






## Article

# Secure Encryption of Biomedical Images Based on Arneodo Chaotic System with the Lowest Fractional-Order Value

Berkay Emin <sup>1,2</sup>, Akif Akgul <sup>3</sup>, Fahrettin Horasan <sup>4</sup>, Abdullah Gokyildirim <sup>5</sup>, Haris Calgan <sup>6</sup>  
and Christos Volos <sup>7,\*</sup>

<sup>1</sup> Department of Electronics and Automation, Osmancık Omer Derindere Vocational School, Hitit University, 19500 Corum, Turkey; berkayemin@hitit.edu.tr

<sup>2</sup> Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Yozgat Bozok University, 66100 Yozgat, Turkey

<sup>3</sup> Department of Computer Engineering, Faculty of Engineering, Hitit University, 19030 Corum, Turkey; akifakgul@hitit.edu.tr

<sup>4</sup> Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kırıkkale University, 71450 Kırıkkale, Turkey; fhorasan@kku.edu.tr

<sup>5</sup> Department of Electrical and Electronics Engineering, Faculty of Engineering and Natural Sciences, Bandirma Onyedi Eylul University, Bandirma, 10200 Balıkesir, Turkey; agokyildirim@bandirma.edu.tr

<sup>6</sup> Department of Electrical and Electronics Engineering, Faculty of Engineering, Balıkesir University, Cagis, 10145 Balıkesir, Turkey; haris.calgan@balikesir.edu.tr

<sup>7</sup> Laboratory of Nonlinear Systems, Circuits, and Complexity (LaNSCom), Physics Department, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

\* Correspondence: volos@physics.auth.gr

**Abstract:** Fractional-order (FO) chaotic systems exhibit richer and more complex dynamic behaviors compared to integer-order ones. This inherent richness and complexity enhance the security of FO chaotic systems against various attacks in image cryptosystems. In the present study, a comprehensive examination of the dynamical characteristics of the fractional-order Arneodo (FOAR) system with cubic nonlinearity is conducted. This investigation involves the analysis of phase planes, bifurcation diagrams, Lyapunov exponential spectra, and spectral entropy. Numerical studies show that the Arneodo chaotic system exhibits chaotic behavior when the lowest fractional-order (FO) value is set to 0.55. In this context, the aim is to securely encrypt biomedical images based on the Arneodo chaotic system with the lowest FO value using the Nvidia Jetson Nano development board. However, though the lowest FO system offers enhanced security in biomedical image encryption due to its richer dynamic behaviors, it necessitates careful consideration of the trade-off between high memory requirements and increasing complexity in encryption algorithms. Within the scope of the study, a novel random number generator (RNG) is designed using the FOAR chaotic system. The randomness of the random numbers is proven by using internationally accepted NIST 800-22 and ENT test suites. A biomedical image encryption application is developed using pseudo-random numbers. The images obtained as a result of the application are evaluated with tests such as histogram, correlation, differential attack, and entropy analyses. As a result of the study, it has been shown that encryption and decryption of biomedical images can be successfully performed on a mobile Nvidia Jetson Nano development card in a secure and fast manner.

**Keywords:** chaos; fractional-order systems; cryptography; embedded systems; security analysis



**Citation:** Emin, B.; Akgul, A.; Horasan, F.; Gokyildirim, A.; Calgan, H.; Volos, C. Secure Encryption of Biomedical Images Based on Arneodo Chaotic System with the Lowest Fractional-Order Value. *Electronics* **2024**, *13*, 2122. <https://doi.org/10.3390/electronics13112122>

Academic Editors: Tommaso Addabbo and Fabio Pareschi

Received: 1 May 2024

Revised: 18 May 2024

Accepted: 24 May 2024

Published: 29 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Advancements in computer networks, storage devices, and imaging technology have led to the widespread utilization of images, videos, and text in a multitude of sectors. However, with the increased exchange of these assets over public networks, they become susceptible to various security risks, including eavesdropping, unauthorized alterations, and duplication, among others [1]. Particularly in the e-healthcare domain, biomedical

images are transmitted through public networks, which makes these images vulnerable to potential attacks by hackers [2].

Biomedical images are often highly sensitive to external influences, where even minor data alterations can yield significant differences in the final outcomes. In certain situations, an incorrect diagnosis can have life-threatening consequences. Therefore, ensuring the security of biomedical data presents a critical challenge, necessitating their concealment during storage, manipulation, and transmission [3]. Furthermore, encryption of medical images necessitates the implementation of effective techniques to ensure patient privacy. Hence, the secure transmission of biomedical images has been addressed in the literature through the application of various encryption algorithms.

In the fields of image encryption and secure communication, chaos systems are often used due to their advantages such as sensitivity to initial values, ergodicity, and complex behavior. These features have expanded the utilization of encryption schemes in medical and biometric images employing chaotic systems [4]. In this context, Pérez et al. introduced a fingerprint image encryption scheme based on a hyperchaotic Rössler map. Their objective was to create a highly secure encryption method using strong pseudorandom sequences to prevent identity theft [5]. Han et al. introduced the utilization of a chaotic sequence derived from multi-scroll chaotic attractors for the encryption of fingerprint images [6]. Hikal and Eid introduced an innovative encryption method that applies a combination of various chaotic maps to enhance the security of palm print images [7]. Boyraz et al. utilized two distinct 1D discrete-time chaotic systems to encrypt dorsal vein images captured from individuals [8].

The previously mentioned biometric encryption applications utilize a range of integer-order chaotic systems. Nevertheless, it is possible to enhance the security of encryption by increasing the complexity of chaotic systems through fractional-order analysis [9]. Therefore, FO chaotic systems have attracted significant interest, particularly in image encryption applications, due to their increased complexity when compared to integer-order systems. To give some examples, Chen et al. proposed a 3D fractional discrete Hopfield neural network for image encryption [10]. Liu et al. generalized the six-degree discrete chaotic map (SODCM) fractionally and implemented image encryption for information security. Then, they proved the encryption efficiency with related tests [11]. Xu et al. proposed a new FO chaotic system based on a four-neuron Hopfield Neural Network (HNN) model. They designed a pseudorandom number generator using the proposed system and realized a new image encryption application [12]. In another study, Kiren et al. proposed a new discrete-time chaotic encryption mechanism for the efficient encryption of large-scale data (images and videos) on the Nvidia Jetson Orin development board [13]. In addition to the Nvidia Jetson board, Raspberry Pi [14], a DSP board [15], and an FPGA board [16] are utilized to digitally implement FO chaotic systems for image encryption.

On the other hand, there are limited studies that utilize fractional calculus in the field of medical image encryption. Doui et al. introduced a new 2D chaotic system based on FO Meixner polynomials and achieved a high level of security with their proposed encryption scheme for biomedical multimedia [17]. The brief literature review on biomedical data security emphasizes the preference for FO chaotic systems because of their inherently complex dynamic behaviors. Consequently, these systems have the potential to enhance the security of encryption schemes. Additionally, the security level of encryption schemes based on chaos depends on the dynamical properties, initial conditions, and system parameters of the chaotic system. In this context, the security level can be enhanced in an encryption scheme when cubic nonlinearity is considered. This is because a chaotic system with a cubic term complicates decryption and becomes more challenging to analyze [18]. Furthermore, it is possible to increase dynamical richness by identifying operating conditions where a system is in chaos at different FO values [19].

Recently, it has been reported in [20] that the FOAR system with cubic nonlinearity has rich dynamics while setting various fractional-order values. However, the applicability of the FOAR system with the lowest fractional-order value has not been tested yet for

cryptography applications, especially in biometric image encryption. On the other hand, fractional-order systems inherently require high memory capacity at low values, while the complexity of the application increases. The trade-off between high memory requirements and complexity in encryption algorithms should be accurately adjusted. These challenges make it difficult to establish low-cost, lightweight encryption structures in places where patient privacy is crucial.

In this study, an encryption scheme for medical images, specifically brain MRI, is proposed utilizing the introduced FOAR system. Unlike most biometric encryption studies, a comprehensive analysis of FO chaotic systems is carried out in this study by examining bifurcation maps, Lyapunov exponents' spectra, and spectral entropy diagrams. Furthermore, an encryption scheme designed for brain MRI is developed based on the Arneodo system. The novelty of our study lies in enhancing encryption security by reducing the FO value of the chaotic system to its minimum applicable level. The remaining significant aspects of the proposed research are described in terms of contributions as follows:

- Detailed analyses of the FOAR chaotic system are conducted using chaotic time series, bifurcation maps, spectral entropies, and Lyapunov exponents. Consequently, the minimum applicable FO value, which is 0.55, is investigated and applied in the encryption application.
- A RNG is designed on the low-cost, lightweight Nvidia Jetson Nano development board.
- Brain MRI images are encrypted using the generated random numbers.
- To assess the accuracy of the RNG, NIST 800-22 and ENT tests are conducted, and histogram, correlation, differential attack, entropy, and time analyses are performed to evaluate the success of the encryption process.

