Lightweight Medical Image Encryption for Secure Healthcare Systems Using Rabbit Stream Cipher and Chebyshev Chaotic Maps

Rasha Subhi Hameed and Noor Ahmed Hameed
Department of Computer Science and Artificial Intelligence, University of Diyala, 32001 Diyala, Iraq
purecomp.rasha.hameed@uodiyala.edu.iq, noorahmadhameed@uodiyala.edu.iq

Keywords: Rabbit Stream Cipher, Chebyshev Chaotic Maps, Lightweight Cryptography, Healthcare Systems, Information Technology (IT).

Abstract: Medical images are very sensitive information that should be well guarded, particularly in the existing healthcare systems, where information is usually shared over open networks. Medical devices that are resource-constrained require lightweight encryption methods, but the security of lightweight encryption relies on the quality of the initialization vectors (IVs). In this paper, a lightweight medical image encryption algorithm, called RabbitCheb, has been proposed that combines the Rabbit stream cipher with one-dimensional Chebyshev chaotic maps to improve IV randomness. In the suggested solution, Chebyshev chaotic sequences are used to create initialization vectors that are highly unpredictable, which enhances the randomness of the keystream generation process and does not increase the computational cost. RabbitCheb performance is tested using two medical image datasets, the Brain MRI Tumor dataset and the HAM10000 dataset. Several evaluation metrics are adopted, including entropy, NPCR, UACI, PSNR, and MSE. Experimental results demonstrate that the proposed method achieves high encryption performance, with an average entropy of 7.86, NPCR of 99.67, and UACI of 33.09, while maintaining low computational overhead. Compared with the original Rabbit cipher, RabbitCheb improves randomness and resistance against statistical and differential attacks. By employing chaos-based strategies to increase the randomness of IVs while preserving computational efficiency, this work enhances the security of lightweight stream ciphers. As a result, the proposed method is well-suited for securing medical image transmission in modern healthcare systems.

1 INTRODUCTION

The recent advances in medical technologies, namely wearable medical devices, telemedicine, and smart diagnostic systems, have greatly improved modern medical services and enabled monitoring patients remotely. These technologies generate and transmit large amounts of medical data, particularly medical imaging, such as computed tomography (CT) scans, magnetic resonance imaging (MRI), and X-rays. Because such images are sensitive and they carry patient data, it is necessary to provide their safe storage and transmission within a healthcare setting.

Medical images are usually transmitted via open communication networks and cloud platforms, which are exposed to cyberattacks, unauthorized access, and data breaches. Recent reports show that healthcare data breaches are the most expensive cybersecurity breaches, with an average financial loss of over USD 9.77 million in 2024. These issues show the need to take effective protection measures in order to guarantee the privacy and confidentiality of medical imaging information [1]. Image encryption has become a promising solution to the issue of medical image security transmission and storage. Most image encryption algorithms rely on two principles: diffusion and confusion. Confusion attempts to blur a relation between the plaintext and the encryption key by modifying the pixel values, and diffusion attempts to cause minor modifications in the plaintext output to cause large modifications in the ciphertext to reduce the statistical relationships and increase the complexity of the cryptanalytic attacks [2], [3].

Statistical and differential metrics like entropy, mean squared error (MSE), peak signal-to-noise ratio (PSNR), number of changing pixels (NPCR), and unified average changing intensity (UACI) are commonly used to assess an image encryption algorithm's efficiency [4]. In addition to strong security, computational performance is a critical need

in healthcare environments, where many medical devices have limited computational resources and power constraints. Therefore, lightweight encryption methods have attracted significant attention as an effective solution for protecting medical information in modern healthcare systems [5].

The conventional cryptographic algorithms, like RSA, offer a high level of security, but are computationally intensive when used with large medical images. They are computationally complex and therefore cannot be used in real time in medical and resource-limited healthcare devices [6]. In addition, these methods focus mainly on the protection of images when transmitting and storing them. After decryption of an image, the information can be spread without due regulation, and this can be a privacy issue in the context of smart healthcare [7], [8]. In that way, it is still quite difficult to design the encryption mechanisms of a high-security level and low-computational complexity in healthcare systems [9].

The past years have witnessed growing interest in lightweight-based encryption techniques for medical images. Hasan et al. [10] developed an encryption algorithm suited for resource-constrained environments, aiming to reduce the cost of computation. Similarly, Chaudhary and Chatterjee [11] introduced a lightweight healthcare system security framework that focuses on authentication and uses a time-stamping mechanism to eliminate replay attacks. In another study, Padma and Prasad [12] suggested a hybrid adaptive encryption scheme in the cloud-based healthcare application, combining S-box generation and elliptic curve cryptography. Ye and Chen [6] suggested a joint encryption and fingerprinting approach with discrete wavelet transform (DWT), cellular automata for confusion and diffusion with singular value decomposition (SVD) to offer confidentiality and traceability, but it is more computationally complex.

Moreover, several studies have examined chaotic and hybrid encryption techniques to enhance the security of medical images. For example, Kumari and Mondal [13] proposed a lightweight stream cipher, specifically an NLFSR with a message authentication code, which provides data confidentiality and integrity in Internet of Medical Things (IoMT) systems. Building on this, Demirkol et al. [14] suggested a hybrid medical image encryption algorithm that combines DNA coding and memristor-based chaotic systems. Along the same lines, Badr et al. [15] developed an improved GIFT-based lightweight encryption algorithm utilizing substitution and permutation. Likewise, Islam and Parah [16] introduced a fast and lightweight encryption technique leveraging modified chaotic systems to achieve effective confusion and diffusion. Continuing this trend, Afzal et al. [17] proposed a hybrid encryption system that combines logistic chaos, SHA-256 pseudorandom generation, dynamic DNA encoding, and Fibonacci diffusion. Finally, Rasheed and Kumar [18] introduced lightweight cryptographic techniques for resource-constrained IoT healthcare devices using hyper-chaotic diffusion and combined transformation techniques.

In recent years, the increasing use of digital healthcare systems has created a strong need for secure and efficient medical image encryption, especially in devices with limited resources. Although most of the currently existing methods enhance security, they tend to be based on complicated designs or demand a higher cost of computation, which restricts their application in practice. Rabbit stream cipher is regarded as a viable lightweight alternative because it is fast and simple, but the quality of the initialization vector (IV) determines its security, because weak or repetitive initialization vectors can decrease the randomness and influence the encryption process. Hence, a problem of generating IV with a reduced computational cost is still a significant challenge.

