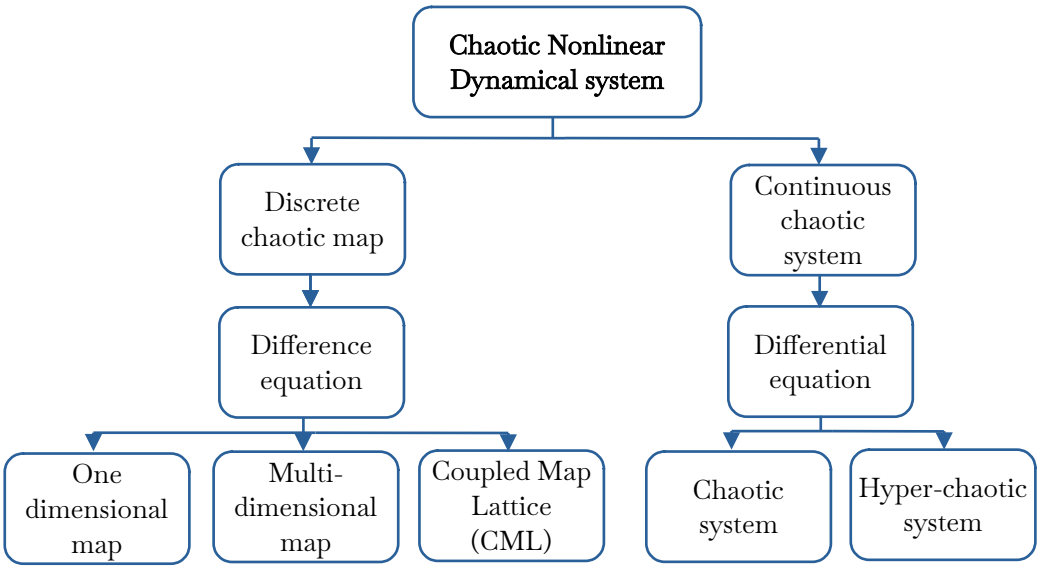


Figure 1: Classification of chaotic non-linear dynamical systems

6.Chaos Detection Techniques

Theorists of chaos have proposed several chaos detection techniques, which examine the behavior of the dynamic system in different dimensions. They are instrumental in quantifying the complexity of the chaotic maps and systems [9].

6.1 Bifurcation Diagram

The bifurcation diagram is very significant as this plot exhibits a sudden appearance of a qualitative behavior as some parameters are varied. This behavioral change is called bifurcation and occurs at bifurcation points [10].

6.2 Lyapunov Exponent

Lyapunov exponent is defined as a measure of the sensitivity and predictability of the dynamic system to changes in its initial conditions [11], It is numerically calculated with the following equation:

$$\lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| \frac{\Delta R_i}{\Delta R_0} \right|$$

Where λ is the Lyapunov exponent, ΔR_i is a distance between two time series X_i and Y_i and ΔR_0 is an intial distance of $\Delta R_0 = \| X_0 - Y_0 \|_2$. A dynamic system with n dimensions has n Lyapunov exponents. A system with a positive Lyapunov exponent defines it to be chaotic.

6.3. Time Series Analysis

The time series analysis is a graphical inspection of the time series data, obtained by plotting the data series of the dynamic system versus time. It is a statistical analysis of the dynamic system, which represents the features of the data series such periodicities and complexity of the data [3].

6.4. Phase Portrait

The phase portrait is obtained by plotting the x_n versus x_{n+1}, which is used to visualize both the discrete maps and the continuous systems. Nevertheless, it is more frequently used in continuous system to observe the butterfly pattern of the attractor [12].

6.5. Poincare Section

The Poincare map is essential in studying the flow of a system near a periodic orbit or a chaotic system. Consider S to be the surface of section in the system, where all the trajectories flow through it. Then, the Poincare map P is obtained by following the trajectories of the dynamical system, from one intersection of the trajectory with S to the next [35]. It is an effective method to study the motion features of the autonomous system, especially when observation of the characteristics of the flow is difficult using the phase portrait.

6.6. Kolmogorov Entropy

The Kolmogorov entropy also known as the Kolmogorov–Sinai (KS) entropy is a measure of the predictability of a system. Kolmogorov entropy K is measured by tracking two points on the attractor with respect to time, as long as they are separated in space by less than an infinitesimally small length scale. With this method, K is obtained with high precision for a large number of pairs. For a predictable system, i.e., fully periodic, K equals zero, whereas for a stochastic system K is infinitely large. In a system featuring deterministic chaos, K is a finite positive value. Hence, KS entropy predicts the chaos and complexity of the motion in the dynamical system[11].

7. Showcase of well-known Chaotic systems and maps

In this section, famous chaotic systems and maps are shown with their corresponding feature characteristics.

7.1 Chaotic Systems

7.1.1 Lorenz System

Considered to be the first in line when the word chaos is mentioned, with an equation described as follow [13]: with $\sigma=10, \rho=28, \beta=\frac{8}{3}$

$$\dot{x}=\sigma(y-x)$$
$$\dot{y}=x(\rho-z)-y$$
$$\dot{z}=xy-\beta z$$

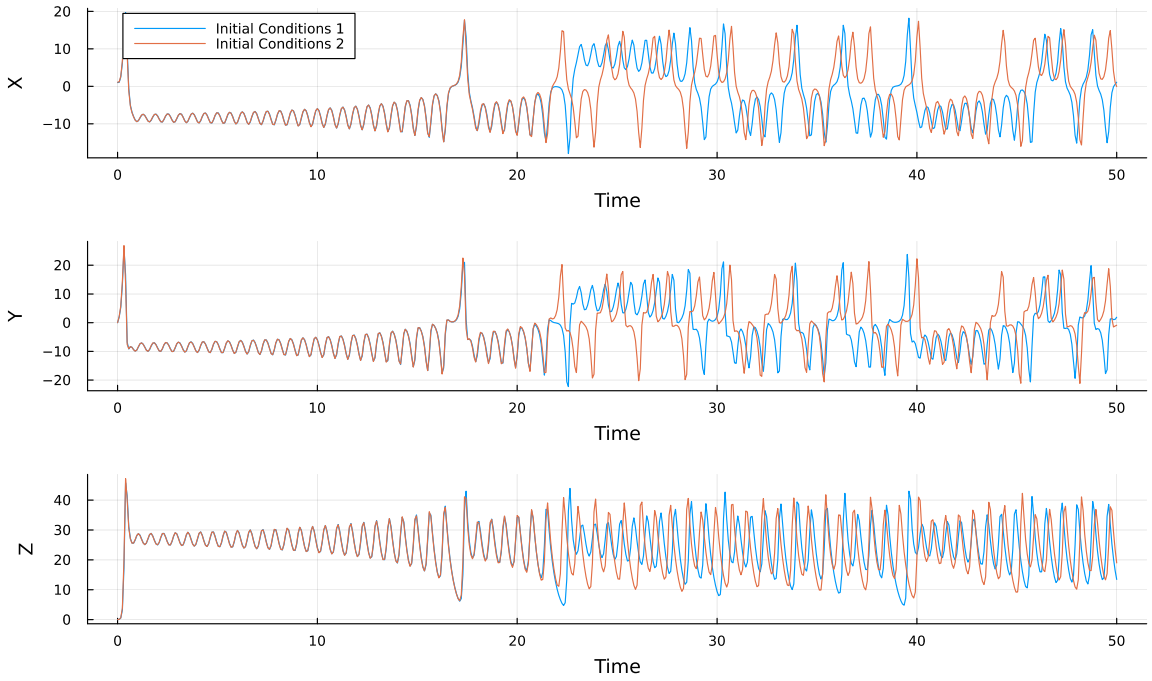


Figure 2: Sensitivity to Initial Conditions for Lorenz system

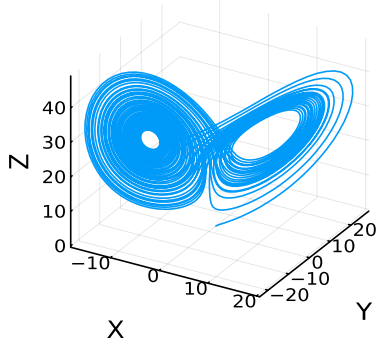


Figure 3: Lorenz System's Phase Portrait

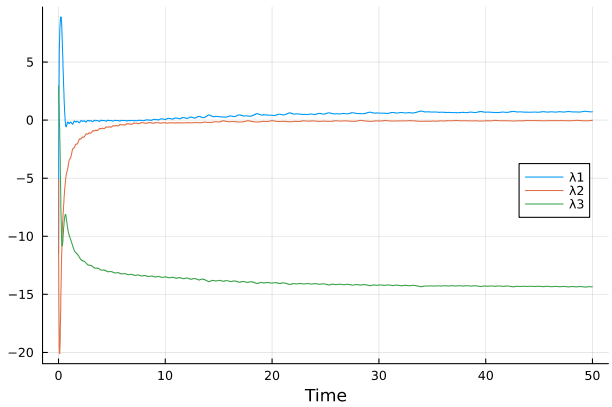


Figure 4: Lyapunov Exponent for the Lorenz system

7.1.2 Chen System

It is described with the following equation [14]: where $a=40, b=3, c=28$

$$\dot{x}=a(y-x)$$
$$\dot{y}=(c-a)x+cy-xz$$
$$\dot{z}=xy-bz$$

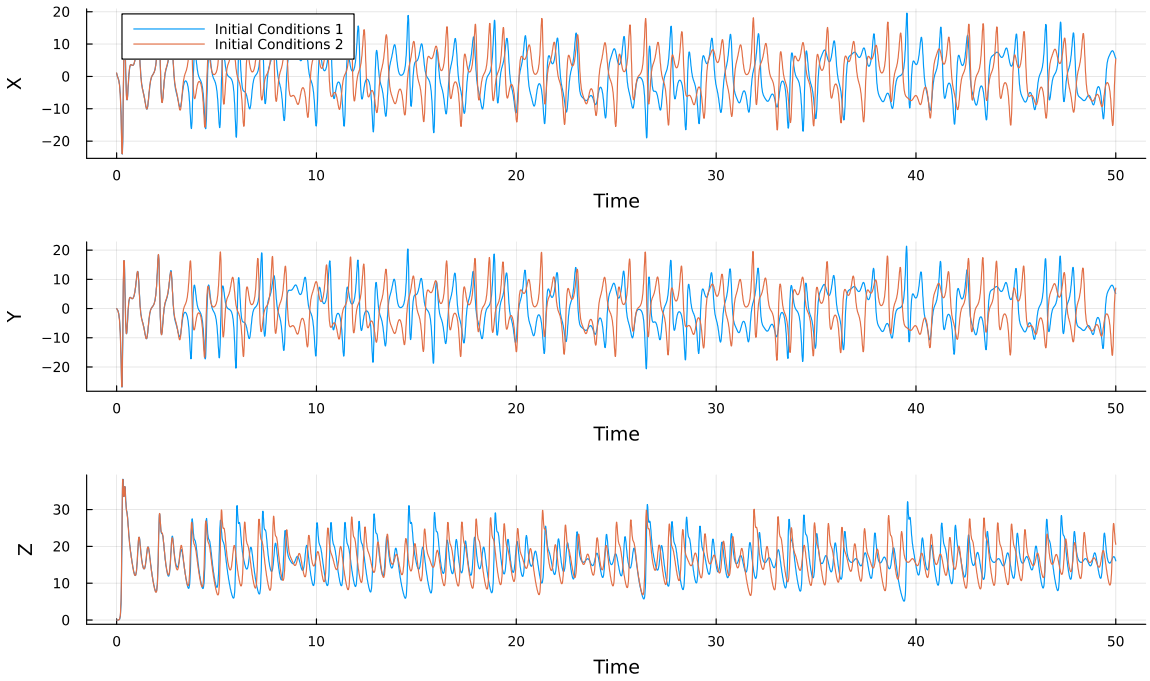


Figure 5: Sensitivity to Initial Conditions for Chen system

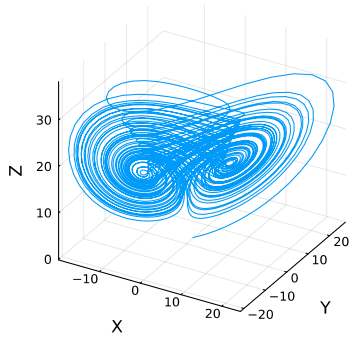


Figure 6: Chen System's Phase Portrait

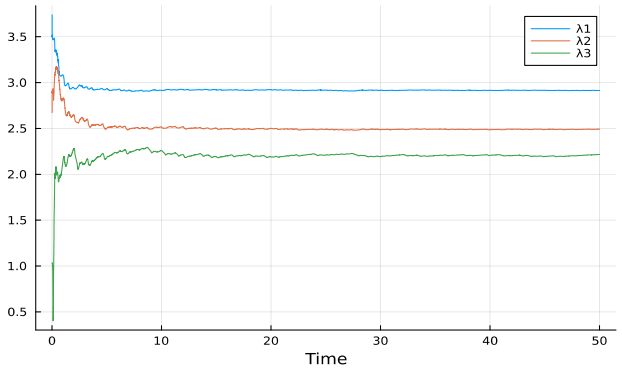


Figure 7: Lyapunov Exponent for Chen system

7.2 Chaotic Maps

7.2.1 Logistic Map Described as follow [15]: $x_{n+1} = r x_n (1 - x_n)$

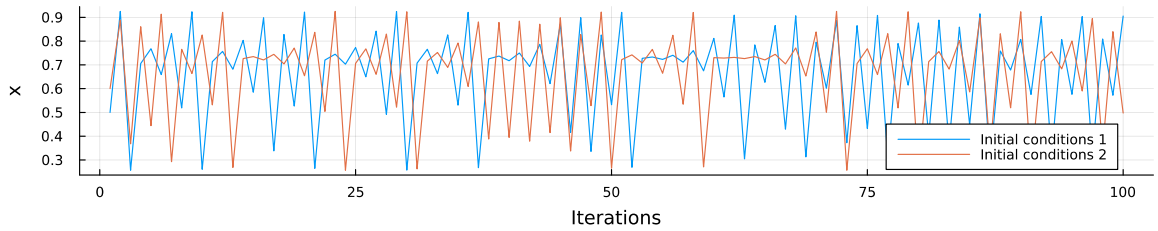


Figure 8: Sensitivity to Initial Conditions for the Logistic Map

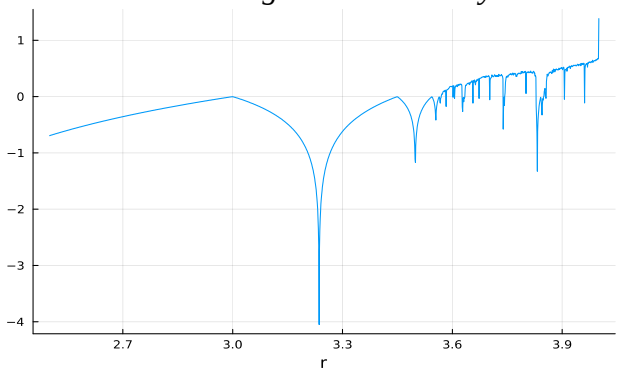


Figure 9: Lyapunov Exponent for the Logistic Map

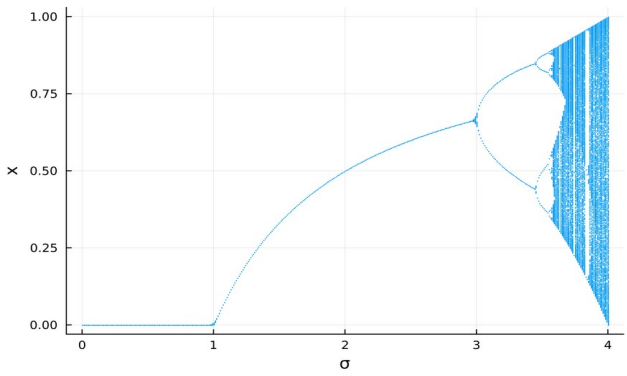


Figure 10: Bifurcation Diagram

7.2.2 Tent Map Described as follow [16]:

$$x_{n+1} = \mu x \quad \text{if } x < 0.5$$
$$x_{n+1} = \mu(1 - x) \quad \text{if } x > 0.5$$