The rest of the paper is organized as follows: The second section provides a brief overview of fractional calculus and parametric FO analyses of the Arneodo system. Section 3 covers the design of the FOAR-based RNG on the development board. Section 4 presents the secure encryption of biomedical images using the FOAR chaotic system. Additionally, Section 4 reports the results of various security analyses. The final section provides the conclusion.

## 2. Fractional-Order Analysis of Arneodo System with Cubic Nonlinearity

In this section, a brief overview of fractional calculus and operators is given. Additionally, a dynamical investigation of the FOAR chaotic system is presented for different fractional-order parameter  $q$  values, including the depiction of bifurcation maps, spectral entropy complexity diagrams, and Lyapunov spectra. Towards the end of the section, chaotic time series run under suitable operating conditions obtained from numerical analyses are shown for the minimum applicable parameter  $q$  value.

### 2.1. A Brief Overview of Fractional Calculus

Fractional derivatives and integrals have become increasingly important in the fields of engineering and mathematics, providing essential tools for scientists and researchers involved in practical real-world applications. Among the various fractional operators, Caputo and Grünwald–Letnikov fractional derivatives are selected for numerical studies in this paper. Caputo's operator simplifies the determination of initial conditions for initial value problems when applied to both continuous-time and discrete-time systems. The definition of Caputo's derivative with fractional order  $q$  is given below [21–23]:

$${}^C D_{t_0}^q f(t) = \begin{cases} \frac{1}{\Gamma(m-q)} \int_{t_0}^t \frac{f^{(m)}(\tau)}{(t-\tau)^{q+1-m}} d\tau & m-1 < q \leq m \\ \frac{d^m}{dt^m} f(t) & q = m \end{cases}, \quad (1)$$

Here,  $m$  represents the integer closest to  $q$ , where  $m > q$ .  $\Gamma(m - q)$  is Euler's Gamma function. The definition of the Grünwald–Letnikov fractional derivative with order  $q$  is as follows [23]:

$${}^{GL}D_t^q f(t) = \lim_{h \rightarrow 0} \frac{1}{h^q} \sum_{j=0}^{\infty} (-1)^j \binom{q}{j} f(t - jh), \quad (2)$$

where  $\binom{q}{j}$  represents the binomial coefficient and is calculated using Euler's Gamma function. Defining the FOAR system is accomplished by employing the  $q$ th order Caputo fractional derivative, as described below:

$$\begin{aligned} {}^*D^{q_1} x &= y \\ {}^*D^{q_2} y &= z \\ {}^*D^{q_3} z &= ax + by + cz - x^3 \end{aligned}, \quad (3)$$

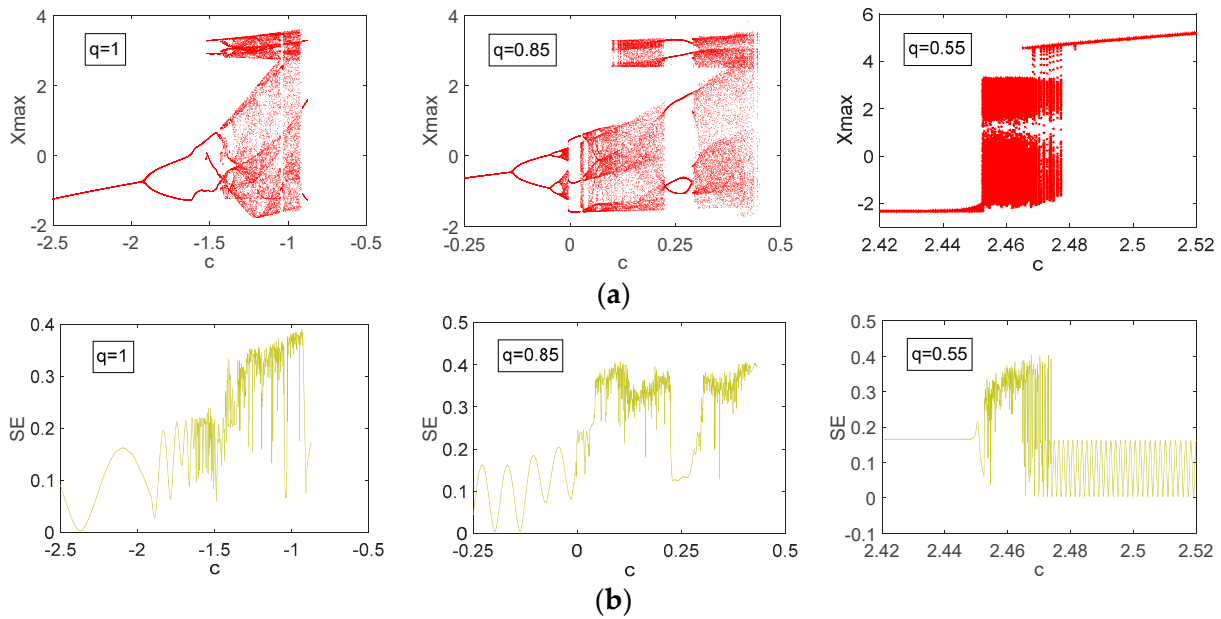
System (3) produces chaotic sequences for integer-order systems when parameters  $a$ ,  $b$ , and  $c$  stand for 5.5,  $-3.5$ , and  $-1$ , respectively. In the following subsection, the system (3) is explored through the examination of bifurcation maps, spectral entropy diagrams, and Lyapunov spectra.

Solving a nonlinear FO system is a significant challenge. Specifically, numerically integrating a FO chaotic system can be even more difficult. Therefore, various approaches have been presented in the literature to solve FO differential equations [24]. The Matlab program provides various toolboxes that can be used in the solution of FO systems, such as "fde12", "ninteger", and "FOMCON". The Adams–Bashforth–Moulton predictor–corrector method, implemented in Matlab with Garrappa's "fde12" code [25], utilizes the Caputo fractional definition. This allows for the specification of initial values, including both the function value and integer-order derivatives. Unlike many ad hoc numerical methods for fractional differential equations that are designed for specific cases, the Adams–Bashforth–Moulton method has been rigorously developed and tested across a wide range of equations. This makes it a versatile and straightforward choice for computational implementation [26].

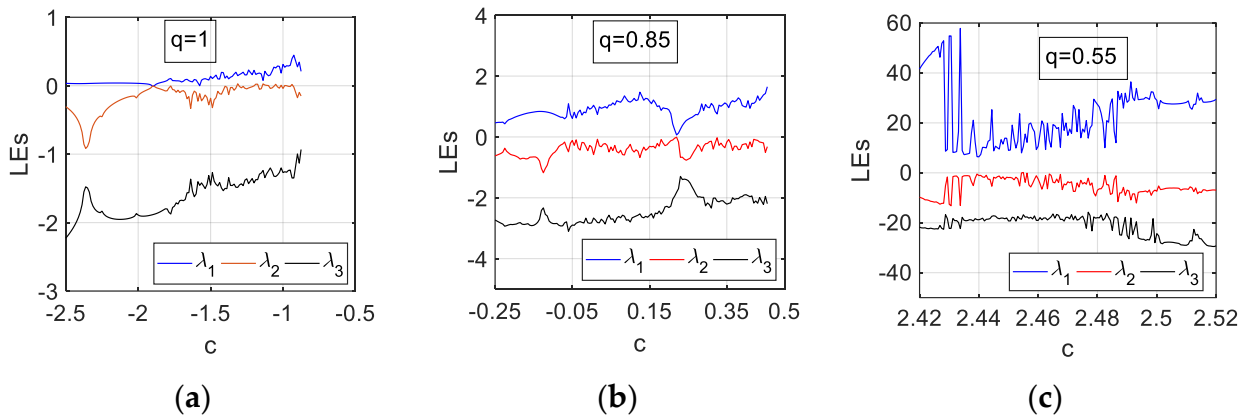
The operating conditions of a FO chaotic system chosen for an engineering application, such as parameter values and initial conditions, can be determined through the execution of numerical simulations, including bifurcation analysis, complexity analysis, and Lyapunov spectra. In this way, the FO value and parameter values at which the system exhibits chaotic behavior are determined, and chaotic time series to be used in the application are obtained. On the other hand, after a series of simulations where the FO value  $q$  is parametrically reduced, the minimal FO chaotic form of the system is identified using bifurcation diagrams corresponding to varying values of parameter  $c$ . Note that the FO Arneodo chaotic system in Equation (3) is considered to have commensurate order, where  $q = q_1 = q_2 = q_3$  in this study.

In this section, bifurcation diagrams, spectral entropy analyses, and phase portraits for the FOAR chaotic system are investigated using the fde12 toolbox. Bifurcation diagrams showing the local maxima of  $x$  concerning the varying values of the parameter " $c$ " for different FO values, along with spectral entropy complexity analysis plots, are presented in Figure 1. In addition, the Lyapunov exponents' spectrum is a useful tool for determining the stability, fractal dimension, and limit set of a dynamical system. Lyapunov exponents can also be utilized to assess the potential presence of chaotic behavior through the observation of sensitivity to initial conditions. However, the phase flow within FO chaotic systems is influenced by their history, making the calculation of Lyapunov exponents more complex. In this regard, an extended Benettin–Wolf algorithm [27] and a method based on memory principles [28] are studied. Because of the nonlocal nature of the FOAR system, the method based on memory principles is utilized, and the corresponding Lyapunov spectra are computed for  $q$  values of 1, 0.85, and 0.55, as depicted in Figure 2. The agreement of Lyapunov exponents with bifurcation maps and complexity diagrams is demonstrated, showing only minor deviations.





