International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SECURE MEDICAL IMAGE ENCRYPTION IN IOT USING RNN-BASED KEY GENERATION WITH AES

Rana Saeed Hamdi [1], Saif Al-alak [2], Elaf Ali Abbood [3]

[1] College of Medicine-University of Babylon

[2,3] College of Science for Women-University of Babylon

*Corresponding author's E-mail: scw564.rana.saeed@student.uobabylon.edu.iq

Abstract:

Secure image transmission is a crucial requirement in Internet of Things (IoT) applications, particularly in the medical field, where privacy and confidentiality are of paramount importance. This paper proposed an image encryption scheme that integrated Recurrent Neural Networks (RNNs) for generating multiple cryptographic keys with the Advanced Encryption Standard (AES) to provide strong image protection. The use of RNN-based key generation enhanced the unpredictability and diversity of the keys, thereby reducing the risk of brute-force and statistical attacks. Experimental results were conducted on medical images to evaluate the scheme's performance using several security metrics. The encrypted images achieved near-ideal entropy values, approaching 8, indicating excellent randomness. PSNR values below 10 dB and high MSE confirmed strong visual distortion in the encrypted images, while the decrypted images exhibited lossless reconstruction with infinite PSNR and zero MSE. Histogram analysis showed uniform pixel intensity distribution, and correlation coefficients were close to zero, demonstrating resilience against statistical and differential attacks. the proposed RNN with AES scheme as a secure and efficient solution for IoT image encryption.

Keywords: Cryptography, RNN, AES, IoT, medical images, security, encryption.

Introduction:

The Internet of Things is composed of a wide range of devices and various link-layer technologies [1]. IOT applications are growing in number in recent times. Smart homes, cities, agriculture, utilities, healthcare monitoring, animal husbandry, water, industrial control, management, security and emergencies, transportation, and environment monitoring are among the applications [2]. It addresses extremely secret information regarding individuals and corporations, which must remain undisclosed to unauthorized individuals or adversaries.

The security of IoT technology presents significant challenges, and it is an important issue that may affect the future advancement of IoT. Additionally, Confidentiality, availability, authenticity, authorization, and access control are security requirements in an IoT environment [3].

A process called cryptography turns readable data into unintelligible (encrypted) data that only the owner of the decryption key can decode. Since the data is frequently confidential or

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

extremely important, the main goal of encryption is to protect data privacy by preventing unwanted access or alteration [4].

Symmetric (where the key for encryption and decryption is identical) and asymmetric (where the keys for encryption and decryption are different) are fundamental categories in cryptography, while lightweight is a modern category aimed at low-end devices. Numerous cryptographic algorithms and security techniques form the basis for many lightweight security solutions in use. These include lightweight public-key cryptography systems built on ECC (Elliptic Curve Cryptography), hash functions such as access control, xor encryption [5], the symmetric AES 128 algorithm, WiMAX (Worldwide Network for Microwave Access), and Bluetooth [6] [7].

To improve IoT security, machine learning techniques are becoming more popular as they make encryption and key generation procedures simpler, requiring less time and complexity. The system performs cryptographic key generation to enhance image communication security by generating stronger and more random keys.

The work combines strong randomness with multiple AES keys to improve image security through the Internet of Things. Measures like correlation, MSE, PSNR, and entropy show that the suggested RNN-AES can achieve higher levels of security and integrity.

Related Work

2.1 Random Key Generation Using Machine Learning

Park et al. aimed to improve the quality of random number generators. In 2022, they developed a hybrid PRNG model that consisted of both convolutional neural networks and reinforcement learning algorithms. That hybrid PRNG model produced a more random sequence, with the highest correlation coefficient [5].

Okada et al. (2022) developed a Learned Pseudo-Random Number Generator (LPRNG) using Wasserstein GANs and gradient penalty, achieving statistically significant pseudo-randomness and passing the NIST test suite [6].

In 2022, Erkana et al. proposed a chaotic image-encryption pipeline that combines deep learning with chaos theory. The approach uses a VGG16 CNN to extract image features, which are flattened and passed through dense layers to generate a dynamic public key. A logistic chaotic map is then applied for permutation and diffusion, ensuring that each encryption run produces a uniqe and secure key [7].

2.2 Image Encryption Using Machine Learning

Ding et al. (2021) proposed DeepKeyGen: A GAN-based framework for Medical Image Encryption, which produced very random private keys and ciphertext, performed strong robustness to attacks, and exhibited very strong resilience to statistical correlations.[8].

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Alslman et al. (2022) developed a hybrid scheme utilizing autoencoders and AES for encrypting medical images, enhancing efficiency and image quality by reducing data size [9].

Panwar et al. (2023) introduced Encipher GAN, an end-to-end image encryption and decryption technique based on Cycle-GAN for the secure transmission of medical images. The approach provides a strong defense against statistical attacks and maintains structural integrity after recovery [10].

Theoretical Background

1. Cryptography

Cryptography is an approach that converts readable information into unreadable (encrypted) information, which can only be decrypted by an individual in possession of the decryption key. A primary objective of encryption is to ensure data privacy, preventing unauthorized access or modifications, as the information is frequently secret or of significant importance [3].

Only those who possess the intended secret key can decipher the information messages' content. The keys used in the cryptosystem are generated from complicated mathematical methods (algorithms). Data encryption algorithms for working in the networks can be divided according to the classification of encryption keys into two primary categories: asymmetric key algorithms and symmetric key algorithms.

The asymmetric key algorithm, sometimes called the public key algorithm, uses a pair of mathematically related keys (a public and a private key). The sender uses the public key to encrypt messages, while the receiver uses the private key to decrypt. The symmetric key algorithm, often called secret key cryptography, employs the same key for both encryption and decryption [11].

2. AES Algorithm

AES is the most widely used block symmetric key algorithm in encryption. When the AES method encrypts data, hackers find it very hard to access the original data. AES supports three key sizes: 128, 192, and 256 bits, with a fixed message block size of 128 bits. AES is based on a principle called substitution-permutation. It involves several rounds depending on the key length. For example, AES with a 128-bit key has 10 rounds, while 192-bit and 256-bit keys have 12 and 14 rounds, respectively. Each round includes four operations: adding a round key, byte substitution, row shifting, and column mixing [12].

AES provides several standard modes of operation that specify the use of block encryption on data larger than 128 bits. Various modes differ in performance, security characteristics, and suitability for different applications, making the chosen mode an important consideration in achieving the required balance between efficiency and security [13].

- **ECB (Electronic Code Book):** The simplest and fastest mode. Each block is independently encrypted.
- **CBC (Cipher Block Chaining):** Every block is XORed with the ciphertext of the previous block before encryption, except for the initial block, which uses an initialization vector (IV).

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **CFB (Cipher Feedback):** converts a block cipher into a stream cipher by XORing plaintext with previously encrypted feedback from a shift register. Padding is unnecessary; however, the blocks rely on the preceding ciphertext, which limits parallelization.
- **OFB (Output Feedback):** Similar to CFB, however, feedback is obtained from the output of the encryption instead of the ciphertext. Errors do not propagate, and it additionally facilitates stream-like encryption with uniform randomness.
- **CTR (Counter Mode):** Uses a counter and nonce to generate a keystream that is XORed with plaintext.

3. Random Key Generation

3.1 RNN (The Recurrent Neural Network)

Recurrent neural networks are a type of artificial neural network where connections are made sequentially between nodes. Nodes in a variant layer can only communicate with one other node in one direction. The data flow in recurrent neural networks is unidirectional, so the network architecture includes three layers (input, hidden, and output). Due to the sequential transfer of data from one node to another, outputs from previous nodes can be used as inputs for subsequent nodes.

Consequently, the model can dynamically construct the past by assembling and transmitting data from previous inputs to subsequent nodes[14] [15].

Classical neural networks perform well when input and output are completely irrelevant to one another; however, that is not always the case [16].

Figure 1 illustrates a diagram for a recurrent neural network. All prior data is compiled and stored in the network's hidden section, which is calculated in equation (1).

A new input xt is introduced to the network at each time step t. [17].