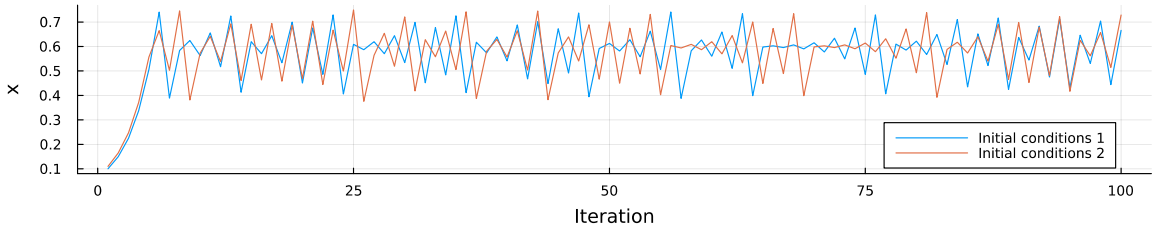


Figure 11: Sensitivity to Initial Conditions for Tent Map

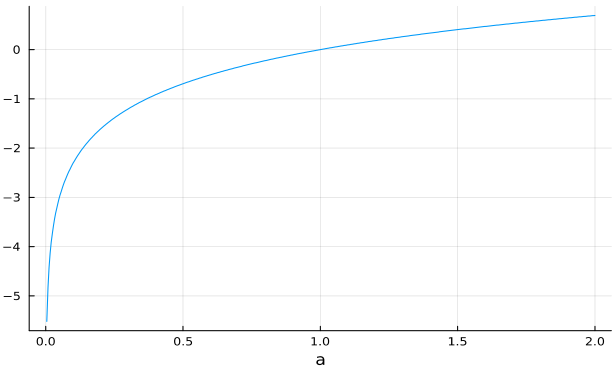


Figure 12: Lyapunov Exponent for Tent Map

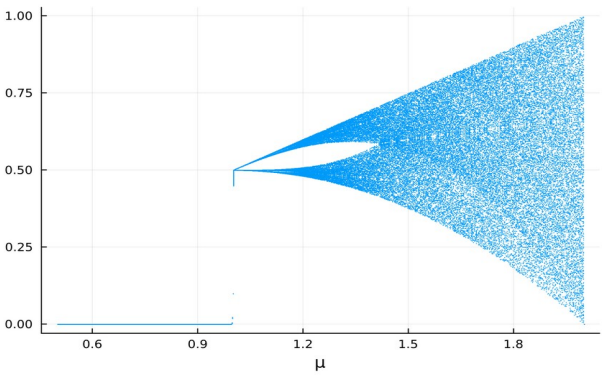


Figure 13: Bifurcation Diagram

8. Chaotic systems in cryptology

Chaos has potential applications in several functional blocks of a digital communication system: compression, encryption and modulation. The possibility for self-synchronization of chaotic oscillations has sparked an avalanche of works on application of chaos in cryptography, as the usage of it in cryptography is to ensure and create secure encryption algorithms. The nature of chaotic systems makes them basically tough to various cryptanalytic attacks, like statistical attacks, and differential attacks. The complexity created by chaotic behavior makes it difficult for attackers to predict or analyze cryptographic keys, ciphertexts, or other sensitive data [17].

9. Conclusion

In conclusion, this research on chaotic systems has revealed their nature and profound impact as well as the central role in cryptography. Highlighting the significant advantages they have in cryptology, and emphasizing their importance in generating robust and secure cryptographic algorithms.

Chapter II

Review of Cryptography

1.Introduction

In light of modern computing systems permeating every facet of daily life, the need for robust security in these systems has grown significantly. The skill of developing protocols for secure communication is known as cryptography, it is traditionally considered as a main information security mechanism, providing several security services such as confidentiality of messages, as well as data and entity authentication. A robust cryptographic method would ensure that sensitive data are only accessible to the intended audience and protect it from being exploited by malicious actors [18].

2.Cryptography Termonology

Plain Text: It is the first structure information that a sender needs to ship off the beneficiary. It is a unique reasonable message that is contribution to the calculation.

Ciphertext: Ciphertext is the mixed substance or message in its coded human incomprehensible structure. The ciphertext is the yield of encryption cycle and contribution of unscrambling measure if two distinct keys utilized for encryption of a message, at that point two diverse cipher texts are created.

Encryption Algorithm: It performs various methods, for example, replacement and change on the plaintext to acquire ciphertext.

Decryption Algorithm: It is the precisely inverse system of encryption strategy. To acquire unique plaintext it utilizes ciphertext and mystery key.

Secret Key: The mystery key is contribution to an encryption cycle. The key worth is free of plaintext and calculation. Contingent upon the key being utilized, the calculation gives different yield. The specific activity performed on that calculation relies upon the key.

Key Management: It deals with the secure generation, distribution and storage of keys. Secure methods of key management are extremely important, once a key is randomly generated, it must remain secret to avoid unfortunate mishaps.

3.Cryptography Branches

The following figure illustrate the different types of cryptography :

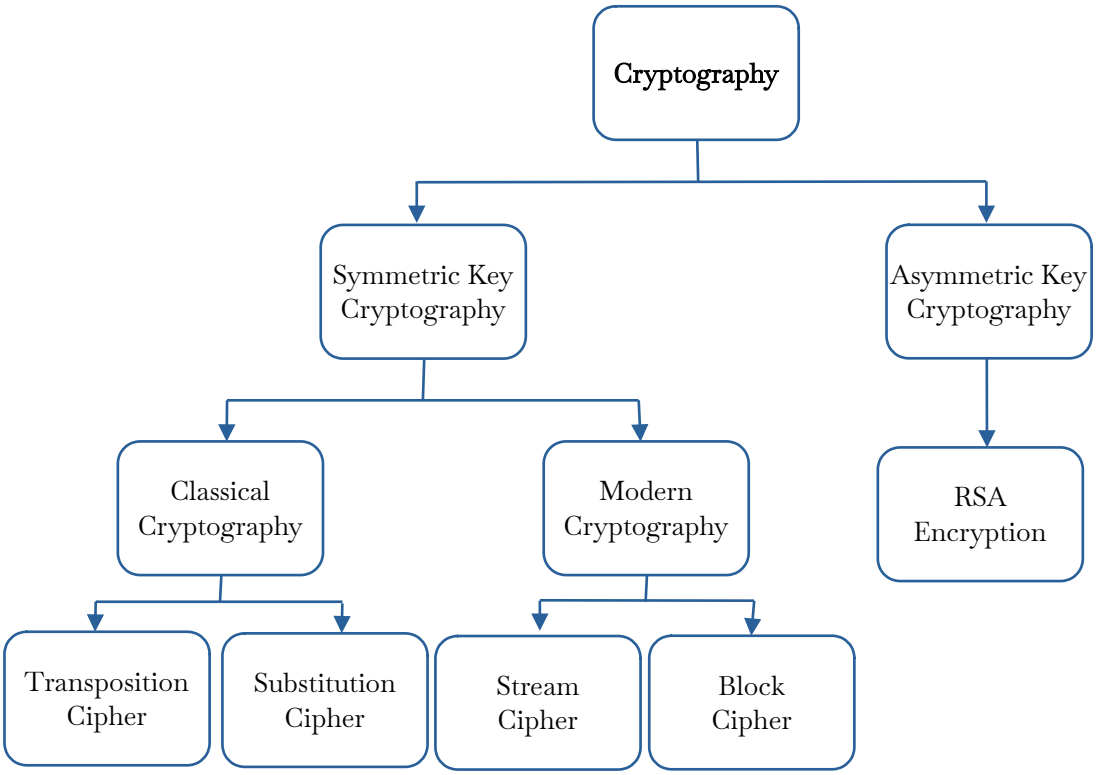


Figure 14: Cryptography Branches

3.1 Asymmetric Key Cryptography

In asymmetric encryption, there are two keys: one to encrypt and another to decrypt. The encryption key is called a public key and is generally considered publicly available to anyone who wants to send you encrypted messages. The decryption key, however, must remain secret and is called a private key. The public key can be computed from the private key, but the private key can't.

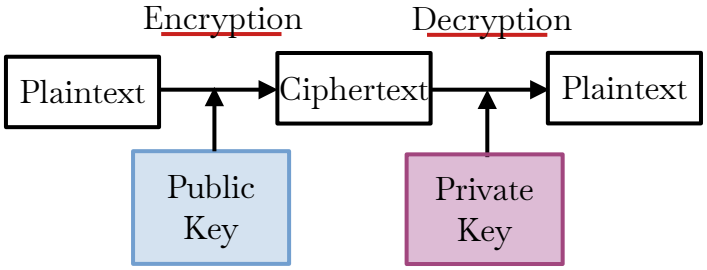


Figure 15: Asymmetric Encryption Scheme

3.2 Symmetric Key Cryptography

Unlike Asymmetric key cryptography, there is only one key to encrypt and decrypt.

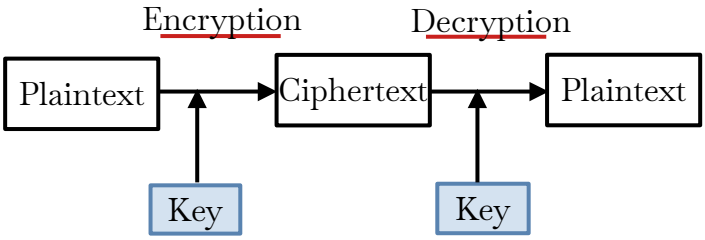


Figure 16: Symmetric Encryption Scheme

3.2.1. Classic Cryptography

3.2.1.I. Substitution Cipher

Each letter of plaintext is substituted for some letter of ciphertext according to some algorithm, a famous example is the Caeser cipher, in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

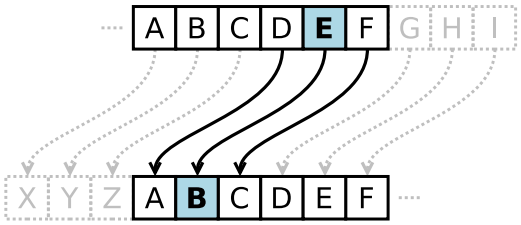


Figure 17: Caeser cipher using a left shift of 3 positions

3.2.1.II. Transposition Cipher

The way of this cipher is to change the position of each letter in the plaintext to a different position, example of that, the rail fence cipher where the plaintext is written downwards and diagonally on successive "rails" of an imaginary fence, then moving up when we get to the bottom. The message is then read off in rows. For example, using three "rails" and a message of 'WE ARE DISCOVERED FLEE AT ONCE', the ciphertext would look like this:

```
W . . . E . . . C . . . R . . . L . . . T . . . E
. E . R . D . S . O . E . E . F . E . A . O . C .
. . A . . . I . . . V . . . D . . . E . . . N . .
```

3.2.2. Modern Cryptography

3.2.2.I. Stream Cipher

A stream cipher encrypts bits individually. This is achieved by generating a bit from a pseudorandom stream of bits called the keystream. The keystream is XORed to a plaintext to encrypt it and then XORed again to the ciphertext to decrypt it [19].

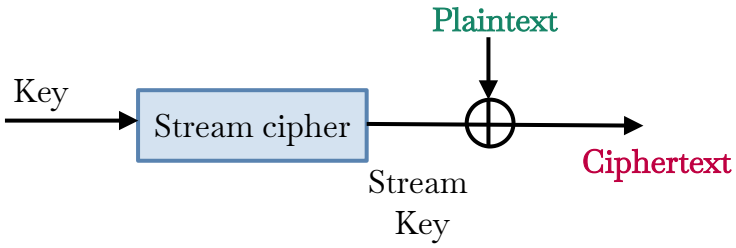


Figure 18: Stream Cipher Scheme

3.2.2.II. Block Cipher

A block cipher consists of an encryption algorithm and a decryption algorithm. The encryption algorithm **(E)** takes a key **K**, and a plaintext block **P**, and produces a ciphertext block **C**. We write an encryption operation as **C = E(K, P)**. The decryption algorithm **(D)** is the inverse of the encryption algorithm and decrypts a message to the original plaintext **P**. This operation is written as **P = D(K, C)** [19].

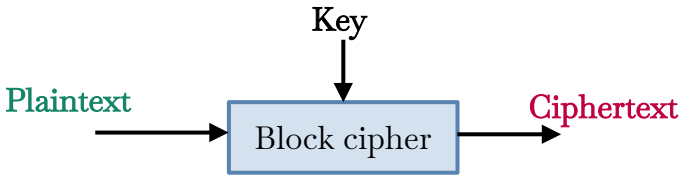


Figure 19: Block Cipher Scheme

4. Modes of Operations

Block ciphers operates in different modes each has its security and function properties, which makes it suitable for general or specific context.

4.1. The Electronic Codebook (ECB) Mode

Considered as the simplest mode, ECB takes plaintext blocks $P_1, P_2, \ldots, P_N$ and processes each independently by computing $C_1 = E(K, P_1)$, $C_2 = E(K, P_2)$, and so on, simple operation but also a highly insecure, the issue lies in that when encrypting the same plaintext, it gives the same ciphertext. A popular illustration is the encryption of an image to the linux mascot Tux [19].

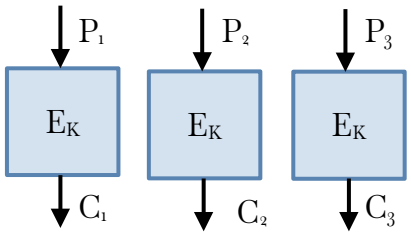


Figure 20: The ECB Mode

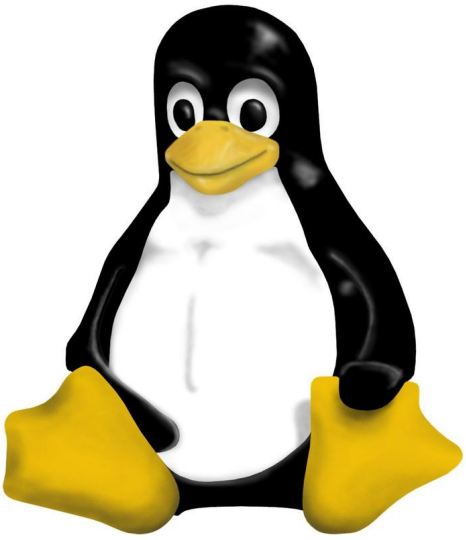


Figure 21: Tux mascot

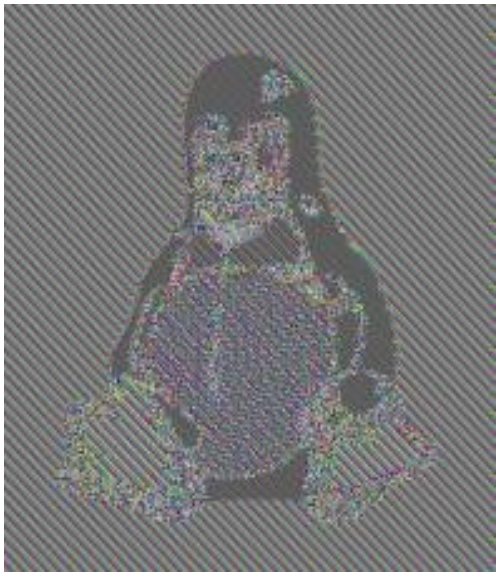


Figure 22: Tux mascot encrypted using ECB mode

4.2. The Cipher Block Chaining (CBC) Mode

Cipher block chaining (CBC) takes a different approach, instead of encrypting the ith block, P_i, as $C_i = E(K, P_i)$, it sets $C_i = E(K, P_i \oplus C_{i-1})$, where C_{i-1} is the previous ciphertext block—thereby chaining the blocks C_{i-1} and C_i , When encrypting the first block, P1, there is no previous ciphertext block to use, so CBC takes a random initial value (IV) [19].