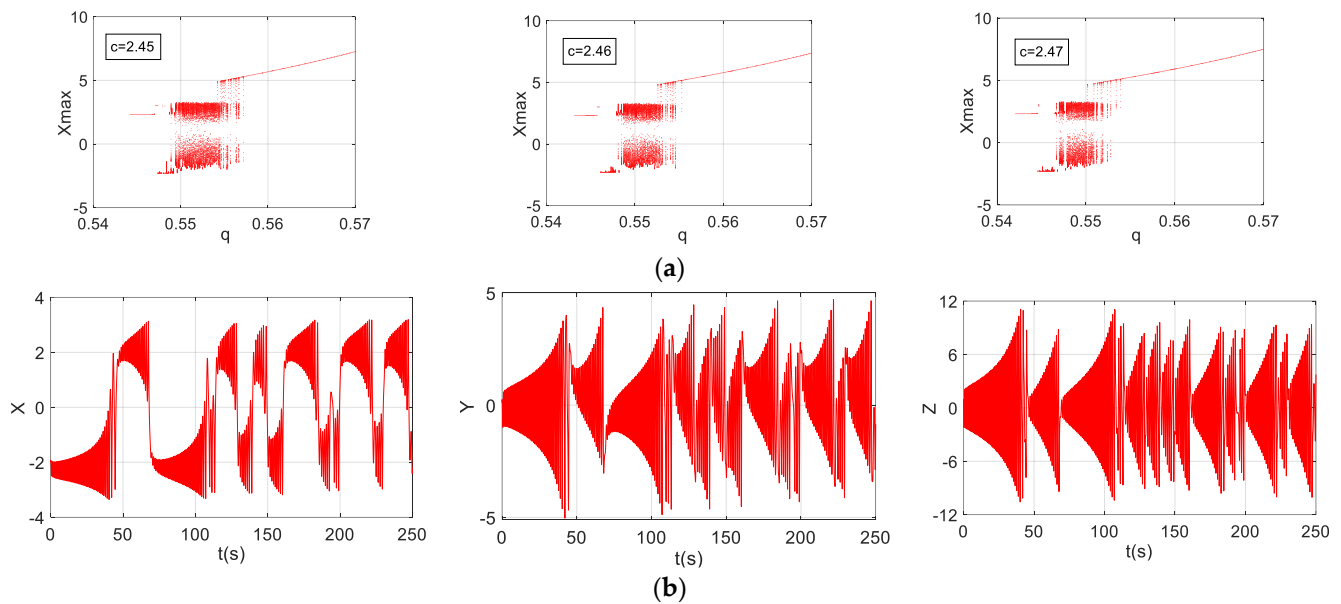
**Figure 1.** Detailed dynamical analyses of the FOAR system for varying FO parameter  $q$  values when  $a = 5.5, b = -3.5, x(0) = -2, y(0) = 0.1, z(0) = 1$ : (a) bifurcation maps; (b) spectral entropy diagrams.



**Figure 2.** Lyapunov exponent spectra of the FOAR system with the varying parameter  $c$  when  $a = 5.5, b = -3.5, x(0) = -2, y(0) = 0.1, z(0) = 1$ : (a)  $q = 1$ ; (b)  $q = 0.85$ ; (c)  $q = 0.55$ .

**2.2. Dynamical Analysis of Arneodo Chaotic System with the Lowest FO Value**

Considering Figures 1 and 2 together, it is determined that the lowest applicable fractional-order value at which the FOAR system exhibits chaotic behavior is approximately  $q = 0.55$ , when  $c$  is between 0.45 and 0.47. However, further investigations are realized to ensure the lowest  $q$ . Since  $c$  is taken as constant and  $q$  as the bifurcation parameters, Figure 3 is obtained. As illustrated in the figure, the chaotic behavior is lost by gradually decreasing  $q$ . Therefore, as the lowest FO value,  $q = 0.55$  is handled to be used in subsections. As seen in Figure 3, at  $q = 0.55$ , the system produces chaotic time series with parameter values  $a = 5.5, b = -3.5$ , and  $c = 2.45$ . In the next section, the encryption application of the system will be simulated under the obtained operating conditions.



**Figure 3.** (a) Bifurcation analyses to determine the lowest  $q$  value of the FOAR system when  $a = 5.5$ ,  $b = -3.5$ ,  $x(0) = -2$ ,  $y(0) = 0.1$ ,  $z(0) = 1$ ; (b) chaotic time series of the FOAR system for  $q = 0.55$  and  $c = 2.45$  when  $a = 5.5$ ,  $b = -3.5$ ,  $x(0) = -2$ ,  $y(0) = 0.1$ ,  $z(0) = 1$ .

### 3. FOAR System-Based Random Number Generator Design

In this section, the determined FOAR system with the minimum applicable  $q$  value, i.e., 0.55, is utilized to design a RNG on the Nvidia Jetson Nano development board. The 4 GB RAM of the Nvidia Jetson Nano suffices for the objectives of implementing the lowest-order FOAR system outlined in this paper. GPGPUs (General Purpose Graphics Processing Units) present a cost-effective option for data processing, employing parallel techniques [29]. Later, the randomness of the RNG based on the FOAR system is tested using various methods.

#### 3.1. A Brief Overview of Fractional Calculus

The algorithm for the RNG implemented on the Nvidia Jetson Nano development board is provided in Algorithm 1.

---

#### Algorithm 1: Pseudo Code of Random Number Generation

---

```

Input : Parameters and initial condition of chaotic system
Output : Tested random number
1: START
2: Entering system parameters, initial condition and fraction order of chaotic system
while minimum 1MBit data do
  | Select "s = 8" bit LSB;
  | Solving the chaotic system ;
  | Obtaining time series x,y and z as float numbers;
  | Convert to 32-bit binary number with IEEE-754 standard;
  | Select s bit from RNG (selectdatax and selectdatay = 8 bit);
  | for i=0;8 do
  | | randombits(i) = selectdatax (i) XOR selectdatay (i);
  | end
end
3: Apply NIST-800-22 and Ent Tests for each minimum 1MBit randombits
if test results == pass then
  | Ready tested random number for RNG
else
  | Test results == false;
end
EXIT

```

---

In Algorithm 1, the design steps can be detailed as follows:

- Step 1. The parameters, initial conditions, and fractional degrees of the fractional chaotic system in Equation (1) are entered into the system.
- Step 2. The equations of the chaotic system are numerically solved at each stage using the Grünwald–Letnikov method.
- Step 3. The decimal values obtained from each phase are converted to 32-bit binary format using the IEEE 754 standard [30], and the least significant bit with the most random distribution (LSB = 8) is selected.
- Step 4. The values obtained from the  $x$  and  $y$  phases are subjected to XOR operation.
- Step 5. The resulting bits are added to the random bit sequence until it reaches 1 million bits.
- Step 6. The generated random numbers are tested with NIST 800-22 and ENT tests. Thus, randomness analysis is performed.
- Step 7. If the test results are successful, the algorithm terminates. Otherwise, the values in Step 1 are changed, and these steps are repeated until randomness is achieved.

Oscilloscope images of the signal of random numbers obtained from the Nvidia Jetson Nano development board are given in Figure 4.



**Figure 4.** Oscilloscope images obtained using the Nvidia Jetson Nano development board to demonstrate the applicability of random numbers generated through the  $x \oplus y$  operation.

### 3.2. Randomness Tests

To perform a randomness analysis on the random numbers obtained from the Nvidia Jetson Nano development board, two distinct testing processes, known as NIST 800-22 and ENT, are performed.

The ENT test suite, developed by John Walker, is employed to assess the randomness of bit sequences. It includes five distinct statistical tests. The test results for the random numbers obtained from the  $x \oplus y$ ,  $x \oplus z$ , and  $y \oplus z$  outputs are given in Table 1.

The NIST 800-22 test suite is used to evaluate various properties related to randomness in both random and quasi-random number generators. This internationally recognized test suite, published by the National Institute of Standards and Technology (NIST), includes 15 distinct statistical tests and requires a bit sequence of 1,000,000. For the resulting bit sequence to be considered successful, it must pass all tests in the NIST 800-22 suite. The test results for the  $x \oplus y$ ,  $x \oplus z$ , and  $y \oplus z$  outputs are provided in Table 2.