- $h_t = f(W_x x_t + W_h h_t - 1)$ (1)

In conclusion, equation (2) generates the output yt at each node sequentially.

- $y_t = g(W_y)$ (2)

Where the selected activated function is represented by g(Wy).

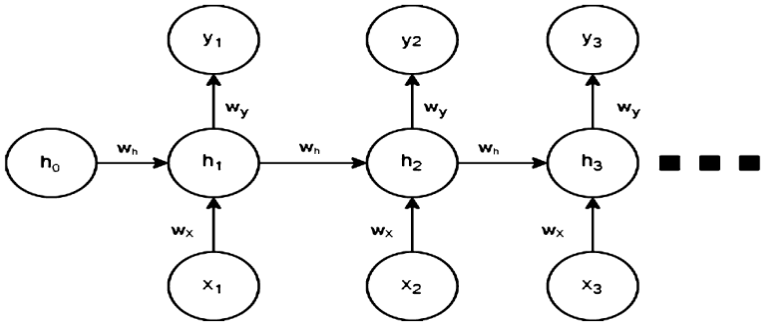


Figure 1: A typical Recurrent Neural Network [17].

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Recurrent neural networks, also known as RNNs, are made up of three layers of neurons that are connected via synapses. Each neuron's input, activation, and output functions can be used to define it. Weight is a value assigned to each synapse between two neurons [18]. The final result is calculated by doing the computation layer by layer.

At first, the input values are fed into the input neurons. In the input layer, the output function is computed by every neuron. The other layer receives the computed values. Arriving at the output layer marks the end of the operation.

3.2 Training Process

In the RNN, a sequential RNN model was constructed using a multi-layer architecture. The model is defined by two Simple RNN Layers, each with 256 and 128 units and ReLU activation. During training, it randomly adjusts 30% of the neurons to zero through dropout regularization. This process prevents overfitting and returns sequences, ensuring that the output is transmitted to the next RNN layer. The dense output layer ensures that the output values are between a specific range by utilizing tanh activation and flattened into a 1D array, and stored as a 256-dimensional representation.

The model is trained iteratively with both real data (sampled from a uniform distribution in [-1, 1]) and fake data (generated by the RNN model). The training loop runs for 50,000 iterations. Progress is printed every 500 iterations, and the batch size is dynamically increased every 5,000 iterations (up to 128) to improve convergence and stability.

Basically, the proposed RNN generator creates random sequences from Gaussian noise input. The generated key, shown as a flattened array, is suitable for cryptographic and security applications. The structure of the proposed RNN model is shown in **Table 1.** It is developed by the following steps:

1. **Noise Generation:** A random noise vector z with latent_dim units is sampled from the Normal (Gaussian) distribution N (0,1). This acts as the input to the generator.

2. **First Recurrent Layer:** The input is passed into a SimpleRNN layer with 256 units and ReLU activation function, and returns sequences. Outputs the full sequence to the next layer.

3. **Dropout Regularization:** A Dropout layer with a rate = 0.3 is applied to prevent overfitting.

4. **Second Recurrent Layer:** consists of a second SimpleRNN with 128 units and ReLU activation. It further processes the output sequence from the previous layer. Unlike layers that return the full sequence, this layer is set with return_sequences=False, meaning it only outputs the final state of the sequence rather than the output at each time step. This setup is ideal for tasks that need a single representation of the entire sequence.

5. **Second Dropout Layer:** Another Dropout layer with a rate of 0.3 is added for regularization.

Received: August 05 2025

International Journal of Applied Mathematics

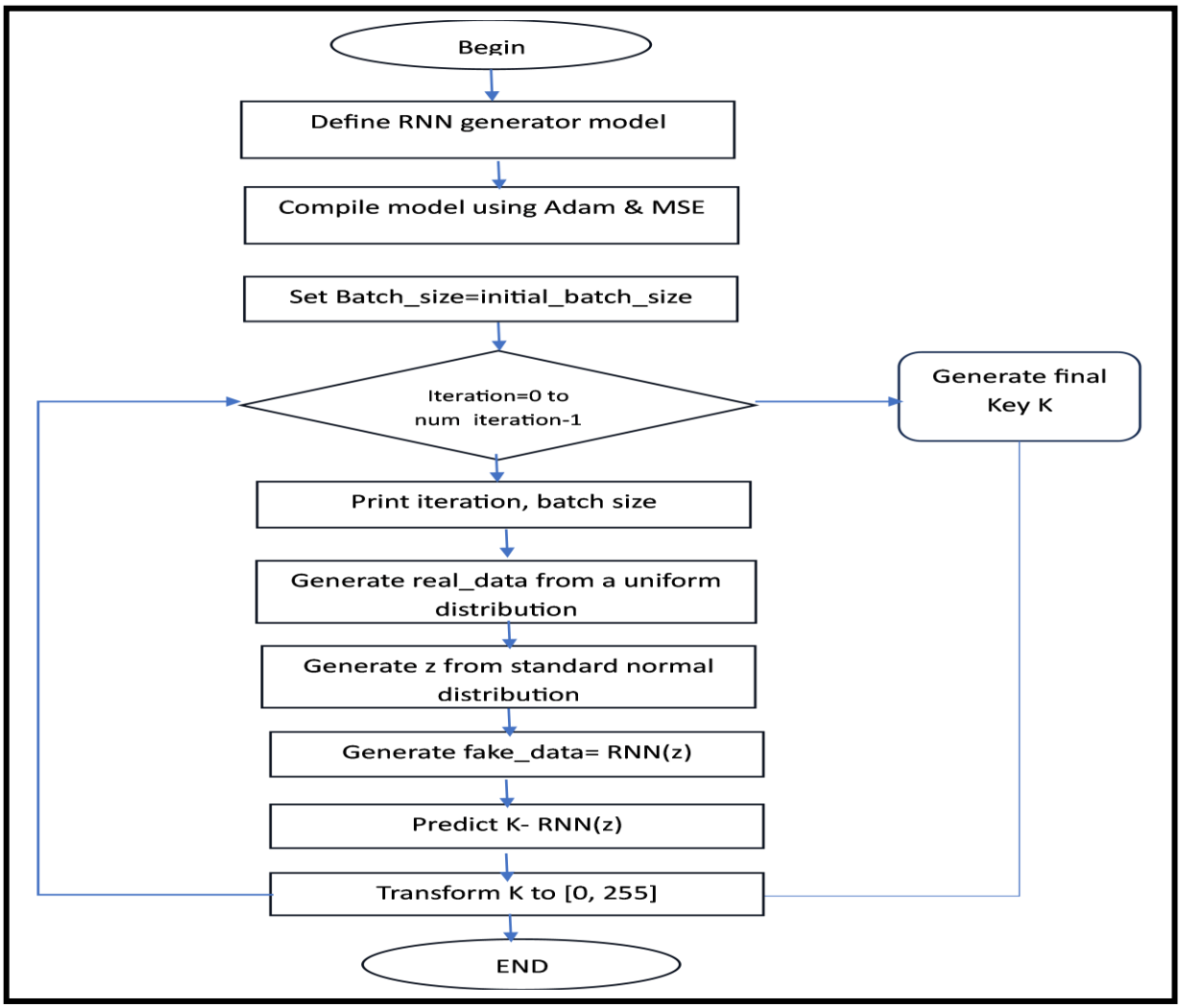
Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6. **Dense Output Layer:** A Dense layer with latent_dim units and tanh activation produces the final output within [-1, 1].

7. **Key Generation:** The output is rescaled to the range [0, 255]. Converted to unsigned integers. Flattened into a 1D key vector.

8. **Training Procedure:** The model runs for 50,000 iterations. Every 500 iterations, the progress is displayed. Every 5000 iterations, the batch size increases (up to 128) for better



performance. Real data (U(-1,1)) and fake data (RNN output) are used to refine the generator. The final result is a 256-dimensional key, which is useful in security applications. **Figure 2 demonstrates the architecture of an RNN.**

Figure 2: The Architecture of an RNN.

Table 1: The Architecture of the Proposed RNN Model.