To address this drawback, a lightweight medical image encryption algorithm, RabbitCheb, a hybrid of the Rabbit stream cipher and the Chebyshev chaotic maps, is proposed in this paper. The presented algorithm has adopted Chebyshev chaotic sequences, which create a high degree of randomness in IVs, which increases the security of the initialization stage and the randomness of the keystream generation process, without raising the complexity of calculations. The proposed method offers efficient and secure encryption that can be applied to the current healthcare systems by combining the efficiency of the Rabbit cipher with the nonlinear nature of the chaotic maps. The main contributions of this work are summarized as follows:

1) A lightweight medical image encryption algorithm, RabbitCheb, is proposed by integrating the Rabbit stream cipher with Chebyshev chaotic maps to improve the randomness of the initialization process while preserving computational efficiency.
2) A chaos-based initialization mechanism is introduced to generate unpredictable initialization vectors, addressing the limitation of conventional Rabbit cipher implementations.

3) A security analysis is conducted to assess the resistance of the proposed algorithm against common cryptographic attacks, including brute-force, statistical, differential, key sensitivity, and noise attacks.
4) The effectiveness of the proposed algorithm in protecting medical images is demonstrated by evaluating its performance using common evaluation metrics, such as entropy, MSE, PSNR, NPCR, UACI, encryption time, decryption time, and computational complexity.

2 MATERIALS AND METHODS

This section describes the main components of the proposed encryption framework. The proposed method integrates the Rabbit lightweight stream cipher with Chebyshev chaotic maps in order to improve the randomness of the initialization vector used during the encryption process. First, the structure and operation of the Rabbit stream cipher are presented. Next, the mathematical formulation of the Chebyshev chaotic map is introduced. Finally, the proposed RabbitCheb encryption algorithm is explained.

2.1 Rabbit Lightweight Stream Cipher

This section describes the structure and operation of the Rabbit's lightweight stream cipher. The cipher operates with a 128-bit key and maintains an internal state of 513 bits composed of eight state variables $x_{j,t}$, eight counters $c_{j,t}$, and one carry bit initialized to zero. During the initialization phase, these values are derived from the secret key. For $t \geq 1$,the internal state is represents as $(x_{0,t}, \dots, x_{7,t}, c_{0,t}, \dots, c_{7,t})$.The counter values are updated iteratively based on their previous states [9], [19].

$$c_{j,t} = \begin{cases} c_{0,t-1} + a_0 + \phi_{7,t-1} \bmod 2^{32}, if\ j = 0 \\ c_{j,t-1} + a_j + \phi_{j,t-1} \bmod 2^{32}, if\ j > 0 \end{cases}. \quad (1)$$

In the relation:

$$\phi_{j,t+1} = \begin{cases} 1, if\ j = 0\ and\ c_{0,t} + a_0 + \phi_{7,t} > 2^{32} \\ 1, if\ j > 0\ and\ c_{j,t} + a_j + \phi_{j-1,t+1} > 2^{32} \\ 0, otherwise. \end{cases} \quad (2)$$

The $\phi_{j,t+1}$ represents the carry bit associated with the counter update operation. It is set to 1 when an overflow occurs during the addition process (i.e., when the result exceeds 2^{32}), and 0 otherwise.

a_i is constant; expressed using hexadecimals: $a_0 = a_3 = a_6 = 0x4D34D34D, a_1 = a_4 = a_7 = 0xD34D34D3, a_2 = a_5 = 0x34D34D34.$ $x_{j,t+1}$ is computed as following:

$$x_{0,t+1} = g_{0,t} + (g_{7,t} \lll 16) + (g_{6,t} \lll 16); \quad (3)$$
$$x_{1,t+1} = g_{1,t} + (g_{0,t} \lll 16) + g_{7,t}; \quad (4)$$
$$x_{2,t+1} = g_{2,t} + (g_{1,t} \lll 16) + (g_{0,t} \lll 16); \quad (5)$$
$$x_{3,t+1} = g_{3,t} + (g_{2,t} \lll 8) + g_{1,t}; \quad (6)$$
$$x_{4,t+1} = g_{4,t} + (g_{3,t} \lll 16) + (g_{2,t} \lll 16); \quad (7)$$
$$x_{5,t+1} = g_{5,t} + (g_{4,t} \lll 8) + g_{3,t}; \quad (8)$$
$$x_{6,t+1} = g_{6,t} + (g_{5,t} \lll 16) + (g_{4,t} \lll 16); \quad (9)$$
$$x_{7,t+1} = g_{7,t} + (g_{6,t} \lll 8) + g_{5,t}. \quad (10)$$

In (3)-(10), the symbol $\lll$ denotes a left rotation operation applied to three bits ($\lll 16$ when j is even and $\lll 8$ when j is odd). The symbol $\oplus$ represents the logical XOR operation, while || indicates the concatenation of two sequences. All arithmetic additions, represented by the symbol +, are performed modulo 232. Furthermore, the function $g_{j,t}$is defined as follows:

$$g_{j,t} = (x_{j,t} + c_{j,t+1})^2 \oplus ((x_{j,t} + c_{j,t+1})^2 \gg 32), 0 \leq j \leq 7. \quad (11)$$

2.1.1 Rabbit Structure

During the initialization phase, the 128-bit key K is represented as $k_7k_6\dots k_0$, where each k_i corresponds to a 16-bit value. Based on this key representation, the internal state of the cipher at time t = - 4 is determined according to the following relationships [19]:

$$x_{j,-4} = \begin{cases} k_{(j+1 \bmod 8)} || k_j, for\ even\ j \\ k_{(j+5 \bmod 8)} || k_{(j+4 \bmod 8)}, for\ odd\ j. \end{cases} \quad (12)$$

$$c_{j,-4} = \begin{cases} k_{(j+4 \bmod 8)} || k_{(j+5 \bmod 8)}, for\ even\ j \\ k_j || k_{(j+1 \bmod 8)}, for\ odd\ j. \end{cases} \quad (13)$$

$$\phi_{7,-4}. \quad (14)$$

After four iterations, these internal states produce the final initial state of the system. At this stage, each $c_{j,0}$ is updated according to (5).

$$c_{j,0} = c_{j,0} \oplus x_{(j+4 \bmod 8),0}. \quad (15)$$

After completing internal state initialization, the next iteration (t = 1) begins to emit the first key stream.