**Table 1.** ENT test results of  $x \oplus y$ ,  $x \oplus z$ , and  $y \oplus z$  outputs.

Test Name	Ideal Result	Average					
		$x \oplus y$	Result	$x \oplus z$	Result	$y \oplus z$	Result
Arithmetic mean	127.5	127.5947	Succeed	127.2477	Succeed	127.4634	Succeed
Entropy	8	7.9984	Succeed	7.9986	Succeed	7.9985	Succeed
Correlation	0	0.0048	Succeed	0.0040	Succeed	−0.0009	Succeed
Chi-square	Between 10% and 90%	276.799	Succeed	250.6154	Succeed	255.7619	Succeed
Monte-Carlo	Pi number	3.1611 (error = 0.0062)	Succeed	3.1418 (error = $6.6456 \times 10^{-5}$ )	Succeed	3.1473 (error = 0.0018)	Succeed

**Table 2.** NIST 800-22 test results of  $x \oplus y$ ,  $x \oplus z$ , and  $y \oplus z$  outputs.

Statistical Tests	$x \oplus y$		$x \oplus z$		$y \oplus z$	
	<i>p</i> -Value	Result	<i>p</i> -Value	Result	<i>p</i> -Value	Result
Frequency (Monobit) Test	0.9553	Succeed	0.0551	Succeed	0.4727	Succeed
Block Frequency Test	0.7984	Succeed	0.0911	Succeed	0.5876	Succeed
Cumulative Sums Test	0.7964	Succeed	0.1011	Succeed	0.5834	Succeed
Runs Test	0.7233	Succeed	0.2510	Succeed	0.3388	Succeed
Longest Run Test	0.3501	Succeed	0.0190	Succeed	0.7649	Succeed
Binary Matrix Rank Test	0.5174	Succeed	0.7089	Succeed	0.7041	Succeed
Discrete Fourier Transform Test	0.2997	Succeed	0.3083	Succeed	0.9560	Succeed
Non-Overlapping Templates Test	0.2762	Succeed	0.1094	Succeed	0.9172	Succeed
Overlapping Templates Test	0.1893	Succeed	0.6285	Succeed	0.7661	Succeed
Maurer’s Universal Statistical Test	0.8212	Succeed	0.4451	Succeed	0.6651	Succeed
Approximate Entropy Test	0.6681	Succeed	0.7978	Succeed	0.4047	Succeed
Random Excursions Test ( $x = -4$ )	0.8103	Succeed	0.7079	Succeed	0.6200	Succeed
Random Excursions Variant Test ( $x = -9$ )	0.4201	Succeed	0.0332	Succeed	0.2085	Succeed
Serial Test-1	0.4323	Succeed	0.6296	Succeed	0.8394	Succeed
Serial Test-2	0.6573	Succeed	0.7507	Succeed	0.7266	Succeed
Linear Complexity Test	0.0455	Succeed	0.8592	Succeed	0.8312	Succeed

Based on the information in Tables 1 and 2, it is evident that the resulting bitstream has successfully passed all the tests and can be securely employed in areas requiring the encryption of medical images.

#### 4. Biomedical Image Encryption and Security Analysis

In this section,  $256 \times 256$  brain MRI images are encrypted using random numbers obtained from the  $x \oplus y$  operation with Nvidia Jetson Nano development board. The dataset “Uncovering Knowledge: A Clean Brain Tumor Dataset for Advanced Medical Research.”, available on <https://www.kaggle.com/datasets/thomasdubail/brain-tumors-256x256> (accessed on 15 March 2024), was used [31]. The Python programming language was used for this task. Then, a security analysis is performed to measure the quality of the encryption algorithm and the security of the encrypted image.

In encryption processes, both scrambling and propagation are fundamental requirements. Though the initial conditions and control parameters in chaotic systems ensure precision, the complexity of these systems ensures randomness, i.e., scrambling. This ensures that the encrypted data have an unpredictable and irregular pattern. In this way, the encrypted data can be securely and reliably protected. The proposed encryption method incurs low computational cost due to its simple XOR operation, making it highly suitable for implementation on the Nvidia Jetson Nano development board. The algorithm used for encryption is presented as pseudo code in Algorithm 2.

**Algorithm 2:** Pseudo code of Biomedical Image Encryption Algorithm

---

```

Input : Tested random bit sequence and image
Output : Encrypted image
1: START
2: Random bit sequence generated from x-y phase (random bits) and entering image data
      (image)
3: Get the dimensions of the image (w=256,h=256)
4: Convert image to GrayScale
5: resize the encryptedimage to w*h
6: t ← 0
for i=0; w * h do
  | decimalrandomnumbers(i)=bintodecimal(randombits (t → t + 8))
  | t = t+8
end
7: Sort decimal array (decimalrandomnumbers) and get indexes (decidxrandomseq)
      decidxrandomseq=argsort(decimalrandomnumbers)
8: for i=0; w * h do
  | confusionimg(i) = image(decidxrandomseq(i))
end
9: for i=0; w * h do
  | encryptedimgpixels(i) = confusionimg(i) XOR decimalrandomnumbers(i)
end
10: Resize encrypted image to size w x h
      encryptedimage = encryptedimgpixels.reshape((w,h))
EXIT

```

---

According to Algorithm 2, randomly generated bits and the image are initially input into the system. Subsequently, the received image is converted to grayscale and resized to uniform dimensions. The random bits are then converted into decimal numbers and sorted in ascending order. The image is scrambled using the indices obtained from this sorting process. Finally, the scrambled image is XORed with the decimal numbers, resulting in the encrypted image.

The algorithm used for decryption is presented as pseudo code in Algorithm 3.

**Algorithm 3:** Pseudo code of Biomedical Image Decryption Algorithm

---

```

Input : Tested random bit sequence and encrypted image
Output : Decrypted image
1: START
2: Entering random bit sequence (randombits) and encrypted image data (encryptedimage)
3: Get the dimensions of the image (w=256,h=256)
4: resize the encryptedimage to w*h
5: t ← 0
for i=0; w * h do
  | decimalrandomnumbers(i)=bintodecimal(randombitseq (t → t + 8))
  | t = t+8
end
6: Sort decimal array (decimalrandomnumbers) and get indexes (decidxrandomseq)
      idxdecrandomseq=argsort(decimalrandomseq)
7: for i=0; w * h do
  | confimage(i) = encryptedimage(i) XOR decimalrandomnumbers(i)
end
8: for i=0; w * h do
  | decryptimagepixels(decidxrandomseq(i)) = confimage(i)
end
9: Resize encrypted image to size w x h
      decryptedimage = decryptimagepixels.reshape((w,h))
EXIT

```

---

According to Algorithm 3, a randomly generated bit sequence (randombits) and the encrypted image (encryptedimage) are input to the system. After obtaining the dimensions



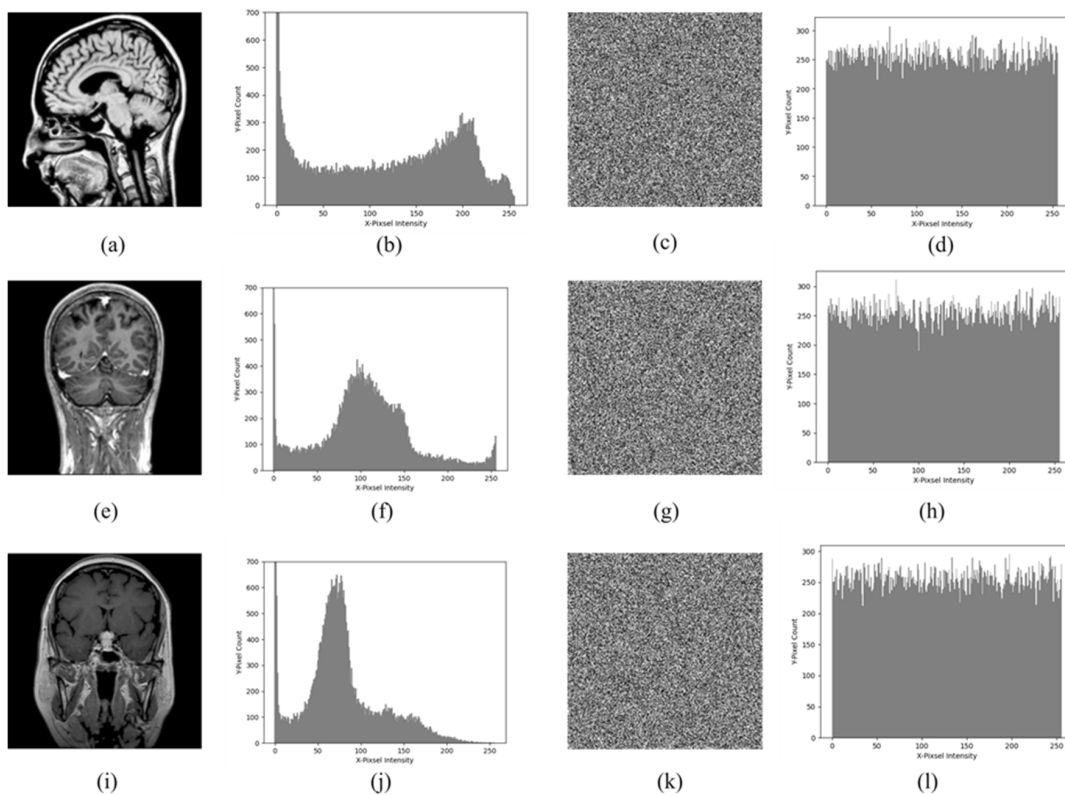
of the image ( $256 \times 256$ ), the encrypted image is converted to uniform size. The random bit sequence is converted into decimal random numbers (decimalrandomnumbers) to obtain the scrambling indices, and this sequence is sorted in ascending order. The encrypted image is then XORed with the transformed decimal numbers to obtain the scrambled image pixels. Using the scrambled image and the indices obtained from the random numbers, the image is decrypted and resized.

