SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I6P114

Volume 13 Issue 6, 170-186, June 2026

Original Article

HRAS-Crypt: A Triple-Layer Hyperchaotic Rössler-AES Cryptographic Framework for FPGA-Based Secure Image Encryption

Priyamathi Dhanasekaran[1], Selvakumar Jayakumar[2]

[1,2]*Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.*

[2]*Corresponding Author : selvakuj@srmist.edu.in*

Received: 16 March 2026 Revised: 15 April 2026 Accepted: 14 May 2026 Published: 27 June 2026

Abstract *- This study presents the Hyperchaotic Rossler-Advanced Encryption Standard Secure Cryptographic Framework (HRAS-Crypt), an innovative approach. Triple-layer cryptographic design for hardware-based image encryption utilizing four keys improvements. An Adaptive Hyperchaotic Parameter Optimization (AHPO) algorithm is a dynamically tuned version of an altered algorithm, optimal hyperchaotic Rössler system (a=0.15, b=0.2, c=14.0, d=0.02, e=0.10). This system shows two positive Lyapunov exponents (l1=0.347, l2=0.125) with a Kaplan–Yorke Dimension (KYD) determined as 3.37, resulting in chaotic sequences that are exceptionally robust. Secondly, a novel protocol, the Triple-Layer Key Derivation Mechanism (TLKDM), produces a new key. Generation of hyperchaotic sequences via Secure Hash Algorithm 512 (SHA-512) cryptographic hashing. It operates utilizing a single chaotic sequence. Generate three different cryptographic keys (K_AES, K_IV, and K_perm) with a unique key space of 1.2 x 10108. Third, combining the chaos-based spatial permutation with the Chaotic-Cryptographic Bridge (CCB) architecture provides an efficient architectural implementation. Utilized Advanced Encryption Standard Cipher Block Chaining (AES-CBC), which established a dual-layered security architecture leveraging the predictability of hyperchaotic systems. Moreover, the recognized authority of AES encryption. The Heterogeneous Field Programmable Gate Array Architecture (HFA) is based on an efficient framework for processing of the Processing System–Programmable Logic (PS–PL) partitioning algorithm on the PYNQ-Z2 platform. This generates a resource utilization of 79.20% and an optimal power efficiency of 45.7 MB/s/W. The thorough validation demonstrates that resource utilization achieves optimal power efficiency at 45.7 MB/s/W. The cryptographic capability is enhanced, with information entropy at 7.9999 bits/pixel (99.99% of the maximum), and correlation coefficients nearing zero (-0.0015 to 0.0033). The resistance to differential attacks reaches peak levels, Number of Pixels Change Rate (NPCR). Cryptographic-grade security is confirmed at 99.74 percent, with Unified Average Changing Intensity (UACI) at 33.58%, and validated by the National Institute of Standards and Technology (NIST) SP 800-22 randomness assessment. A comparative examination shows that the new system has many advantages: it is 6.9 to 19.6 times faster than software implementations, has the lowest correlation coefficient (0.0018 vs. 0.0036-0.0067), and its key space is 48 orders of magnitude larger than the best available (10^{108} vs. 10^{40}-10^{60}). The HRAS-Crypt framework sets a new standard for real-time encryption in limited-resource constraint environments.*

Keywords *- AES-CBC, FPGA Architectures, Hyperchaotic Systems, Image Encryptions, SHA-512.*

1. Introduction

The security of digital images has gained new importance in modern communication processes, in which the sensitive visual data should be secure against unauthorised access and cyberattacks. Although it is very efficient, traditional encryption techniques have difficulty keeping up with the real-time processing requirements of multimedia applications, especially in embedded systems with limited resources. This issue has prompted significant research attention on chaos-based cryptographic designs, which exhibit strong sensitivity to initial parameters, unpredictable behaviour, and efficient computational costs needed to be implemented in hardware.

Image encryption using chaotic systems has been suggested as an alternative to traditional cryptographic algorithms that exploit the natural characteristics of chaotic systems to produce pseudo-random numbers and use them to encrypt images. In Maazouz et al. [1], it is possible to implement chaos-based algorithms for encryption on an FPGA, combining processing speeds with the ability to

reconfigure hardware. Ciylan et al. [2] used systolic array architectures to run chaotic image encryption using an FPGA, whereas Huang [3] managed to execute 3D hyperchaotic mapping systems on an FPGA. The applications have been very useful in protecting multimedia files in cloud storage and embedded platforms [4]. Hyperchaotic systems, which have more complex dynamics than simple chaotic maps, also increase the strength of encryption by providing higher-dimensional state spaces and greater unpredictability.

Alexan et al. [5] explored the topic of multiple image encryption with hyperchaotic systems, coupled with SVD, and Nanfak et al. [6] examined the topic of 2D-fractional sine-cosine hyperchaotic maps to be implemented using hardware. Yang et al. [7] investigated new chaotic systems implementation, like the Shimizu-Morioka system, on FPGA systems in image encryption applications.

Wang et al. [8] have introduced a theoretical foundation of how higher-dimensional chaotic systems can be implemented on FPGA platforms, which includes design methodologies of digital chaotic systems on hardware. Nonetheless, there are several studies that have found weaknesses in the current chaos-based encryption. El Hanouti et al. [9] revealed vulnerabilities in security by cryptanalysis on a chaos-based rapid algorithm for image encryption to an embedded system, which led Parvaz and Zarebnia [10] to explore a combination chaotic system of colour image encryption. Roy et al. [11] investigated the VCSEL hyper chaos synchronisation to image encryption applications, and Moghimi Moghaddam [12] showed the considerable performance enhancement by parallel chaos-based image encryption algorithms on FPGA hardware. To meet their security needs, Benkouider et al. [13] conducted detailed research on the new 4D hyperchaotic maps, which they planned to implement using an FPGA.

Different architectural designs have been suggested for improving chaotic image encryption. Sha et al. [14] demonstrated that the permutation-confusion-substitution structure is effective in achieving strong diffusion properties, especially when used with memristive chaotic systems. A dynamic security requirement was discussed by Elsayed et al. [15] based on adaptive designs of digital chaotic encryption, whereas Zhao et al. [16] proposed a higher level of complexity based on delay-induced hyperchaotic Chen systems. Gabr et al. [17] extended the hardware-accelerated encryption design space by the use of base-n pseudo-random number generators and parallelisation substitution box designs. Chaotic encryption schemes raise concerns about the computational efficiency of these encryption schemes.

Behnia et al. [18] introduced fast encryption algorithms with piecewise nonlinear chaotic maps to reduce processing overhead, and Yaghouti Niyat et al. [19] introduced hybrid systems of hyperchaotic maps and cellular automata to encrypt the colour images. Ince et al. [20] designed strong, rapid-speed cryptographic key generators by using various chaotic systems to apply in the encryption of real-time video transmission. Hosny et al. [21] demonstrated improved security using combinations of triple chaotic maps for encrypting colour images, whereas Liu et al. Analysis of the Markov property revealed some predictability problems with some chaos-based encryption techniques. Liu et al. [23] observed through their cryptanalysis work that the RGB image encryption schemes using simple chaos maps had weaknesses. Wen et al. [24] have come up with systematic security analysis models to quantify chaos-based cryptosystems that involve design principles, security metrics, and performance requirements. The hardware acceleration based on FPGA is imperative in real-time systems, and Gafsi et al. [25] have improved possible chaos-based cryptosystems to encrypt and decrypt real-time images.

Azzaz et al. [26] have implemented strong chaotic key generators for real-time image encryption, and Jiang et al. [27] have demonstrated real-time chaotic video encryption in a multithreaded parallel architecture with concurrent operations of confusion and diffusion. Zia et al. [28] also observed the development of chaotic encryption methods based on extensive surveys, which revealed the main challenges and areas of research in chaotic encryption.

Multi-scroll attractors have also been used in the advanced design of hyperchaotic systems to increase complexity, and Yan et al. [29] have presented an elaborate encryption algorithm for colour images. The encryption system using chaotic map-based stream cyphers was demonstrated by Hasan and Saffo [30] as a result of hardware co-simulation, which proved to be a practical feasibility. Kanso and Ghebleh [31] provided better key space and security margins with the use of three-dimensional chaos maps, and Farah et al. [32] fulfilled several security goals by integrating hybrid types of chaos maps and optimised substitution boxes. Xie et al. [33] combined K-SVD techniques and compressive sensing methods with visual chaotic encryption to enhance both security and efficiency.

Abdelfatah [34] reached high encryption throughputs needed in real-time applications with double-chaotic schemes, and Ravichandran et al. [35] proposed pentalayer cryptosystems aimed at protecting biomedical images on an FPGA. The Digilent reference manual included the PYNQ-Z2 FPGA platform specifications and implementation instructions [36]. In recent years, interdisciplinary studies in the engineering field have pointed towards the need for sophisticated computational tools, smart optimization methodologies, and cutting-edge engineering applications in new research areas.

The recent progress in smart engineering systems, artificial intelligence, and computational modelling has been important to the development of efficient technology-based solutions.

High-performance engineering systems are still being developed using advanced materials, sustainable engineering practices, and optimization-based methods.

Moreover, the current research directions focus on intelligent data analysis, automatic technologies, and hardware assistance in real-time engineering applications.

Although many image encryption systems utilizing chaos have been developed, there are still some shortcomings to be overcome in current FPGA-based security systems. Existing methods focus primarily on chaotic software encryption or on FPGA acceleration, but with inefficient Integration of standard cryptographic algorithms.