2.1.2 Key Stream Generation

When $t \geq 1$, 128-bit key streams s_t are computed using the following relations [20]:

$$s_t^{[15\ldots0]} = x_{0,t}^{[15\ldots0]} \oplus x_{5,t}^{[31\ldots16]}; \quad (16)$$
$$s_t^{[31\ldots16]} = x_{0,t}^{[31\ldots16]} \oplus x_{3,t}^{[15\ldots0]}; \quad (17)$$
$$s_t^{[47\ldots32]} = x_{2,t}^{[15\ldots0]} \oplus x_{7,t}^{[31\ldots16]}; \quad (18)$$
$$s_t^{[63\ldots48]} = x_{2,t}^{[31\ldots16]} \oplus x_{5,t}^{[15\ldots0]}; \quad (19)$$
$$s_t^{[79\ldots64]} = x_{4,t}^{[15\ldots0]} \oplus x_{1,t}^{[31\ldots16]}; \quad (20)$$
$$s_t^{[95\ldots80]} = x_{4,t}^{[31\ldots16]} \oplus x_{7,t}^{[15\ldots0]}; \quad (21)$$
$$s_t^{[111\ldots96]} = x_{6,t}^{[15\ldots0]} \oplus x_{3,t}^{[31\ldots16]}; \quad (22)$$
$$s_t^{[127\ldots112]} = x_{6,t}^{[31\ldots16]} \oplus x_{1,t}^{[15\ldots0]}. \quad (23)$$

The notation $s_t^{[a\ldots b]}$ refers to the segment of bit s_t starting at the bit and ending at the b^{th} bit. The length of this segment is a-b+1 bits, assuming that a>b.

2.1.3 Encryption/Decryption Rabbit

The encryption and decryption process relies on the XOR operation. Encryption is performed as $c_t = p_t \oplus$, while decryption using $p_t = c_t \oplus s_t$. In these expressions, c_t and p_t denote the t^{th} ciphertext block and plaintext block, respectively [19]

2.2 Chebyshev Chaotic Maps

The definition of the Chebyshev chaotic map for an integer n and x ∈ [-1,1] is [21]:

$$T_n(x) = \cos(n \times \arccos(x)). \quad (24)$$

Where:

- $T_n(x)$ represents the Chebyshev polynomial;
- x is the input variable;
- n is the polynomial degree.

The primary characteristics of Chebyshev polynomials are preserved in this formulation, especially the semigroup property that permits the creation of extremely complicated and sensitive sequences. This characteristic increases randomness and unpredictability, which is advantageous in cryptography applications.

2.3 RabbitCheb: Lightweight Medical Image Encryption Algorithm

The proposed RabbitCheb algorithm is a combination of the Rabbit stream cipher and Chebyshev chaotic maps to create a very unpredictable 64-bit key for the initialization of the cipher (IV). This integration maintains the lightweight structure of the original Rabbit stream cipher and increases randomness of the initialization process, making it appropriate for use in healthcare systems with limited resources. The RabbitCheb includes two main phases: (i) Chebyshev chaotic map –based Rabbit secret key and IV generation. A256-bit master key (MK) is the source of the Rabbit key (k). The Chebyshev map is an iteration for a pre-determined burn-in period (N_0), which helps eliminate transitory effects and stabilize the output. Because Chebyshev maps are highly sensitive to initial conditions, even small variations in parameters can result in significantly different outputs, making the IV unpredictable.

Algorithm 1 describes the generation of the Rabbit key and the chaotic initialization vector. After the burn-in phase, the generated chaotic values are normalized and converted into 32-bit words, which are concatenated until the required length of the IV is obtained. In the second phase, the medical image is encrypted using the Rabbit stream cipher. The input image is first converted into a one-dimensional byte sequence to enable stream-based processing. The Rabbit key and the IV generated in Algorithm 1 are used to initialize the internal state of the cipher. Algorithm 2 presents the complete RabbitCheb encryption procedure. The algorithm starts by converting the input medical image into a one-dimensional byte sequence to enable stream-based processing. The Rabbit key and the initialization vector generated in Algorithm 1 are then used to initialize the internal state of the Rabbit cipher.

```
Input: MK:256-bit master key
x0: initial chaotic value
n: Chebyshev polynomial degree
N0:number of burn-in iterations
Output: K:128-bit Rabbit key
    IV: 64-bit initialization vector
Begin
1. Derive the 128-bit Rabbit key K
from MK.
2. Initialize x ← x0
3. For i = 1 to N0 do
    x ← cos(n × arccos(x))  // using
Equation 24
 End For
4. Initialize IV ← Empty
5. While length(IV) < 64 bits do
      x ← cos(n × arccos(x))  //
using Equation 24
      Normalize x into the range
[0,1]
      Convert x to a 32-bit integer
value
```

```
      Concatenate the 32-bit value to
the IV
  End While
6. Truncate IV to 64 bits
7. Return (K, IV)
End Algorithm
```

Algorithm 1: Chebyshev-based key and IV generation.

During the initialization and post –initialization phases, the internal counters and state variables are iteratively updated according to the Rabbit state equations, ensuring proper diffusion and mixing of the internal state. After initialization, the cipher generates keystream blocks, which are combined with the plaintext bytes using the XOR operation to produce the encrypted image. The use of Chebyshev chaotic maps in the initialization stage enhances the randomness of the IV, resulting in a more unpredictable keystream and improved resistance against statistical and differential attacks.

Where P represents the plaintext, S is the keystream, C is the ciphertext, $x_{j,t}$ denotes the state variables,$c_{j,t}$, the counter, and $\phi_{j,t}$ the carry bit.

3 RESULTS AND DISCUSSION

All experiments were conducted on a computer running Windows 11 Pro (version 25H2) equipped with a 13th-generation Intel Core i7-13620H processor operating at 2.40 GHz. The RabbitCheb encryption algorithm recommended was implemented in Python 3.10.4 environments. The experimental setup was intended to test the functionality of the recommended algorithm, as well as to compare the proposed algorithm to the initial Rabbit stream cipher.

The experiments were conducted with two publicly available datasets on medical images, the Brain MRI Tumor dataset [22] and the HAM10000 skin lesion dataset [23]. The datasets consist of medical images of various sizes, formats, and color values, which enables the proposed method to be tested in various imaging conditions (Table 1).