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

LAYER-TYPE	OUTPUT SHAPE	PARAM #
SimpleRNN_2	**(none, 10, 256)**	**131, 328**
Dropout_6	**(none, 10, 256)**	**0**
SimpleRNN_3	**(none, 128)**	**49, 280**
Dropout_7	**(none, 128)**	**0**
Dense_7	**(none, 265)**	**33, 024**
The Total params: 213,632 (834.50 KB) **Trainable params:** 213,632 (834.50 KB) **Non-trainable params:** 0 (0.00 B)		

3.3 Randomness Validation of RNN-Generated Keys

Before performing the image encryption process, the RNN-generated keys were subjected to the Diehard statistical test suite to verify their randomness quality. The results confirmed that the generated sequences satisfied the randomness requirements, making them suitable for use in cryptographic key generation.

3.3.1 Diehard Tests:

The Diehard test is used to measure randomness. The results of diehard tests are p-values, which belong between 0 and 1. The Diehard tests are (birthday spacing, overlapping permutations, ranks of matrices, monkey tests, count the ones, parking lot test, minimum distance test, random spheres test, the squeeze test, overlapping sums test, runs test, the craps test). These p-values are divided into three kinds of areas, as follows: [19]

- **The Failure Areas:** $0 <$ p-value ≤ 0.1 or $0.9 \leq$ p-value ≤ 1.
- **The Doubt Areas:** $0.1 <$ p-value ≤ 0.25 or $0.75 \leq$ p-value < 0.9.
- **The Safe Areas:** $0.25 <$ p-value < 0.75.

4. The Proposed Work (RNN with AES):

The main goals of this research are to improve the security and accuracy of images transmitted through the Internet of Things. In other words, this work aims to leverage machine learning applications like the RNN shown below. The RNN algorithm is responsible for generating secret keys, which are then used with the Advanced Encryption Standard (AES) to further enhance encryption effectiveness and key generation.

Additionally, this approach decreases the total time needed to complete these tasks and reduces the complexity involved.

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Using the keys generated by the RNN method in combination with the RNN-AES approach appears to significantly enhance the current level of security. The RNN- AES method is proposed, and has two focuses: the RNN method, which generates the secret keys; and the AES method, which allows for encryption/decryption with the secret keys.

This work implements the Advanced Encryption Standard (AES) using the Cipher Block Chaining (CBC) mode. CBC enhances AES security by linking each cipher block to the previous one. Specifically, each plaintext block is XORed with the previous cipher block before encryption, and the first block is XORed with a randomly generated Initialization Vector (IV). This chaining process ensures that identical plaintext blocks produce different cipher blocks, preventing patterns that could be exploited in statistical attacks. CBC mode is especially suitable for image encryption because it greatly increases diffusion and offers resistance against differential and chosen-plaintext attacks.

At first, the image is separated into image color matrices (R, G, and B), and the image can be encrypted (using the RNN-AES method). However, to complete the encryption, each of these matrices, R, G, and B, must be split into an equal number of n blocks. If the total number of secret keys is fewer than the number of image blocks, the blocks are randomly grouped by the secret keys. The number of secret keys is equal to the number of recognized groups. All blocks in a group use the same secret key (I) for encryption. The decryption operation reverses the previous operation. Figure 3 is a summary of the work performed using A-encryption and B-decryption. Algorithms 1 and 2 are the encryption operation and the decryption operation of the proposed life form.

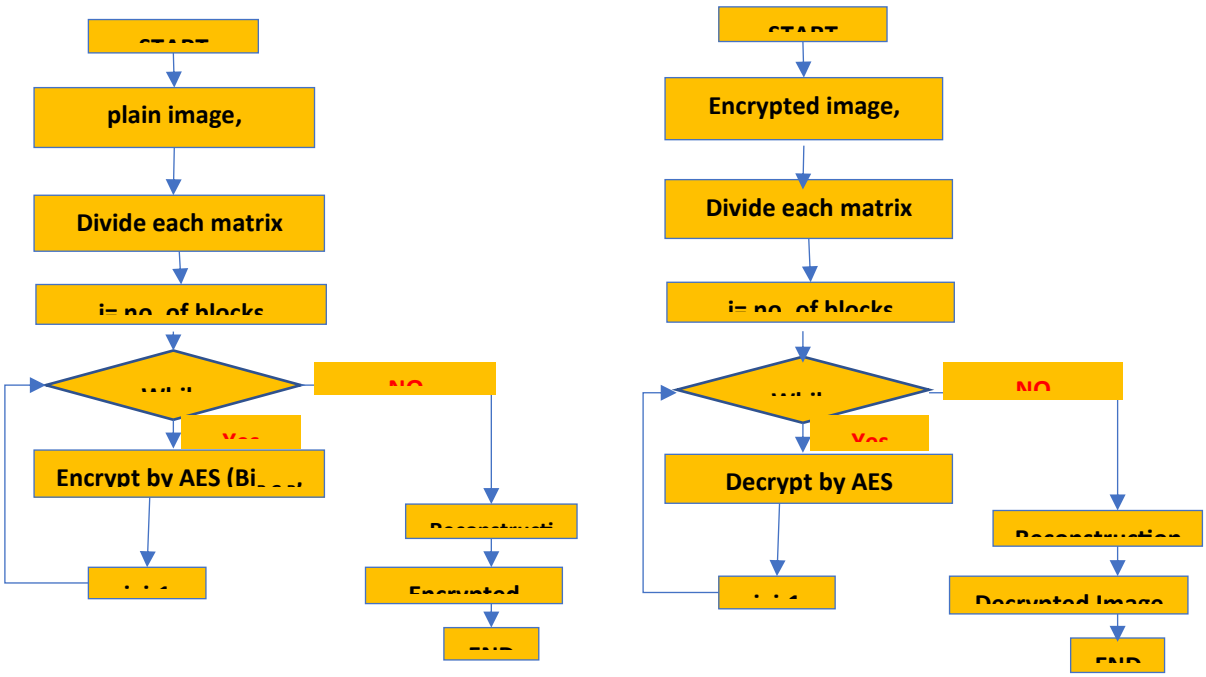


Figure 3: The flowchart of a Proposed Work (A) The Encryption, (B) The Decryption Operations.

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Algorithm 1: Encryption Process

Input:
- RGB image I
- secret key K (generated using RNN)
- number of encryptions rounds r

output:
- encrypted image Ci

```
1:  Begin
2:     load RGB image I
3:     convert I to array form
4:     separate I into channels: R, G, B
5:     for each channel C in {R, G, B} do
6:        convert channel C to 1D byte array B[]
7:        divide B[] into blocks of 16 bytes (AES block size)
8:        If the last block < 16 bytes, apply PKCS7 padding
9:        generate random 16-bytes Initialization Vector (IV)
10:       initialize AES cipher in CBC mode using key K and IV
11:       encrypt each block sequentially → C_enc
12:       concatenate (IV || C_enc) → Encrypted_C[]
13:    End For
14:    reconstruct encrypted R, G, B from Encrypted_C[]
15:    stack encrypted channels → C_i
16:    save C_i as PNG format
17: End
```

Algorithm 2: Decryption Process

- **Input:**
  -Ci Encrypted image
  -Secret key K (generated using RNN)
  -r Number of decryption rounds
- **Output:**
  I Decrypted RGB image

```
1:  Begin
2:     Load encrypted image Ci
3:     Convert Ci to array form
4:   Separate Ci into encrypted the channels: R_enc, G_enc, B_enc
5:     For each encrypted the channel C_enc in{R_enc, G_enc, B_enc} do
6:        Extract first 16 bytes as Initialization Vector (IV)
7:        Extract remaining bytes as cipher image
8:        Divide cipher image into blocks of 16 bytes
9:        Initialize  AES cipher in CBC mode using key K and IV
10:       Decrypt each block sequentially → B_padded[]
11:       Remove PKCS7 padding from B_padded[] → B[]
12:       Reshape B[] to original channel dimensions
13:    End For
14:    Combine decrypted R, G, B channels
15:    Merge channels to reconstruct RGB image I
16:    Save I as PNG format
17: End
```

Performance Measures:

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The metrics used to evaluate the encryption power of any image cryptography scheme are Entropy, PSNR & MSE, Histogram, and Correlation, which are demonstrated as follows:

5.1 The Entropy:

In communication theory, entropy (H) is a statistical measure of unpredictability that signifies unreliability [20]. Information theory (Entropy) encompasses a wide range of topics, including error correlation, communication systems, network security, encryption, and data compression. The calculation of entropy is performed using Equation (2) as outlined below:

$$H=-\Sigma p\,(xi)\log 2\, p(xi) \quad \text{(2)}$$

In this context, p of (xi) represents a probable density function for the appearance of symbol xi. An image's entropy shows the distribution of grey-level values. An ideal result would be an entropy of 8 when all image probabilities are equal. For a cryptosystem to be resistant to entropy attacks, the entropy must be smaller than eight (8). This predictability is achieved to a certain degree [21].