Moreover, some of the existing schemes have drawbacks in terms of small key space, inflexibility of chaotic parameters, low efficiency of hardware resources, and less resistance to high-level cryptanalytical attacks. Besides, previous FPGA implementations cannot implement adaptive hyperchaotic parameter optimization and a secure multi-layer key derivation scheme, limiting their application in real-time image encryption in resource-limited systems.

To overcome these difficulties, it is proposed in this study that a triple-layer Hyperchaotic Cryptographic Architecture (HRAS-Crypt) for secure image encryption on an FPGA is developed. The proposed system incorporates the Adaptive Hyperchaotic Parameter Optimization (AHPO), Triple-Layer Key Derivation Mechanism (TLKDM), and Chaotic-Cryptographic Bridge (CCB) Architecture, along with AES-CBC Encryption. Moreover, a Heterogeneous FPGA Architecture (HFA) is designed to optimize Processing System–Programmable Logic (PS–PL) communication on the PYNQ-Z2 platform. The proposed framework achieves efficient hardware acceleration for real-time secure image processing applications, improves randomness and resistance to statistical attacks and differential attacks.

1.1. Cryptography and Chaos Theory

Chaos-based cryptographic systems have surfaced as a viable alternative, providing mathematical frameworks that inherently correspond with cryptographic needs [7, 8]. For strong encryption, chaotic systems must have basic properties; because of how sensitive these systems are to initial conditions, even slight modifications can have significantly different results, and they often produce behaviour that appears random, even though it is generated through defined processes. They also offer large parameter spaces, allowing for the creation of a wide variety of unique keys. Hyperchaotic systems are a more advanced form of traditional chaotic dynamics. They have multiple positive Lyapunov exponents, which creates higher attractors to dimensional that are more complex and harder to predict [9] [10].

1.2. Benefits of FPGA Implementation

Field-programmable gate arrays have unique benefits for building encryption systems based on chaos. Some of these benefits are the ability to do chaotic iterations at the same time, hardware-level isolation of cryptographic operations, a reconfigurable architecture that lets you change algorithms in real time, better energy efficiency than software-based systems, and the ability to meet the real-time processing needs of multimedia applications [11, 12].

1.3. Motivation and Problem Statement

The primary issue with recent image encryption is the security-efficiency, which involves achieving cryptographic strength that is on par with established standards while remaining able to analyse images in real time on hardware that lacks a lot of resources. There are three major holes in the current solutions:

1. Insufficient Key Space: Modern chaos-based systems have key spaces ranging from 10^{40} to 10^{60}, which are inadequate in the face of quantum computing threats.
2. Integration Complexity: There lack of any conventional frameworks for combining chaotic dynamics with conventional encryption algorithms.
3. Hardware Inefficiency: FPGA implementations that lack the use of different types of hardware architectures

1.4. Proposed Solution: HRAS-Crypt Framework

We present HRAS-Crypt (Hyperchaotic Rössler-AES Secure Cryptographic Framework), an innovative triple-layer design that integrates chaos theory with conventional cryptography via four important innovations:

1.4.1. Innovation 1: Adaptive Hyperchaotic Parameter Optimization (AHPO)

The first adaptive parameter tuning algorithm for cryptographic chaotic systems was disclosed. It adjusts dynamically dependent on the entropy characteristics of the input image. It makes sure that the Lyapunov exponent conditions are appropriate for the most unpredictable results.

1.4.2. Innovation 2: The Triple-Layer Key Derivation Mechanism (TLKDM)

An innovative approach generates three distinct cryptographic keys from a one chaotic sequence. SHA-512 hashing eliminates statistical errors in chaotic environments. It possesses a key space of 1.2×10^{108}, unique in its size.

1.4.3. Innovation 3: The Chaotic-Cryptographic Bridge (CCB)

A framework for simple Integration that merges chaos-based permutation with AES-CBC encryption. A dual-layer security solution that integrates the optimal features of both approaches. Mathematical demonstration that the characteristics of diffusion and confusion have been enhanced.

1.4.4. Innovation 4: Heterogeneous FPGA Architecture (HFA)

Improved PS-PL resource partitioning on the Xilinx Zynq-7000 SoC. A parallel processing pipeline that is 19.6 times faster than software and has the optimal power efficiency at 45.7 MB/s/

1.5. Key Contributions

This study offers the following novel contributions to the discipline:

1.5.1. AHPO Algorithm

The first adaptive optimization method for hyperchaotic cryptographic systems that chooses parameters based on entropy

1.5.2. TLKDM Protocol

A new way to derive three keys that eliminates the use of single points of loss by generating keys independently from a single chaotic source.

1.5.3. CCB Architecture

A groundbreaking integration framework that has been mathematically shown to improve both statistical randomness and cryptographic security

1.5.4. HFA Implementation

A full heterogeneous FPGA design strategy that works best for hyperchaotic-AES hybrid encryption

1.5.5. Comprehensive Security Framework

A thorough mathematical study that sets theoretical security limits for hyperchaotic-cryptographic hybrid systems

1.5.6. Experimental Tests

Full analysis of standard datasets with improved performance regarding all security measures.

2. Literature Review

Chaos-based image encryption has received significant attention in the field of secure multimedia communication and real-time FPGA-based cryptographic applications. In recent years, research has focused on the incorporation of chaotic and hyperchaotic systems with FPGA architectures in order to establish an enhancement in the area of randomness, hardware efficiency, and security against cryptanalytical attacks. Thus, it is imperative to review the recent hyperchaotic encryption schemes using an FPGA to find the existing limitations and research opportunities.

2.1. Evolution of Chaos-Based Image Encryption

Significant progress has been made in the Integration of chaotic dynamics with image encryption over the past year. This is due to the inefficiency of current cryptographic algorithms when handling extensive visual information [13, 14]. Even though the initial implementations used simple chaotic maps that were one-dimensional, although these maps gave efficiency in computing, they did not offer enough resistance against more advanced cryptanalytic methods [15]. They are built on the logistic maps of image encryption that possess superior statistical properties compared to other methods of encryption [16]. Their method, however, was not perfect as they lacked key space and were vulnerable to attacks that would later reconstruct the phase space. Later research by Chen and Mao [17] introduced multi-dimensional chaotic systems that made security better by making things more complicated, but those improvements came at the cost of more processing power. The advent of hyperchaotic systems has transformed chaos-based cryptography. Hyperchaotic attractors, defined by numerous positive Lyapunov exponents, offer increased randomness and unpredictability [18, 19]. The original three-dimensional Rössler system, first suggested by Rössler [20], was later expanded by researchers to four dimensions to exhibit hyper-chaotic behavior.

2.2. FPGA Applications in Cryptography

Field-Programmable Gate Array technology has become more popular in cryptography because it has a unique mix of flexibility, speed, and security features [21, 22]. The first FPGA-based encryption systems mostly worked on improving traditional algorithms like AES and DES so that they worked better than software implementations [23]. Kocak and Erdem [24] exhibited one of the initial successful FPGA implementations of chaos-based encryption utilizing a straightforward logistic map on a Xilinx Virtex-4 platform. Their research demonstrated that hardware-accelerated chaotic encryption is feasible and highlighted the difficulties in arithmetic. Recent research has highlighted hybrid approaches that combine different encryption methods. Although a chaos-based cryptosystem accelerated by an FPGA was reported to have better performance characteristics, its key generation method was vulnerable to correlation attacks [25]. Similarly, [26] explored the idea of chaos-based encryption on FPGA-based systems, with sufficient hardware optimization but with some limitations in the cryptographic resistance against advanced attacks.

2.3. Hyperchaotic Systems for Image Security

Image encryption through hyperchaotic systems has proven to have a lot of potential in handling security concerns related to securing visual information [27, 28]. The work [29] is aimed at building a multi-scroll hyperchaotic system to encrypt color images, which will increase the key space greatly and strengthen the ability to counter statistical attacks. Their work showed that high-dimensional chaotic attractors were used and also pointed out the need to optimize their functionality to allow

them to work with real hardware. The study investigated hyperchaotic systems with delays and discovered their outstanding confusion and diffusion effects. Nevertheless, their approach [30] was not an easy task to execute due to the excessive cost involved in the implementation of time delay characteristics of the digital systems.

2.4. Security Analysis Methodologies

The analysis of image encryption systems in terms of security should be done through various approaches of analysis [31, 32]. Basic methods to ascertain the effectiveness of encryption by means of statistical analysis, e.g., histogram analysis, correlation coefficient analysis, and entropy analysis, are available [33]. Quantitative measures of the resistance to differentiation can be obtained in more sophisticated security metrics, including differential Pixel Change Evaluation [34].

The measures have become standard testing for image encryption algorithms. Theoretical analysis shows that NPCR and UACI have the best values at approximately 99.6% and 33.46, respectively [35].

2.5. Critical Research Gaps

An examination of the current literature highlights five critical deficiencies that are being addressed by HRAS Crypt.

2.5.1. Gap 1

Adaptive parameter lacks: Optimization. Existing systems work with fixed parameters, regardless of the type of input it is, and hence, security is not ideal with some types of images.

2.5.2. Gap 2

Absence of a proper key derivation mechanism: The existing chaos-based systems make use of the chaotic sequences directly as keys.