Figure 1 shows the output sequences generated by the Chebyshev chaotic map for different parameter values over 50 iterations.

```
Input: I (H × W): input medical image
          MK: 256-bit master key
    x0initial chaotic value
          n: Chebyshev degree
    N0:burn-in iterations
Output: C: Cipher image
Begin
 1. Convert the input image I into a
    one-dimensional byte array P.
 2. Let L be the length of P.
 3. (K, IV) ← Call Algorithm 1(MK,
    x0,n, N0)
 4. Split the K into eight 16-bit
    Subkeys (k0, k1, …, k7).
 5. Initialize the Rabbit internal
    state variables xj and counter cj,t
    using Equations 12–14.
 6. For t = -4 to -1 do
    Update the counters cj,tusing
    Equation 1 and update the carry
    bit ϕj,t using Equation 2
    Compute nonlinear functions gj,t
    using Equation 11
    Update the state variables xj,t
    using Equations 3–10
    End For
 7. Modify counters using IV according
    to Equation 15
 8. For i = 1 to 4 do
    Update the counters and state
    variables using Equations 1-2 and
    Equation 11, and Equations 3-10
    End For
 9. Initialize the keystream array S
    ← to be empty
10. While length(S) < L do
    Perform one iteration of Rabbit
    next update using Equations 1-2
    and Equation 11, and Equations 3-
    10
    Generate one 128-bit keystream
    block St using Equations 16–23.
    Append St to S
    End While
11. Truncate S to length L
12. For i = 0 to L-1 do
    Ci ← Pi ⊕ Si
     End For
13. Reshape C into the image of size
    H × W
14. Return C
End Algorithm
```

Algorithm 2: RabbitCheb medical image encryption.

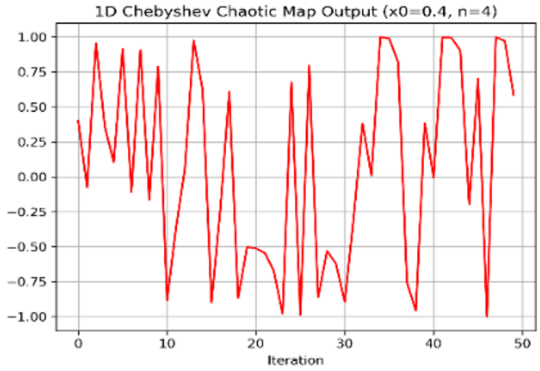
Table 1: Characteristics of medical image datasets used for experiment evaluation.

Dataset name	Total image	color	Image size in	File format
Brain MRI tumor dataset [22]	3,264	gray	240×240 pixel	JPG
HAM10000 (ISIC skin) dataset [23]	10,015	RGB	600×450 pixel	JPG/ PNG

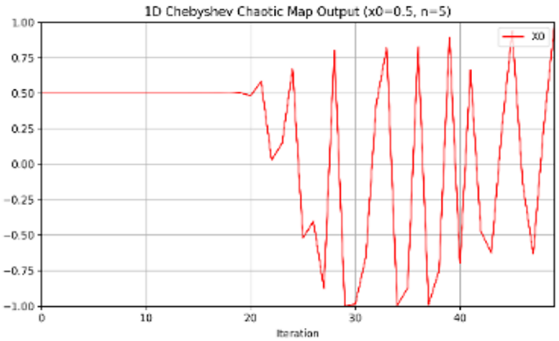
The irregular, non-periodic behavior within the interval [-1,1] reflects strong chaotic properties and high sensitivity to initial conditions. This sensitivity enhances IV randomness, leading to a more unpredictable keystream and improved resistance to statistical and differential attacks.

Figure 2 presents the statistical results of some of the initialization vectors generated by the proposed Chebyshev-based IV generating method. The resulting built sequences have an entropy of 3.7 to 3.9 bits per hexadecimal character. Given that the theoretical maximum number of bits of hexadecimal symbols is 4 bits, these findings demonstrate that the produced values are near-perfectly distributed and hence have high randomness properties, which enhances the value of the IV of the suggested algorithms. This increase in IV randomness increases the unpredictability of the keystream and increases the overall resistance of the RabbitCheb algorithm to cryptographic attacks. Moreover, the time to generate a chaotic IV is 0.014 ms to 0.035 ms, which proves that the chaotic IV generation process has minimal computational overhead.

The statistical characteristics of encrypted images are evaluated using histogram analysis [21]. A secure technique yields cipher images that are different from the original and have almost uniform histograms. The encrypted image produced by RabbitCheb has a uniform distribution, as seen in Figure 3, demonstrating better resilience to statistical attacks and successful statistical information concealment.



(a)



(b)

Figure 1: Chebyshev chaotic maps output: a) Chebyshev parameter x0=0.4 and n=4; b) Chebyshev parameter x0=0.5 and n=5.

Key File Preview & Statistics

Raw Data | Key Statistics

	Generated_Key	Generation_Time	Entropy
0	774d36d75c563f283abf850c94a4b4c7fe775176b54b97e1b234bd887b076a90	0.029	3.8356
1	4e262faebef3fc455a3b79fafaeda448ef6037a1f1ea87fc5bc83657f3a6a49b	0.0186	3.761
2	789e5be598534f82f57651ee064009ed5fe8eead20773dd76e87fa7b282da0dc	0.022	3.7721
3	3d691d0c5dc2a35273b9b80d0376273a4dbd192117feb1e1a6b508e0a9ea6568	0.021	3.8778
4	8ed44fd16808f0e337866e0ef2b57064775d1074d6c54cbc76b804755e333aef	0.0191	3.7495
5	0bfe6fe8fb057469f421ce59e0dc388c5e2687b339211f7507681a54f4b74af2	0.0275	3.9068
6	ea17f19ccc1fcd012b8ce6007fa1e0c33589b69d35c2de8638025900479762df	0.014	3.8863
7	4b24276a10865c9da7b0108f31ad07814e32c57afcc56e42bc1296c96f87ce5e	0.021	3.932
8	e05f8aefd66695f610452910f651fd8e1298ac6cca3463fefedc1ac730df62d6	0.035	3.7502
9	e41a2bca2cf4af2c06be4c6417f0e5c9a225c909585806622596794f812ff4c5	0.019	3.6927

Figure 2: An example of generation 20 rabbitcheb iv_{64} values using chebyshev-iv generation.

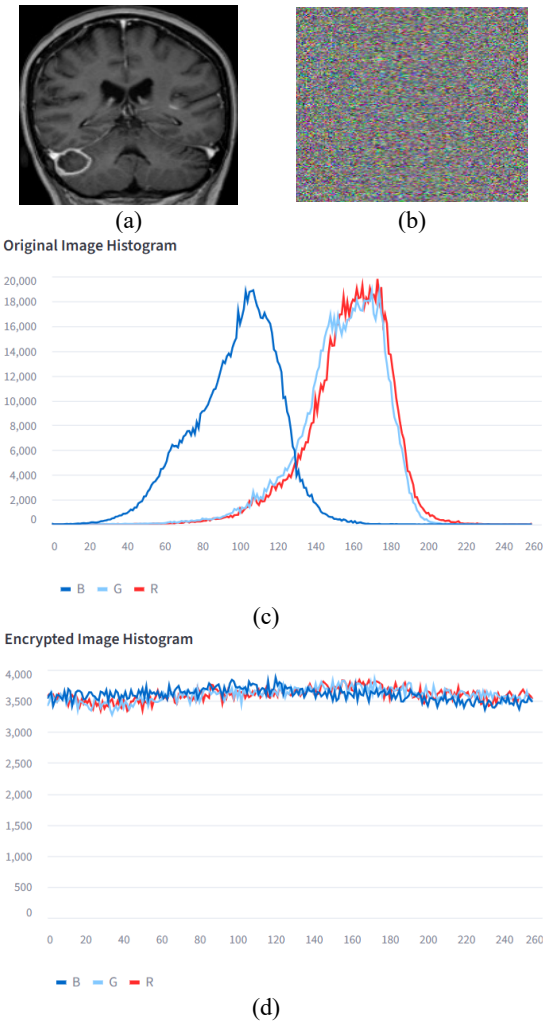


Figure 3: Performance of the rabbitcheb algorithm on medical images 256×256 with its histogram: a) original image; b) cipher image; c) histogram of the original image; d) histogram of cipher image.