#### 4.1. Security Analyses

To ensure the confidentiality and security of image data, it is crucial to perform a security analysis after image encryption. These analyses comprehensively evaluate the security and robustness of the encryption algorithm and measure its resistance to potential attacks and cryptanalysis methods. An effective encryption method should demonstrate the ability to resist various cryptanalytic, statistical, and security attacks. In this section, we conduct a series of analyses including histogram, correlation, differential attack, information entropy, key space, and time analyses, along with occlusion and noise attacks.

#### 4.2. Histogram Analysis

Histogram analysis reveals the distribution of pixel values of an image. This distribution reflects the characteristics of the image. The histogram of an encrypted image should be completely different from the histogram of the original image. Source images may have different pixel values and colors, so the histogram may show different peaks and a wide distribution. However, since a secure encryption process is achieved by randomizing or secretly scrambling the image data, the histograms of encrypted images should have a more regular and uniform structure [32]. The histogram plots obtained in the study are given in Figure 5.



**Figure 5.** Histogram analysis. (a) Source image Img1. (b) Histogram of (a). (c) Encrypted image (a). (d) Histogram of (c). (e) Source image Img2. (f) Histogram of (e). (g) Encrypted image (e). (h) Histogram of (g). (i) Source image Img3. (j) Histogram of (i). (k) Encrypted image (i). (l) Histogram of (k).

Figure 5 shows that the histogram distributions of the encrypted images are in a uniform homogeneous distribution. In order to prove that the histogram of the encrypted image is uniform, an additional chi-square ( $X^2$ ) test was performed. The chi-square value is calculated by the formula given in Equation (4) [33].

$$X^2 = \sum_{i=1}^{256} \frac{(OV - EV)^2}{EV}, \tag{4}$$

For significance level  $\alpha = 0.05$  and degrees of freedom  $d = 255$ , the  $X^2_{(\alpha,d)}$  value is 293. The  $X^2$  values of the encrypted images are given in Table 3. All of the values obtained are less than 293, indicating that the histogram of the encrypted images is uniform.

**Table 3.** Chi-square analysis.

Image	Chi-Square Value
Img1 (256 × 256)	271.7539
Img2 (256 × 256)	285.9960
Img3 (256 × 256)	273.1328

#### 4.3. Correlation Analysis

Correlation coefficient analysis is used to examine the similarity between the pixel values of an original and an encrypted image. In the source image, neighboring pixel values are strongly correlated in the horizontal, diagonal, and vertical directions. However, good image encryption aims to minimize these correlations in the encrypted image. In this way, the information in the encrypted image can be stored securely, making it difficult for unwanted persons to access the original data. Mathematically, the correlation coefficient between two adjacent pixels is calculated by the formula given in Equation (5) [34].

$$r_{x,y} = \frac{C(x,y)}{\sqrt{D(x)} \cdot \sqrt{D(y)}}, \tag{5}$$

Here,

$$C(x,y) = \frac{\sum_{i=1}^K (x_i - E(x))(y_i - E(y))}{K}, \tag{6}$$

$$D(x) = \frac{1}{K} \sum_{i=1}^K (x_i - E(x))^2, \tag{7}$$

$$D(y) = \frac{1}{K} \sum_{i=1}^K (y_i - E(y))^2, \tag{8}$$

where  $x, y$  are the coordinates of the image,  $C(x, y)$  is the covariance between  $x, y$  samples,  $K$  is the number of pixel pairs  $(x_i, y_i)$ , and  $D(x)$  and  $D(y)$  are the standard deviation of  $x$  and  $y$ .

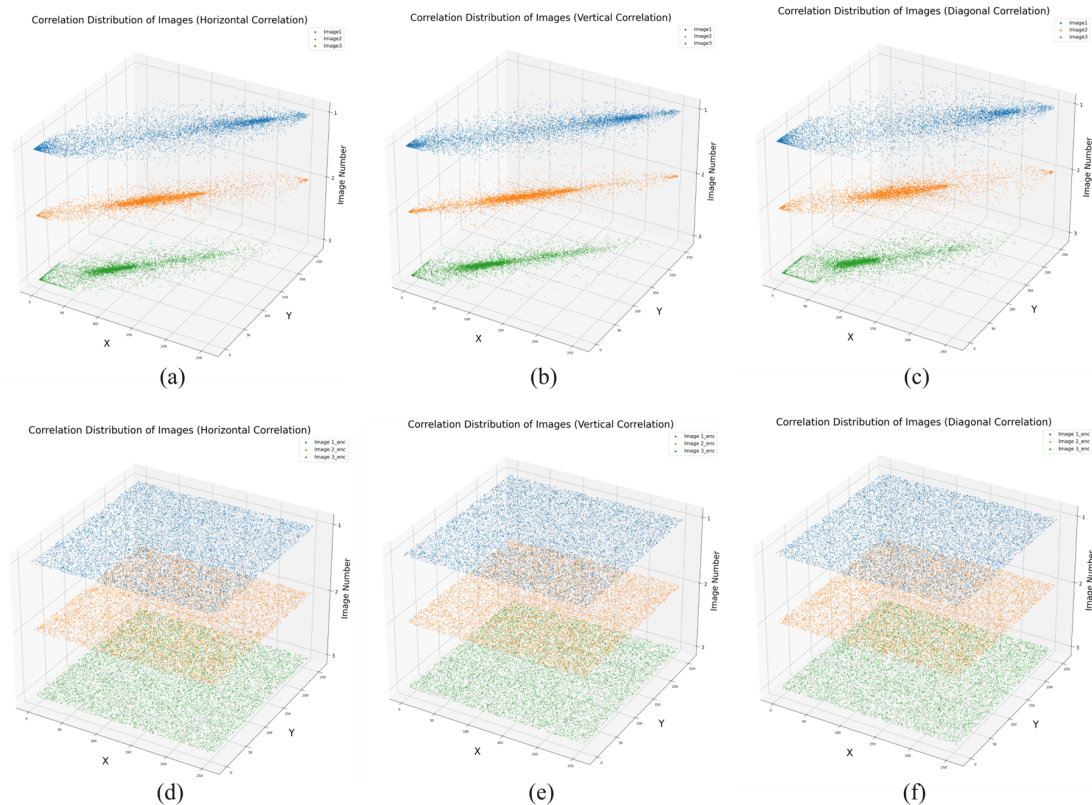
The correlation coefficient value is in the range of  $[-1,1]$ , and this value is expected to be close to 0 in encrypted images [35]. Table 4 lists the correlation coefficient values obtained in similar studies in the literature and in this study.

Table 4 shows that the correlation coefficient values of the encrypted images are close to zero. Thus, it is shown that the proposed encryption method is quite successful. In addition, Figure 6 shows the correlation maps of the biomedical images.

In the correlation maps, it is seen that the distribution in the source images is diagonal, and the distribution in the encrypted images is homogeneous. This proves that the encryption is successful.

**Table 4.** Correlation coefficient values.

Test Image	Direction	Source Image	Encrypted Image
Img1 (256 × 256)	V	0.9644	0.0135
	H	0.9709	0.0237
	D	0.9380	0.0232
Img2 (256 × 256)	V	0.9774	0.0006
	H	0.9807	0.0023
	D	0.9637	−0.0170
Img3 (256 × 256)	V	0.9578	0.0029
	H	0.9651	−0.0048
	D	0.9315	0.0068
Kamal et al. [35] (256 × 256)	V	0.9841	0.0063
	H	0.9543	−0.0044
	D	0.9414	0.0081
Chang et al. [36] (256 × 256)	V	0.9391	0.0050
	H	0.9172	0.0018
	D	0.8708	−0.0028
Sha et al. [37] (256 × 256)	V	0.9344	0.0006
	H	0.9588	0.0100
	D	0.9588	0.0064



**Figure 6.** The examined images’ correlations are as follows: (a) source images/horizontal, (b) source images/vertical, (c) source images/diagonal, (d) encrypted images/horizontal, (e) encrypted images/vertical, and (f) encrypted images/diagonal (the analyzed images are indexed as 1—Img1, 2—Img2, 3—Img3).

**4.4. Differential Attack Analysis**

*NPCR* (Number Of Changing Pixel Rate) and *UACI* (Unified Averaged Changed Intensity) parameters are commonly used in the literature to analyze differential attacks.