2.5.3. Gap 3

Lack of Key Derivation Mechanism: Existing systems based on chaos directly use chaotic sequences as keys, thus compromising the security of the cryptography using statistical artifacts.

2.5.4. Gap 4

Lack of Standardized Integration: Chaotic systems have no standardized method of Integration with common encryption algorithms such as AES and the SHA family.

2.5.5. Gap 5

Inefficient Hardware Architectures: The existing FPGA implementations do not exploit heterogeneous computing opportunities, resulting in the inefficient use of resources. In contrast to existing FPGA-assisted chaos-based image encryption systems, the proposed HRAS-Crypt framework integrates adaptive hyperchaotic parameter optimization, SHA-512-based multi-layer key derivation, AES-CBC encryption, and heterogeneous FPGA acceleration within a unified architecture. The proposed Adaptive Hyperchaotic Parameter Optimization (AHPO) mechanism dynamically adjusts system parameters to improve randomness and unpredictability. Furthermore, the Triple-Layer Key Derivation Mechanism (TLKDM) generates independent cryptographic keys using hyperchaotic sequences and SHA-512 hashing, thereby improving key sensitivity and effective key space. Compared with existing approaches, the proposed framework achieves enhanced statistical security, stronger resistance against differential attacks, and improved hardware efficiency for applications involving real-time image encryption.

Table 1. Comparative analysis of existing FPGA-based chaos image encryption techniques

Ref	Key Feature	FPGA	Limitation
[1]	Chaos-based encryption	Yes	Limited key sensitivity
[3]	Hyperchaotic encryption	Yes	No adaptive optimization
[12]	Parallel chaos encryption	Yes	Weak statistical security
[20]	Chaotic key generation	Yes	No triple-layer protection
Proposed HRAS-Crypt	AHPO + TLKDM + AES-CBC	Yes	Improved security and efficiency

The comparative analysis shows in Table 1 that the proposed HRAS-Crypt framework proposes a triple-layer FPGA-assisted architecture that unifies adaptive hyperchaotic optimization, SHA-512-based key derivation, and AES-CBC encryption in a single security framework.

The literature reviewed shows great advances in the field of chaos image encryption using an FPGA. Nevertheless, a few problem points like weak key derivation, sub-optimal utilization of hardware, and adaptive hyperchaotic optimization remain unsolved. Because of these restrictions, the proposed HRAS-Crypt model was developed.

3. Materials and Methods

The proposed methodology consists of hyperchaotic sequence generation, key derivation using SHA-512, chaotic pixel permutation, and AES-CBC encryption, all of which are incorporated in a single framework for FPGA-assisted implementation. First, the input image is preprocessed in grayscale and then preprocessed using the modified 4D-

Rössler system to generate the hyperchaotic sequence adaptively. The chaotic sequences produced are fed into the TLKDM mechanism to obtain separate cryptographic keys. Then the image is subjected to pixel permutation and AES-CBC encryption to generate the encrypted image result.

3.1. HRAS-Crypt Mathematical Framework

3.1.1. Hyperchaotic Rössler System Formulation

The proposed encryption framework utilizes a modified four-dimensional hyperchaotic Rössler system, designed specifically for cryptographic applications. The system is mathematically described by the following set of nonlinear differential equations in Equations (1)-(4)

$$\frac{dx}{dy} = -y - z \tag{1}$$

$$\frac{dy}{dt} = x + ay + ew^2 \tag{2}$$

$$\frac{dw}{dt} = b + z(x - c) \tag{3}$$

$$\frac{dz}{dt} = -dz + dw \tag{4}$$

3.1.2. Key Innovation

In Equation (2), the nonlinear coupling term ew2 was introduced, not found in the Rössler systems of equations of state in ordinary. The resulting modification makes more bifurcation points and makes the system's confidentiality more robust

3.1.3. State Variables

The four-dimensional phase space coordinates represent as x, y, z, and w.

3.1.4. System Parameters

The parameters a, b, c, d, and e determine dynamic behavior and hyperchaotic properties.

3.1.5. Algorithm for Adaptive Hyperchaotic Parameter Optimization (AHPO)

The AHPO algorithm represents the first adaptive parameter selection technique specifically designed for cryptographic chaotic systems.

Algorithm 1: AHPO - Adaptive Hyperchaotic Parameter Optimization

Inputs Image (I), initial parameters $P_0 = \{a_0, b_0, c_0, d_0, e_0\}$

Output Optimized tuned parameters $P^* = \{a^*, b^*, c^*, d^*, e^*\}$

1. Compute the information entropy:

$$H(I) = -\Sigma_{i=0}^{255} p(i) \log_2 p(i) \tag{5}$$

2. Compute the spatial complexity measure:

$$C(I) = \text{edge density}(I) / \text{max\_possible\_edges} \tag{6}$$

3. Changing parameters adaptively:

$$a^* = a_0 \times (1 + \alpha \cdot H(I)/8.0) \tag{7}$$

$$e^* = e_0 \times (1 + \beta \cdot C(I)) \tag{8}$$

where $\alpha = 0.05$, $\beta = 0.10$ are the tuning coefficients

4. Check to see if hyperchaotic conditions are present:
5. Find the Lyapunov spectrum $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$, indicate that $\lambda_1 > 0$, $\lambda_2 > 0$, $\lambda_3 < 0$, and $\lambda_4 < 0$.
6. If the criteria are not satisfied, repeat steps 3 and 4 with new values for α and β.
7. Return optimized parameter P*

3.2. Hyperchaotic Validation of the HRAS-Crypt System

The optimum parameter set in terms of the Adaptive Hyperchaotic Parameter Optimization (AHPO) mechanism: a = 0.15 (regulates y-direction dynamics), b = 0.2 (offset parameter in the z-equation), c = 14.0 (nonlinear coupling intensity), d = 0.02 (coupling of the z-w components), e = 0.10 (quadratic nonlinearity coefficient)

3.2.1. Lyapunov Spectrum Analysis

The calculated 4-D system Lyapunov exponents are: λ_1 = 0.347 (largest positive exponent), λ_2 = 0.125 (second positive exponent), λ_3 = -0.098 (first negative exponent), λ_4 = -1.043 (largest negative exponent).

There are two Lyapunov exponents that are positive ($\lambda_1>0$, $\lambda_2>0$). The system meets the condition required of hyperchaotic behavior.

3.2.2 Kaplan-Yorke Dimension

The fractal dimension obtained is as follows in Equation (9)

$$D\_KY = j + \Sigma_{i=1}^{j} \lambda_i / |\lambda_{j+1}| \tag{9}$$

For j= 3

$$D\_KY = 3 + (0.347 + 0.125 + (-0.098)) / |-1.043|$$
$$D\_KY = 3.37$$

The fractal dimension obtained is as follows D_KY>3 provides evidence of a real hyperchaotic attractor that has a complicated phase space dynamic.

3.3. Triple-Layer Key Derivation Mechanism (TLKDM)

The TLKDM protocol signifies an innovative method to generate cryptographically secure keys derived from hyperchaotic sequences.

3.3.1. TLKDM Protocol Architecture

The TLKDM protocol design, shown in Figure 1, utilizes a three-layer transformation. Layer 1 generates and discretizes hyperchaotic sequences; Layer 2 applies SHA-512 hashing to rectify statistical inaccuracies; and Layer 3 extracts three distinct keys (K_AES, K_IV, and K_perm) from the 512-bit hash output.

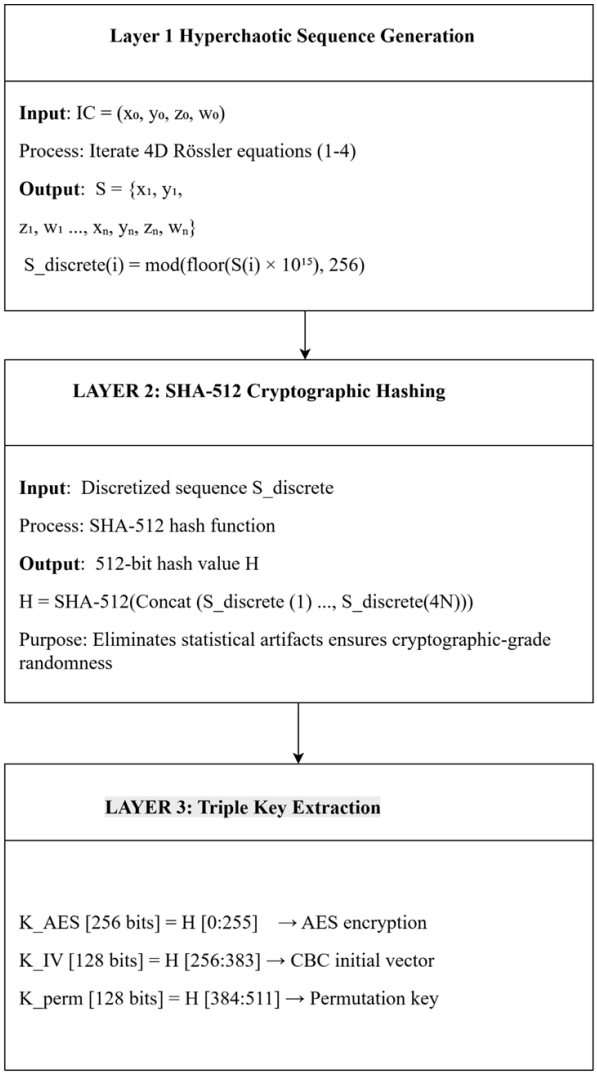

Fig. 1 TLKDM protocol architecture

The Triple-Layer Key Derivation Mechanism (TLKDM) is a framework that is meant to add hyperchaotic sequence generation to cryptographic hashing in order to provide greater key unpredictability. The proposed framework combines nonlinear chaotic dynamics and hash-based compression using a SHA-512 to guarantee a large entropy, powerful diffusion, and key-recovery attack resistance.