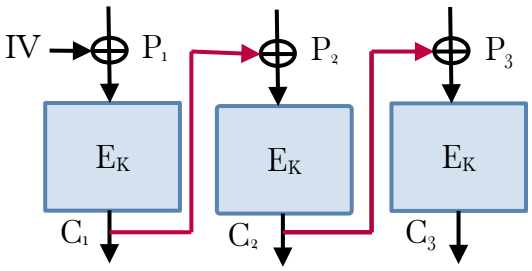


Figure 23: The CBC Mode

4.3. The Counter (CTR) Mode

In CTR mode, the block cipher algorithm won't transform plaintext data. Instead, it will encrypt blocks composed of a counter and a nonce. A counter is an integer that is incremented for each block. No two blocks should use the same counter within a message, but different messages can use the same sequence of counter values (1, 2, 3, ...). A nonce is a number used only once. It is the same for all blocks in a single message, but no two messages should use the same nonce [19].

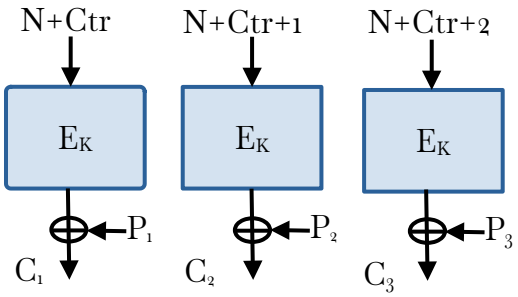


Figure 24: The CTR Mode

5. Substitution–Permutation Networks

Confusion and *Diffusion,* two that words summarize how cryptography work. Confusion means that the input (plaintext and encryption key) undergoes complex transformations, and diffusion means that these transformations depend equally on all bits of the input. At a high level, confusion is about depth whereas diffusion is about breadth. In the design of a block cipher, confusion and diffusion take the form of substitution and permutation operations, which are combined within substitution–permutation networks.

Substitution often appears in the form of S-boxes, or substitution boxes, which are small lookup tables that transform chunks of 4 or 8 bits. The permutation in a substitution–permutation network can be as simple as changing the order of the bits, which is easy to implement but doesn't mix up the bits very much. Instead of a reordering of the bits, some ciphers use basic linear algebra and matrix multiplications to mix up the bits [19].

6 Substitution Box

Non-linearity is an important concept in cryptography. The goal of any cipher is to have the output look as much like a random number as possible, and still be something we can decrypt later to get back the original plaintext. As s-boxes provide the primary source of nonlinearity in many symmetric ciphers, they are central elements in cryptography.

Considering a new S-box design, there are certain features that any s-box must exhibit in order to be an effective s-box. It is not the case that you can simply put together any substitution scheme you wish and have a good s-box. It is not enough that it simply substitutes values, the s-box needs to provide confusion and diffusion. The efficacy of an s-box is usually measured by examining three separate criteria. Each of these features contributes to the security of the s-box. Those features are Strict Avalanche Criteria, Balance, and Bit Independence Criteria [20].

6.1. Bijectivity Property

The bijectivity of an S-box refers to its property of being both injective and surjective, ensuring a one-to-one correspondence between input and output elements. In simpler terms, each input value maps to a unique output value, and conversely, each output value has a unique pre-image in the input space. This property is crucial in cryptographic applications as it ensures the invertibility of the S-box transformation, allowing for the successful decryption of ciphertext back to its original plaintext.

6.2. Non-linearity Property

The non-linearity $N(f)$ of a Boolean function $f: B^n \rightarrow B$ with n input bits is calculated using the formula [21]:

$$N(f)=2^{n-1}-\frac{1}{2}max\sum(-1)^{f(x)+a.x} \qquad a,\ x\in B^n$$

6.3. Strict Avalanche Criterion (SAC)

The strict avalanche criterion (SAC) is an important property for symmetric encryption algorithms. It states that each input bit should cause every output bit to change with a probability of half. This means that if a single input bit is flipped, each output bit should change with a 50% probability [22].

6.4. Bit Independence Criterion (BIC)

The Output Bit Independence Criterion (BIC) is a measure used in cryptographic analysis to evaluate the independence of output bits in a cryptographic function, particularly in the context of substitution boxes (S-boxes) used in symmetric encryption algorithms [23].

The equation for the Bit Independence Criterion (BIC) is as follows:

$$BIC(f)=\frac{1}{2^{2n}}\sum_{x,y\in 0,1}(-1)^{f(x)\oplus f(x\oplus y)}$$

7. Feistel Networks

In 1973, IBM engineer Horst Feistel designed a block cipher as follows [19]:

1. Split the 64-bit block into two 32-bit halves, L and R.
2. Set L to L ⊕ F(R), where F is a substitution–permutation round.
3. Swap the values of L and R.
4. Iterate through step 2 to step 3, 15 times.
5. Swap the values of L and R.
6. Merge L and R into the 64-bit output block.

This construction became known as a Feistel scheme. For encryption, note that the first F takes a first round key K1, and the second F takes another round key K2 and so on. As for decryption, the same algorithm is performed on the ciphertext but the keys are in reverse order.

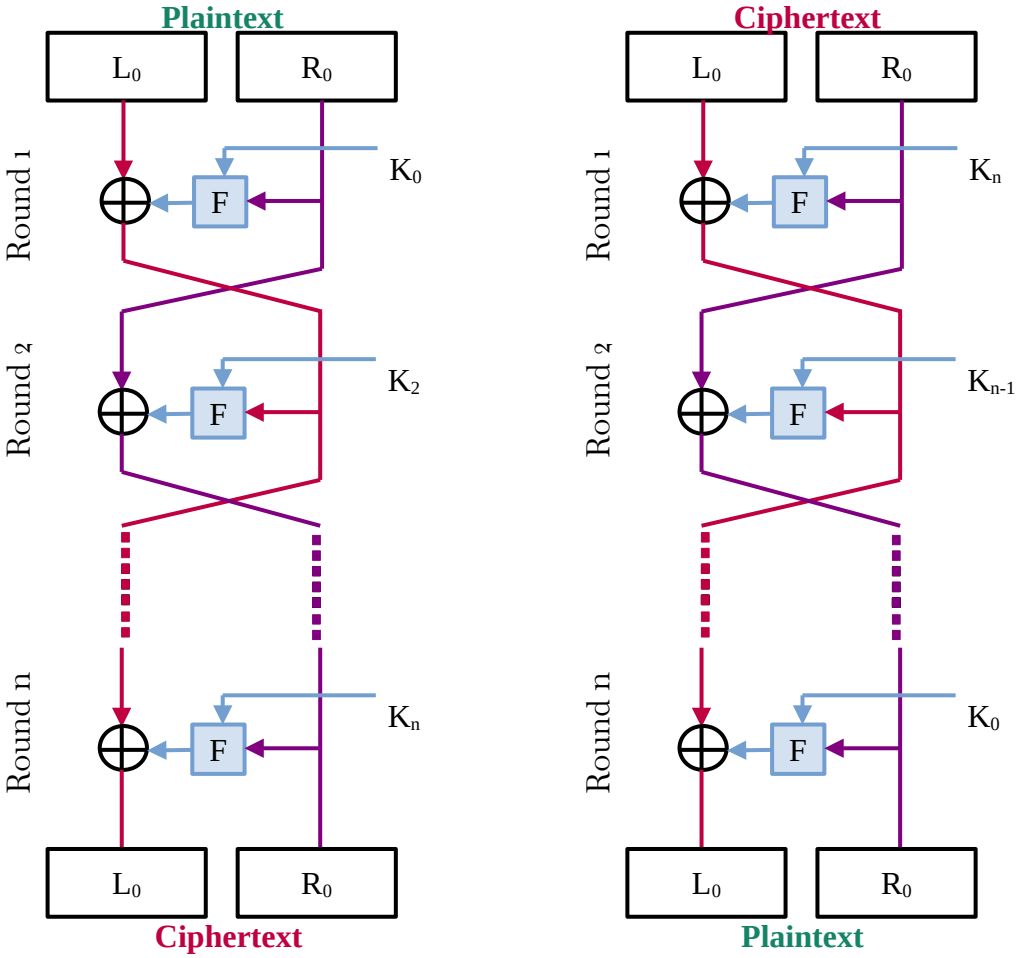


Figure 25: Feistel encryption/decryption structure

8.Conclusion

This chapter provided a comprehensive overview of cryptography, covering key terminologies and branches of cryptography from classical to modern encryption techniques, including symmetric and asymmetric encryption methods. Moving forward, subsequent chapters will delve into practical applications, proposing a cipher block algorithm for image encryption.

Chapter III

Chaos Based Cryptosystem

1.Introduction

As of today, it is no exaggeration to say that most encryption algorithms are designed based on chaotic maps. A typical way is to set the chaotic map as a pseudo-random number generator (PRNG). In this chapter, a new chaotic system is proposed by introducing a linear state feedback controller to the chaotic system by Guanrong Chen [14], then extensive simulation is done to produce an S-Box with robust features.

2.Background on constructing chaotic systems

Researchers tend to adapt simple 1D chaotic maps such as Logistic map to achieve high efficiency PRNG [24]. However, these simple systems have some drawbacks which may cause security hazards to encryption process. Firstly, most of them have a narrow and discontinuous chaotic range on their control parameter that is often served as a secret key in encryption. It means that the key space is limited, which makes the cryptosystem vulnerable to be attacked. Secondly, some chaotic maps are quite sensitive to the finite precision effect of computer and may degrade to non-chaotic immediately. Besides, the chaotic sequences generated by iterating the chaotic maps, usually have a non-uniform distribution, which reduces the randomness of cipher. Nevertheless, the principal features of chaos are exhibited for limited or discontinuous range in most of the conventional chaotic maps or systems. To overcome this deficiency, novel maps and systems are proposed in several works to make it more suitable for the key generation of a cryptosystem. These novel dynamical systems are developed predominantly by two methods, generalization and extended forms of the conventional maps and systems. Generalization of discrete maps is done by introducing extra parameters and thereby adding extra degrees of freedom. There are two methods of introducing the extra parameters in the map equations, which are scaling parameters and shaping parameters. In scaling, the parameters are introduced using additions and multiplications, whereas in shaping, the parameters are

introduced using the powering function. As for the extended forms, it is by adding more dimensions to the system.

2.Construction of a new chaotic system

Let consider the following system:

$$\begin{aligned}\dot{x}&=a(-y-x)\\ \dot{y}&=-xz+cy\\ \dot{z}&=xy-bz+u\end{aligned}$$

Where u is a linear state feedback controller as follow: $u=k_1x+k_2y+k_3z$

Our system becomes:

$$\begin{aligned}\dot{x}&=a(-y-x)\\ \dot{y}&=-xz+cy\\ \dot{z}&=(k_1+y)x+k_2y+(k_3-b)z\end{aligned}$$

The Jacobian of this equation evaluated at (x,y,z) is given by:

$$J_{(x,y,z)}=\begin{pmatrix}-a & -a & 0\\ -z & c & -x\\ y+k_1 & x+k_2 & k_3-b\end{pmatrix}$$

The Jacobian at $(x,y,z)=(0,0,0)$ gives :

$$J_{(0,0,0)}=\begin{pmatrix}-a & -a & 0\\ 0 & c & 0\\ k_1 & k_2 & k_3-b\end{pmatrix}$$

The $det(J_{(0,0,0)}-\lambda I)=0$ is given by:

$$(k_3-b-\lambda)(-a-\lambda)(c-\lambda)=0$$

We may choose $k_1=k_2=0$ since the characteristic equation is free from k_1 and k_2.

Replacing k_3 with d, the proposed new chaotic system lead to only having four parameters (a,b,c,d).

The newly proposed chaotic system:

$$\begin{aligned}\dot{x}&=a(-y-x)\\ \dot{y}&=-xz+cy\\ \dot{z}&=xy+y+(d-b)z\end{aligned}$$

After extensive simulation and trials, the parameters of the system are choosen as follow: $a=49$, $b=18.2$, $c=35$, $d=12.2$

3.Performance analysis of the proposed chaotic system

Numerical simulation is done using Julia programming language.

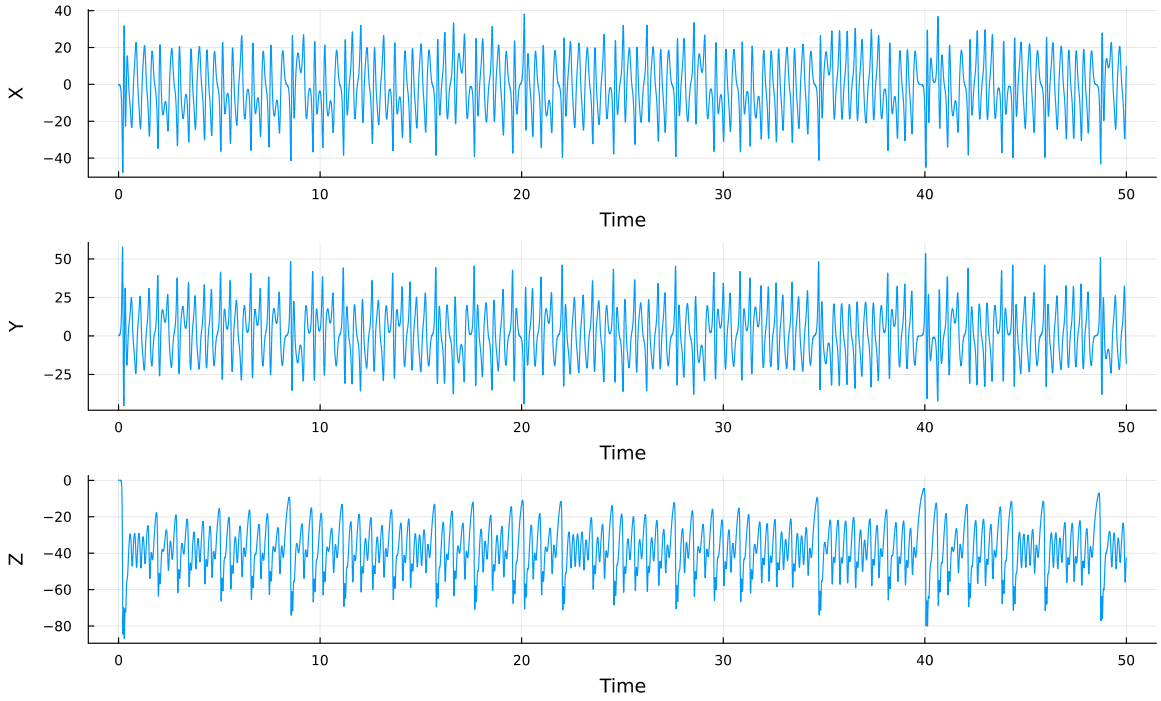


Figure 26: Time Series Analysis

Initial conditions 1:[0.1, 0.1, 0.1], Initial conditions 2: [0.1000000000001, 0.1, 0.1]