5.2 PSNR & MSE

PSNR, which stands for Peak Signal to Noise Ratio, shows how much the encrypted and unencrypted images differ in terms of pixel values [22]. If the PSNR result, which is a single decibel (dB) value, is less than 30 dB, it indicates that the encrypted image differs from the original.

The PSNR can be determined using the formula 3,4 by calculating the MSE (mean squared error) relative to the maximum luminance value, which for an 8-bit value is 28 - 1 = 255.

$$MSE = (1/m\,n)\,\Sigma_{i=0}^{m-1}\,\Sigma_{j=0}^{n-1}\,(f(i,j) - F(i,j))^2 \quad \text{(3)}$$

$$PSNR = 10\log_{10}(m^2 / MSE)\ dB \quad \text{(4)}$$

The image dimensions are represented by M × N, and f (i,j) is the original pixel at coordinates (i,j) or an encrypted pixel. As can be seen from equation 3, MSE provides a measurable way to assess how similar two signals are to one another [23].

5.3 Histogram

Within an image, the frequency of pixel intensity values is displayed via a histogram, which is an essential statistical aspect. In contrast to the histogram of the original image, an efficient encryption system generates a histogram that is consistent with the image that has been encrypted [24].

5.4 Correlation

A strong connection among pixels in basic image data is inherently guaranteed, nearing unity, along vertical, horizontal, or diagonal directions. The encryption procedure aims to reduce the significant connection between neighboring pixels of the encrypted image (approaching 0). The coefficient of correlation for each pixel pair is computed in equation (5) as follows:

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$D(x) = \frac{1}{N}\sum^{N}(x\,i - E(x))^2 \qquad \text{.....(5)}$$

Where,

$$\textbf{Cov (x,y)} = \frac{1}{N}\sum_{i=1}^{N}\left(xi - E\,(x)\right)\left(yi - E(y)\right) \qquad \text{..... (6)}$$

$$\textbf{E(x)} = \frac{1}{N}\sum_{i-1}^{N} x\,i$$

……..(7)

$$\textbf{r\{x,y\}} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

The terms cov (x:y), d(x), and d(y) represent the covariances and variances of the variables x and y, respectively. E(x) and E(y) denote the expected values of the variables x and y, respectively. Let N denote the total number of pixels in an image, defined as N = rows × cols. Let x be a vector of length N, where x_i represents the intensity value of the original image [25].

Results and Discussion:

The effectiveness of a method of encryption is evaluated utilizing a medical image, specifically a retinal image of the eye. Figure 4 illustrates that the input image is a retinal image with dimensions of 600 by 600 pixels, a file size of 23.1 KB. encrypted images resulting from implementing the proposed method applied to the original retinal image. Figure 5 shows the encrypted images, each with a varying number of keys that belong to the original RGB retinal image after applying the proposed scheme.

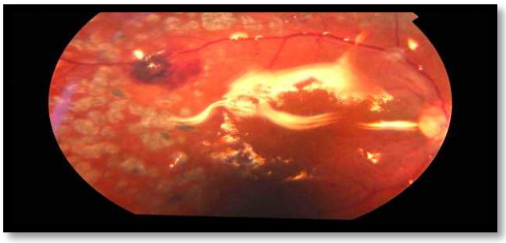


Figure 4: The Original of the Retinal Image.

Ciphered one key	**Ciphered ten keys**	**Ciphered twenty keys**
Ciphered thirty keys	**Ciphered forty keys**	**Ciphered fifty keys**

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

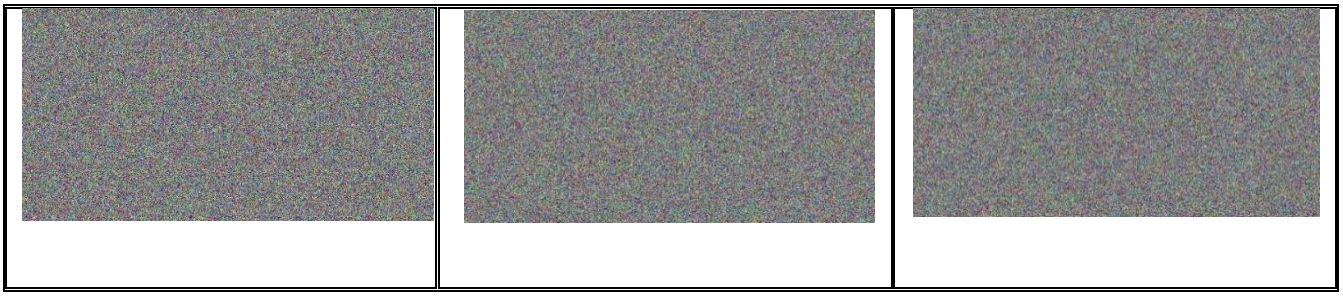


Figure 5: The Encrypted Images with Various Numbers of Keys for the retinal Image.

1. Randomness Test of Encrypted Image:

This test was performed on the six experiments for the retinal image ciphered by the RNN with the AES method by using the NIST tests, which are described in the following table:

Table 4.21: The Randomness of Encrypted Retinal Image.

Test Name	Frequency	Frequency Test within a Block	Longest Run of Ones in a Block	Binary Matrix Rank	Approximate Entropy	Cumulative Sums
ciphered 1 key	0.9707508	0.7511982	0.4911832	0.0541648	0.7372877	0.8760386
ciphered 10 keys	0.1388732	0.5700212	0.3843816	0.8551069	0.0576131	0.1088764
ciphered 20 keys	0.9867025	0.3177215	0.7514608	0.2785312	0.2402344	0.9632102
ciphered 30 keys	0.7514965	0.7247162	0.5978197	0.9566496	0.1203319	0.4307786
ciphered 40 keys	0.2542863	0.1450488	0.7295501	0.0605712	0.7321469	0.4174222
ciphered 50 keys	0.6647726	0.8657196	0.8273608	0.7566106	0.3749883	0.8509901

The NIST tests indicate that the encrypted retinal image produced with RNN-generated keys shows robust randomness overall. Most of the p-values fall within the range often considered an indication of acceptable randomness, and an increased number of keys (30 - 50) demonstrated more consistent and uniform behavior throughout the whole suite of tests. These

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

tests perform quite similarly in terms of statistical quality and will have very little effect on the overall statistical quality. The results indicate that keys generated by RNNs may successfully generate ciphered outputs exhibiting substantial randomness.

2. Entropy

As shown in Table 2, the entropy analysis for the retinal image demonstrates a clear and progressive enhancement in statistical randomness with encryption, further reinforced by increasing the number of keys.

For the retinal image, the original average entropy was approximately 6.12, indicating strong pixel correlation. Encryption with 1 key raised the entropy in all channels to ~7.9994, and as the number of keys increased to 10, 20, 30, 40, and 50, the values maintained a stable high range between 7.9994 and 7.9995, ensuring consistently optimal randomness at every stage.

This consistent behavior across different medical imaging modalities highlights that increasing the number of keys not only achieves but also sustains peak statistical randomness, providing enhanced assurance against entropy-based cryptanalysis.

Table 2: Entropy Evaluation of the Retinal Image.