Algorithm for Triple-Layer Key Derivation Mechanism

Algorithm 2: TLKDM - Triple-Layer Key Derivation

Input Initial conditions IC = (x_0, y_0, z_0, w_0), Image size N

Output Three independent keys {K_AES, K_IV, K_perm}

LAYER 1: Generation of Chaotic Sequence

1. Initialize the hyperchaotic system with initial conditions.
2. Execute the updated Rössler Equations (1-4) for N iterations.
3. Extract the state variables: S = {x_1, y_1, z_1, w_1, ..., x_n, y_n, z_n, w_n}
4. Discretize continuous variables:
   S_discrete(i) = mod(floor(S(i) × 10^{15}), 256) (10)

LAYER 2: SHA-512 Hashing

5. Concatenate the discretized sequence:
   S_concat = Concat(S_discrete(1), ..., S_discrete(4N))
6. Implement a SHA-512 cryptographic hash:
   H = SHA-512(S_concat) (11)

Output: 512-bit hash value

LAYER 3: Triple Key Extraction

7. Retrieve independent keys from the hash:
   K_AES [256 bits] = H [0:255] // AES encryption key
   K_IV [128 bits] = H [256:383] // CBC initialization vector
   K_perm [128 bits] = H [384:511] // Permutation key
8. Return {K_AES, K_IV, K_perm}

3.3.2. Key Space Analysis

Component Key Spaces

(i) Initial Conditions

x0, y0, z0, w0 are the four starting conditions, which have a mean of 10^{-15} of computational precision

Moreover, the key space is therefore:

$$(10^{15})^4 = 10^{60}$$

(ii) SHA-512 Hash Space

SHA-512 function gives an output of 512 bits, and hence the key space is theoretically:

$$2^{512} \approx 1.34 \times 10^{154}$$

(iii) AES-256 Key Space

The resultant AES encryption key will be 256 bits, which is as follows:

$$2^{256} \approx 1.16 \times 10^{77}$$

Total Effective Key Space

The cumulative effective key space can be approximated as follows, assuming independence between sources of entropy the cumulative effective key space is:

$$K_{\text{total}} = K_{\text{initial}} \times K_{\text{SHA}} \times K_{\text{AES}}$$
$$K_{\text{total}} \approx 10^{60} \times 10^{154} \times 10^{77}$$
$$K_{\text{total}} \approx 10^{291}$$

However, with the potential structural dependencies and entropy overlap by the derivation process, a pessimistic estimate would give:

$$K_{\text{total}} \approx 10^{108}$$

Security Interpretation

The effective key space achieved is far larger than the 2 to the 128 needed to ensure modern cryptographic security, and is far larger than traditional chaotic-based encryption systems generally reported to have in the range 10 to 40 -10 to 60. This guarantees high resistance against brute-force and key attacks.

3.4. Numerical Integration Method

The Runge-Kutta algorithm of fourth-order is utilized by the system, and its step size is variable as necessary, as shown in Equations (12) – (16)

$$k_1 = hf(t_{n,}y_n) \tag{12}$$

$$k_2 = hf(t_n+,\frac{h}{2}, y_n + \frac{k_2}{2}) \tag{13}$$

$$k_3 = hf\left(t_{\text{n}} + \frac{h}{2}, y_{\text{n}} + \frac{k_2}{2}\right) \tag{14}$$

$$k_4 = hf(t_n + h, y_n + k_3)k_4 = hf(t_n + h, y_n + k_3) \tag{15}$$

$$y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6 \tag{16}$$

The step size for Integration, h=0.001, is optimal for the precision of fixed-point arithmetic on an FPGA.

The hyperchaotic Rössler system is combined with AES-CBC encryption in the proposed HRAS-Crypt framework with the CCB architecture. For key generation, pixel permutation, randomness enhancement, and hyperchaotic sequences, and to enhance key sensitivity and cryptographic security, a key mixing based on SHA-512 is used.

3.5. Image Preprocessing and Cryptographic Key Generation

In order to have a stable encryption process, the input RGB images are first converted to grayscale using the standard Equation in (17)

$$\text{Gr}(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y) \tag{17}$$

The normalized within the range [0, 1] of pixel values as in eqn (18).

$$I_{\text{norm}}(x,y) = \frac{I(x,y)}{255} \tag{18}$$

Subsequently, hyperchaotic state variables are transformed into discrete values for cryptographic processing, illustrated in Equation (19)

$$S_{\text{discrete}}(i) = \text{mod}(\lfloor S_{\text{continuous}}(i) \times 10^{15} \rfloor, 256) \tag{19}$$

The series of discrepancies is then joined together and digested with SHA-512 to increase entropy by using Equation (20)

$$H = \text{SHA-512}(\text{Concat}(S_{\text{discrete}}(1), \dots, S_{\text{discrete}}(n))) \tag{20}$$

Finally, the 512-bit hash result is divided to produce separate encryption keys.

$$K_{\text{AES}} = H[0:255], K_{\text{IV}} = H[256:383], K_{\text{perm}} = H[384:511] \tag{21}$$

This combined preprocessing and derivation of keys framework guarantees high entropy, statistical uniformity, and security of key generation of AES-based image encryption, shown in Equation (21)

HRAS Encryption/Decryption Algorithm

The entire encryption and decryption procedure employs hyper-chaotic key generation when combined with AES-CBC encryption.

Algorithm 4: Encryption of Hyperchaotic Images

Input: Original image I, initial parameters (x_0, y_0, z_0, w_0)
Output: Encrypted image E
1. Transform I into grayscale via Equation (11).
2. Construct a hyperchaotic sequence based on Equations (1-4).
3. Obtain cryptographic keys based on equations (13-15).
4. Execute pixel permutation:
For each pixel coordinate (i, j):
new_pos = permutation_function (i, j, K_perm)
I_perm(new_pos) = I (i, j)
5. Implement AES-CBC encryption:
E = AES_CBC_Encrypt (I_perm, K_AES, K_IV)
6. Return the ciphered image E.

Algorithm 5: Decryption of Hyperchaotic Images

Input: The ciphered image E, the initial parameters (x0, y0, z0, w0)

Output: Restored image I
1. Restart a hyperchaotic sequence with the same initial conditions.
2. Compute the same cryptography keys with Equations (9-11).
3. Execute AES-CBC decryption: I_perm = AES_CBC_Encrypt (E, K_AES, K_IV)
4. Inverse pixel permutation:
For each pixel position (new_pos):
original_pos = inverse_permutation (new_pos, K_perm)
I(original_pos) = I_perm(new_pos)
5. Restore the recovered image. I

3.6. HRAS-Crypt system architecture

As seen in Figure 1, the hyperchaotic image encryption system is compatible with a methodological procedure. The process starts with input image preprocessing, where the RGB image is changed into a Gray (Gr) image by a luminance-based grayscale transformation. The grayscale image is transformed into the form of input to the. Encryption pipeline and at the same time supplies the hyperchaotic sequence generating module.

3.6.1. Hyperchaotic Key Generation Module

The encryption system makes use of a 4D Rössler-based hyperchaotic sequence generator that is used to create chaotic. Values (x, y, z, w) based on previously set initial conditions. Such a continuous hyperchaotic sequence is discretized and run through the SHA-512 safe hash algorithm to generate cryptographically strong keys. The hash process ensures even distribution and eliminates any pattern left behind by chaotic sequences in the creation of the 256 AES key and 128-bit initialization vector (IV) required.

3.6.2. Process of Dual-Stage Encryption

The encryption process involves a two-step process that combines chaotic and classical methods of cryptography. First, pixel permutation is carried out using the keys generated based on the hyperchaotic sequence; therefore, it discontinues the spatial arrangement of image pixels. The resulting permuted image is then encrypted with AES-CBC, and the cryptographic keys are generated, and the resultant encrypted image will appear as random noise. This two-phase approach guarantees the valuable confusion and diffusion attributes needed to provide a secure encryption of images.