Encryption and decryption time were used to assess the computational rate of the proposed RabbitCheb algorithm. Table 2 demonstrates the comparison between the proposed algorithm and the original Rabbit cipher on the medical images of size 2561 256. The findings indicate that RabbitCheb is able to preserve computational performance that is similar to the initial Rabbit cipher. The mean encryption and decryption times are about 0.0415 s and 0.0401 s, respectively, whereas the Rabbit original cipher has about 0.0389 s and 0.0375 s. This minor change suggests that the use of Chebyshev-based IV generation does not contribute significantly to the cost of computing.

Both the Rabbit cipher and the proposed RabbitCheb scheme have computational complexity, which is linear, with respect to the size of the image (N = H × W). The overriding activities in the two algorithms are the generation of a key stream and the encryption procedure by the XOR, which needs only one pass over the image bytes, amounting to a time complexity of (O(N)).

Though RabbitCheb uses Chebyshev chaotic iterations in order to generate the activation vectors, this extra step does not depend on the image size and only adds constant-time overhead (O(1)). Thus, the suggested improvement does not change the asymptotic complexity and maintains the lightweight and efficient character of the initial Rabbit algorithm.

To determine the overall security and encryption performance of the proposed RabbitCheb algorithm compared with the original Rabbit, entropy, MSE, PSNR, NPCR, and UACI are used to measure the algorithm's effectiveness.

1) Encryption Entropy [24] is calculated as:

$$H(\mathrm{x}) = -\sum_{\mathrm{i}=0}^{255} \mathrm{p(i)}\log_2 \mathrm{p(i)}, \qquad (25)$$

where the maximum entropy for an 8-bit image is 8-bit, and p(i) is the probability that grey level I will occur.

2) Mean Squared Error (MSE) is calculated using Equation 26 between the original image and the encryption image [16]:

$$MSE = \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{[I(i,j) - C(i,j)]^2}{N \times M}. \qquad (26)$$

Table 2: Encryption and decryption time (s) of the proposed RabbitCheb algorithm and its corresponding original Rabbit for a medical image of size 256×256.

Image No.	Proposed RabbitCheb		Original Rabbit	
	Encryption (s)	Decryption (s)	Encryption(s)	Decryption(s)
1	0.0482	0.0465	0.0458	0.0441
2	0.0334	0.0322	0.0311	0.0298
3	0.0398	0.0384	0.0371	0.0359
4	0.0446	0.0431	0.0417	0.0402
Avg.	0.0415s	0.0401s	0.0389s	0.0375s

Where:

- N and M represent image size,
- I(i, j)refers to the original image,
- C(i, j) denotes the cipher image.

3) Peak-Signal-to-Noise Ratio (PSNR) is employed to evaluate the quality of images [10]. Stronger encryption is indicated by lower PSNR values, which can be computed as:

$$PSNR = 10\frac{255^2}{MSE}\ . \qquad (27)$$

4) Number of Changing Pixels Rate (NPCR) calculates the proportion of encrypted image pixel changes caused by a one-pixel change in the source image [5]:

$$NPCR = \frac{Nc}{N} \times 100\%. \qquad (28)$$

Where N is the total number of pixels in the images, and N_c is the number of pixels that differ between the two encrypted images.

5) Unified Average Changing Intensity (UACI) is a statistic that quantifies how much the original image and the associated ciphered image differ [6].

$$NPCR = \frac{Nc}{N} \times 100\%. \qquad (29)$$

Where I(x, y) and K(x, y) denote intensity values of the pixels located at position $y\ in$ in the two images. The variable n represents the total number of pixels in the image, while MAX_i corresponds to the highest possible pixel intensity value, such as 255 in the case of an 8-bit image.

Tables 3 and 4 show the encryption performance for the proposed RabbitCheb and original Rabbit algorithms using four test images. The entropy has a range of values between 7.6087 and 7.9978, and is closer to the theoretical value of 8. MSE ranges between 7814.2278- 12098.6921, whereas PSNR ranges between 6.9847- 8.9126 dB, which is high and distorted. The values of NPCR are 99.5214-99.7013, and UACI 26.9814-35.8621 are in the range of the theoretical 33.

A comparison of the RabbitCheb algorithm and the Rabbit algorithm is shown in Figure 4. The results indicate that RabbitCheb achieves a higher average entropy value (7.8602) compared with Rabbit (7.7541), suggesting an improvement in randomness. In addition, RabbitCheb produces a higher MSE of 10358.04 and a lower PSNR of 7.6057 dB, whereas Rabbit records an MSE of 9258.78 and a PSNR of 8.4960 dB.

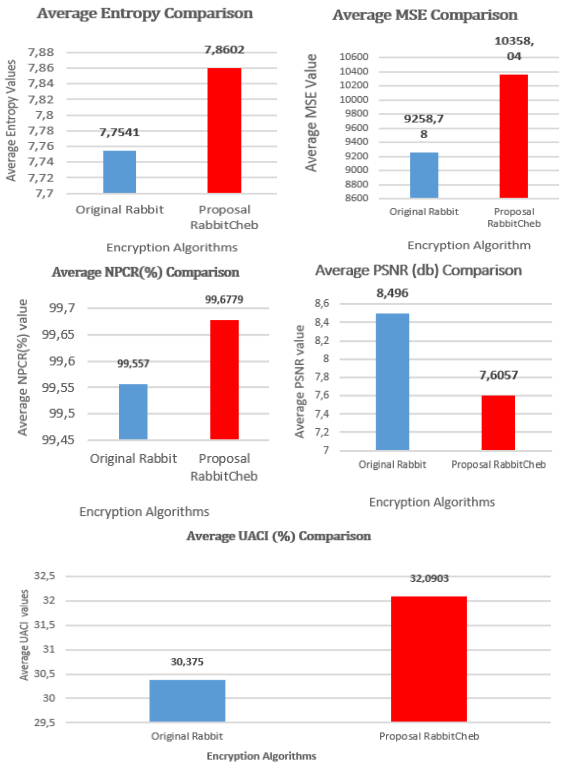


Figure 4: Comparison between the original rabbit and rabbitcheb based on entropy, MSE, PSNR, NPCR, and UACI.