These parameters are used to evaluate the sensitivity of encryption algorithms to the smallest changes in the plain image. For an acceptable encryption algorithm, a small change in the plain image should result in a large difference in the encrypted image. The *NPCR* and *UACI* are calculated by the formula given in Equations (9) and (11).

$$NPCR = \frac{\sum_{i=1}^M \sum_{j=1}^N W(i, j)}{MN} \times 100, \tag{9}$$

Here,

$$W(i, j) = \begin{cases} 0, & C_1(i, j) = C_2(i, j) \\ 1, & C_1(i, j) \neq C_2(i, j) \end{cases} \tag{10}$$

$$UACI = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \left[ \frac{|C_1(i, j) - C_2(i, j)|}{(L - 1)} \right] \times 100, \tag{11}$$

In Equation (9), *MN* represents the dimensions of the image. In Equation (10), *C*<sub>1</sub> and *C*<sub>2</sub> indicate two different images. If *C*<sub>1</sub>(*i, j*) is not equal to *C*<sub>2</sub>(*i, j*), then *W*(*i, j*) = 1; otherwise, *W*(*i, j*) = 0 [38].

*L* is the grayscale level of the image. The *NPCR* value is expected to be close to 99.61% [39], and the *UACI* value is expected to be close to 33.46% [40]. Table 5 shows the obtained *NPCR* and *UACI* values, and the obtained values are compared with the studies in the literature. It is seen that the *NPCR* and *UACI* values in the encrypted images are very close to the expected values.

**Table 5.** NPCR and UACI comparisons.

Test Image	NPCR	UACI
Img1 (256 × 256)	99.6124	33.3149
Img2 (256 × 256)	99.6063	33.6300
Img3 (256 × 256)	99.6032	33.4785
Kamal et al. [35] (256 × 256)	99.6002	33.4535
Chang et al. [36] (256 × 256)	99.5834	33.4535
Sha et al. [37] (256 × 256)	99.6067	33.4675

#### 4.5. Information Entropy Analysis

Information entropy is used to measure the average information per bit in an image. Information entropy is calculated by the formula given in Equation (12).

$$H(s) = \sum_{i=1}^w P(s_i) \log_2 \frac{1}{P(s_i)}, \tag{12}$$

where *P*(*s*<sub>*i*</sub>) denotes the probability of *s*<sub>*i*</sub> in the image sequence. The entropy value is in the range [0, 8], and the closer this value is to 8, the higher the encryption performance in encrypted images. The entropy values obtained are given in Table 6.

**Table 6.** Entropy comparisons.

Test Image	Entropy	
	Source Image	Encrypted Image
Img1 (256 × 256)	6.1286	7.9969
Img2 (256 × 256)	5.1538	7.9968
Img3 (256 × 256)	5.2883	7.9969

Table 6 shows that the entropy values are very close to 8, indicating that the random distribution in the encryption is statistically strong.

#### 4.6. Key Space Analysis

The key space of a strong image encryption algorithm should be large enough to make brute force attacks impossible. Fractional degree chaotic systems have a larger key space than integer degree chaotic systems, since fractional degrees can be used as parameters [41]. The key length must be greater than  $2^{100}$  to make the encryption process resistant to brute force attacks [42]. According to the IEEE floating point standard, the computational accuracy of a double-precision 64-bit number is approximately  $10^{15}$  [30]. In this paper, the key space of the proposed encryption algorithm consists of the parameters  $a, b, c, q, x_1(0), x_2(0),$  and  $x_3(0)$ . The entire key space of the proposed encryption algorithm is  $2^{349}$ , which is greater than  $2^{100}$ . Table 7 shows the comparison results of the proposed method with the studies in the literature.

**Table 7.** Key space comparisons.

Method	Key Space
Proposed Method	$2^{349}$
Kocak et al. [43]	$2^{200}$
Chai et al. [44]	$2^{232}$
Talhaoui et al. [45]	$2^{224}$

Table 7 shows that the key space of the proposed encryption algorithm is large enough to provide strong resistance to attacks.

#### 4.7. Occlusion Attacks

Occlusion occurs when the encrypted data are partially or completely lost. In order to evaluate how effective the encryption method used is against occlusion, the encrypted images are cut at  $\frac{1}{4}$  (corner),  $\frac{1}{4}$  (middle), and  $\frac{1}{2}$  and reconstructed. Figure 7 shows the test results for data loss attacks.

Distortion is usually evaluated using the peak signal-to-noise ratio (PSNR), defined in Equation (13), which measures the ratio between the signal's highest potential power and the distortion that occurs. The higher the PSNR, the better the image quality.

$$PSNR = 10 \log \left( \frac{255^2}{MSE} \right), \quad (13)$$

Here, MSE is the mean square error and is calculated as follows:

$$MSE = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (e_{ij} - f_{ij})^2 \quad (14)$$

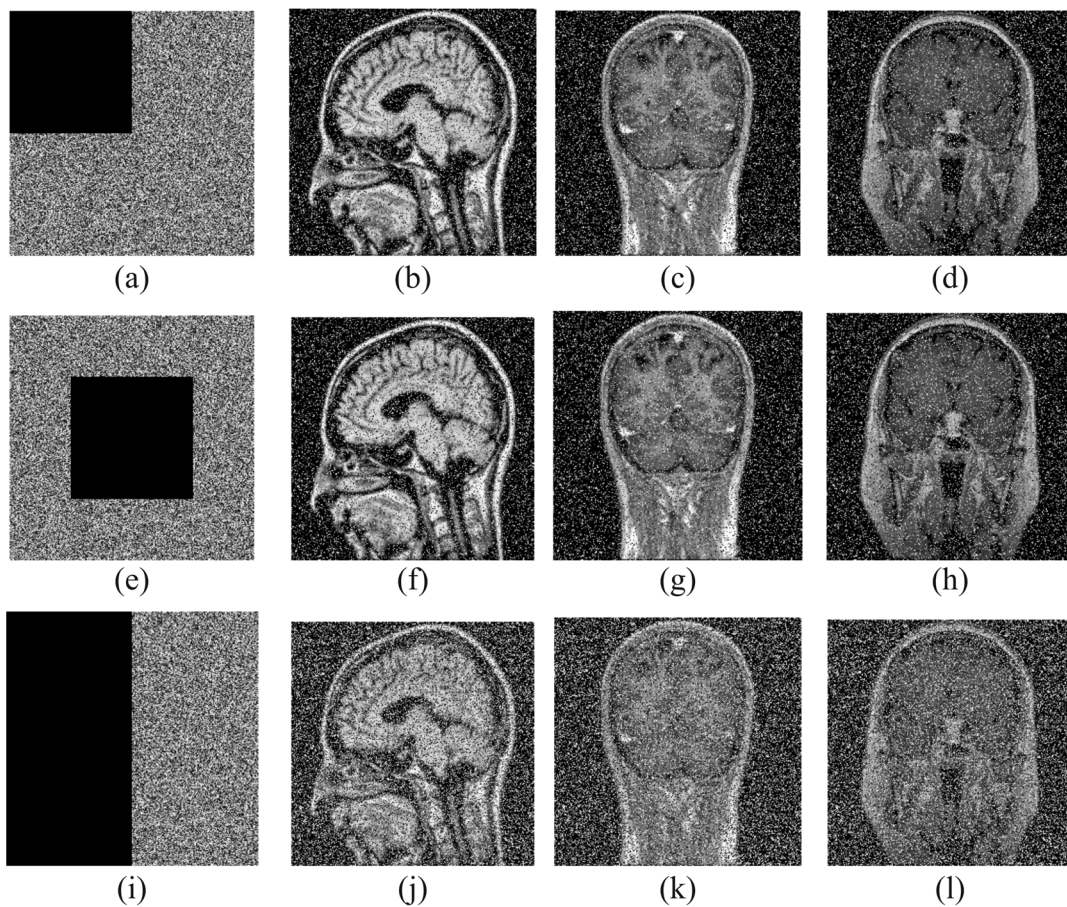
where  $e_{ij}$  is the source image, and  $f_{ij}$  is the decrypted image. Table 8 shows the PSNR values obtained as a result of occlusion.

Although there are some distortions on the reacquired images, the main information of the images can still be recognized. Therefore, the proposed method is proved to be robust against data loss attacks.

**Table 8.** PSNR value of images obtained as a result of occlusion.

	Img1	Img2	Img3
$\frac{1}{4}$ (corner) occlusion	12.5146	12.5514	12.5896
$\frac{1}{4}$ (middle) occlusion	12.4963	12.6142	12.5935
$\frac{1}{2}$ occlusion	9.4886	9.5517	9.5774





**Figure 7.** Decryption results with  $\frac{1}{4}$ (corner),  $\frac{1}{4}$ (middle), and  $\frac{1}{2}$  occlusion ratios, respectively: (a)  $\frac{1}{4}$ (corner) occlusion and its decryption images (b–d); (e)  $\frac{1}{4}$ (middle) occlusion and its decryption images (f–h); (i)  $\frac{1}{2}$  occlusion and its decryption images (j–l).