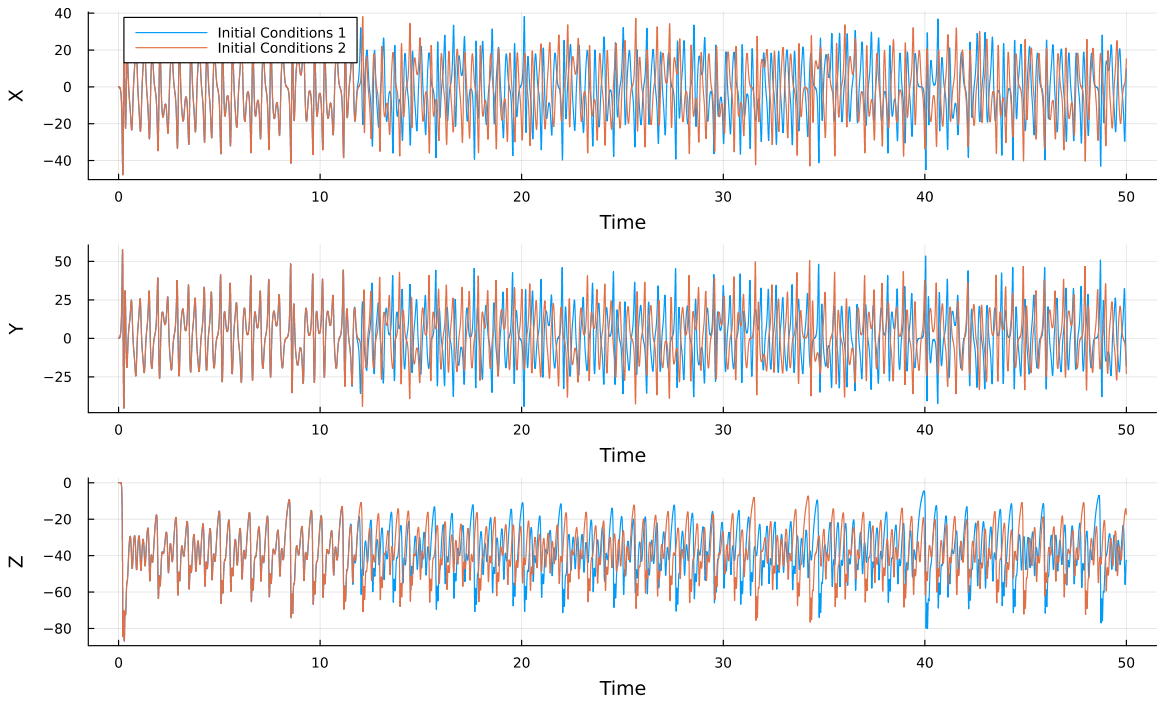


Figure 27: Sensetivity to Initial Conditions for the Proposed system

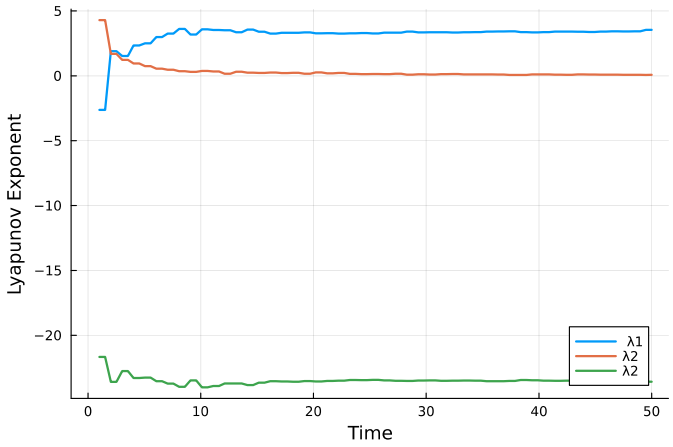


Figure 29: Lyapunov Exponent for the Proposed system

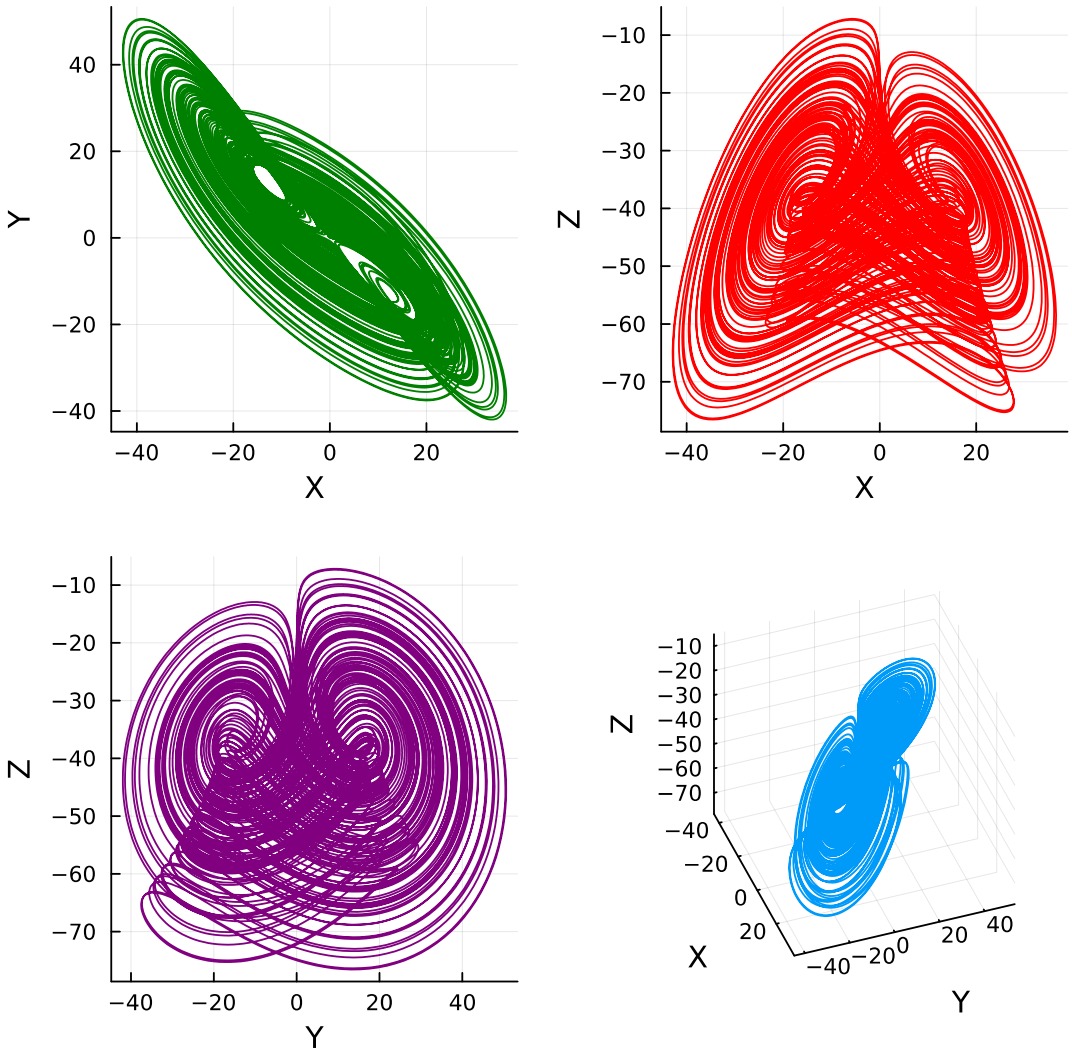


Figure 28: Phase Portrait of the Proposed system

4.The proposed S-Box generation algorithm

In this section, the method conceived to construct an S-box matrix is based on state variables outcomes of the chaotic system proposed in section 2. These steps can be summarised as follows:

Algorithm 1:

1. Choose initial conditions for the chaotic system.
2. Initiate an empty vector $S[256]$ and $j=0$, with j the current cursor position on the vector.
3. Implement a discretization function using Runge-Kutta method.
4. Take the output of the chaotic system as x, y, z.
5. Consider $f(x)=(|x| \bmod 1)\times 256, \quad X=f(x),\ Y=f(y),\ Z=f(z)$
6. If $X \notin S$ and $X \neq j$, then push X to S. If $Y \notin S$ and $Y \neq j$, then push Y to S. If $Z \notin S$ and $Z \neq j$, then push Z to S. Else pass to the next iteration.
7. Iterate through 4 to 6 until the vector S is filled.

Table 1: S-Box Generated using Algorithm 1

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	0x43	0x18	0xE5	0x60	0x3F	0xD0	0xF5	0xA6	0x1A	0xCE	0x12	0x70	0x27	0xB5	0x13	0xC8
10	0x58	0xBF	0x82	0xB2	0x89	0x1D	0xD9	0x7B	0x46	0xED	0x77	0x57	0x0C	0xCD	0x67	0x9E
20	0x3C	0x62	0xA7	0x30	0x97	0x76	0xA1	0x55	0x9A	0xB6	0x8F	0xAE	0x5D	0x3D	0x1B	0x8E
30	0x7E	0x3B	0xB3	0x8D	0x0F	0x66	0xD8	0xF7	0x22	0x31	0x7F	0x4A	0xE3	0x01	0xF3	0x47
40	0xBD	0x98	0xA4	0x2D	0xBC	0x2E	0x86	0xFD	0xE9	0x21	0x93	0x9C	0x99	0x10	0x74	0x44
50	0xEB	0x50	0x9B	0x1E	0x5C	0xFF	0x00	0x29	0xF0	0xE0	0x94	0xA8	0x1F	0xBB	0xA5	0x6A
60	0xBE	0xCA	0xEC	0x5E	0xC0	0x42	0xF6	0x35	0x8C	0xB7	0x49	0x4B	0xB4	0x20	0xF4	0x04
70	0x1C	0x54	0x5F	0xDC	0x2F	0xFB	0x79	0xC6	0x0A	0x4F	0xC1	0xAB	0xD4	0x0E	0x5B	0xDB
80	0x9F	0x32	0xF2	0x41	0x40	0xDA	0x23	0x4D	0x48	0x06	0x15	0xA9	0x88	0x2B	0x64	0x6F
90	0x95	0x34	0xBA	0xE4	0xB0	0x8B	0xC4	0x85	0x25	0xFC	0xA2	0x45	0x05	0x51	0x07	0x37
A0	0x96	0xB1	0xA3	0x56	0x38	0x7C	0x6B	0x52	0x19	0x39	0x33	0xDD	0x28	0xEF	0x0B	0xE1
B0	0x75	0xA0	0x65	0xAA	0x83	0x5A	0xE8	0xE2	0xC7	0x6D	0x53	0x71	0x09	0xFA	0x08	0x73
C0	0xCF	0x26	0xF1	0x17	0x69	0x80	0x3A	0x84	0x91	0x24	0xAC	0x36	0xEE	0xC9	0x90	0x78
D0	0x61	0xD5	0x59	0xC2	0x87	0x7D	0xF9	0x11	0x92	0xEA	0x72	0xD6	0x7A	0x2A	0x03	0xD2
E0	0x4E	0x4C	0x14	0xAF	0x6E	0xAD	0x16	0x02	0xDE	0xCB	0xF8	0xD1	0xFE	0xC3	0xB9	0xE7
F0	0xE6	0x3E	0xC5	0xCC	0x0D	0xD7	0xDF	0x68	0xB8	0x81	0x9D	0x6C	0x63	0xD3	0x8A	0x2C

5.Performance evaluation of the proposed S-box

To ascertain the robustness and resilience of the generated S-Box against attacks like linear and differential cryptanalysis, it's imperative to subject it to various performance tests. These tests primarily include assessing non-linearity, evaluating the BIC, and applying the SAC. Furthermore, a performance comparisons between the generated S-Box and other similar work.

Table 2: Performance test of the SBox

SBox	Non-linearity			BIC-SAC	BIC Non-linearity	SAC		
	Min	Max	Average			Min	Max	Average
Proposed	**106**	**110**	**107.5**	**0.4944**	**103.91**	**0.375**	**0.5938**	**0.4995**
Ref[25]	104	108	106.25	0.4996	103.643	0.4453	0.5546	0.5009
Ref[26]	106	108	106.75	0.4951	100	0.4219	0.625	0.5034
Ref[27]	106	108	106.7	0.4957	103.5	0.3909	0.6094	0.4941
Ref[28]	104	110	106	0.4988	103.857	0.4018	0.5781	0.4946

Table 3: Detailed SAC test result

0.5156	0.4531	0.4688	0.4688	0.4688	0.4531	0.5156	0.4531
0.4844	0.5156	0.4688	0.5156	0.5156	0.4844	0.5469	0.5781
0.4062	0.5312	0.4219	0.5	0.4531	0.5156	0.5469	0.5
0.4531	0.4688	0.4844	0.4219	0.4531	0.5156	0.4531	0.5312
0.5	0.5781	0.4531	0.5469	0.4688	0.4375	0.5156	0.5469
0.5625	0.5469	0.5156	0.5156	0.5469	0.4844	0.5312	0.5469
0.4531	0.5156	0.4844	0.4844	0.5156	0.5	0.5312	0.4844
0.5781	0.375	0.4844	0.5469	0.5938	0.4688	0.5469	0.5469

Table 4: Detailed result of the BIC-Non-linearity test

0	106	106	100	104	104	100	104
106	0	102	100	102	104	106	106
106	102	0	100	104	104	104	106
100	100	100	0	104	102	106	108
104	102	104	104	0	106	102	104
104	104	104	102	106	0	102	106
100	106	104	106	102	102	0	102
104	106	106	108	104	106	102	0

Table 2 indicates the different values of NL, BIC, SAC for the S-Box. The average of Non-Linearity is 107.5 which is the highest among other S-Box, SAC averaged a value of 0.4995 which is close to the ideal value of 0.5. For the BIC-SAC values, It is also around the desired value of 0.5 (average value: 0.4944). Comparison with other S-Boxes allows us to conclude that the proposed S-Box exhibits remarkable performance results, making it more robust against attacks and therefore more suitable for encryption algorithms.

6. Cipher Block Algorithm

This algorithm is based on a Feistel network, which includes elements inspired by the AES standard, such as: *SubByte*, *ShiftRows*, *MixColumns* and *AddRoundKey*. In addition, a Counter Mode (CTR) is adopted for more simplicity and efficiency.