Table 3: Results of original rabbit based on entropy, MSE, PSNR, NPCR, and UACI.

Metrics	Original Rabbit			
	Image1	Image2	Image3	Image4
Entropy (≈ 8)	7.7200	7.9908	7.8215	7.4839
MSE↑	9345.12	9021.77	7814.23	10853.99
PSNR↓ (*db*)	8.9126	8.3472	8.8817	7.8425
NPCR↑ (%)	99.5214	99.5336	99.5742	99.5988
UACI≈ 33	29.7439	28.9125	26.9814	35.8621

These results indicate that the proposed method introduces greater visual distortion between the plain and encrypted images. Moreover, RabbitCheb demonstrates a little better differential attack metrics, with an average NPCR of 99.6779% and UACI of 32.0903, which comes much closer to the theoretical optimal value of 33% than the values that Rabbit demonstrates (30.3750% in UACI). In general, the

findings indicate that the RabbitCheb algorithm has a greater diffusion strength and resistance to differential attacks than the original Rabbit algorithm.

Table 4: Results of proposed rabbitcheb based on entropy, MSE, PSNR, NPCR, and UACI.

Metrics	Proposed RabbitCheb			
	Image1	Image2	Image3	Image4
Entropy (≈ 8)	7.8707	7.9637	7.9978	7.6087
MSE↑	11234.58	9815.70	8283.18	12098.69
PSNR↓ (db)	7.6842	8.0325	7.7214	6.9847
NPCR↑ (%)	99.6728	99.6485	99.6891	99.7013
UACI≈ 33	32.8841	31.5649	30.6376	33.2745

The test results indicate that the RabbitCheb algorithm is effective for medical image encryption. The values of entropy are near the theoretical maximum of 8, which means that the ciphertext is highly random. The values of NPCR and UACI are within their expected ranges, which indicates that it is well diffused and resistant to any form of differential attack. The histograms of the encrypted images are nearly uniform, and the correlation between pixels is close to zero, which means that the statistical properties of the original images have been properly concealed. Moreover, the algorithm maintains low computational cost, as the chaotic initialization adds only a small overhead. The results also show that RabbitCheb performs better than the original Rabbit cipher. This improvement is mainly due to the use of Chebyshev chaotic maps in generating the initialization vector (IV), which leads to a more unpredictable keystream.

Tables 5 and 6 include the comparative analysis of the current lightweight medical image encryption in a healthcare system and the proposed RabbitCheb algorithm. The results indicate that the suggested algorithm obtains an effective compromise between the high level of security and low level of computation complexity, which is why it is a good, efficient solution to deploy in healthcare settings.

Table 5: Comparison analysis of existing lightweight medical image encryption and proposed rabbitcheb in healthcare systems.

Ref. No	Entropy	NPCR (%)	UACI(%)	PSNR(dB)	Enc. Time (s)
[10]	-	-	-	39	-
[11]	-	-	-	-	1.194
[12]	-	-	-	-	15.4
[6]	-	-	-	>30	-
[13]	7.23	-	-		-
[14]	7.99	99.6	-	7-19	-
[15]	7.99	99.62	31.22	8.37	-
[16]		99.6	33.4	-	0.03
[17]	7.99	99.73	33.72	-	-
[18]	8.13-11.12	99.57-99.62	35.45-38.89	-	-
RabbitCheb	7.86	99.67	33.09	7.61	0.14

Note: N/A indicates that the value was not available in the reported results of the referenced study.

Table 6: Comparison Analysis of Existing Lightweight Medical Image Encryption and Proposed RabbitCheb based on the strength and limitations of the key.

Ref. No	Year	strength	limitation
[10]	2021	High entropy	Low speed
[11]	2022	Low computation	High encryption time
[12]	2023	Real-time support	Low robustness
[6]	2023	Attack resistance	Moderate entropy
[13]	2024	Large key space	High complexity
[14]	2024	High security	Complex design
[15]	2024	Strong key generation	Limited testing
[16]	2024	Fast encryption time	Devices limitations
[17]	2025	High entropy	Sorting overhead
[18]	2025	Strong diffusion	Limited validation
RabbitCheb	2025	Fast and low complexity O(N)	Tested on medical images only

3.1 Security Analysis

The security of the suggested RabbitCheb algorithm will be methodically assessed in this section.

3.1.1 Brute-Force Attack

The suggested RabbitCheb algorithm uses a one-dimensional Chebyshev chaotic map to generate the IV while maintaining the Rabbit stream cipher's 128-bit secret key. The resultant key space is at least 2^{128}, making a brute-force attack computationally infeasible. In addition, the effective search domain is expanded by the sensitivity of the Chebyshev-based IV generation. As a result, RabbitCheb maintains security in line with contemporary 128-bit cryptographic standards while offering robust defense against brute-force attacks.

3.1.2 Statistical (Entropy-Based) Attacks

As shown in Table 7, RabbitCheb achieves entropy values close to the ideal entropy of 8 bits and higher than those of the original Rabbit cipher. Additionally, the correlation coefficients between adjacent pixels (horizontal, vertical, and diagonal) are closer to zero compared with the original Rabbit cipher. This indicates a higher degree of statistical decorrelation. These results show that the use of the one-dimensional Chebyshev map improves ciphertext randomness and resistance to statistical attacks.

Table 7: Statistical comparison between proposed rabbitcheb and original rabbit algorithms.

#	Metrics	Proposed RabbitCheb	Original Rabbit
Image 1	Entropy	7.9991	7.9124
	Correlation -H	0.0004	0.0125
	Correlation -V	0.0021	0.0108
	Correlation -D	0.0002	0.0097
Image 2	Entropy	7.9993	7.9048
	Correlation -H	0.1009	0.0109
	Correlation -V	0.0048	0.0124
	Correlation -D	0.0003	0.0086

3.1.3 Key Sensitivity Test

To evaluate key sensitivity, a one-bit change was applied to a 128-bit key while keeping all other parameters unchanged. The resulting ciphertexts were compared using NPCR and UACI metrics. The results show that the two encrypted images are substantially different, with NPCR values close to 99% and UACI values approaching the expected value of 33%. These results indicate a strong key sensitivity.