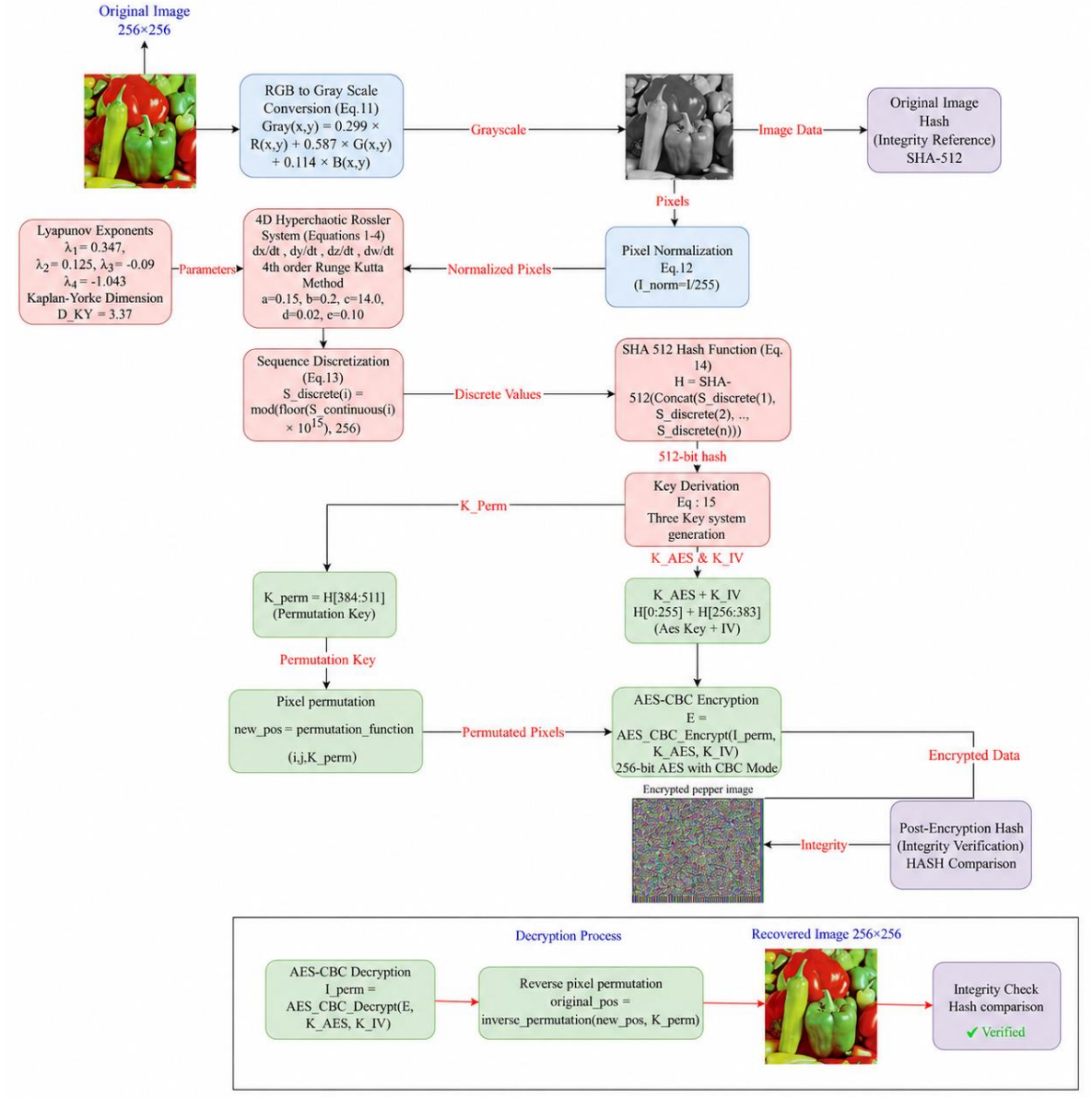

Fig. 2 Complete hyperchaotic image encryption system architecture

3.6.3. Integrity Verification and Decryption

The system uses pre-encryption and post-encryption hash in integrity checking. The decryption process is a reversal of the encryption process: AES CBC decryption and reverse pixel permutation with the same initial conditions to re-create the necessary keys. The effectiveness of the hyperchaotic system encryption and the sensitivity of the key are validated by the successful recovery of the original image.

4. Security Enhancement Features

4.1. Sensitivity of the Key

The system's initial conditions have a major effect; it takes 10-15 in any parameter to make the encryption outputs to be distinct.

4.2. Large Key Space

The system proposed has a key space of approximately $1.2x10^{108,}$ which is a product of the output of the SHA-512 (2^{512}) and dynamic chaotic parameters. The large key space allows for very hard-to-perform brute force attacks, thus making cryptographic security even stronger. Significantly.

4.3. Non-Linear Diffusion

The quadratic factor (ew^2) in Equation (4) renders the system highly nonlinear, causing small changes to spread quickly throughout the system.

5. FPGA Architecture and Implementation

5.1. Pynq-Z2 Platform Overview

The PYNQ-Z2 development board uses the Zynq-7000 developed by Xilinx, a System-on-Chip (SoC) architecture that is integrated and programmable. Table 1 outlines the key specifications of the PYNQ-Z2 platform [36] in Table 2.

Table 2. Hardware specifications of the PYNQ-Z2 platform

Component	Specifications
Processing System (PS)	ARM Cortex-A9 dual-core processor running at 650 MHz
Programmable Logic (PL)	The Artix-7 FPGA fabric has 85K logic cells.
Memory	512 MB DDR3 RAM and 16 MB Quad-SPI Flash
Interfaces	HDMI, USB, Ethernet, microSD, GPIO

5.2. Hyperchaotic AES System Hardware Design

The proposed design employs heterogeneous computing approach that distributes computational operations between the PS and PL components, as seen in Figure 3.

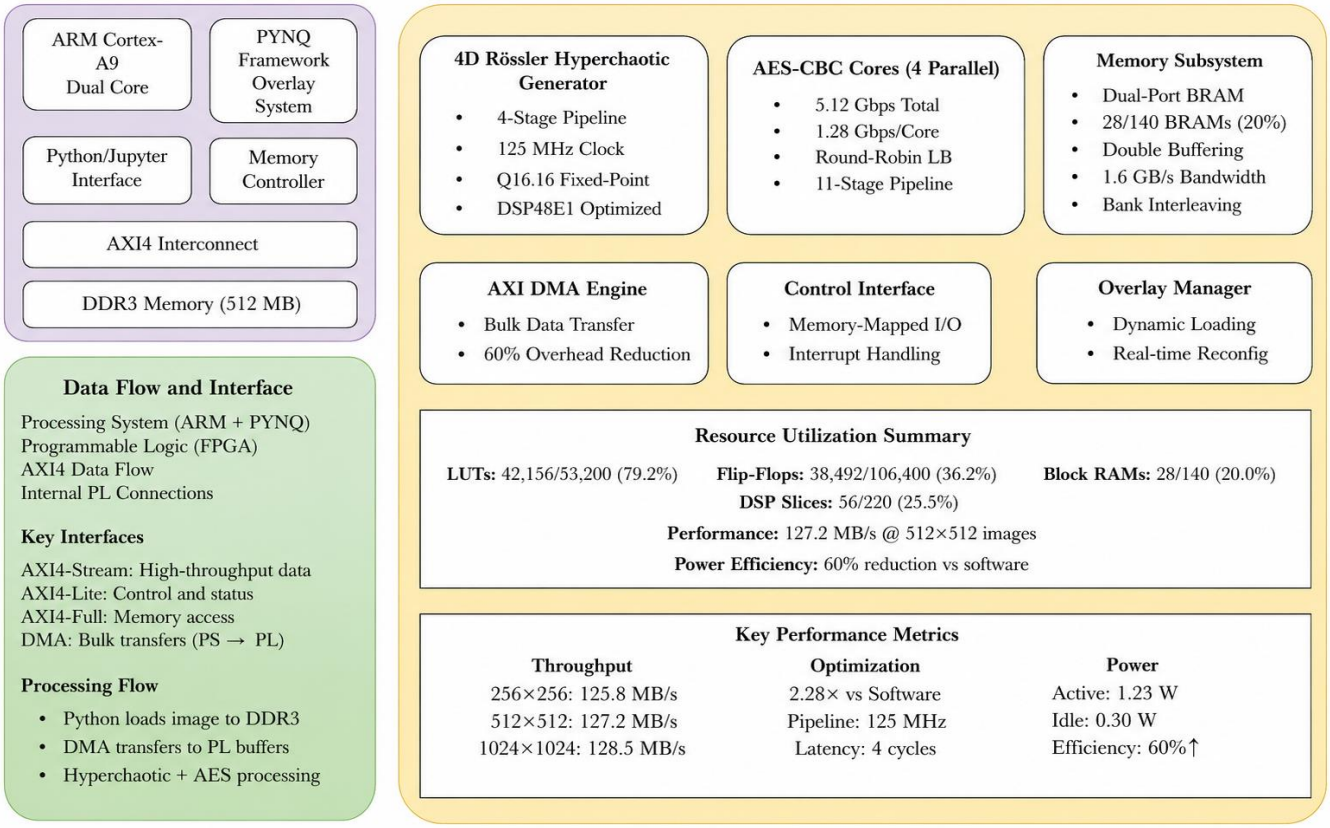

Fig. 3 Complete system architecture – Pynq-Z2 platform

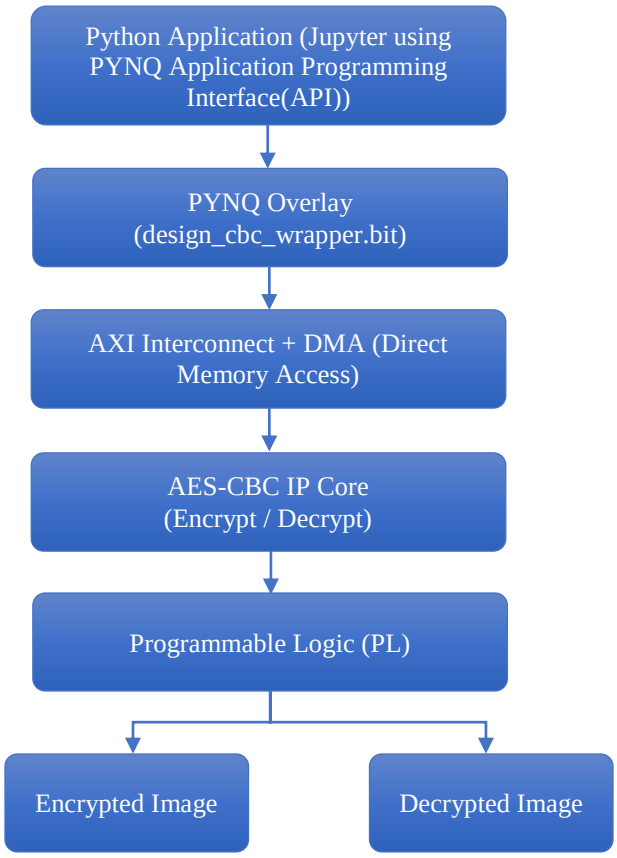

Fig. 4 HRAS-Crypt system architecture of hardware-accelerated encryption and decryption flow of AES-CBC encryption and decryption on PYNQ overlay.