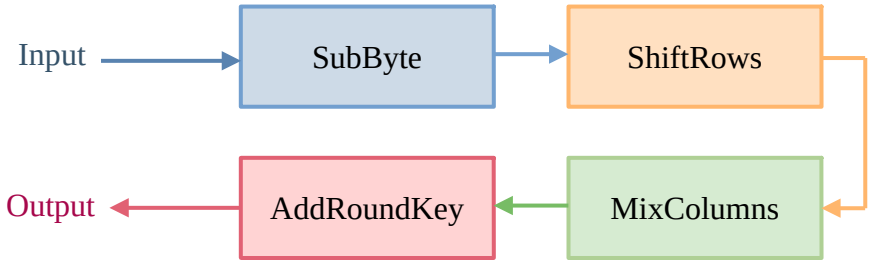


Figure 30: Function used in Feistel Network

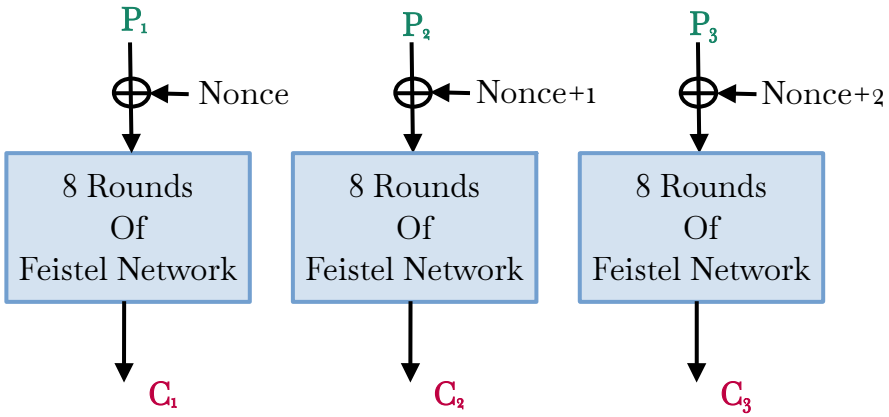


Figure 31: Encryption Scheme

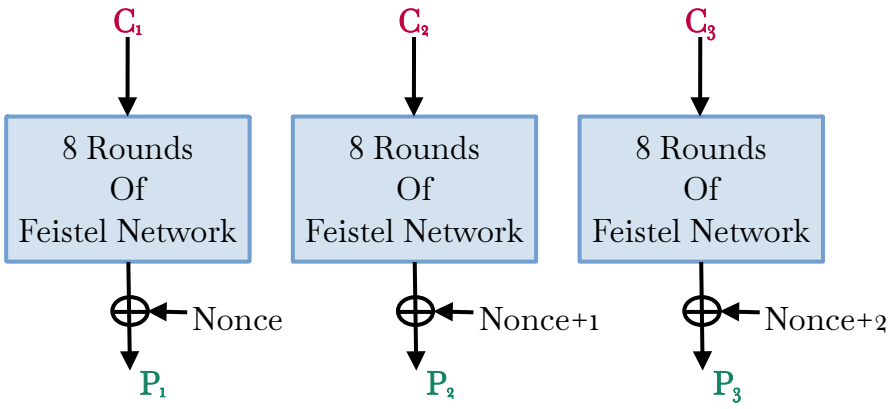


Figure 32: Decryption Scheme

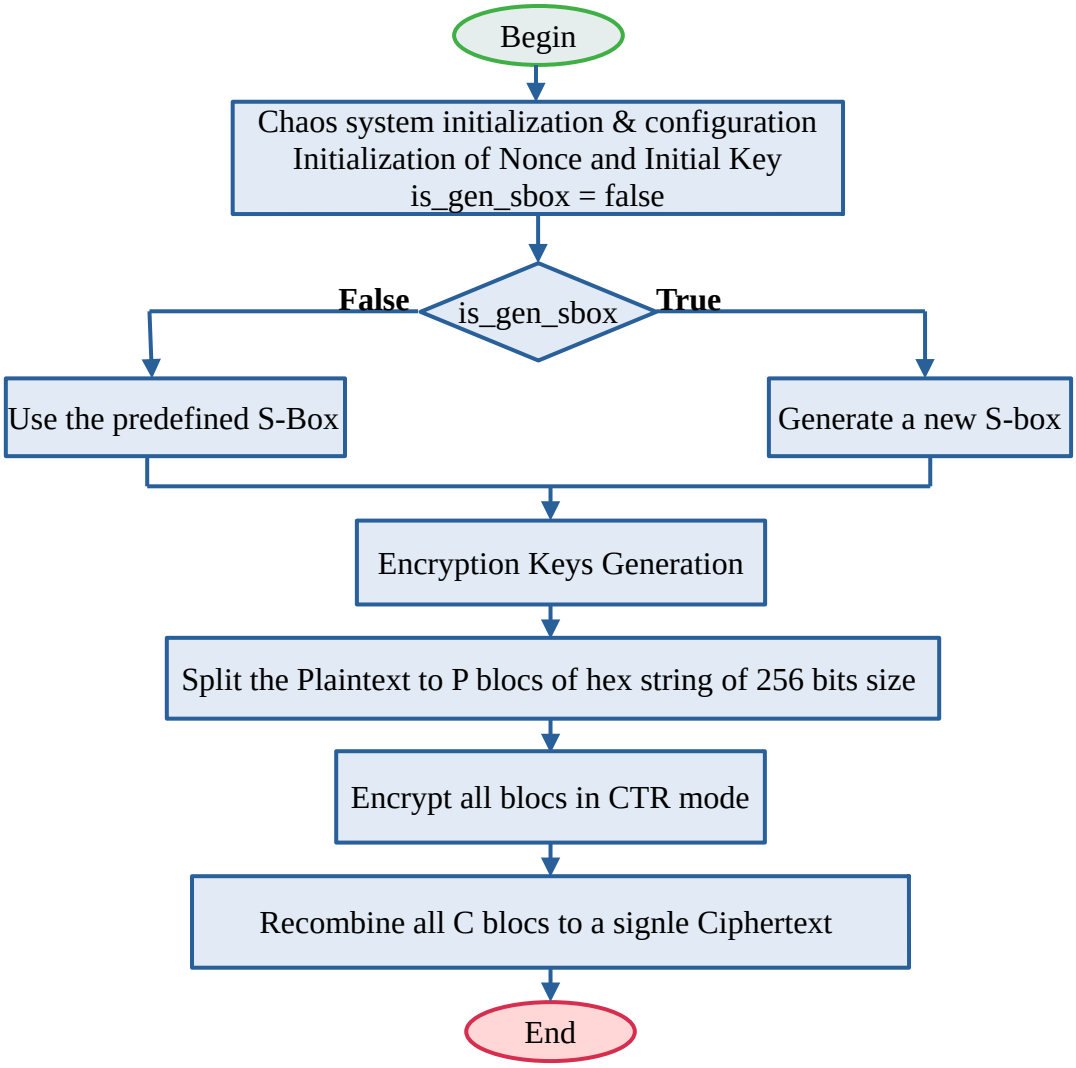


Figure 33: Flowchart of the Proposed Block Cipher

7.Conclusion

This chapter was a detailed description on the theoretical aspect of the cryptosystem, by constructing a new chaotic system and using it as a framework to produce robust S-Box and finally designing a scheme of the cipher block algorithm. All these are the foundation for the practical implementation process in the next chapter.

Chapter IV
FPGA Implementation
Results and Analysis

1.Introduction

This chapter is the bearing fruit of all the previous chapters, describing goals to achieve, implementation process, design requirements, illustration of the results and finally a comparison to similar state-of-art works

2.Overview of FPGA

Field-programmable gate arrays (FPGAs) are reconfigurable computer chips that can be programmed to implement any digital hardware circuit. FPGAs consist of an array of different types of programmable blocks (logic, IO, and others) that can be flexibly interconnected using pre-fabricated routing tracks with programmable switches between them. The functionality of all the FPGA blocks and the configuration of the routing switches are controlled using millions of static random access memory (SRAM) cells that are programmed (i.e. written) at runtime to realize a specific function. The user describes the desired functionality in a hardware description language (HDL) such as Verilog or VHDL, or possibly uses high-level synthesis to translate C or OpenCL to HDL. The HDL design is then compiled using a complex computer-aided design (CAD) flow into the bitstream file used to program all the FPGA's configuration SRAM cells [18]. Compared to building a custom application-specific integrated circuit (ASIC), FPGAs have a much lower non-recurring engineering cost and shorter time-to-market. A pre-fabricated off-the-shelf FPGA can be used to implement a complete system in a matter of weeks, skipping the physical design, layout, fabrication, and verification stages that a custom ASIC would normally go through. They also allow continuous hardware upgrades to support new features or fix bugs by simply loading a new bitstream after deployment in-field, thus the name *field-programmable* [29].

3.Design Requirements

3.1.Hardware

An FPGA card and a computer is used with the following specifications:

- *CPU*: AMD Ryzen 7 4800U (16) @ 1.80 GHz.
- *OS*: Arch Linux
- *RAM*: 16 GB.

FPGAs consist of three types of modules:

- *Configurable Logic Block (CLB)*: It is a fundamental building block of an FPGA. Each CLB contains multiple slices. Each slice consists of function generators (LUTs), registers, multiplexers, carry logic etc.
- *Switch matrix or interconnection wires*: Signals between CLBs and from CLBs to IOBs are routed through Switch matrices.
- *IO blocks (IOB)*: An IOB is the basic input/output function block.

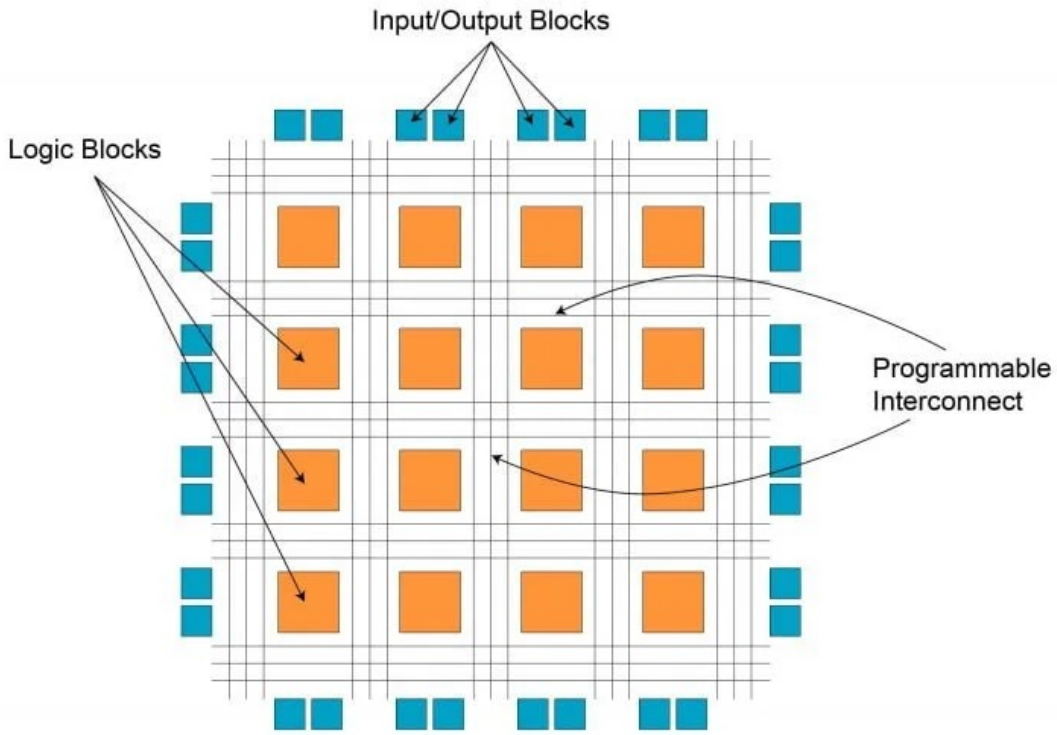


Figure 34: Architecture of an FPGA Device

An FPGA Nexys A7 100T card is used in practical implementation, this versatile platform offers a combination of power and flexibility making it a compelling solution for medium and small volume designs [30].

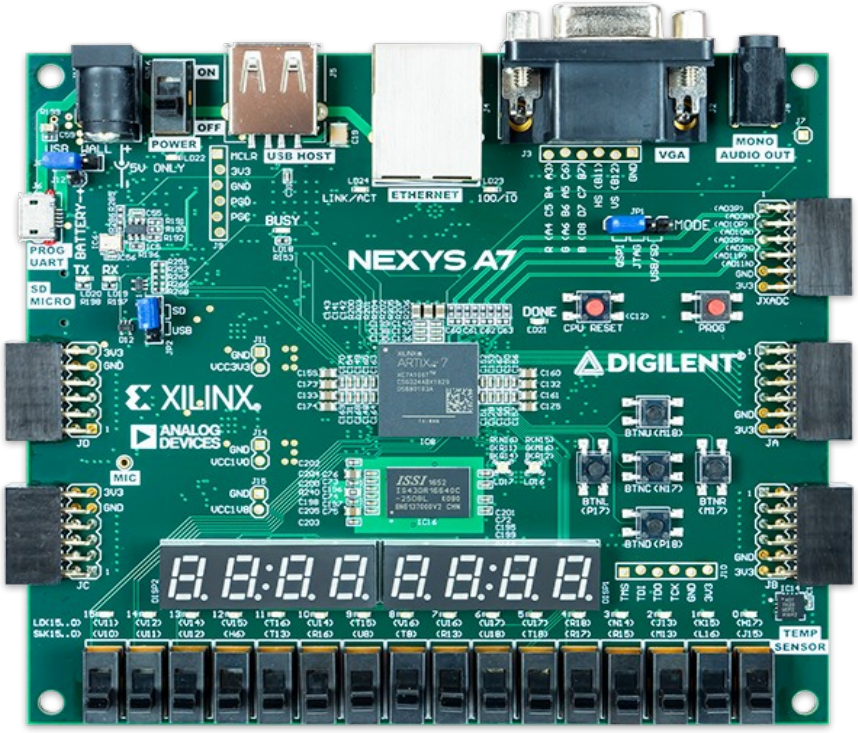


Figure 35: The FPGA card Nexys A7 100T

Table 5: Characteristics of the FPGA Nexys A7 100T.

FPGA Part Number	XC7A100T-1CSG324C
Look-up Tables (LUTs)	63,400
Flip-Flops	126800
Block RAM	4,860 Kb
DSP Slices	240
Clock Management Tiles	6

3.2.Software

For the design implementation, Vivado Design Suite is used. It provide three core tools, namely, Vivado HLS, Vivado HLx and Vivado SDK.

Vivado HLS

Vivado HLS (High-Level Synthesis) streamlines FPGA development by allowing engineers to write code in high-level languages like C or C++ instead of traditional HDLs. It automatically synthesizes this code into optimized hardware, reducing design time and offering advanced optimization and debugging features.

Vivado HLx

Vivado HLx is an advanced edition of Xilinx's Vivado software suite tailored for high-level FPGA development. It offers enhanced features and capabilities compared to the standard edition, including advanced optimization techniques, improved debugging tools, With Vivado HLx, engineers can efficiently design and implement FPGA-based solutions for a wide range of applications.

Vivado SDK

The Software Development Kit (SDK) is the Integrated Design Environment for creating embedded applications on various soft-core microprocessor. The SDK is the first application IDE to deliver true homogenous and heterogeneous multiprocessor design, debug, and performance analysis.

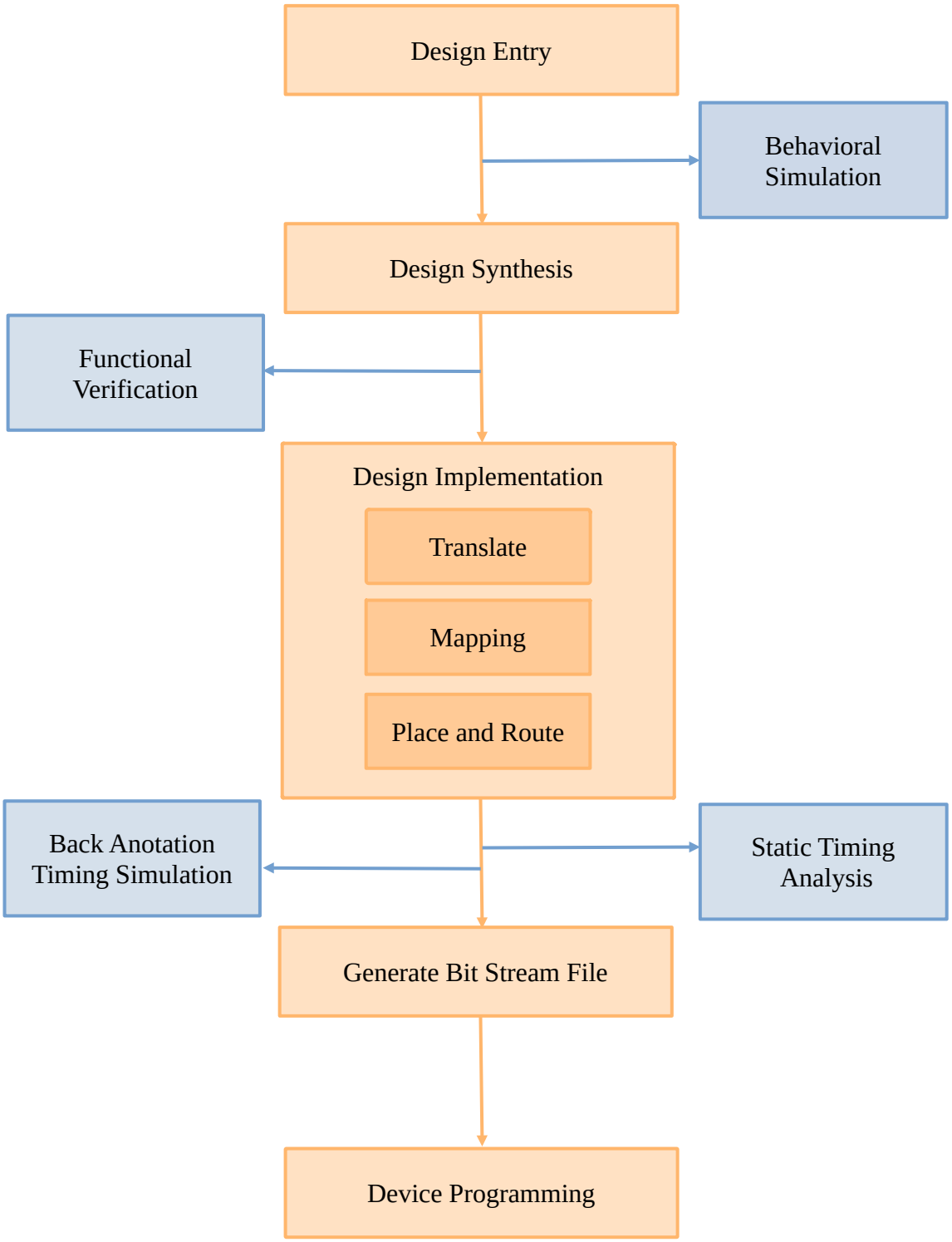


Figure 36: Typical FPGA Design Flow

2.3.IP Core Generation

The implementation begins by writing C++ source code and its corresponding test bench as the foundation of the IP Core in Vivado HLS. These codes allowed us to simulate our Cryptosystem, providing an opportunity to validate its functionality before proceeding to generate the IP core.