The entropy	Channel-Red	Channel-Green	Channel-Blue	Average-RGB
Retinal-original	6.20486265	6.30141768	5.87343936	6.1265732
Ciphered One Key	7.99950921	7.99940640	7.99944395	7.9994531
Ciphered Ten Keys	7.99945592	7.99941198	7.99949629	7.9994547
Ciphered twenty Key	7.99947401	7.99945430	7.99950557	7.9994779
Ciphered Thirty Key	7.99949710	7.99951023	7.99951034	7.9995058
Ciphered Forty Key	7.99950777	7.99950994	7.99949821	7.9995053
Ciphered Fifty Key	7.99946881	7.99954550	7.99951341	7.999509

PSNR & MSE

As shown in Table 4.24, the evaluation of PSNR and MSE for the retinal image confirms the effectiveness and stability of the proposed encryption method across different numbers of keys. The PSNR between the original and decrypted images is infinite with zero MSE, indicating a perfect reconstruction and lossless decryption process.

For the retinal image, the PSNR values for the ciphered images ranged from approximately 6.71 dB to 6.72 dB, with MSE values around 105, reflecting a high degree of visual distortion

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

from the original, essential for secure encryption. Increasing the number of keys from 1 to 50 maintained this strong distortion level while ensuring consistent encryption quality.

Overall, the results show that increasing the number of keys ensures sustained encryption strength, uniform distortion, and complete reversibility, making the cipher highly reliable for diverse medical imaging modalities.

Table 3: PSNR and MSE Evaluation for Retinal Image.

Test Name	Average (MSE)	Average (PSNR)
Original and Decrypted Retinal	Zero	Inf - dB
one key	105.56502778	6.72131571
ten keys	105.54460185	6.71236869
twenty key	105.47763611	6.71294993
thirty key	105.31892407	6.71647716
forty key	105.47949630	6.72202319
fifty key	105.49635741	6.71524826

3. Histogram

Figure 9 shows the retinal image in its original form exhibits sharp peaks in the histograms of the(red, green, blue) channels, reflecting a structured nature and distinct patterns typical of unencrypted medical data.

After encryption using a single key, these histograms transform into nearly flat distributions across all channels, indicating that pixel values have reached an ideal state of randomness. As the number of keys increases (20, 30, 40, and 50 keys), this flat distribution remains consistent, demonstrating that key expansion preserves high randomness while adding an extra layer of security.

This stability across all channels and varying key counts highlights the proposed system's effectiveness in eliminating any exploitable statistical features, rendering the image completely unpredictable.

Test Name	RED	GREEN	BLUE	RGB

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

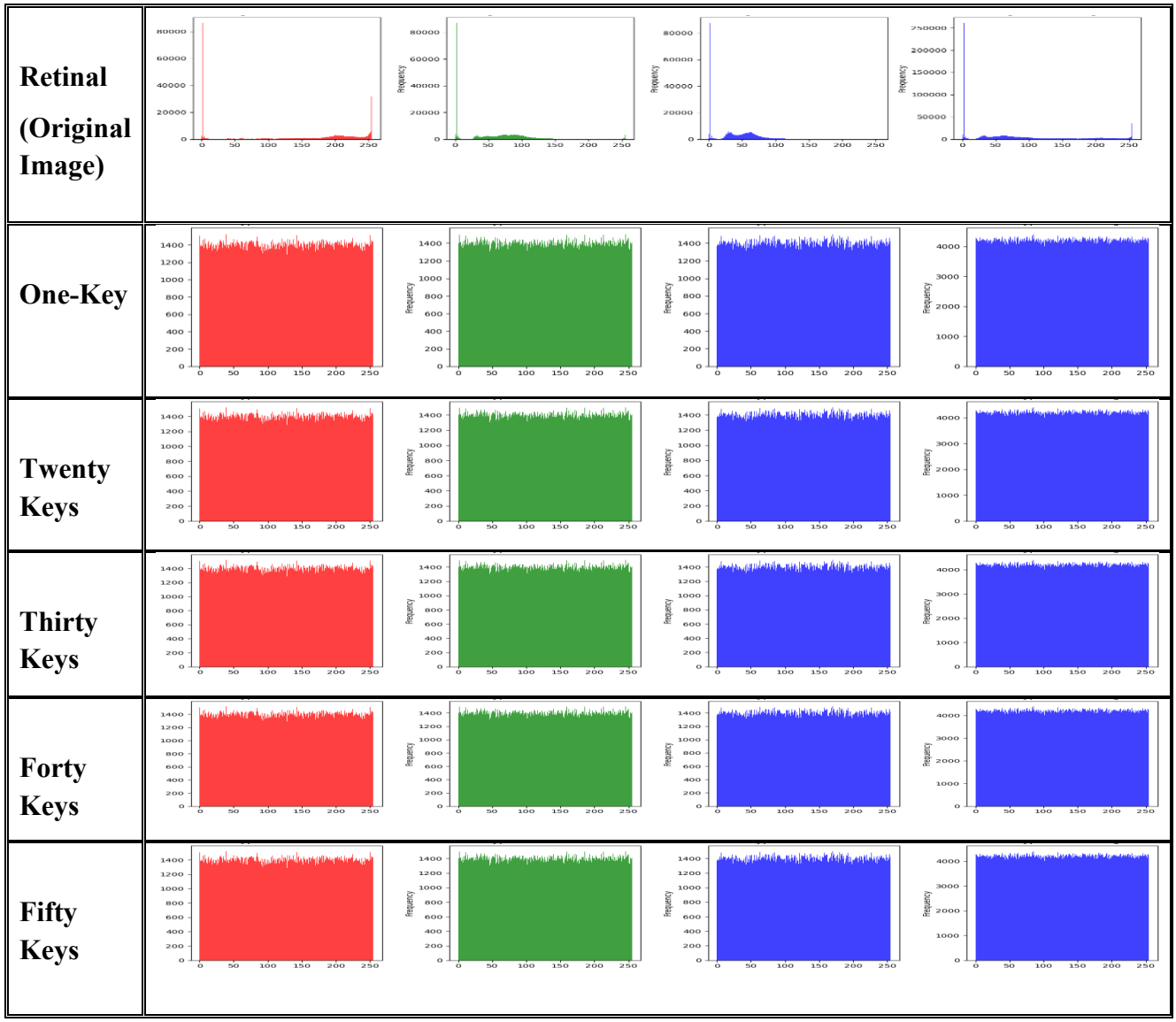


Figure 6: The Histogram of The Retinal Image.

4. Correlations

Tables 4, 5, and 6 show the correlation coefficient analysis of the retinal image in (red, green, blue) channels, measured in horizontal, vertical, and diagonal directions. In the original image, the correlation values are near 1.0 across all channels and orientations (e.g., 0.995 in the vertical direction of the red channel, 0.997 in the vertical direction of the green channel, and 0.999 in the vertical direction of the blue channel), indicating a high level of pixel dependency typical of natural images .

After encryption with a single key, these coefficients decrease sharply toward zero in all channels and directions, showing that the encryption operation effectively breaks the spatial relationships between neighboring pixels. As the number of keys increases from 10 to 50, the correlation values consistently stay close to zero, with minor fluctuations around small positive

Received: August 05 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

and negative values, confirming that the method keeps strong decorrelation regardless of the number of keys used.

Figures 7, 8, and 9 provide visual evidence to confirm these results. These figures show a significant drop in correlation for the encrypted images in all channels (Red, Green, and Blue) compared with the original image, illustrating the encryption algorithm's ability to disrupt statistical correlations effectively. This behavior indicates that the proposed encryption scheme successfully eliminates statistical dependencies, thereby enhancing resistance to statistical attacks.

Table 4: Evaluation of the correlation metric for the red channel thyroid inferno image across horizontal, vertical, and diagonal channels.

RED	HORIZONTAL	VERTICAL	DIAGONAL
The Original	0.9870999	0.9951248	0.9829458
1 Key	0.0004737	-0.000149	0.003756
10 Keys	0.0012174	-0.000494	-0.002227
20 Keys	-0.00147	-0.000177	0.0003872
30 Keys	-0.001902	-0.000727	-0.000698
40 Keys	0.0019545	-0.001065	0.0017619
50 Keys	0.0014117	-0.000897	-0.000894

Table 5: Evaluation of the correlation metric for the green channel thyroid inferno image across horizontal, vertical, and diagonal channels.