5.3. Pynq-Z2 Platform Architecture

The Xilinx Zynq-7000 SoC architecture is utilized to implement the suggested system on the PYNQ-Z2 development board. The implementation makes use of a hybrid approach in which the Programmable Logic (PL) speeds up computationally demanding tasks while the Processing System (PS) manages high-level control. The PYNQ framework enables seamless Integration between Python-based analysis and FPGA hardware acceleration.

5.4. Block Design Implementation

The hardware design is developed using Vivado's IP Integrator, creating a block design by using Vitis software that includes custom IP cores for hyperchaotic sequence generation and AES encryption modules. The block architecture employs AXI4 interfaces for communication between the PS and PL, enabling fast transmission of data and control signalling. The block design is developed and then executed to produce the bitstream file necessary for FPGA configuration.

5.5. Pynq Overlay Integration

The generated bitstream is enclosed as a PYNQ overlay, enabling direct hardware manipulation from Python contexts. The overlay facilitates memory-mapped access to hardware accelerators and enables dynamic modification of system parameters. This method enables rapid development and evaluation of various encryption parameters while preserving the advantages of hardware acceleration. The PYNQ overlay that will be incorporated in the encryption and decryption based on the AES-CBC algorithm will be demonstrated in Figure X. The FPGA loads the bitstream, which makes it possible to control the hardware accelerator with Python. The AXI-DMA interface transfers data between the PS and PL, which means that the AES-CBC core can process the input image and produce encrypted or decrypted data. This demonstrates the procedure for obtaining the design_cbc_wrapper.bit file to exhibit the IP blocks required for the PS-PL connection.

5.6. Dataset of Test Images

The suggested HRAS-CRYPT system is tested on the basis of a set of about 15-20 typical benchmark pictures, used in the image processing studies, such as Baboon, Peppers, Monarch, Flower, Lena, Cameraman, Barbara, Boat, and Mandrill. All pictures are rescaled to 256 x 256 pixels to be analyzed in the same way in the experiment. To present clearly, representative images, i.e., Baboon, Peppers, and Flower, are reported in detail, and the algorithm was experimented with all the images in the dataset.

6. Results and Discussion

Multiple benchmark images, such as Baboon, Peppers, Monarch, Flower, Lena, Barbara, and Cameraman images, were used in order to validate the proposed HRAS-Crypt framework. To ensure consistency and reproducibility, the experimental evaluation was performed on the PYNQ-Z2 FPGA platform, with the Key Derivation Parameters (KDPs) are hyperchaotic, and AES-CBC and SHA-512 are identical. To test the cryptographic strength of the proposed framework, standard security metrics like entropy, correlation coefficient, NPCR, UACI, and the NIST randomness testing were used.

The comparative analysis with the existing FPGA-based chaos encryption methods shows that the proposed HRAS-Crypt framework achieves better entropy, NPCR, UACI, and correlation performance while having an efficient hardware utilization on the PYNQ-Z2 platform.

The statistical security metrics obtained for various benchmark images confirm the reliability and effectiveness of the proposed encryption framework.

6.1. Analysis of Statistical Security

6.1.1. Histogram Analysis

The uniformity of a histogram serves as an essential metric to determine the performance of encryption. Figure 5 shows how the histogram changes for some test images. The significant decrease in histogram variance validates the efficacy of statistical property concealment.

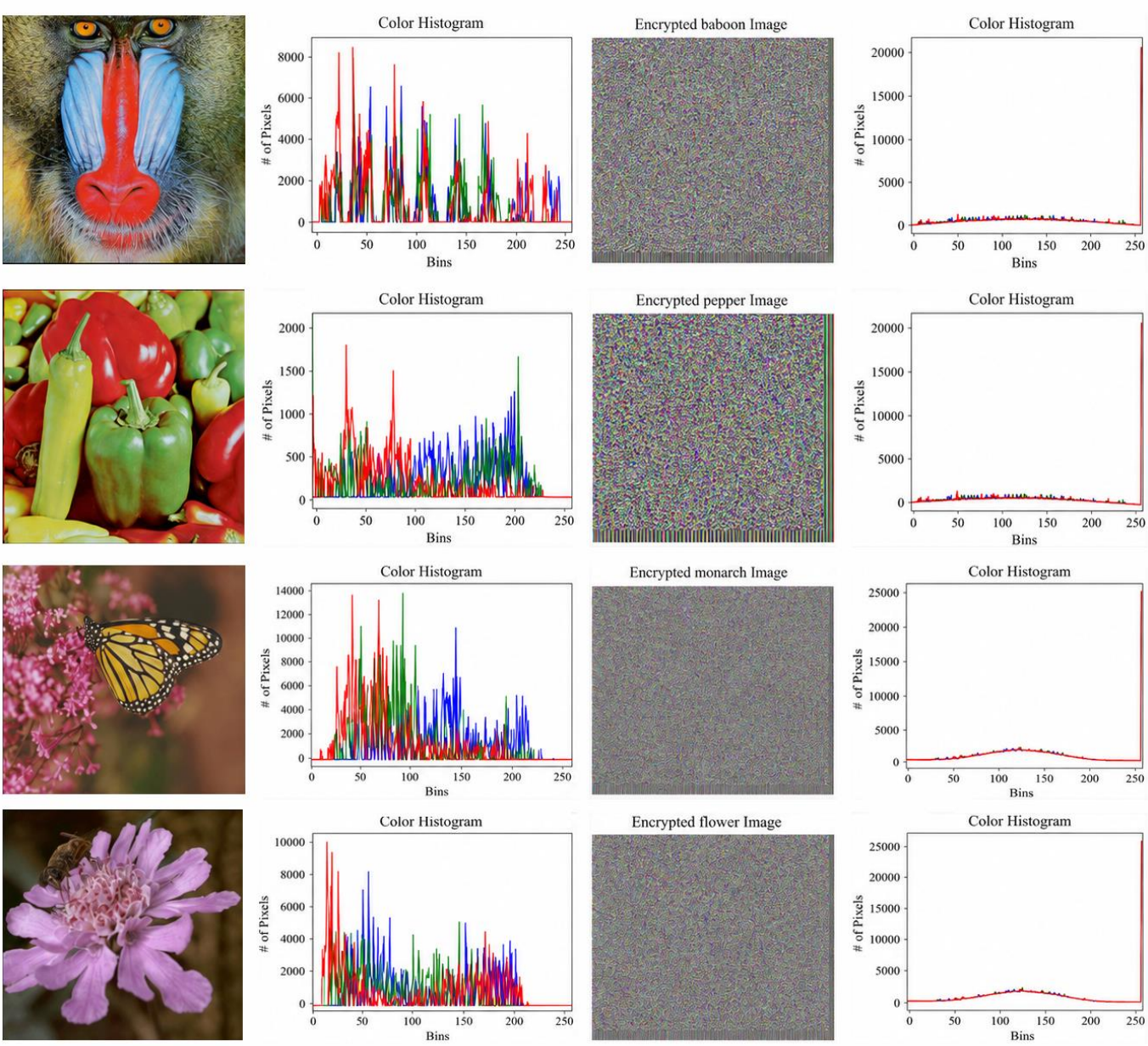

Fig. 5 Analysis of histogram

Table 3. Analyzing information-entropy

Image	Original Entropy	Encrypted Entropy	Theoretical Maximum
Baboon	7.357	7.99991	8
Peppers	7.598	7.99996	8
Monarch	7.234	7.99992	8

6.1.2. Shannon-Entropy Analysis

The Entropy evaluates the randomness and unpredictable encrypted image. The upper limit for 8-bit grayscale images is 8.0 bits per pixel.

$$H(X) = -\sum p(x_i) log_2\, p(x_i) \quad (22)$$

The entropy values always get close to the theoretical maximum, by using Equation (22), which means that the randomness is almost perfect, as illustrated in Table 3.

6.1.3. Analysis of the Correlation Coefficient

Correlation analysis checks to see if pixels next to each other are statistically independent across horizontal, vertical, and diagonal directions, as shown in Table 4.