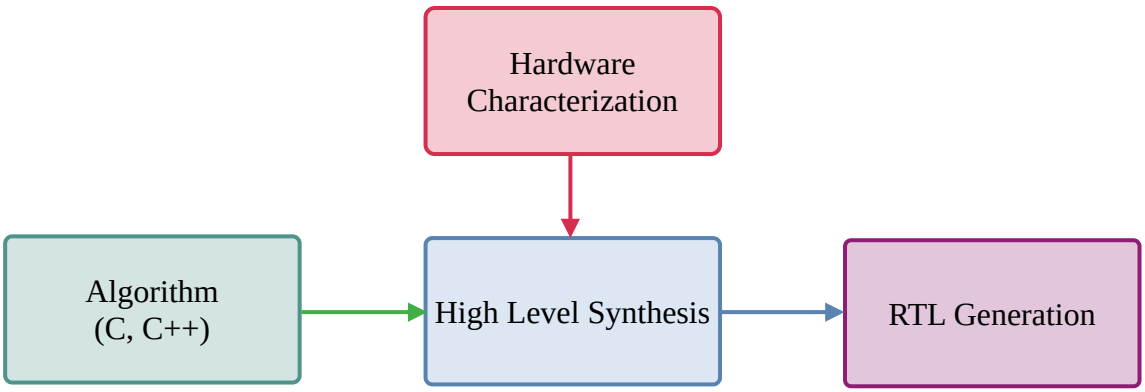


Figure 37: High Level Synthesis Flow

Table 6: Utilization Estimates Summary

	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	1254
FIFO	-	-	-	-
Instance	8	75	16362	20801
Memory	5	-	0	0
Multiplexer	-	-	-	1218
Register	-	-	2830	-
Total	13	75	19192	23273
Available	270	240	126800	63400
Utilization (%)	4	31	15	36

4.Block Design

Designing an IP Block in Vivado HLx to use the exported RTL is crucial, as the IP Core by itself is useless without proper data processing and storage.

List of the used component:

- The cipher block IP Core: This is the exported RTL from Vivado HLS.
- Microblaze: it is AMD 32/64-bit RISC Harvard architecture soft processor core with a rich instruction set optimized for embedded applications. The MicroBlaze soft processor solution delivers complete flexibility to select the combination of peripheral, memory and interface features that will give you the exact system you need at the lowest cost possible on a single FPGA.
- AXI UART: The UART (Universal Asynchronous Receiver/Transmitter) primary function is to manage the flow of data asynchronously.
- AXI DMA: The AXI Direct Memory Access IP provides high-bandwidth direct memory access between memory and AXI4-Stream-type target peripherals.
- AXI BRAM Controller: The core is designed as an AXI endpoint slave IP for integration with the AXI interconnect and system master devices to communicate to local BRAM. The core supports both single and burst transactions to the BRAM and is optimized for performance.
- AXI Timer: The time would be used to calculate the overall runtime cycles.

To finalize the design, all the wires connecting different parts should be correctly configured. Fortunately, Vivado software handles the arrangement of most wires automatically. Additionally, the software assists by generating extra components that facilitate smooth operation. These components manage memory, handle communication between ports and devices, generate internal clock signals, and perform other tasks.

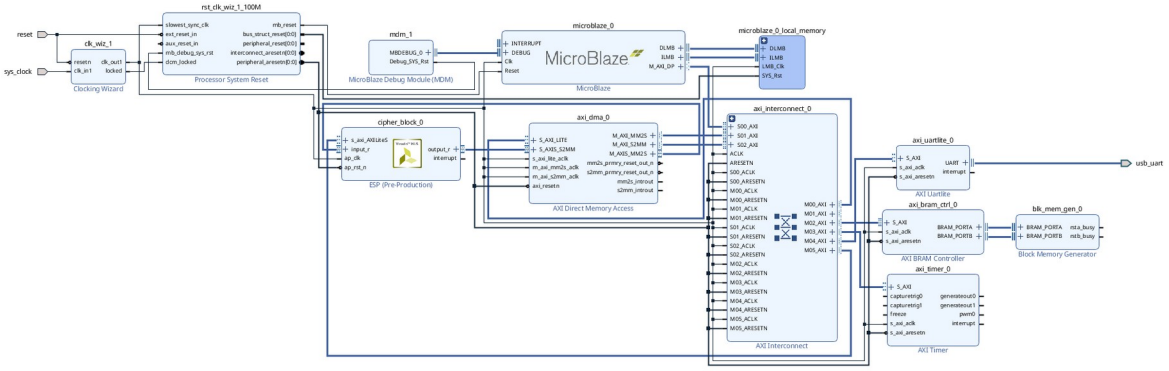


Figure 38: Block Design Overview

5.Implementation

For testing the designed cryptosystem, a C source code is written whithin Vivado SDK. This program is designed to load a grayscale image (256x256 resolution) into the BRAM memory and then transfer it to our IP core in 256-bit blocks using DMA. The encrypted blocks produced by our IP core are organized into a buffer to compose the encrypted image.

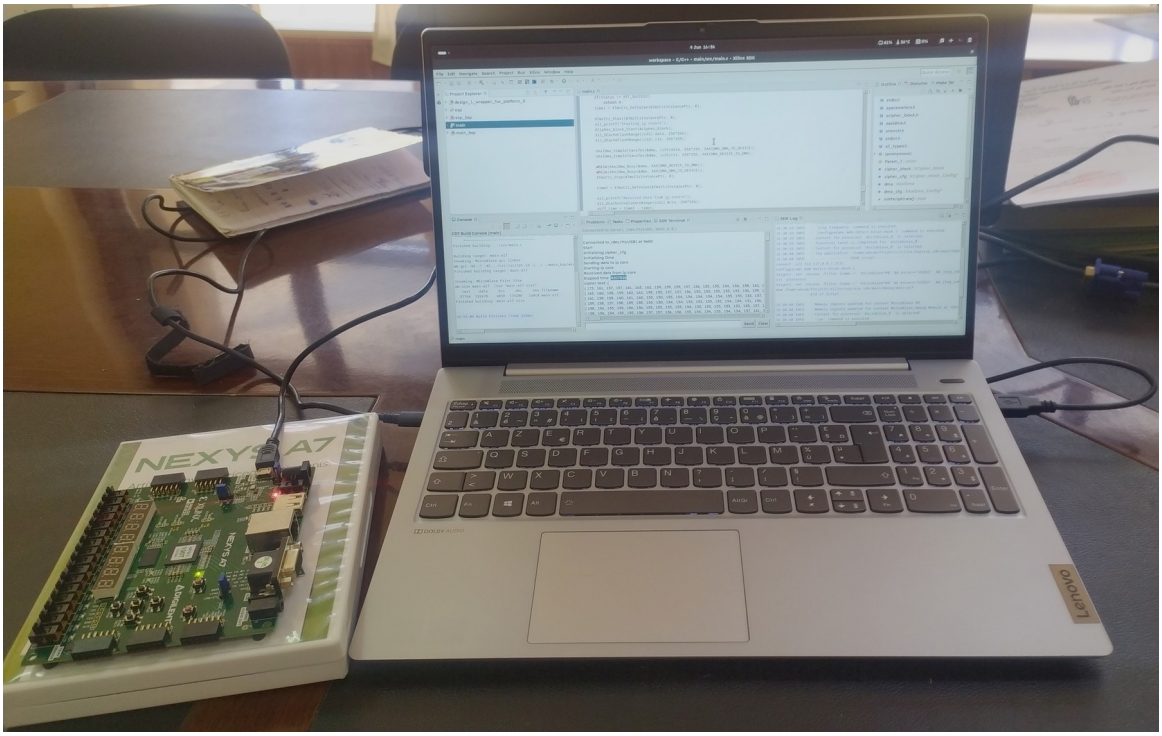


Figure 39: FPGA Implementation Process

6.Results

To evaluate the efficiency of the cryptosystem, several standard testing images are used including "Cameraman", "Peppers", "Tree" and "Clock" at a resolution of 256x256. Additional tests are conducted, namely, entropy, histogram, correlation coefficient, NPCR and UACI.

6.1.Correlation Coefficient

Another important test in statistical analysis is the correlation coefficient. For a given image, the correlation coefficient measures the degree of similarity of adjacent pixels. In cryptography, the dependence of adjacent pixels must be broken, hence a correlation coefficient close to zero is always desirable for a cryptosystem [36]. The correlation coefficient can be calculated using the following formula:

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

Where N represents the number of pixels, x_i and y_i are two adjacent pixels, $\bar{x}=\frac{1}{N}\sum_{i=1}^{N} x_x$ and $\bar{y}=\frac{1}{N}\sum_{i=1}^{N} y_i$ are the average values of the pixels x_i and y_i respectively. Figs 40, 41, 42 and 43 reveal the correlation in the three directions (vertical, horizontal and diagonal) of the test images and their numbers. In addition, the numerical results of the correlation test are reported in Table 7. Thus, we notice that for the different directions, the correlation coefficient is approximately zero, which shows the decorrelation of the pixels of the encrypted image. Therefore, it is plausible to confirm the effectiveness of the proposed encryption algorithm.

Table 7: Correlation Coefficient analysis

Image	Direction	Correlation in Clear Image	Correlation in Encrypted Image
Cameraman	Horizontal	0.9337	-0.0025
	Vertical	0.9592	-0.0063
	Diagonal	0.9086	0.0041
Peppers	Horizontal	0.9666	-0.0038
	Vertical	0.9721	0.0016
	Diagonal	0.9418	0.0051
Clock	Horizontal	0.9565	0.0020
	Vertical	0.9740	0.0029
	Diagonal	0.9389	-0.0011
Tree	Horizontal	0.9676	-0.0058
	Vertical	0.9449	0.0023
	Diagonal	0.9299	0.0017