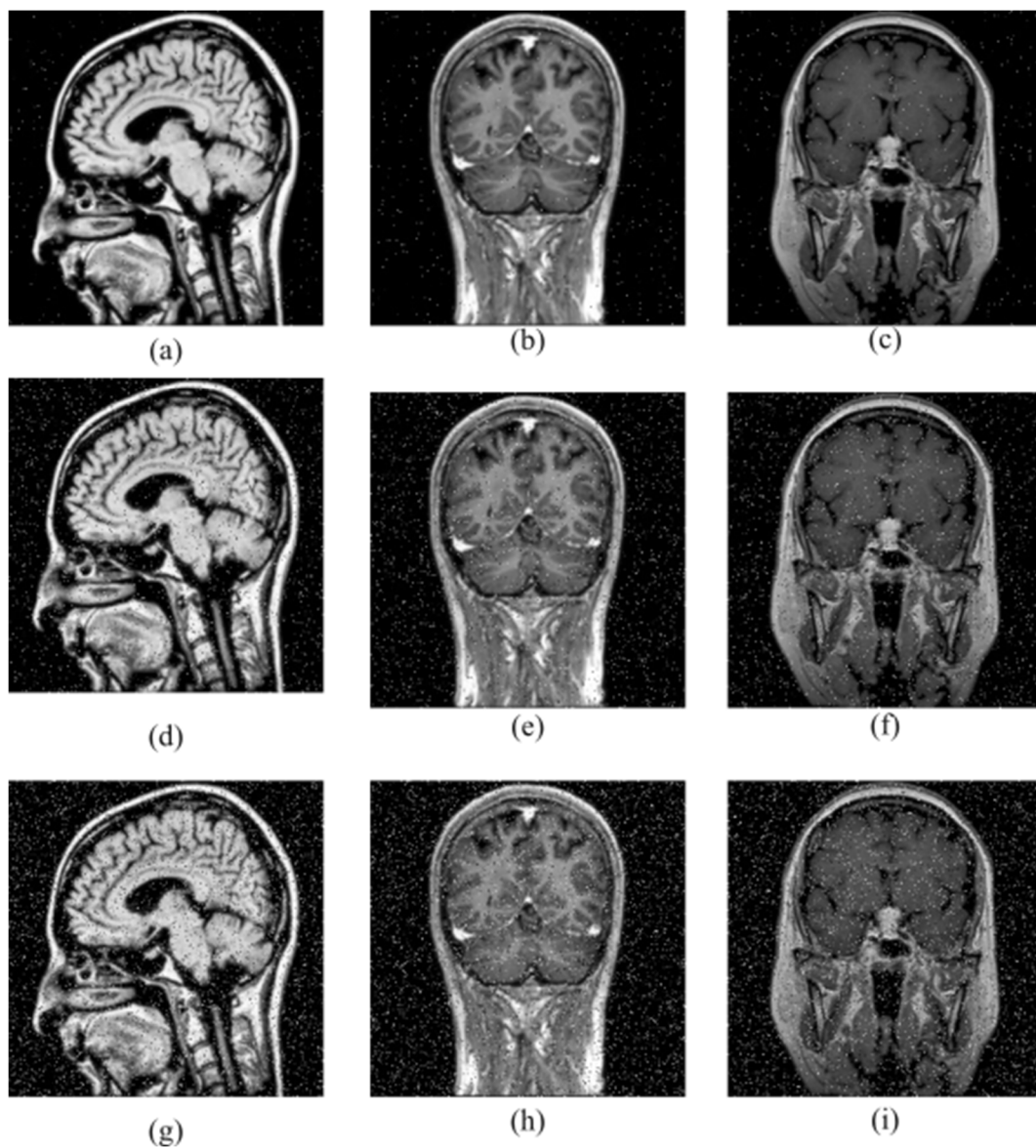
#### 4.8. Noise Attacks

Various forms of noise in the communication channel often affect encrypted images. A good encryption system should be able to successfully recover the source image from the noisy encrypted image. In our work, salt-and-pepper noises of different intensities are added to the encrypted image to measure the robustness against noise. Decryption was then performed. The results are shown in Figure 8. Table 9 shows the PSNR values obtained as a result of salt-and-pepper noise attacks.

**Table 9.** PSNR value of images obtained as a result of salt-and-pepper noises.

Salt-and-Pepper Noises	Img1	Img2	Img3
0.01 intensity	26.3505	26.5974	26.5329
0.05 intensity	19.4308	19.4979	19.6187
0.1 intensity	16.3862	16.5001	16.5861

According to the obtained results, it is clear that the source image can be successfully restored even if the encrypted images are subjected to noise. Therefore, it is seen that the proposed method is robust against noise attacks.



**Figure 8.** Decryption images under salt-and-pepper noises of different intensities: (a–c) intensity is 0.01, (d–f) intensity is 0.05, (g–i) intensity is 0.1.

#### 4.9. Time Analysis

The biomedical image encryption applications and analyses performed in this study were performed using the Python 3.10 programming language on a Nvidia Jetson Nano development board with 4-core ARM A57 CPU 1.43 GHz and 4 GB 64 Bit LPDDR4 memory on Linux-based Ubuntu operating system. Table 7 shows the encryption times in seconds for biomedical images. In Table 10, the times obtained in similar studies in the literature are also listed, and it is concluded that the proposed method is fast.

**Table 10.** Encryption duration comparisons.

		Encryption Duration (Unit: s)
Proposed (256 × 256)	Img1	0.5320
	Img2	0.5329
	Img3	0.5262
Xuejing and Zihui [46]		2.2703
Ding and Ding [47]		0.54

## 5. Conclusions

In this study, time series obtained from the FOAR system, utilizing the minimum applicable FO parameter  $q$  value, are employed for the encryption of biomedical images, ensuring patient privacy protection. Suitable operating conditions of the system are determined through bifurcation maps, spectral entropy diagrams, and Lyapunov spectra. A RNG is designed on the Nvidia Jetson Nano development board using the FOAR chaotic system with  $q = 0.55$ . The generated random numbers successfully passed internationally recognized NIST 800-22 and ENT tests, confirming their randomness. Subsequently, an encryption process is performed using the generated random numbers. To validate the success of the encryption process, histogram, correlation, differential attack, entropy, and time analyses are performed. As a result of these analyses, the utilization of random numbers obtained from the FOAR system, which exhibits increased unpredictability through fractional order analysis, enhances both the security and effectiveness of the encryption process. On the other hand, though the history-dependent nature of FO systems enhances robustness in encryption algorithms, it also complicates their implementation on development boards with limited memory such as the Nvidia Jetson Nano. Nevertheless, the encryption of biomedical images on the Nvidia Jetson Nano development board, which is cost-effective due to its mobility, is successfully achieved. Incorporating incommensurate fractional-order analysis may be the focus of future work to enhance the complexity of the encryption algorithm while also addressing memory limitations.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** Data is contained within the article.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest.

## References

- Gupta, R.; Agrawal, R.K. A Comprehensive Survey on Image Security using Encryption Techniques. In Proceedings of the 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 26–28 May 2023; pp. 739–743.
- Kaur, M.; Singh, D.; Kumar, V.; Gupta, B.B.; El-Latif, A.A.A. Secure and energy efficient-based E-health care framework for green internet of things. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 1223–1231. [[CrossRef](#)]
- Jabeen, T.; Ashraf, H.; Ullah, A. A survey on healthcare data security in wireless body area networks. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 9841–9854. [[CrossRef](#)] [[PubMed](#)]
- Vaseghi, B.; Mobayen, S.; Hashemi, S.S.; Fekih, A. Fast reaching finite time synchronization approach for chaotic systems with application in medical image encryption. *IEEE Access* **2021**, *9*, 25911–25925. [[CrossRef](#)]
- Abundiz-Pérez, F.; Cruz-Hernández, C.; Murillo-Escobar, M.A.; López-Gutiérrez, R.M.; Arellano-Delgado, A. A Fingerprint Image Encryption Scheme Based on Hyperchaotic Rössler Map. *Math. Probl. Eng.* **2016**, *2016*, 2670494. [[CrossRef](#)]
- Han, F.; Hu, J.; Yu, X.; Wang, Y. Fingerprint images encryption via multi-scroll chaotic attractors. *Appl. Math. Comput.* **2007**, *185*, 931–939. [[CrossRef](#)]
- Hikal, N.A.; Eid, M.M. A new approach for palmprint image encryption based on hybrid chaotic maps. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 870–882. [[CrossRef](#)]
- Boyraz, O.F.; Guleryuz, E.; Akgul, A.; Yildiz, M.Z.; Kiran, H.E.; Ahmad, J. A novel security and authentication method for infrared medical image with discrete time chaotic systems. *Optik* **2022**, *267*, 169717. [[CrossRef](#)]
- Yang, Y.-G.; Guan, B.-W.; Zhou, Y.-H.; Shi, W.-M. Double image compression-encryption algorithm based on fractional order hyper chaotic system and DNA approach. *Multimed. Tools Appl.* **2021**, *80*, 691–710. [[CrossRef](#)]
- Chen, L.; Yin, H.; Huang, T.; Yuan, L.; Zheng, S.; Yin, L. Chaos in fractional-order discrete neural networks with application to image encryption. *Neural Netw.* **2020**, *125*, 174–184. [[CrossRef](#)]
- Liu, Z.; Xia, T.; Wang, T. Dynamic analysis of fractional-order six-order discrete chaotic mapping and its application in information security. *Optik* **2023**, *272*, 170356. [[CrossRef](#)]