GREEN	HORIZONTAL	VERTICAL	DIAGONAL
The Original	0.9936465	0.9974435	0.9913166
1 Key	0.0026802	0.0001404	0.0007268
10 Keys	0.0004368	0.0016231	-3.84E-05
20 Keys	0.0008644	-0.000933	0.0007259
30 Keys	-0.001791	0.0005022	-0.000623
40 Keys	0.0017778	-0.003171	0.0006169
50 Keys	-0.003996	0.004014	0.0022046

Received: August 05 2025

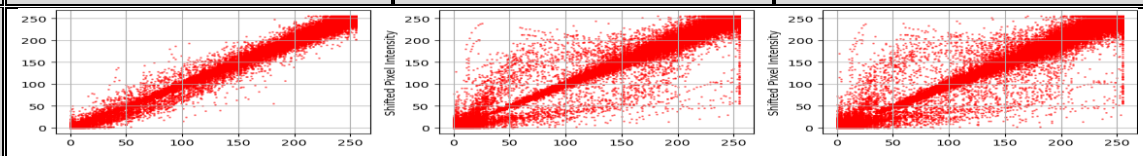
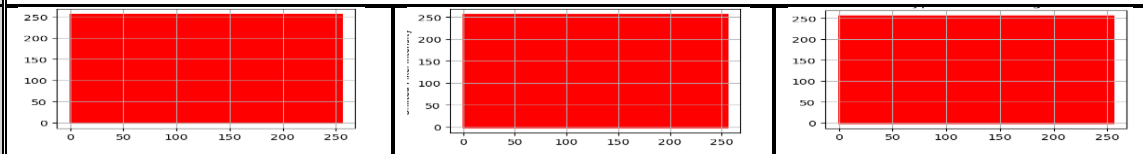
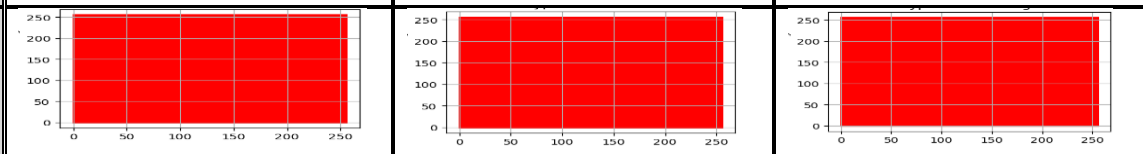
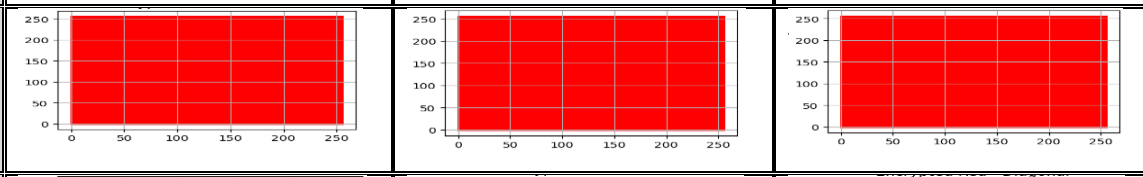
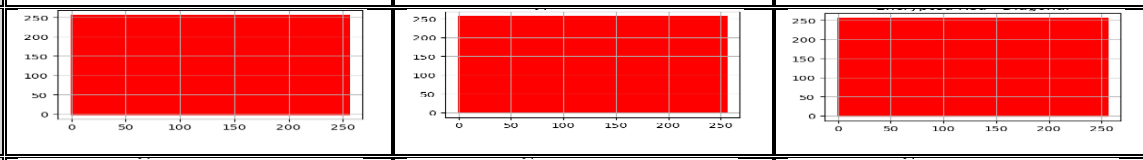
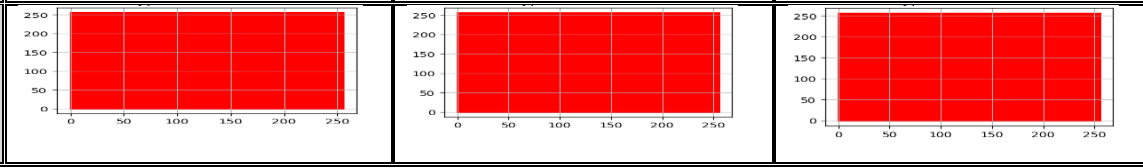
International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 6: Evaluation of the correlation metric for the blue channel thyroid inferno image across horizontal, vertical, and diagonal channels.

BLUE	HORIZONTAL	VERTICAL	DIAGONAL
The Original	0.997335	0.999376	0.996743
1 Key	-0.000257	-0.000461	-0.002418
10 Key	-0.001856	-0.000136	7.05E-05
20 Key	0.0004963	-0.002399	-0.003717
30 Key	-0.000583	-6.17E-05	-0.000232
40 Key	-0.001896	0.0022668	-0.000287
50 Key	0.000832	-0.001533	-0.000355

Test Name	HORIZONTAL	VERTICAL	DIAGONAL
The original			
one key			
ten keys			
twenty keys			
thirty keys			
forty keys			

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

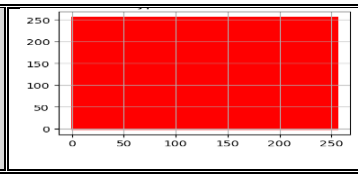
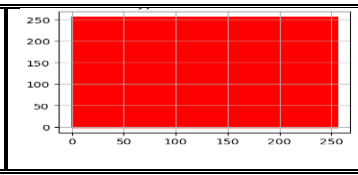
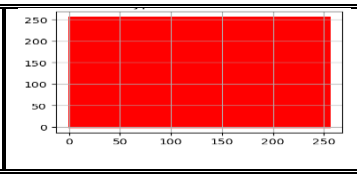
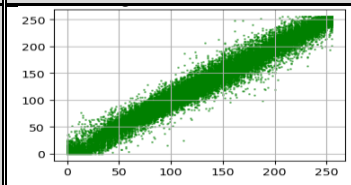
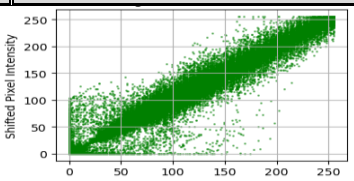
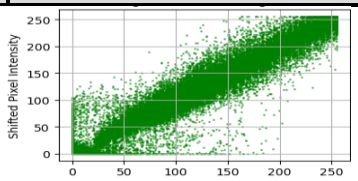
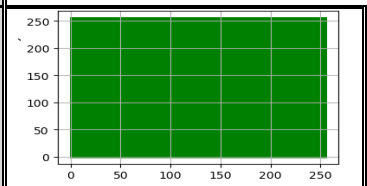
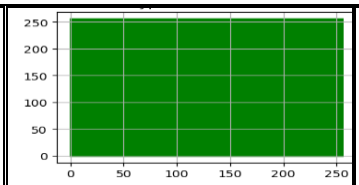
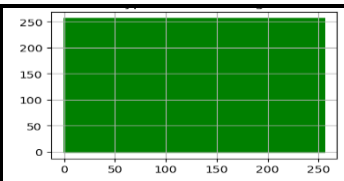
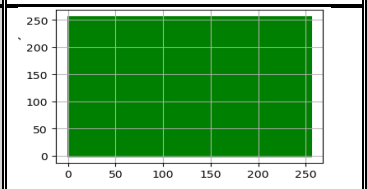
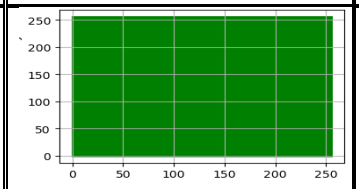
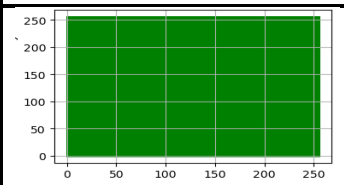
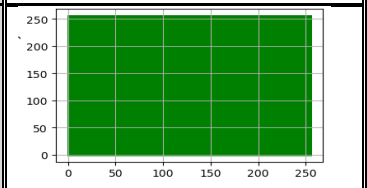
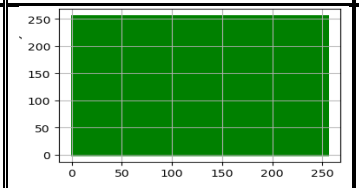
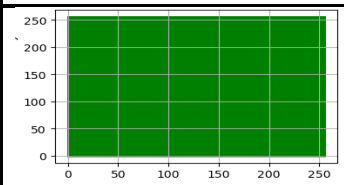
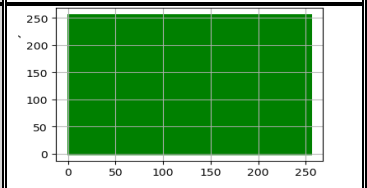
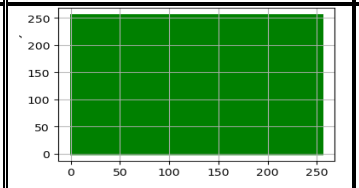
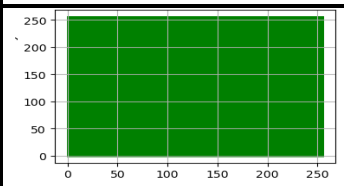
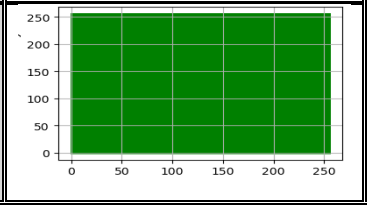
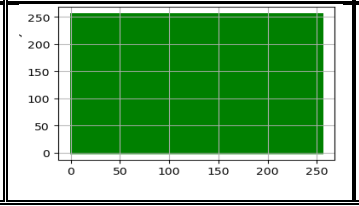
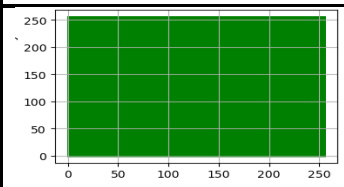
fifty keys			

