Ingénierie des Systèmes d'Information
Vol. 30, No. 10, October, 2025, pp. 2793-2805
Journal homepage: http://iieta.org/journals/isi

A Novel Chaos-Based CDT Map for Digital Image Encryption

Ari Rosemalatriasari[1*], Suryadi MT[2], Sarifuddin Madenda[1]

[1] Department of Information Technology, Universitas Gunadarma, Depok 16424, Indonesia
[2] Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia

Corresponding Author Email: arirosemala@gmail.com

https://doi.org/10.18280/isi.301024

Received: 15 September 2025
Revised: 17 October 2025
Accepted: 24 October 2025
Available online: 31 October 2025

Keywords:
chaotic map, Circle–Dyadic Transformation Map, image encryption, keystream generator, NIST SP800-22, security analysis

ABSTRACT

Securing digital images remains a major challenge in modern communication systems due to increasing risks of interception, manipulation, and unauthorized access. This study proposes a novel chaotic function called the Circle–Dyadic Transformation Map (CDT Map), constructed through the composition of the Circle Map and the Dyadic Transformation Map, to serve as a keystream generator for image encryption. The methodology consists of four key stages: (1) formulating the CDT Map via function composition, (2) validating its chaotic behavior through Lyapunov exponent analysis, bifurcation diagrams, and the NIST SP800-22 test suite, (3) designing a keystream-based encryption–decryption algorithm using XOR operations, and (4) evaluating performance through statistical, differential, and quality metrics. Experimental results show that the CDT Map achieves a 100% pass rate on NIST randomness tests, a key space of 5.832×10^{650}, high key sensitivity (10^{-16}), and superior NPCR (99.6%) and UACI (≈40%) values compared to existing chaotic maps. The proposed approach ensures perfect decryption (MSE = 0, PSNR = ∞) and strong resistance to brute-force, statistical, and differential attacks. The main contribution of this work is the development of a new composite chaotic function that significantly enhances randomness, security strength, and computational efficiency for digital image encryption.

1. INTRODUCTION

Manipulation of digital image data poses a threat to data owners, especially when the images are used for malicious purposes. To protect confidential data, it is necessary to design storage and transmission systems that can prevent unauthorized access and modification, both when images are stored on computers and when they are sent via online services such as email or cloud storage. Digital images such as medical images are highly sensitive and require a robust security system to maintain confidentiality in accordance with medical ethics [1]. The security of medical record data is an obligation as stipulated in Law Number 29 of 2004 concerning Medical Practice and Article 28G paragraph (1) of the 1945 Constitution, which guarantees the protection of individual rights to privacy and personal security [2].

The necessity for data security necessitates the adoption of cryptographic methods. Historically, cryptography is categorized into classical and modern forms. Both necessitate keys for the processes of encryption and decryption. Classical encryption prioritizes the confidentiality of algorithms, whereas current cryptography emphasizes the confidentiality of keys. Cryptography is categorized into symmetric and asymmetric keys based on the encryption key. In symmetric cryptography, a single key is employed for both encryption and decryption, whereas asymmetric cryptography utilizes distinct key pairs. Commonly utilized conventional encryption techniques encompass the Data Encryption Standard (DES), the Advanced Encryption Standard (AES), and the Rivest-Shamir-Adleman (RSA) algorithm. This approach, while effective in encryption, is rendered less appropriate for digital image encryption due to its limited key space and sluggish processing speed [3]. This is due to the substantial duplication and significant correlation among pixels in multiple directions within digital images. The prolonged encryption duration and limited key space render typical text encryption algorithms inefficient [4].

Efforts to create digital picture encryption with chaotic functions are conducted by sequential or ordered implementation of two or more encryption phases, or by composing functions. This is undertaken to enhance resilience against diverse attackers. Certain research incorporates a chaos-based diffusion-transposition method for the security of digital images [5]. During the diffusion phase, the Logistic Map function is employed, while the transposition phase utilizes Arnold's Cat map. The technique produces a key space of 1.84×10^{49}, with key sensitivity attaining 10^{-16}. This indicates that the algorithm is resistant to brute-force attacks. The algorithm for constructing the Logistic Map and the Chebyshev Map is arranged sequentially [6]. This technique is employed to encrypt medical photographs. Initially, the image undergoes encryption via the Logistic Map, resulting in an encrypted image that is subsequently re-encrypted using the Chebyshev Map. This approach employs two functions to

encrypt medical images in a two-step process. The decryption process is conducted in an identical manner. This algorithm comprises four parameters. Altering the parameter values of the Logistic Map to infeasible levels compromises the security of the image. Furthermore, the encryption process requires double the duration.

The Circle Map has unlimited potential for chaos and is composed with the Gauss Map. The algorithm for composing the two sequentially [5]. The possibility of having greater chaotic properties is investigated in this way. The sensitivity diagram of this algorithm is much larger with respect to the initial value. Only four out of 16 NIST tests passed the randomness test, meaning the randomness level is only 25% [7]. Therefore, if this Gauss-Circle Map is used for cryptographic purposes, the cryptographic system that uses it will have strong resistance to brute-force attacks, but it may also be weak against statistical attacks. Another encryption algorithm is composing MS Map and Dyadic Transformation Map [8, 9]. Based on the bifurcation diagram, it can be seen that for values in the range $\lambda \in (0.3, 5)$, the density is better when r = 3.8 [10]. The Lyapunov exponent results show that non-negative values are satisfied for r values [8]. The NIST test results with initial values and parameter values $x_0 = 0.6$, r = 3.8, and $\lambda = 3.5$, and a randomness level of 82.4% of the NIST randomness test results.