12. Xu, S.; Wang, X.; Ye, X. A new fractional-order chaos system of Hopfield neural network and its application in image encryption. *Chaos Solitons Fractals* **2022**, *157*, 111889. [[CrossRef](#)]
13. Kiran, H.E.; Akgul, A.; Yildiz, O.; Deniz, E. Lightweight encryption mechanism with discrete-time chaotic maps for Internet of Robotic Things. *Integration* **2023**, *93*, 102047. [[CrossRef](#)]
14. Guillén-Fernández, O.; Tlelo-Cuautle, E.; de la Fraga, L.G.; Sandoval-Ibarra, Y.; Nuñez-Perez, J.C. An image encryption scheme synchronizing optimized chaotic systems implemented on raspberry pis. *Mathematics* **2022**, *10*, 1907. [[CrossRef](#)]
15. Li, X.; Mou, J.; Banerjee, S.; Wang, Z.; Cao, Y. Design and DSP implementation of a fractional-order detuned laser hyperchaotic circuit with applications in image encryption. *Chaos Solitons Fractals* **2022**, *159*, 112133. [[CrossRef](#)]
16. Ávalos-Ruiz, L.F.; Zúñiga-Aguilar, C.J.; Gómez-Aguilar, J.F.; Cortes-Campos, H.M.; Lavín-Delgado, J.E. A RGB image encryption technique using chaotic maps of fractional variable-order based on DNA encoding. *Chaos Solitons Fractals* **2023**, *177*, 114306. [[CrossRef](#)]
17. Daoui, A.; Yamni, M.; Karmouni, H.; Sayyouri, M.; Qjidaa, H.; Ahmad, M.; Abd El-Latif, A.A. Biomedical Multimedia Encryption by Fractional-Order Meixner Polynomials Map and Quaternion Fractional-Order Meixner Moments. *IEEE Access* **2022**, *10*, 102599–102617. [[CrossRef](#)]
18. Gokyildirim, A.; Kocamaz, U.E.; Uyaroglu, Y.; Calgan, H. A novel five-term 3D chaotic system with cubic nonlinearity and its microcontroller-based secure communication implementation. *AEU-Int. J. Electron. Commun.* **2023**, *160*, 154497. [[CrossRef](#)]
19. Gokyildirim, A. Dynamical Analysis and Electronic Circuit Implementation of Fractional-order Chen System. *Chaos Theory Appl.* **2023**, *5*, 127–132. [[CrossRef](#)]
20. Gokyildirim, A.; Akgul, A.; Calgan, H.; Demirtas, M. Parametric fractional-order analysis of Arneodo chaotic system and microcontroller-based secure communication implementation. *AEU-Int. J. Electron. Commun.* **2024**, *175*, 155080. [[CrossRef](#)]
21. Gokyildirim, A. Circuit Realization of the Fractional-Order Sprott K Chaotic System with Standard Components. *Fractal Fract.* **2023**, *7*, 470. [[CrossRef](#)]
22. Rajagopal, K.; Akgul, A.; Pham, V.T.; Alsaadi, F.E.; Nazarimehr, F.; Alsaadi, F.E.; Jafari, S. Multistability and coexisting attractors in a new circulant chaotic system. *Int. J. Bifurc. Chaos* **2019**, *29*, 1950174. [[CrossRef](#)]
23. Clemente-López, D.; Muñoz-Pacheco, J.M.; Rangel-Magdaleno, J.D.J. A Review of the Digital Implementation of Continuous-Time Fractional-Order Chaotic Systems Using FPGAs and Embedded Hardware. *Arch. Comput. Methods Eng.* **2023**, *30*, 951–983. [[CrossRef](#)]
24. Gokyildirim, A.; Calgan, H.; Demirtas, M. Fractional-Order sliding mode control of a 4D memristive chaotic system. *J. Vib. Control* **2024**, *30*, 1604–1620. [[CrossRef](#)]
25. Garrappa, R. On linear stability of predictor–corrector algorithms for fractional differential equations. *Int. J. Comput. Math.* **2010**, *87*, 2281–2290. [[CrossRef](#)]
26. Diethelm, K.; Ford, N.J.; Freed, A.D. Detailed error analysis for a fractional Adams method. *Numer. Algorithms* **2004**, *36*, 31–52. [[CrossRef](#)]
27. Danca, M.-F.; Kuznetsov, N. Matlab Code for Lyapunov Exponents of Fractional-Order Systems. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850067. [[CrossRef](#)]
28. Li, H.; Shen, Y.; Han, Y.; Dong, J.; Li, J. Determining Lyapunov exponents of fractional-order systems: A general method based on memory principle. *Chaos Solitons Fractals* **2023**, *168*, 113167. [[CrossRef](#)]
29. Routis, G.; Michailidis, M.; Roussaki, I. Plant Disease Identification Using Machine Learning Algorithms on Single-Board Computers in IoT Environments. *Electronics* **2024**, *13*, 1010. [[CrossRef](#)]
30. *IEEE Std 754-2019*; IEEE Standard for Floating-Point Arithmetic. (Revision of IEEE 754-2008). IEEE: Piscataway, NJ, USA, 2019.
31. CC0: Public Domain, A Clean Brain Tumor Dataset for Advanced Medical Research. Available online: <https://www.kaggle.com/datasets/thomasdubail/brain-tumors-256x256> (accessed on 15 March 2024).
32. Thangavel, S.; Venkatesan, R. A Novel Image Encryption Using Calligraphy Based Scan Method and Random Number. *KSII Trans. Internet Inf. Syst.* **2015**, *9*, 2317–2337. [[CrossRef](#)]
33. Ravichandran, D.; Praveenkumar, P.; Rayappan, J.B.B.; Amirtharajan, R. Chaos based crossover and mutation for securing DICOM image. *Comput. Biol. Med.* **2016**, *72*, 170–184. [[CrossRef](#)]
34. Chai, X.; Gan, Z.; Yang, K.; Chen, Y.; Liu, X. An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations. *Signal Process. Image Commun.* **2017**, *52*, 6–19. [[CrossRef](#)]
35. Kamal, S.T.; Hosny, K.M.; Elgindy, T.M.; Darwish, M.M.; Fouda, M.M. A New Image Encryption Algorithm for Grey and Color Medical Images. *IEEE Access* **2021**, *9*, 37855–37865. [[CrossRef](#)]
36. Chang, H.; Wang, E.; Liu, J. Research on Image Encryption Based on Fractional Seed Chaos Generator and Fractal Theory. *Fractal Fract.* **2023**, *7*, 221. [[CrossRef](#)]
37. Sha, Y.; Mou, J.U.N.; Wang, J.U.E.; Banerjee, S.; Sun, B.O. Chaotic image encryption with hopfield neural network. *Fractals* **2023**, 2340107. [[CrossRef](#)]
38. Wang, X.; Chen, S. An image encryption algorithm based on pixel bit operation and nonlinear chaotic system. *Vis. Comput.* **2023**, *39*, 3123–3144. [[CrossRef](#)]
39. Belazi, A.; El-Latif, A.A.A.; Belghith, S. A novel image encryption scheme based on substitution-permutation network and chaos. *Signal Process.* **2016**, *128*, 155–170. [[CrossRef](#)]

40. Budiman, F.; Andono, P.N.; Setiadi, D.R.I.M. Image Encryption using Double Layer Chaos with Dynamic Iteration and Rotation Pattern. *Int. J. Intell. Eng. Syst.* **2022**, *15*, 57–67. [[CrossRef](#)]
41. He, G.; Luo, M. Dynamic behavior of fractional order Duffing chaotic system and its synchronization via singly active control. *Appl. Math. Mech.* **2012**, *33*, 567–582. [[CrossRef](#)]
42. Chai, X.; Fu, X.; Gan, Z.; Lu, Y.; Chen, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **2019**, *155*, 44–62. [[CrossRef](#)]
43. Kocak, O.; Erkan, U.; Toktas, A.; Gao, S. PSO-based image encryption scheme using modular integrated logistic exponential map. *Expert Syst. Appl.* **2024**, *237*, 121452. [[CrossRef](#)]
44. Chai, X.; Yang, K.; Gan, Z. A new chaos-based image encryption algorithm with dynamic key selection mechanisms. *Multimed. Tools Appl.* **2017**, *76*, 9907–9927. [[CrossRef](#)]
45. Talhaoui, M.Z.; Wang, X. A new fractional one dimensional chaotic map and its application in high-speed image encryption. *Inf. Sci.* **2021**, *550*, 13–26. [[CrossRef](#)]
46. Xuejing, K.; Zihui, G. A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **2020**, *80*, 115670. [[CrossRef](#)]
47. Ding, L.; Ding, Q. A Novel Image Encryption Scheme Based on 2D Fractional Chaotic Map, DWT and 4D Hyper-chaos. *Electronics* **2020**, *9*, 1280. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.