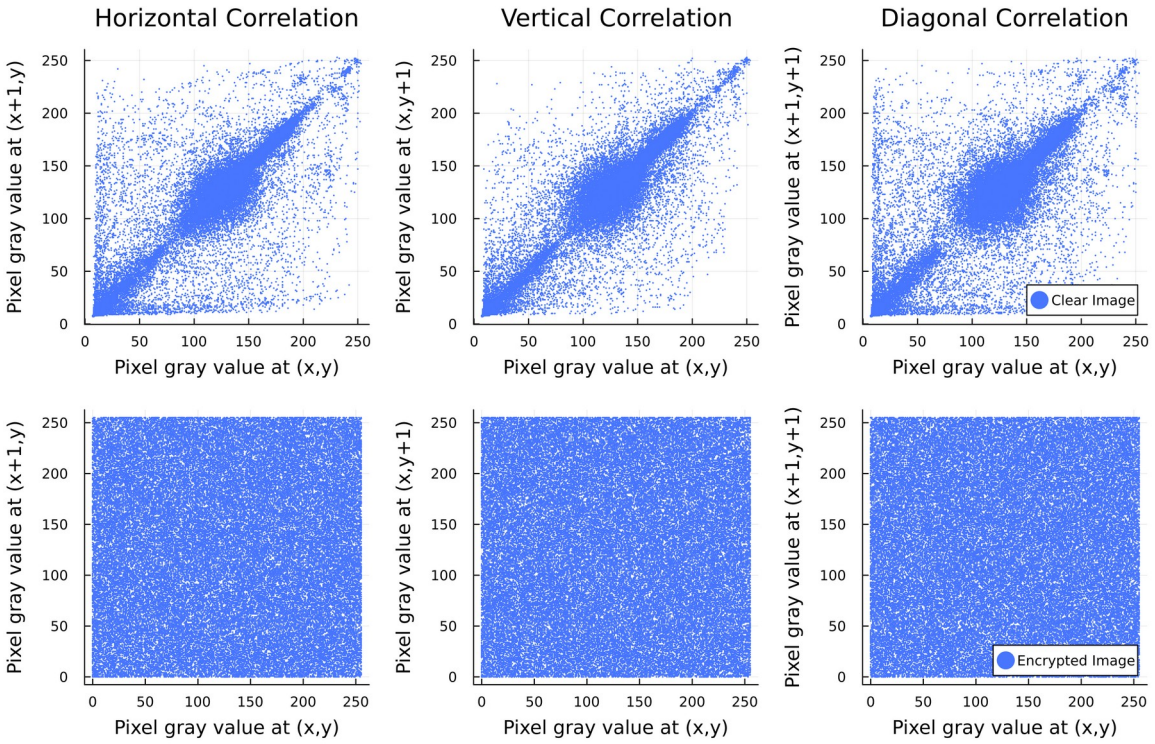


Figure 40: Correlation test on Cameraman Image

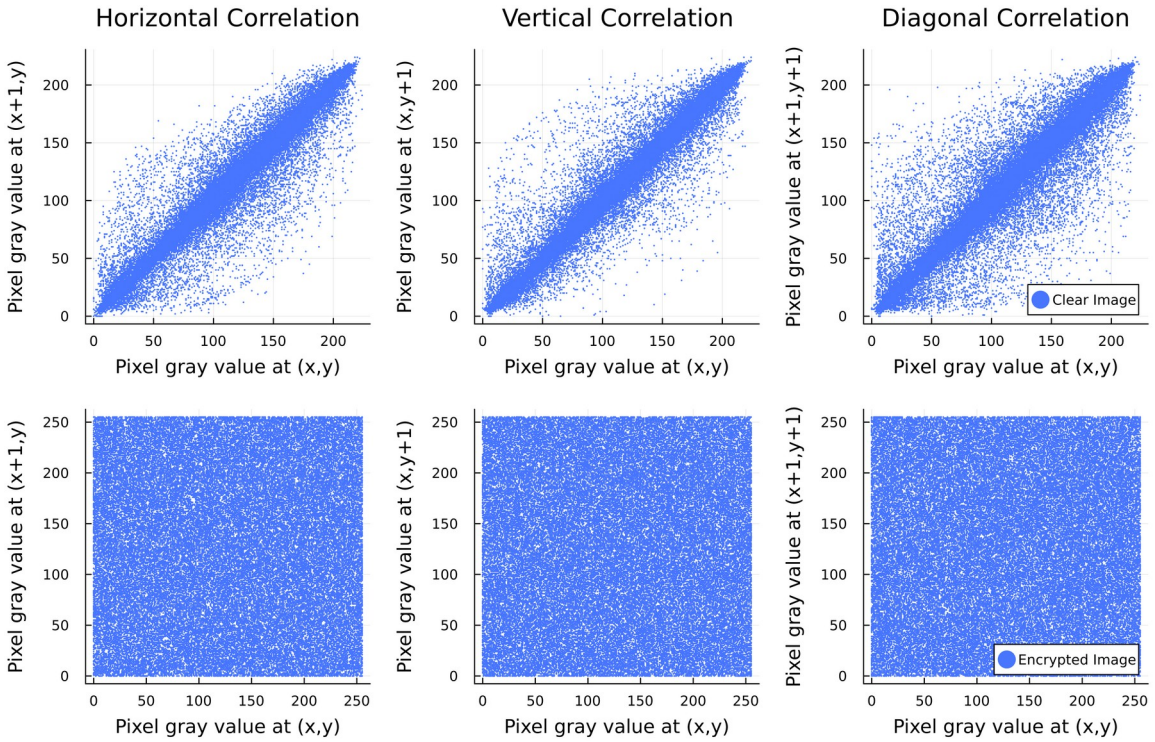


Figure 41: Correlation test on Peppers Image

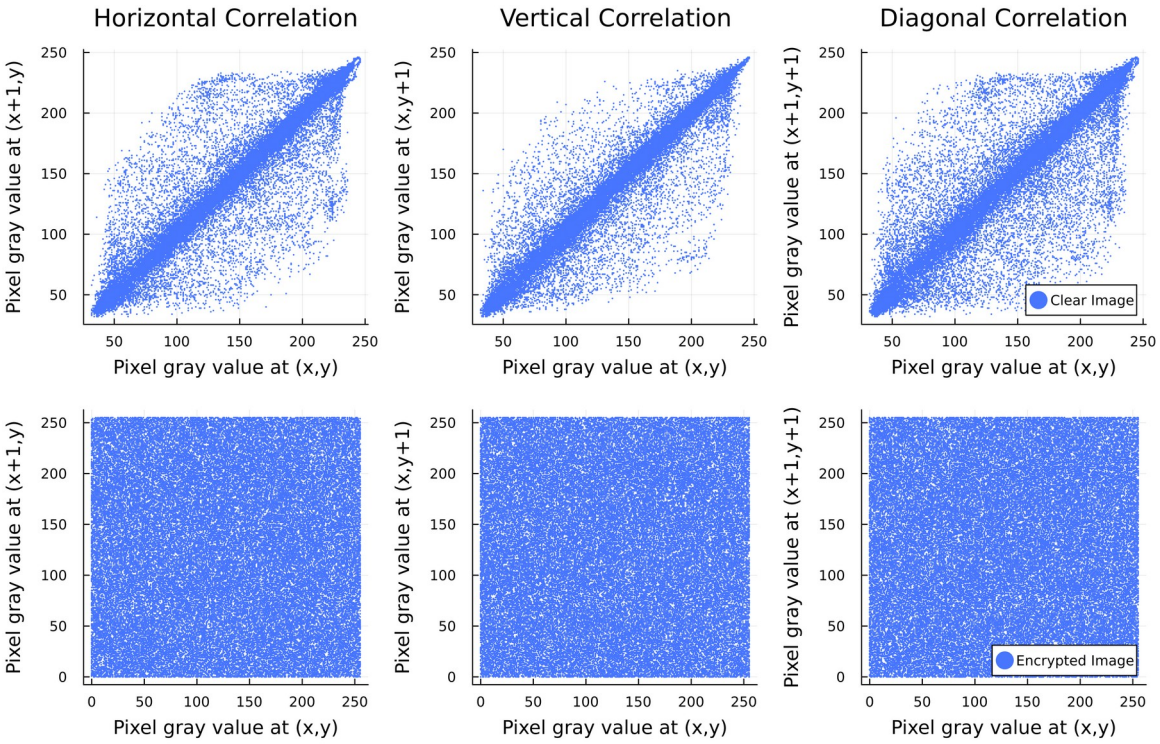


Figure 42: Correlation test on Clock Image

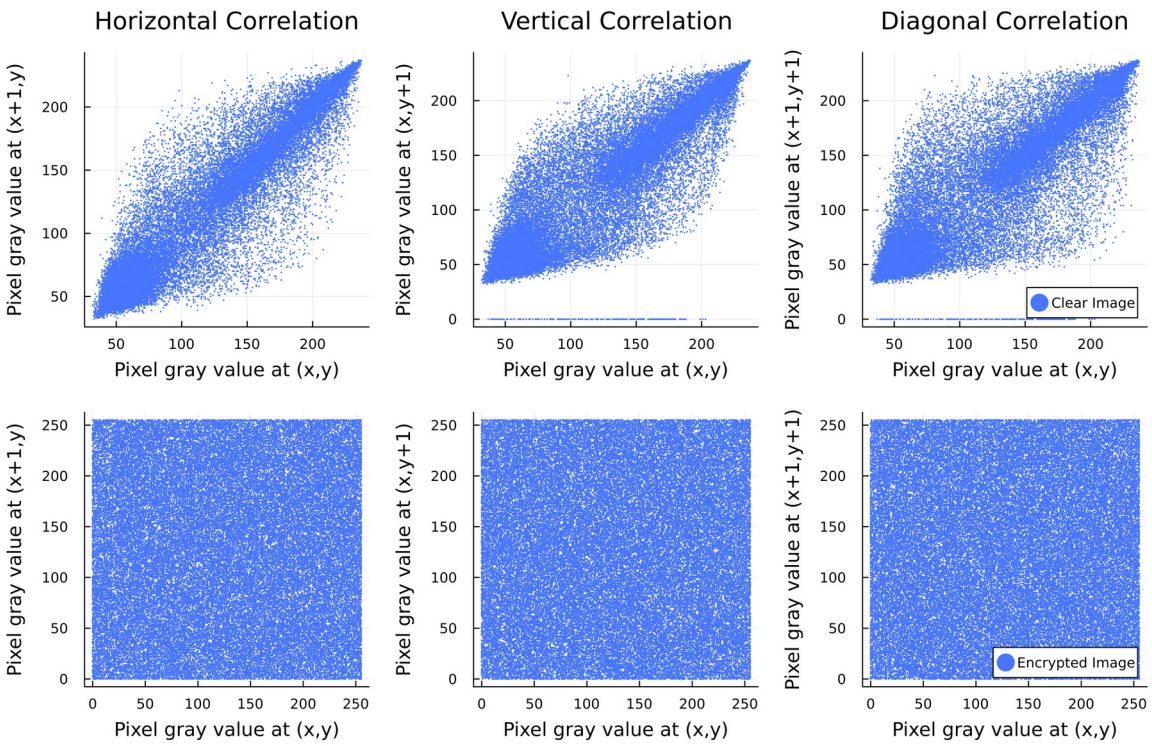


Figure 43: Correlation test on Tree Image

6.2.Key Sensitivity

Original initial key: 0x9DADBECFD0E1F2039DADBECFD0E1F203

Changed initial key: 0x8DADBECFD0E1F2039DADBECFD0E1F203

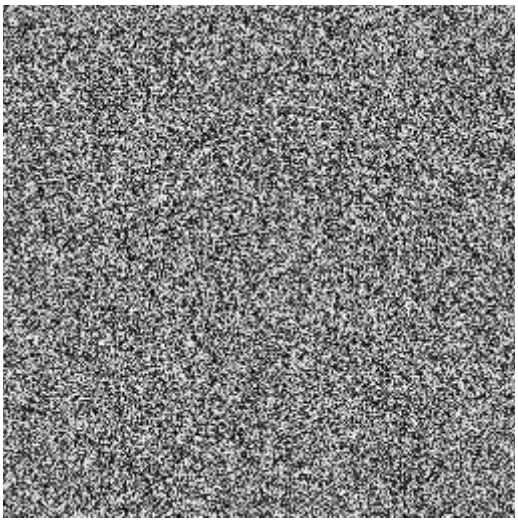


Figure 44: Cameraman Image decrypted with small change in Initial Key

6.3.Histogram

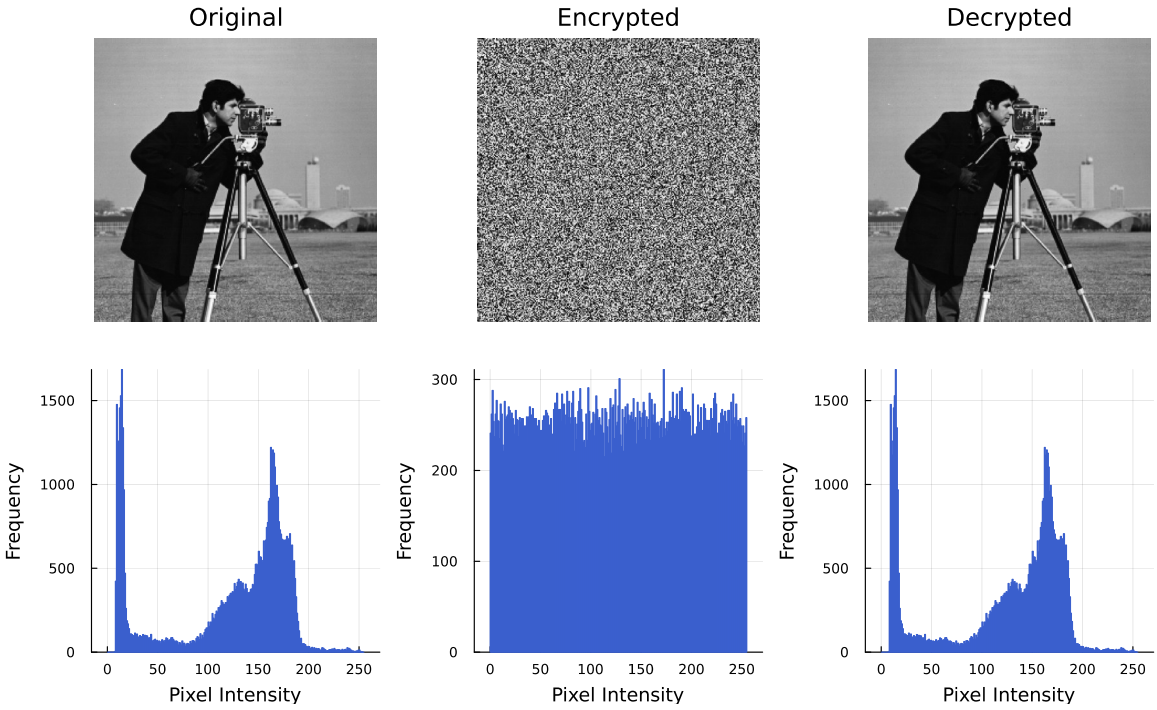


Figure 45: Histogram Analysis of Cameraman Image.

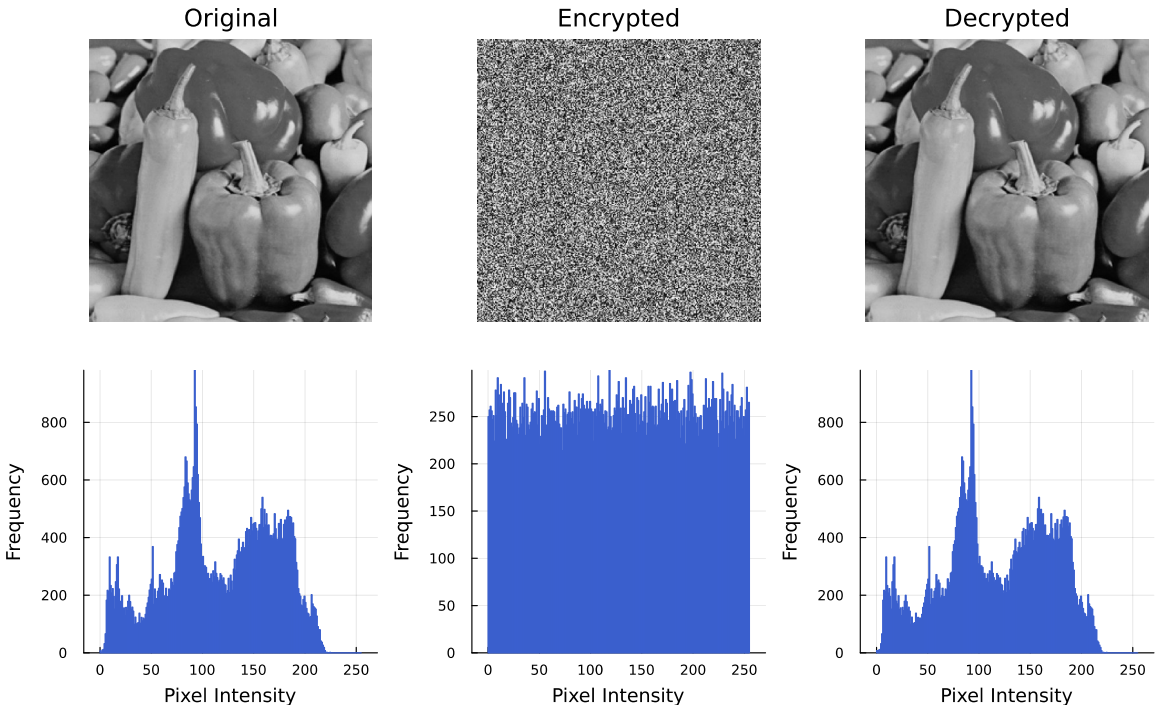


Figure 46: Histogram Analysis of Peppers Image

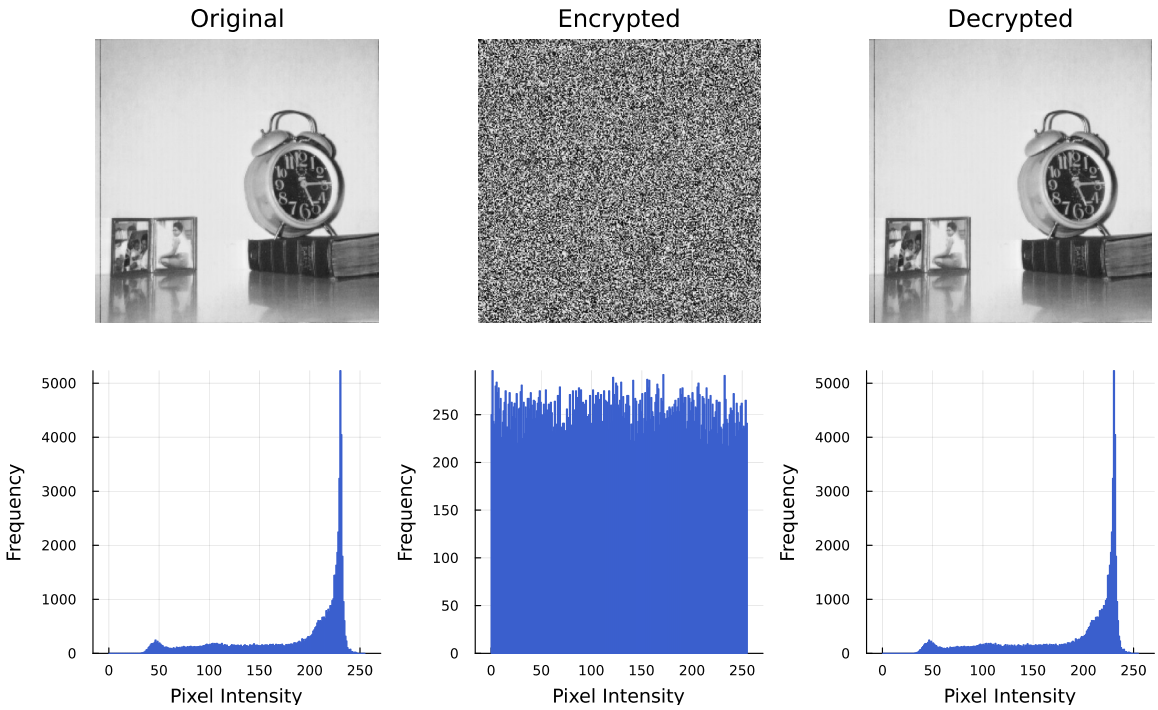


Figure 47: Histogram Analysis of Clock Image

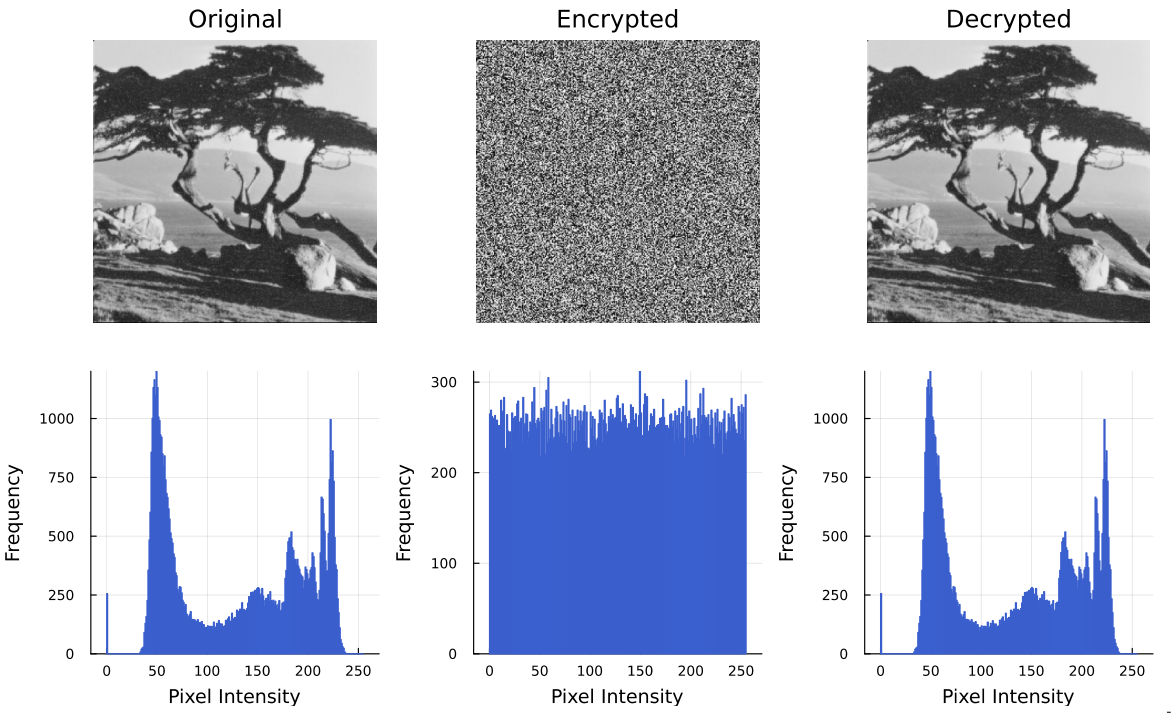


Figure 48: Histogram Analysis of Tree Image

6.4.Diffusion Analysis

For the diffusion test, a differential attack is used. This is based on the comparison between the clear and encrypted image, To measure resistance against differential attacks, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are a standard [31].