3.1.4 Differential Attack Analysis

The RabbitCheb algorithm's robustness against differential attacks was tested by changing a single pixel in the plaintext image and encrypting both images with the same secret key and IV. The ciphertext images were compared using NPCR and UACI measures. The results show that the proposed algorithm achieves strong diffusion, as NPCR and UACI are close to their theoretical values.

3.1.5 Noise Attack Analysis

The robustness of the proposed method was examined by adding Salt-and-Pepper noise and Gaussian noise at low and moderate levels, followed by decryption using the correct key. Figure 5 presents the reconstructed images obtained under different noise conditions. It can be observed that when the noise level is low, the reconstructed images remain visually recognizable. However, increasing the noise level results in more noticeable degradation.

PSNR and SSIM metrics were used to assess the image quality in order to quantify the impact of noise on the reconstructed images, see Table 8.

Table 8: PSNR and SSIM results of the proposed RabbitCheb under noise conditions.

Noise type	Noise level	PSNR(dB)	SSIM	Visual observation
Salt & Pepper	0.01	28.45	0.91	Slight distortion
Salt & Pepper	0.05	21.30	0.74	Noticeable degradation
Gaussian	0.001	30.12	0.94	Minor visual change
Gaussian	0.01	22.05	0.77	Clear degradation

The results indicate that the proposed RabbitCheb scheme can handle low levels of transmission noise, whereas increasing noise levels result in a gradual degradation of the reconstructed image quality.

Table 9 summarizes the outcomes of the additional security evaluations, including tests on key sensitivity, resistance to differential and chosen-plaintext attacks, and the behaviour of the algorithm under noise and cropping disturbances.

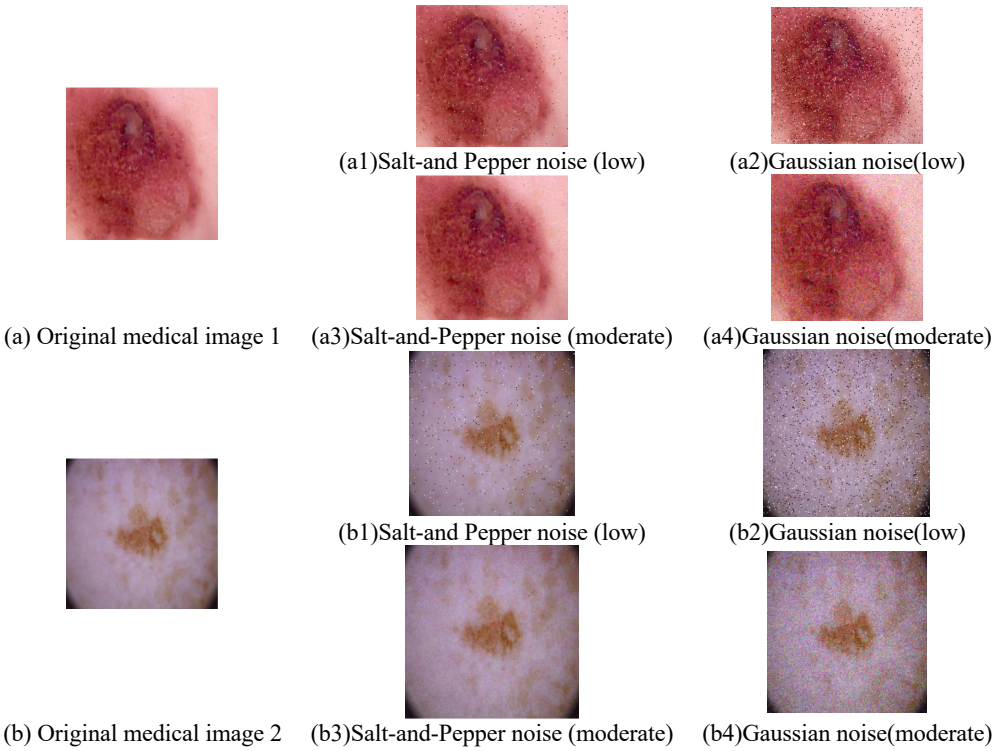


Figure 5: Noise attack analysis using a medical image.

Table 9: Security analysis of the proposed RabbitCheb.

Test	Evaluation Method	Expected secure behavior	RabbitCheb observation
Key sensitivity	One-bit key change	Large ciphertext difference	High sensitivity observed
Differential attack	One-pixel plaintext change	NPCR≈ 99%, UACI≈ 33%	Strong diffusion
Chosen-plaintext	Structure image	no visible plaintext pattern	Uniform cipher output
Noise attack	Add noise before decryption	Partial robustness	acceptable

4 CONCLUSIONS

This paper introduced RabbitCheb, an improved lightweight medical image encryption algorithm based on Rabbit stream cipher and a one-dimensional Chebyshev chaotic map to generate the initialization vectors. The proposed approach will enhance the randomness of the initialization stage but will preserve the computational efficiency of the original Rabbit cipher, which applies to healthcare systems with limited computational resources. The findings demonstrate that the RabbitCheb has a good security performance (high randomness, good diffusion, and low correlation between encryption pixel) and has low computational complexity. These results demonstrate that incorporating Chebyshev chaotic maps improves the Rabbit Cipher's diffusion and randomness characteristics without losing its lightweight design. However, the algorithm has been evaluated only on medical image datasets. Future work will concentrate on extending the algorithm to accommodate secure encryption of medical video streams and multimodal healthcare data, investigating hardware-level optimization for embedded medical devices, and implementing the suggested method into practice in real-time IoMT scenarios.

ACKNOWLEDGMENT

The University of Diyala in Iraq provided the facilities and academic setting that enabled this study, for which the authors are grateful. The University of Diyala's Department of Computer Science and Artificial Intelligence and the College of Basic Education's Department of Computer Science are also

acknowledged by the authors for their assistance in finishing this work.

REFERENCES

[1] K. Lata, C. Gupta, and L. R. Cenkeramaddi, "A Cryptographic Framework for Secure Medical Imaging in Smart Healthcare Environments," Results in Engineering, vol. 25, Art. no. 106780, 2025, [Online]. Available: https://doi.org/10.1016/j.rineng.2025.106780.

[2] J. Arif, M. A. Khan, B. Ghaleb, J. Ahmad, A. Munir, and U. Rashid, "A Novel Chaotic Permutation-Substitution Image Encryption Scheme Based on the Logistic Map and Random Substitution," IEEE Access, vol. 10, pp. 12966-12982, 2022, [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3146792.