Figure 7: The Correlation between the original and the encrypted image in red channel of the retinal image in various key numbers.

Test Name	**HORIZONTAL**	**VERTICAL**	**DIAGONAL**
The original			
one key			
ten keys			
twenty keys			
thirty keys			
forty keys			

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

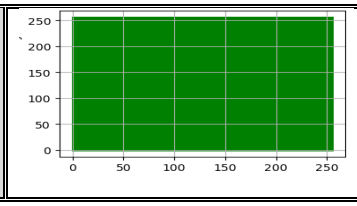
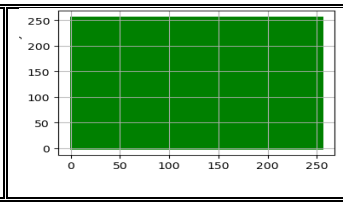
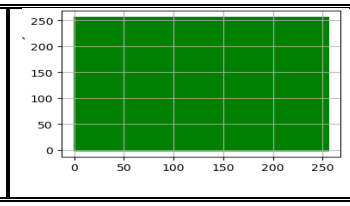
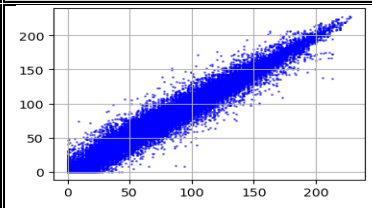
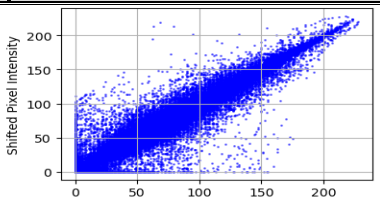
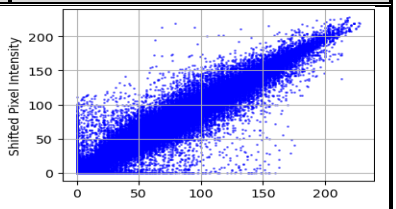
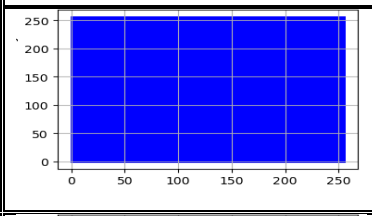
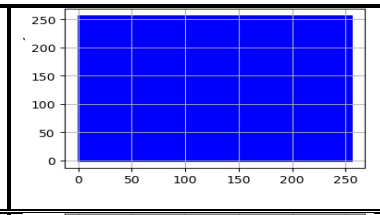
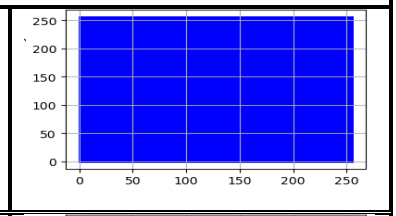
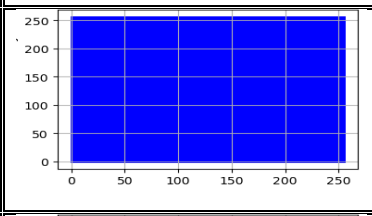
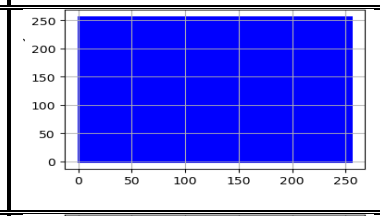
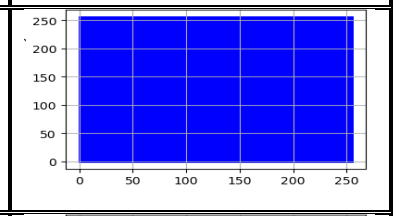
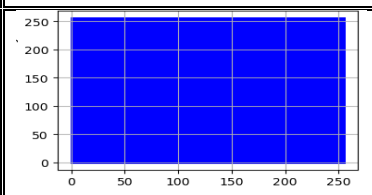
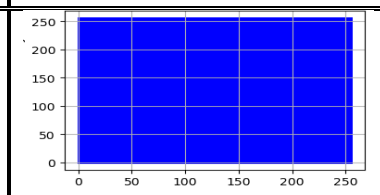
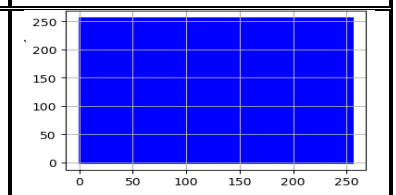
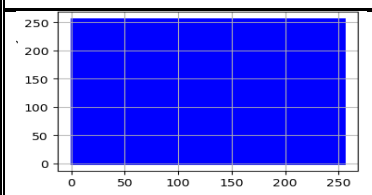
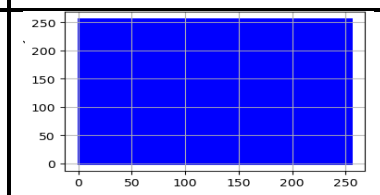
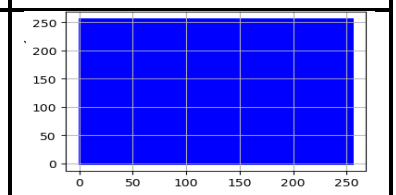
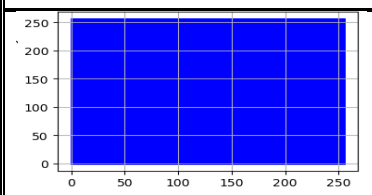
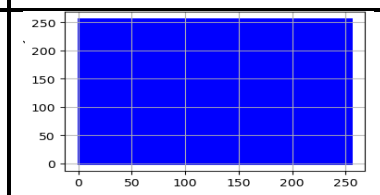
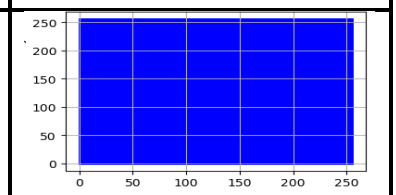
fifty keys			

Figure 8: The Correlation between the original and encrypted image in the green channel of the retinal image in various key numbers.