These two indices are defined as follows:

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

$$UACI(\%)=\frac{1}{M\times N}\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|I_1(i,j)-I_2(i,j)|}{255}\right]\times 100$$

Where $D(i,j)=\begin{cases}1 \ if\ I_1(i,j)\neq I_2(i,j)\\ 0 if\ I_1(i,j)\neq I_2(i,j)\end{cases}$ and $I_1(i,j)$, $I_2(i,j)$ are pixel intensity in (i, j) for clear image and encrypted image respectively. M, N are resolution size.

Table 8: Diffusion analysis

	Cameraman	Peppers	Clock	Tree
NPCR	99.62%	99.61%	99.60%	99.56%
UACI	49.90%	49.94%	49.64%	50.09%

6.5.Entropy

An entropy test of an image is a statistical measure used to quantify the randomness or complexity of the image's pixel values. Entropy is often used in image processing to assess the information content or texture complexity of an image. Higher entropy values indicate more complexity and randomness, while lower entropy values indicate more uniformity and less information content. A perfect score would be equal to eight for pixels encoded on 8 bits.

It is denoted as : $H=\sum_{i=1}^{N}p(x_i)\times\log_2\left(\frac{1}{p(x_i)}\right)$

where N is the total number of pixels and $p(x_i)$ denotes the probability of occurrence of pixels x_i.

Table 9: Entropy analysis

		Cameraman	Peppers	Clock	Tree
Entropy	Original	7.0097	7.5647	6.7056	7.3102
	Encrypted	7.9970	7.9970	7.9970	7.9971

6.6.Encryption/decryption Runtime

Runtime is a very important characteristic for cipher block. To measure it, the numbers of cycles needed to complete an encryption/decryption is considered, knowing that the Microblaze processor is clocked at the frequency of 100 Mhz.

Runtime is noted as: $T_{Runtime}=\frac{N_c}{CPU_{Freq}}$

Test results of the cameraman image at a resolution of 256x256 shows a encryption and decryption processes take approximately 4207956 and 4207826 respectively, resulting in $T_{Runtime}=42.07\ ms$ when encrypting or decrypting and $84.14\ ms$ for a complete cycle of encryption and decryption.

7.Comparison to state-of-art works

Table 10: Performance Comparison using Peppers image as refrence

Algorithm	Correlation			Entropy	NPCR	UACI
	H	V	D			
Proposed	**-0.0038**	**0.0016**	**0.0051**	**7.9970**	**99.61%**	**49.94%**
Ref[32]	0.0052	-0.0028	-0.0143	7.9971	99.61%	33.46%
Ref[33]	0.0020	0.0080	0.0008	7.9972	99.64%	33.59%
Ref[34]	0.0139	0.0054	0.0153	7.9992	99.61%	33.44%

Table 11: Encryption/decryption time comparison

Algorithm	Encryption time in s	Encryption+Decryption
Proposed	**0.04207**	**0.08414**
Ref[28]	0.55540	1.04850
Ref[33]	-	2.90000
Ref[34]	0.39490	-

8.Conclusion

In this chapter, the experimental aspect of the project is studied, noting the necessary software and hardware required for the development and the implementation on the FPGA board. Finally, multiple performance tests are conducted to validate the efficiency of the algorithm.

General Conclusion

General Conclusion

This study is centered on chaos-based cryptosystem, taking advantages of the features and complexity of chaotic systems to employ a robust S-Box as it is the only non-linear component of the cryptosystem. A pioneering cipher block algorithm was developed using Feistel network and element of the well-known AES algorithm as a foundation for securing sensitive data.

Initially, foundational insights into chaotic dynamical systems were provided. Subsequently, fundamental cryptographic principles are to be recognized to establish a theoretical framework for the investigation.

To probe chaotic phenomena within dynamic systems, extensive computational simulations and trials were made. Through these simulations, an assessment to the susceptibility of chaotic systems to minute perturbations in initial conditions, calculations of lyapunov exponent and an interesting attractor whithin the phase portrait, is a discerning signature of chaotic behavior.

Leveraging a chaotic signal, a systematic approach for generating a substitution matrix (S-Box) is conceived as it is a pivotal component within our encryption methodology. A comprehensive run of performance tests was then executed to scrutinize the efficacy of the S-Box, encompassing evaluations of non-linearity, Bit Independence Criterion (BIC), Strict Avalanche Criterion (SAC), and Differential Probability (DP). These empirical assessments underscored the robustness and superiority of our proposed methodology compared to existing approaches.

Subsequently, a cipher block algorithm was proposed. This algorithm is based on a Feistel Network and AES, where row shifting operations, substitution, and key addition were employed as the feistel function.

The size of the blocks manipulated by the cipher block algorithm is 256 bits, the encrypted image undergoes a series of tests to validate the robustness of the proposed algorithm. These tests primarily include examination of the secret key space, sensitivity to the secret key, histogram analysis, correlation analysis, differential analysis, information entropy, and noise attack analysis. Based on the results of these tests, the cipher block algorithm is deemed reliable and resilient against various types of attacks.

To enrich the research and validate the simulated results, a practical implementation of the cipher block algorithm on a Nexys 7 FPGA board.

This opens up opportunities for its utilization in various applications, particularly in embedded systems.

For an improved work, different perspectives are suggested:

- Extending the cipher block algorithm to support colored and larger image sizes to meet diverse application needs.
- Implementation within a communication chain, transmission / reception.

Bibliography

[1] L. Gardini, C. Grebogi, and S. Lenci, 'Chaos theory and applications: a retrospective on lessons learned and missed or new opportunities', *Nonlinear Dyn.*, vol. 102, no. 2, pp. 643–644, Oct. 2020, doi: 10.1007/s11071-020-05903-0.

[2] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*. in Textbooks in Mathematical Sciences. New York, NY: Springer New York, 1996. doi: 10.1007/b97589.

[3] J. S. Muthu and P. Murali, 'Review of Chaos Detection Techniques Performed on Chaotic Maps and Systems in Image Encryption', *SN Comput. Sci.*, vol. 2, no. 5, p. 392, Sep. 2021, doi: 10.1007/s42979-021-00778-3.

[4] G. Dowek, 'The physical Church thesis and the sensitivity to initial conditions'. arXiv, 2023. doi: 10.48550/ARXIV.2304.13318.

[5] A. R. Nieto, J. M. Seoane, and M. A. F. Sanjuán, 'Final state sensitivity in noisy chaotic scattering', 2021, doi: 10.48550/ARXIV.2105.05755.

[6] H. Ghane, A. E. Sterk, and H. Waalkens, 'Chaotic dynamics from a pseudo-linear system', *IMA J. Math. Control Inf.*, vol. 37, no. 2, pp. 377–394, Jun. 2020, doi: 10.1093/imamci/dnz005.

[7] H. G. Schuster, 'Deterministic Chaos: Introduction and Recent Results', in *Nonlinear Dynamics in Solids*, Deutsche Physikalische Gesellschaft and H. Thomas, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 22–30. doi: 10.1007/978-3-642-95650-8_2.

[8] T. O. Roy-Layinde, K. A. Omoteso, D. O. Ogooluwa, H. T. Oladunjoye, and J. A. Laoye, 'Chaotic and Periodic Behavior in a Fractional-order Biological System', *Acta Phys. Pol. B*, vol. 51, no. 9, p. 1885, 2020, doi: 10.5506/APhysPolB.51.1885.

[9] 'Chaos', in *Chaos*, in Textbooks in Mathematical Sciences. , New York, NY: Springer New York, 1996, pp. 105–147. doi: 10.1007/0-387-22492-0_3.

[10] W. S. Sayed, S. M. Ismail, L. A. Said, and A. G. Radwan, 'On the Fractional Order Generalized Discrete Maps', in *Mathematical Techniques of Fractional Order Systems*, Elsevier, 2018, pp. 375–408. doi: 10.1016/B978-0-12-813592-1.00013-1.

[11] M. Rüdisüli, T. J. Schildhauer, S. M. A. Biollaz, and J. R. Van Ommen, 'Measurement, monitoring and control of fluidized bed combustion and gasification', in *Fluidized Bed Technologies for Near-Zero Emission Combustion and Gasification*, Elsevier, 2013, pp. 813–864. doi: 10.1533/9780857098801.3.813.

[12] K. Rajagopal, A. Akgul, I. M. Moroz, Z. Wei, S. Jafari, and I. Hussain, 'A Simple Chaotic System With Topologically Different Attractors', *IEEE Access*, vol. 7, pp. 89936–89947, 2019, doi: 10.1109/ACCESS.2019.2922164.

[13] N. A. Kudryashov, 'Analytical solutions of the Lorenz system', *Regul. Chaotic Dyn.*, vol. 20, no. 2, pp. 123–133, Mar. 2015, doi: 10.1134/S1560354715020021.

[14] J. Lü and G. Chen, 'A NEW CHAOTIC ATTRACTOR COINED', *Int. J. Bifurc. Chaos*, vol. 12, no. 03, pp. 659–661, Mar. 2002, doi: 10.1142/S0218127402004620.

[15] B. Fiedler, *Handbook of Dynamical Systems*. Burlington: Elsevier, 2002.
[16] C. Li, G. Luo, K. Qin, and C. Li, 'An image encryption scheme based on chaotic tent map', *Nonlinear Dyn.*, vol. 87, no. 1, pp. 127–133, Jan. 2017, doi: 10.1007/s11071-016-3030-8.
[17] L. Kocarev, 'Chaos-based cryptography: a brief overview', *IEEE Circuits Syst. Mag.*, vol. 1, no. 3, pp. 6–21, 2001, doi: 10.1109/7384.963463.
[18] K. Lata and L. R. Cenkeramaddi, 'FPGA-Based PUF Designs: A Comprehensive Review and Comparative Analysis', *Cryptography*, vol. 7, no. 4, p. 55, Nov. 2023, doi: 10.3390/cryptography7040055.
[19] J.-P. Aumasson, *Serious cryptography: a practical introduction to modern encryption*. San Francisco: No Starch Press, 2018.
[20] W. Easttom, *Modern Cryptography: Applied Mathematics for Encryption and Information Security*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-63115-4.
[21] E. Bellini, M. Sala, and I. Simonetti, 'Nonlinearity of Boolean Functions: An Algorithmic Approach Based on Multivariate Polynomials', *Symmetry*, vol. 14, no. 2, p. 213, Jan. 2022, doi: 10.3390/sym14020213.
[22] M. M. Matalgah, W. Y. Zibideh, and A. M. Magableh, 'Alleviating the effect of the strict avalanche criterion (SAC) of symmetric-key encryption in wireless communication channels', in *2011 International Conference on Communications and Information Technology (ICCIT)*, Aqaba, Jordan: IEEE, Mar. 2011, pp. 138–141. doi: 10.1109/ICCITECHNOL.2011.5762665.
[23] Alamsyah, A. Bejo, and T. B. Adji, 'AES S-box construction using different irreducible polynomial and constant 8-bit vector', in *2017 IEEE Conference on Dependable and Secure Computing*, Taipei, Taiwan: IEEE, Aug. 2017, pp. 366–369. doi: 10.1109/DESEC.2017.8073857.
[24] G. Hu and B. Li, 'Coupling chaotic system based on unit transform and its applications in image encryption', *Signal Process.*, vol. 178, p. 107790, Jan. 2021, doi: 10.1016/j.sigpro.2020.107790.
[25] M. A. B. Farah, A. Farah, and T. Farah, 'An image encryption scheme based on a new hybrid chaotic map and optimized substitution box', *Nonlinear Dyn.*, vol. 99, no. 4, pp. 3041–3064, Mar. 2020, doi: 10.1007/s11071-019-05413-8.
[26] D. Lambić, 'A novel method of S-box design based on chaotic map and composition method', *Chaos Solitons Fractals*, vol. 58, pp. 16–21, Jan. 2014, doi: 10.1016/j.chaos.2013.11.001.
[27] F. Özkaynak, 'Construction of robust substitution boxes based on chaotic systems', *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3317–3326, Aug. 2019, doi: 10.1007/s00521-017-3287-y.
[28] X. Wang *et al.*, 'S-Box Based Image Encryption Application Using a Chaotic System without Equilibrium', *Appl. Sci.*, vol. 9, no. 4, p. 781, Feb. 2019, doi: 10.3390/app9040781.
[29] A. Boutros and V. Betz, 'FPGA Architecture: Principles and Progression', *IEEE Circuits Syst. Mag.*, vol. 21, no. 2, pp. 4–29, 2021, doi: 10.1109/MCAS.2021.3071607.

[30] 'Nexys A7 Reference Manual - Digilent Reference'. Accessed: Jun. 19, 2024. [Online]. Available: https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual
[31] R. Saidi, N. Cherrid, T. Bentahar, H. Mayache, and A. Bentahar, 'Number of Pixel Change Rate and Unified Average Changing Intensity for Sensitivity Analysis of Encrypted inSAR Interferogram', *Ingénierie Systèmes Inf.*, vol. 25, no. 5, pp. 601–607, Nov. 2020, doi: 10.18280/isi.250507.
[32] Y. Zhang, 'Test and Verification of AES Used for Image Encryption', *3D Res.*, vol. 9, no. 1, p. 3, Mar. 2018, doi: 10.1007/s13319-017-0154-7.
[33] A. Arab, M. J. Rostami, and B. Ghavami, 'An image encryption method based on chaos system and AES algorithm', *J. Supercomput.*, vol. 75, no. 10, pp. 6663–6682, Oct. 2019, doi: 10.1007/s11227-019-02878-7.
[34] G. Ye, H. Wu, K. Jiao, D. Mei, and Faculty of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang 524088, China, 'Asymmetric image encryption scheme based on the Quantum logistic map and cyclic modulo diffusion', *Math. Biosci. Eng.*, vol. 18, no. 5, pp. 5427–5448, 2021, doi: 10.3934/mbe.2021275.
[35] S. H. Strogatz, *Nonlinear Dynamics and Chaos*, 0 ed. CRC Press, 2018. doi: 10.1201/9780429492563.
[36] M. Maazouz, A. Toubal, B. Bengherbia, O. Houhou, and N. Batel, 'FPGA implementation of a chaos-based image encryption algorithm', *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9926–9941, Nov. 2022, doi: 10.1016/j.jksuci.2021.12.022.