[3] J. Ferdush, M. Begum, and M. S. Uddin, "Chaotic Lightweight Cryptosystem for Image Encryption," Advances in Multimedia, vol. 2021, Art. no. 5527295, 2021, [Online]. Available: https://doi.org/10.1155/2021/5527295.

[4] A. Alanezi, B. Abd-El-Atty, H. Kolivand, A. A. Abd El-Latif, B. Abd El-Rahiem, and S. Sankar, "Securing Digital Images Through Simple Permutation-Substitution Mechanism in Cloud-Based Smart City Environment," Security and Communication Networks, vol. 2021, Art. no. 6615512, 2021, [Online]. Available: https://doi.org/10.1155/2021/6615512.

[5] P. Kumari and B. Mondal, "Lightweight Image Encryption Algorithm Using NLFSR and CBC Mode," Journal of Supercomputing, vol. 79, no. 17, pp. 19452-19472, 2023, [Online]. Available: https://doi.org/10.1007/s11227-023-05415-9.

[6] C. Ye and C. Chen, "Secure Medical Image Sharing for Smart Healthcare System Based on Cellular Neural Network," Complex and Intelligent Systems, vol. 9, no. 2, pp. 1653-1670, 2023, [Online]. Available: https://doi.org/10.1007/s40747-022-00881-9.

[7] S. Alsaraireh, A. Ahmad, and Y. AbuHour, "New Step in Lightweight Medical Image Encryption and Authenticity," Mathematics, vol. 13, no. 11, Art. no. 1799, 2025, [Online]. Available: https://doi.org/10.3390/math13111799.

[8] M. A. Al-Fayoumi, A. Odeh, I. Keshta, and A. Ahmad, "Techniques of Medical Image Encryption Taxonomy," Bulletin of Electrical Engineering and Informatics, vol. 11, no. 4, pp. 1990-1997, 2022, [Online]. Available: https://doi.org/10.11591/eei.v11i4.3850.

[9] T. H. Obaida, A. S. Jamil, and N. F. Hassan, "Improvement of Rabbit Lightweight Stream Cipher for Image Encryption Using Lévy Flight," International Journal of Health Sciences, vol. 6, pp. 1628-1641, 2022, [Online]. Available: https://doi.org/10.53730/ijhs.v6nS8.11630.

[10] M. K. Hasan, M. T. Islam, M. M. Rahman, M. S. Hossain, A. Ghoneim, and M. Guizani, "Lightweight Encryption Technique to Enhance Medical Image Security on Internet of Medical Things Applications," IEEE Access, vol. 9, pp. 47731-47742, 2021, [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3061710.

[11] R. R. K. Chaudhary and K. Chatterjee, "A Lightweight Security Framework for Electronic Healthcare System," International Journal of Information Technology, vol. 14, no. 6, pp. 3109-3121, 2022, [Online]. Available: https://doi.org/10.1007/s41870-022-01034-4.

[12] B. P. V. Dev and K. V. Prasad, "An Adaptive Lightweight Hybrid Encryption Scheme for Securing Healthcare Data in Cloud-Assisted Internet of Things," Wireless Personal Communications, vol. 130, no. 4, pp. 2959-2980, 2023, [Online]. Available: https://doi.org/10.1007/s11277-023-10411-6.

[13] P. Kumari and B. Mondal, "Lightweight Encryption With Data and Device Integrity Using NLFSR and PUF for the Internet of Medical Things," Internet of Things, vol. 25, Art. no. 101041, 2024, [Online]. Available: https://doi.org/10.1016/j.iot.2023.101041.

[14] A. S. Demirkol, M. E. Sahin, B. Karakaya, H. Ulutas, A. Ascoli, and R. Tetzlaff, "Real-Time Hybrid Medical Image Encryption Algorithm Combining Memristor-Based Chaos With DNA Coding," Chaos, Solitons and Fractals, vol. 183, Art. no. 114923, 2024, [Online]. Available: https://doi.org/10.1016/j.chaos.2024.114923.

[15] A. M. Badr, L. C. Fourati, and S. Ayed, "An Improved GIFT Lightweight Encryption Algorithm to Protect Medical Data in IoT," in Proceedings of IEEE Symposium on Computers and Communications (ISCC), 2023, pp. 1-7.

[16] M. O. U. Islam and S. A. Parah, "Fast and Lightweight Image Cryptosystem for IoMT Applications," Internet of Things, vol. 25, Art. no. 101083, 2024, [Online]. Available: https://doi.org/10.1016/j.iot.2024.101083.

[17] S. Afzal, M. Usman, M. Waqas, A. Rehman, M. Imran, and F. Al-Turjman, "A Hybrid Encryption Model for Medical Image Security in IoT Healthcare," Discover Computing, vol. 28, no. 1, Art. no. 321, 2025, [Online]. Available: https://doi.org/10.1007/s10791-025-09876-9.

[18] A. M. Rasheed and R. M. S. Kumar, "Efficient Lightweight Cryptographic Solutions for Enhancing Data Security in Healthcare Systems Based on IoT," Frontiers in Computer Science, vol. 7, Art. no. 1522184, 2025, [Online]. Available: https://doi.org/10.3389/fcomp.2025.1522184.

[19] K. Chain, "The Security Analysis on the Rabbit Stream Cipher," International Journal on Information Technologies and Security, vol. 16, no. 2, pp. 91-102, 2024, [Online]. Available: https://doi.org/10.59035/SWYF4934.

[20] M. Boesgaard, M. Vesterager, and E. Zenner, "The Rabbit Stream Cipher," in New Stream Cipher Designs: The eSTREAM Finalists, Berlin, Germany, Springer, 2008, pp. 69-83, [Online]. Available: https://doi.org/10.1007/978-3-540-68351-3_7.

[21] C. Meshram, C. Lee, I. Bahkali, and A. L. Imoize, "An Efficient Fractional Chebyshev Chaotic Map-Based Three-Factor Session Initiation Protocol for the Human-Centered IoT Architecture," Mathematics, vol. 11, no. 9, Art. no. 2085, 2023, [Online]. Available: https://doi.org/10.3390/math11092085.

[22] N. Navoneel, "Brain MRI Images for Brain Tumor Detection," Kaggle Dataset, [Online]. Available: https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection, [Accessed: 2026].

[23] K. Mader, "Skin Cancer MNIST: HAM10000 Dataset," Kaggle Dataset, [Online]. Available: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000, [Accessed: 2026].

[24] S. Kumar and R. Revathi, "An Efficient Image Encryption Algorithm Using a Discrete Memory-Based Logistic Map With a Deep Neural Network," Journal of Engineering and Applied Science, vol. 71, no. 1, Art. no. 41, 2024, [Online]. Available: https://doi.org/10.1186/s44147-023-00349-8.