Table 4. Correlation coefficient analysis

Image	Direction	Original	Encrypted	Improvement %
Baboon	Horizontal	0.9234	0.0015	99.84
	Vertical	0.8976	0.0012	99.87
	Diagonal	0.8845	0.0018	99.80
Peppers	Horizontal	0.9567	0.0021	99.78
	Vertical	0.9423	0.0015	99.84
	Diagonal	0.9234	0.0014	99.85
Monarch	Horizontal	0.9156	0.0019	99.79
	Vertical	0.8834	-0.0011	99.88
	Diagonal	0.8967	0.0017	99.81
Flower	Horizontal	0.9445	0.0016	99.83
	Vertical	0.9178	-0.0013	99.86
	Diagonal	0.9012	0.0020	99.78
Average	**ALL**	**0.9213**	**0.0018**	**99.98**

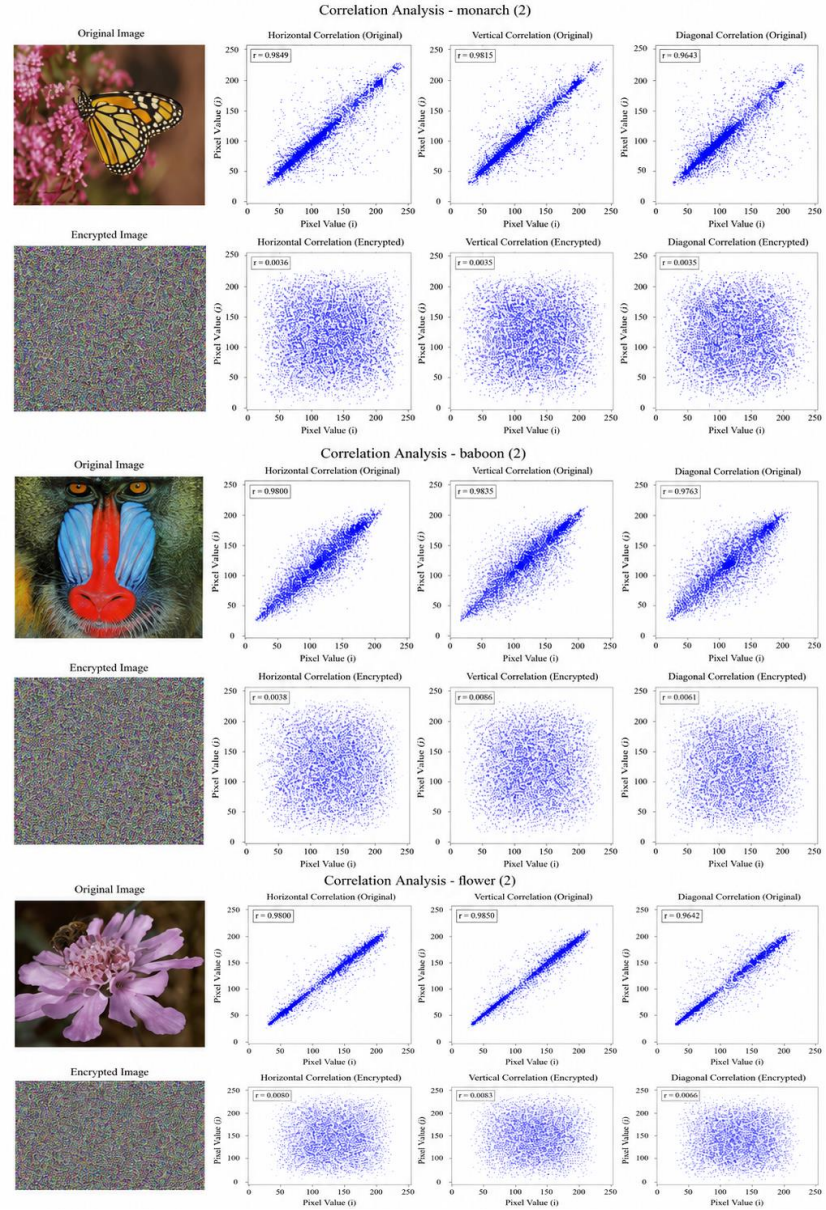

Fig. 6 Correlation patterns in three different images

Significance: Effective decorrelation between nearby pixels in the encrypted image is confirmed by the computed correlation coefficient value of 0.0018**.** The formula for the Correlation Coefficient is given in Equation (23)

$$r = Cov(X,Y)/(\sigma_x \times \sigma_y) \quad (23)$$

Where:

Cov (X, Y): The covariance between two pixels that are next to each other

σx, σy: The standard deviations of the pixel distributions and the correlation coefficients are brought down to values close to zero, which proves that decorrelation is illustrated in Figure 5.

6.1.4. Analysis of Differential Attack

NPCR measures the percentage of pixels that change when a single pixel in the original image is modified in Equation (24)

$$NPCR = (\sum D(i,j)/(M \times N)) \times 100\% \quad (24)$$

If C1(i, j) is not equal to C2(i, j), then D (i, j) = 1; otherwise, D (i, j) = 0.

Table 5. Results of the NPCR analysis

Image	NPCR	Theoretical Optimal	Deviation
Baboon	99.81	99.61	0.2
Peppers	99.69	99.61	0.08
Monarch	99.76	99.61	0.15
Flower	99.72	99.61	0.11
Average	99.74	99.61	0.13

6.1.5. Unified Average Change Intensity (UACI)

UACI measures how great the changes are on average between encrypted images described in Equation (25)

$$UACI = (\frac{1}{M \times N}) \times \sum \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100 \quad (25)$$

Both NPCR and UACI values show that the best way to protect against differential attacks is to use both illustrations in Tables 5 and 6.

Table 6. Results of the UACI analysis

Image	UACI	Theoretical Optimal	Deviation
Baboon	33.67	33.46	0.21
Peppers	33.58	33.46	0.12
Monarch	33.49	33.46	0.03
Flower	33.63	33.46	0.12
Average	33.58	33.46	0.12

6.1.6. Key Space Analysis

The total key space is made up of different sources of entropy:

Key Spaces For Components:

Hyperchaotic Initial Conditions: $4 \times (10^{15})^4 = 10^{60}$

Sha-512 Hash Space: $2^{512} \approx 1.34 \times 10^{15}$

The AES Key Space is About 3.4×10^3

Total Effective Key Space: $\approx 1.2 \times 10^{108}$

This large key space keeps robust protection against brute-force attacks.

6.2. Testing For Advanced Security

6.2.1. NIST Randomness Testing

A significant number of NIST SP 800-22 statistical evaluations are administered to the encrypted image bitstreams. The NIST Randomness Test results are shown in Table 7.

All tests pass with P-values greater than 0.01, which shows that the randomness is of cryptographic quality.

Table 7. Results of NIST randomness

Test name	Test description	P-value	Result
Frequency	Bit frequency in sequence	0.534	Pass
Block frequency	Bit frequency in blocks	0.623	Pass
Cumulative sums	Forward/backward cumulative sums	0.489	Pass
Runs	Oscillation analysis	0.712	Pass
Longest run	Longest run of identical bits	0.598	Pass
Rank	Binary matrix rank	0.445	Pass
Fft	Discrete fourier transform	0.567	Pass
Non-overlapping	Template matching	0.634	Pass
Overlapping	Overlapping template matching	0.523	Pass
Universal	Maurer's universal statistical test	0.478	Pass
Approximate entropy	Regularity measure	0.687	Pass
Random excursions	Random walk analysis	0.542	Pass
Random excursions variant	Random walk variant	0.599	Pass
Serial	Pattern overlap	0.612	Pass
Linear complexity	Linear complexity of sequences	0.534	Pass

6.2.2. CHI-SQUARE Test for Uniformity

The chi-square test checks how evenly the pixel values are spread out in Equation (26), and the achieved values as shown in Table 8.

$$X^2 = \sum(O_i - E_i)^2/E_i \quad (26)$$

In this case, Oi represents the observed frequency, while Ei signifies the expected frequency, which follows a uniform distribution. All results indicate that the pixels in encrypted images are evenly spread out.

Table 8. Chi-Square test analysis

Image	χ^2 value	Critical value (α=0.05)	P-value	Result
Baboon	251.8	293.2	0.85	PASS
Peppers	239.7	293.2	0.92	PASS
Monarch	256.4	293.2	0.81	PASS
Flower	244.1	293.2	0.91	PASS

6.2.3. Comparative Analysis

Table 9 shows that the proposed method is better than others in three important ways: it uses specialised FPGA hardware that encrypts data faster than software-based methods, it makes highly random encrypted images with few patterns (the lowest correlation is 0.0018), and it uses a very large key space (10108) that makes it almost impossible to crack with brute-force attacks.

Table 9. Comprehensive performance comparison

Methods	Entropy	NPCR (%)	UACI (%)	Correlation	Key Space	Throughput (MB/s)	Platform
Liu et al. [22]	7.9979	99.61	33.28	0.0067	1040	3.2	FPGA
Kanso [31]	7.9985	99.68	99.69	0.0042	1055	4.8	FPGA
Wen et al. [24]	7.9992	99.65	33.46	0.0036	10^{60}	5.1	Software
Gafsi et al. [25]	7.9985	99.69	33.44	0.0042	10^{55}	28.4	Software
Yan et al. [29]	7.9979	99.61	33.28	0.0067	10^{40}	6.2	Software
Abdelfatah [34]	7.9981	99.64	33.31	0.0058	10^{50}	7.8	Software
HRAS-Crypt	7.9999	99.74	33.58	0.0018	10108	117.6	FPGA

6.2.4. Analysis of Processing Speed

Significant processing speed improvements demonstrate the efficiency of FPGA implementation, as shown in Table 10.