Test Name	**HORIZONTAL**	**VERTICAL**	**Blue _diagonal**
The Original			
one key			
ten keys			
twenty keys			
thirty keys			
forty keys			

Received: August 05 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

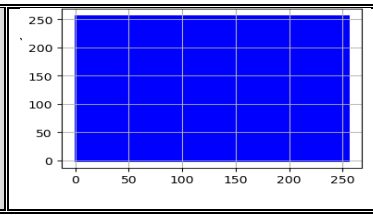
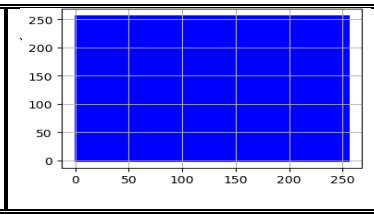
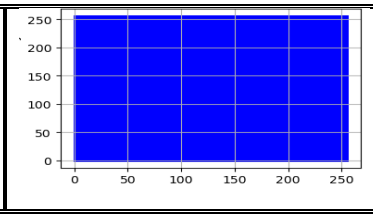
fifty keys			

Figure 9: The Correlation between the original and encrypted image in the blue channel of the retinal image in various key numbers.

The Conclusion

This work introduced a hybrid encryption approach that combines RNN-based key generation with AES to strengthen the security of image transmission in IoT environments. The proposed scheme successfully achieved high randomness in encryption keys and strong data confidentiality, as confirmed by entropy, PSNR/MSE, histogram, and correlation analyses. The results show that the encrypted images are highly resistant to statistical and spatial attacks, while the decrypted images are identical to the originals, proving the method's reliability and efficiency. The combination of machine learning-based key generation with a robust symmetric encryption algorithm demonstrates great potential for medical image protection and can be extended to other IoT security applications. The Future work will focus on optimizing the RNN model for real-time key generation and exploring its integration with lightweight cryptographic protocols for resource-constrained IoT devices.

References

[1] K. G. Salim, S. M. Kh. Al-alak, and M. J. Jawad, "Improved Image Security in Internet of Thing (IOT) Using Multiple Key AES," *Baghdad Sci. J.*, vol. 18, no. 2, p. 0417, June 2021, doi: 10.21123/bsj.2021.18.2.0417.

[2] Gunjan, S. Agarwal, D. Rai, and S. Talreja, "Applications of IoT in Smart Homes and Cities," in *IoT Based Smart Applications*, N. Sindhwani, R. Anand, M. Niranjanamurthy, D. Chander Verma, and E. B. Valentina, Eds., in EAI/Springer Innovations in Communication and Computing. , Cham: Springer International Publishing, 2023, pp. 55–70. doi: 10.1007/978-3-031-04524-0_4.

[3] V. Gugueoth, S. Safavat, S. Shetty, and D. Rawat, "A review of IoT security and privacy using decentralized blockchain techniques," *Comput. Sci. Rev.*, vol. 50, p. 100585, Nov. 2023, doi: 10.1016/j.cosrev.2023.100585.

[4] M. Kaur *et al.*, "EGCrypto: A Low-Complexity Elliptic Galois Cryptography Model for Secure Data Transmission in IoT," *IEEE Access*, vol. 11, pp. 90739–90748, 2023, doi: 10.1109/ACCESS.2023.3305271.

[5] S. Park, K. Kim, K. Kim, and C. Nam, "Dynamical Pseudo-Random Number Generator Using Reinforcement Learning," *Appl. Sci.*, vol. 12, no. 7, p. 3377, Mar. 2022, doi: 10.3390/app12073377.

Received: August 05 2025

Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[6] K. Okada, K. Endo, K. Yasuoka, and S. Kurabayashi, "Learned Pseudo-Random Number Generator: WGAN-GP for Generating Statistically Robust Random Numbers," June 27, 2022, *In Review*. doi: 10.21203/rs.3.rs-1695121/v1.

[7] U. Erkan, A. Toktas, S. Enginoğlu, E. Karabacak, and D. N. H. Thanh, "An Image Encryption Scheme Based on Chaotic Logarithmic Map and Key Generation using Deep CNN".

[8] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K. R. Choo, and Z. Qin, "DeepKeyGen: a deep learning-based stream cipher generator for medical image encryption and decryption," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4915–4929, 2021.

[9] Y. Alslman, E. Alnagi, A. Ahmad, Y. AbuHour, R. Younisse, and Q. Abu Al-haija, "Hybrid Encryption Scheme for Medical Imaging Using AutoEncoder and Advanced Encryption Standard," *Electronics*, vol. 11, no. 23, p. 3967, Nov. 2022, doi: 10.3390/electronics11233967.

[10] K. Panwar, A. Singh, S. Kukreja, K. K. Singh, N. Shakhovska, and A. Boichuk, "Encipher GAN: An End-to-End Color Image Encryption System Using a Deep Generative Model," *Systems*, vol. 11, no. 1, p. 36, Jan. 2023, doi: 10.3390/systems11010036.

[11] S. S. Chouhan, U. P. Singh, and S. Jain, "Applications of Computer Vision in Plant Pathology: A Survey," *Arch. Comput. Methods Eng.*, vol. 27, no. 2, pp. 611–632, Apr. 2020, doi: 10.1007/s11831-019-09324-0.

[12] S. B. Atim and M. Afdhaluddin, "Advanced Encryption Standard (AES) Cryptography Application Design," *J. Comput. Sci.*, vol. 6, no. 1.

[13] S. Almuhammadi and I. Al-Hejri, "A comparative analysis of AES common modes of operation," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON: IEEE, Apr. 2017, pp. 1–4. doi: 10.1109/CCECE.2017.7946655.

[14] M. Boden, "A guide to recurrent neural networks and backpropagation".

[15] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," *Phys. Nonlinear Phenom.*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[16] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 240–254, Mar. 1994, doi: 10.1109/72.279188.

[17] N. Alsadi, S. A. Gadsden, and J. Yawney, "Intelligent estimation: A review of theory, applications, and recent advances," *Digit. Signal Process.*, vol. 135, p. 103966, Apr. 2023, doi: 10.1016/j.dsp.2023.103966.

[18] S. Melacci, L. Sarti, M. Maggini, and M. Bianchini, "A Neural Network Approach to Similarity Learning," in *Artificial Neural Networks in Pattern Recognition*, vol. 5064, L.

Received: August 05 2025

Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Prevost, S. Marinai, and F. Schwenker, Eds., in Lecture Notes in Computer Science, vol. 5064. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 133–136. doi: 10.1007/978-3-540-69939-2_13.

[19] S. Mohammed, S. M. K. Al-Alak, and H. A. Lafta, "ECC and AES Based Hybrid Security Protocol for Wireless Sensor Networks".

[20] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Opt. Laser Technol.*, vol. 82, pp. 121–133, 2016.

[21] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.

[22] M. A. Fadhil Al-Husainy, "A novel image encryption algorithm based on the extracted map of overlapping paths from the secret key," *RAIRO-Theor. Inform. Appl.-Inform. Théorique Appl.*, vol. 50, no. 3, pp. 241–249, 2016.

[23] M. K. Hussein, K. R. Hassan, and H. M. Al-Mashhadi, "The quality of image encryption techniques by reasoned logic," *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 18, no. 6, p. 2992, Dec. 2020, doi: 10.12928/telkomnika.v18i6.14340.

[24] "Image Encryption with The Cross Diffusion of Two Chaotic Maps," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 2, Feb. 2019, doi: 10.3837/tiis.2019.02.031.

[25] M. J. Rostami, A. Shahba, S. Saryazdi, and H. Nezamabadi-pour, "A novel parallel image encryption with chaotic windows based on logistic map," *Comput. Electr. Eng.*, vol. 62, pp. 384–400, Aug. 2017, doi: 10.1016/j.compeleceng.2017.04.004.

Received: August 05 2025