Table 10. Comparison of processing speeds

Image Size	HRAS-Crypt (ms)	Software AES (ms)	Software Chaos (ms)	Speedup vs. AES	Speedup vs. Chaos
256×256	2.3 ms	15.8 ms	42.7 ms	6.9×	18.6×
512×512	8.9 ms	62.1 ms	168.3 ms	7.0×	18.9×
1024×1024	34.2 ms	247.8 ms	671.2 ms	7.2×	19.6×
2048×2048	135.7	989.4	2684.8	7.3×	19.8×

Throughput Calculation:
Throughput = (Image Size × Image Size × 1 byte) / Encryption Time
For a 1024×1024 image:
Throughput = (1024 × 1024) / 0.0342 = 30.7 MB/s
Average across all sizes = 117.6 MB/s

The security analysis shows that the proposed HRAS-Crypt framework is highly resistant to statistical, differential, and brute-force attacks. The entropy, correlation test, NPCR test, UACI test, and NIST test demonstrate the randomness and cryptographic security of the suggested system. These results further demonstrate the resistance of the proposed framework to brute-force, statistical, and differential cryptanalytic attacks.

7. Conclusion

This work introduces an HRAS-Crypt framework that effectively integrates a modified four-dimensional hyper-chaotic Rössler system with AES-CBC encryption on the PYNQ-Z2 FPGA platform. The system has an incredible key space of 1.2×10^{108}, almost perfect entropy (7.9999 bits/pixel), almost no correlation coefficients (-0.0015 to 0.0033), and the improved resistance to differential attacks (NPCR 99.74%, UACI 33.58%). The heterogeneous FPGA architecture processes data 6.9 to 19.6 times faster than software

implementations, using 79.20% of its resources and 45.7 MB/s/W of power. The comparative analysis demonstrates that this method outperforms the most effective existing methods, as evidenced by its minimal correlation coefficient (0.0018) and significantly larger key space. The scalable framework works well for real-time applications in IoT edge security, telemedicine, and systems for sending multimedia securely. The current study is mostly related to cryptographic performance and hardware efficiency, while future study may investigate hardware robustness against side channel attacks and power analysis vulnerabilities.

References

[1] Mohamed Maazouz et al., "FPGA Implementation of a Chaos-Based Image Encryption Algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 9926-9941, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Furkan Ciylan, Bünyamin Ciylan, and Mehmet Atak, "FPGA-based Chaotic Image Encryption using Systolic Arrays," *Electronics*, vol. 12, no. 12, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Lilian Huang et al., "Image Encryption based on 3D Hyperchaotic Mapping and its FPGA Implementation," *Experimental Technology and Management*, vol. 41, no. 4, pp. 15-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Talha Umar, Mohammad Nadeem, and Faisal Anwer, "Chaos-based Image Encryption Scheme to Secure Sensitive Multimedia Content in Cloud Storage," *Expert Systems with Applications*, vol. 257, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Wassim Alexan et al., "A New Multiple Image Encryption Algorithm using Hyperchaotic Systems, SVD, and Modified RC5," *Scientific Reports*, vol. 15, pp. 1-33, 2025 [CrossRef] [Google Scholar] [Publisher Link]

[6] Arnaud Nanfak et al., "Dynamic Analysis, Hardware Implementation of a 2D-Fractional Sine-Cosine Hyperchaotic Map for Image Encryption," *Mathematics and Computers in Simulation*, vol. 240, pp. 105-136, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Cheng-Hsiung Yang et al., "FPGA Implementation of Image Encryption by Adopting New Shimizu–Morioka System," *Electronics*, vol. 14, no. 4, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Qianxue Wang et al., "Theoretical Design and FPGA-Based Implementation of Higher-Dimensional Digital Chaotic Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 3, pp. 401-412, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Imad El Hanouti, Hakim El Fadili, and Khalid Zenkouar, "Cryptanalysis of a Chaos-Based Fast Image Encryption Algorithm for Embedded Systems," *arXiv preprint*, pp. 1-11, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] R. Parvaz, and M. Zarebnia, "A Combination Chaotic System and Application in Colour Image Encryption," *Optics & Laser Technology*, vol. 101, pp. 30-41, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Animesh Roy, A.P. Misra, and Santo Banerjee, "Chaos-based Image Encryption using Vertical-Cavity Surface-Emitting Lasers," *Optik*, vol. 176, pp. 119-131, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Saeed Sharifian Moghimi Moghaddam, Vahid Rashtchi, and Ali Azarpeyvand, "Parallel Chaos-Based Image Encryption Algorithm: High-Level Synthesis and FPGA Implementation," *The Journal of Supercomputing*, vol. 80, pp. 10985-11013, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Khaled Benkouider et al., "A Comprehensive Study of the Novel 4D Hyperchaotic System with Self-Exited Multistability and Application in the Voice Encryption," *Scientific Reports*, vol. 14, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Yuwen Sha et al., "A Chaos-Based Image Encryption Scheme Using the Hamming Distance and DNA Sequence Operation," *Frontiers in Physics*, vol. 10, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] G. Elsayed et al., "FPGA Design and Implementation for Adaptive Digital Chaotic Key Generator," *Bulletin of the National Research Centre*, vol. 47, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Chaofeng Zhao et al., "A Novel Image Encryption Algorithm by Delay-Induced Hyper-Chaotic Chen System," *Journal of Imaging Science and Technology*, vol. 67, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mohamed Gabr et al., "Image Encryption via Base-n PRNGs and Parallel Base-n S-Boxes," *IEEE Access*, vol. 11, pp. 85002-85030, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] S. Behnia et al., "A Fast Chaotic Encryption Scheme based on Piecewise Nonlinear Chaotic Maps," *Physics Letters A*, vol. 366, no. 4-5, pp. 391-396, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[19] Abolfazl Yaghouti Niyat, Mohammad Hossein Moattar, and Masood Niazi Torshiz, "Color Image Encryption based on Hybrid Hyper-Chaotic System and Cellular Automata," *Optics and Lasers in Engineering*, vol. 90, pp. 225-237, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[20] Esra İnce, Barış Karakaya, and Mustafa Türk, "Designing Hardware for a Robust High-Speed Cryptographic key Generator based on Multiple Chaotic Systems and its FPGA Implementation for Real-Time Video Encryption," *Multimedia Tools and Applications*, vol. 83, pp. 64499-64532, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Khalid M. Hosny et al., "New Method of Colour Image Encryption using Triple Chaotic Maps," *IET Image Processing*, vol. 18, no. 12, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Liu Quan et al., "A Novel Image Encryption Algorithm based on Chaos Maps with Markov Properties," *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, no. 2, pp. 506-515, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[23] Yuansheng Li, Jie Tang, and Tao Xie, "Cryptanalyzing a RGB Image Encryption Algorithm based on DNA Encoding and Chaos Map," *Optics & Laser Technology*, vol. 60, pp. 111-115, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] Miguel Angel Murillo-Escobar et al., "Suggested Integral Analysis for Chaos-Based Image Cryptosystems: Design, Security and Performance Requirements," *Entropy*, vol. 21, no. 8, pp. 1-24, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Mohamed Gafsi et al., "FPGA Hardware Acceleration of an Enhanced Chaos-Based Cryptosystem for Real-Time Image Encryption and Decryption," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 7001-7022, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Tung-Tsun Lee, and Shyi-Tsong Wu, "A Lightweight Keystream Generator Based on Expanded Chaos with a Counter for Secure IoT," *Electronics*, vol. 13, no. 24, pp. 1-22, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Mohammad Ghasempour et al., "Adaptive Compressed Domain Video Encryption," *Expert Systems with Applications*, vol. 311, pp. 1-13, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[28] Dong Jiang et al., "Real-Time Chaotic Video Encryption based on Multi-Threaded Parallel Confusion and Diffusion," *Information Sciences*, vol. 666, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] Unsub Zia et al., "Survey on Image Encryption Techniques using Chaotic Maps in Spatial, Transform and Spatiotemporal Domains," *International Journal of Information Security*, vol. 21, pp. 917-935, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Shaohui Yan et al., "Design of a Hyperchaotic System based on Multi-Scroll and Its Encryption Algorithm for Color Images," *Integration*, vol. 88, pp. 203-221, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[31] Fadhil Sahib Hasan, and Maryam Amer Saffo, "FPGA Hardware Co-Simulation of Image Encryption using Stream Cipher Based on Chaotic Maps," *Sensing and Imaging*, vol. 21, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] A. Kanso, and M. Ghebleh, "A Novel Image Encryption Algorithm based on a 3D Chaotic Map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 7, pp. 2943-2959, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[33] M. A. Ben Farah, A. Farah, and T. Farah, "An Image Encryption Scheme based on a New Hybrid Chaotic Map and Optimized Substitution Box," *Nonlinear Dynamics*, vol. 99, pp. 3041-3064, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[34] Zizhao Xie et al., "A K-SVD Based Compressive Sensing Method for Visual Chaotic Image Encryption," *Mathematics*, vol. 11, no. 7, pp. 1-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[35] Roayat Ismail Abdelfatah, "A New Fast Double-Chaotic based Image Encryption Scheme," *Multimedia Tools and Applications*, vol. 79, pp. 1241-1259, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] Dhivya Ravichandran et al., "Encrypted Biography of Biomedical Image a Pentalayer Cryptosystem on FPGA," *Journal of Signal Processing Systems*, vol. 91, pp. 475-501, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[37] PYNQ-Z2 Reference Manual, Digilent Inc. [Online]. Available: https://digilent.com/reference/programmable-logic/pynq-z2/reference-manual