The MS Gauss Map algorithm is the result of developing two chaotic functions, namely composing the MS Map function and the Gauss Iterated Map [11]. The key space generated is 1.8×10^{79} and the key sensitivity reaches 10^{-16}, making it resistant to brute-force attacks. Additionally, testing on encrypted image data is also resistant to statistical and differential attacks. Another encryption algorithm based on chaos is the composition of the Bernoulli Map and the Logistic Map [12]. The key space of this algorithm is $[(2^{31} - 1) - b \times k \times d] \times 1.6 \times 10^{634}$, and the key sensitivity up to 10^{-18} makes this function composition difficult to break with a brute-force attack. Based on testing using two NIST tests on the random number sequence generated from the Bernoulli Logistic Map, the sequence is random, and the distribution of pixel intensity values is uniform [13]. The test results with PSNR on the original image data and the decrypted image data are ∞.

Brute-force attacks, statistical attacks, and differential attacks are some types of cryptographic attacks that can occur during the storage and transmission of secret data or information [14, 15]. A brute-force attack is a type of attack that tries all possible combinations of encryption keys to obtain the plaintext from the ciphertext they know. Statistical attack is an attack that exploits statistical data (e.g., correlation and distribution data between pixels of the ciphertext) to obtain the original text (plaintext) [16]. Differential attack is when the attacker looks for a relationship between the ciphertext and a number of related plaintexts. The analysis was performed on each of these pairs with several rounds of analysis based on the patterns found from the encryption process.

The rapid growth of digital communication systems has intensified concerns regarding the security of sensitive image data, particularly in fields such as medical imaging, surveillance, and forensics [17]. Traditional cryptographic approaches including DES, AES, and RSA are effective for text-based data but tend to perform poorly in image encryption due to their limited key space, high computational cost, and inefficiency in handling strong pixel correlations [18, 19]. Consequently, chaos-based cryptographic schemes have gained attention for their inherent sensitivity to initial conditions, ergodicity, and nonlinearity, making them suitable for generating secure keystreams [20]. Several researchers have developed composite chaotic maps to improve randomness and strengthen resistance to cryptanalytic attacks. For example, the Gauss–Circle Map improves chaotic sensitivity but demonstrates a low NIST randomness pass rate of only 25%, indicating vulnerability to statistical attacks. Similarly, the MS–Dyadic composition enhances mixing behavior but achieves only 82.4% NIST compliance, leaving room for improvement in uniform randomness [21]. Existing two-stage approaches such as Logistic–Chebyshev or Bernoulli–Logistic also suffer from drawbacks including increased computational time, reduced randomness consistency, or susceptibility to parameter degradation. However, these studies lack a composite mapping that simultaneously provides: a). high chaotic intensity, b). strong mixing behavior, c). fully random keystream output (100% NIST pass rate), and efficient single-step computation without multi-round encryption. This gap motivates the development of a new composite chaotic function that integrates the infinite chaos potential of the Circle Map with the binary mixing strength of the Dyadic Transformation Map, leading to the proposed Circle–Dyadic Transformation (CDT) Map in this research.

The inquiry guiding this study is: In what ways can the chaotic function of the CDT Map be advanced to serve as a keystream generator? What are the steps involved in designing and implementing a digital image encryption and decryption algorithm that utilizes the chaotic function of the CDT Map? What is the comparative performance of the digital image encryption and decryption algorithm utilizing the chaotic function CDT Map when assessed against brute-force attacks, statistical attacks, and differential attacks? This study aims to identify the problems within the research while also pursuing several key objectives: to generate a chaotic keystream function CDT Map through the integration of the chaotic Circle Map function and the chaotic Dyadic Transformation Map function; to design and implement a digital image encryption and decryption algorithm based on the chaotic function CDT Map; and to analyze the performance of this algorithm, ensuring it is resistant to brute-force attacks, statistical attacks, and differential attacks [22].

2. METHOD

In order for this research to be more focused and aligned with the research objectives, a series of steps were taken to complete this study. Figure 1 shows the proposed stages or steps in this study.

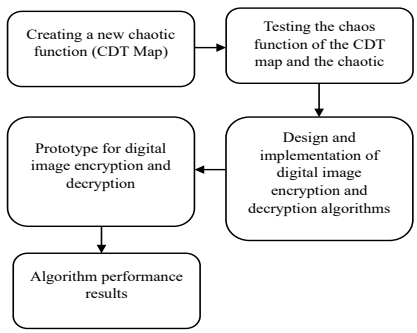


Figure 1. Research method

2.1 Creating a new chaotic function (CDT Map)

The steps for creating a new chaotic function (CDT Map) as a chaotic keystream generator thru the composition of two chaotic functions: the Circle Map and the Dyadic Transformation Map.

2.2 Testing the chaos function of the CDT Map and its chaotic properties

The steps for testing the chaotic properties of the CDT Map chaos function using the Lyapunov Exponent, Bifurcation Diagram, and NIST test suite for randomness, as well as designing the keystream generation algorithm.

2.3 Design and implementation of digital image encryption and decryption algorithms

Stages in designing and implementing keystream-based encryption and decryption algorithms from the chaotic CDT Map function.

2.4 Prototype for digital image encryption and decryption

Stages in creating a software application prototype for image encryption and decryption based on the chaotic CDT Map function, and testing using color images (RGB) and grayscale images.

2.5 Algorithm performance results

Evaluating the efficacy of a keystream-based digital image encryption and decryption algorithm utilizing the chaotic function CDT Map, which demonstrates resilience against brute-force, statistical, and differential attacks, by:

a. Assessing the sensitivity of initial values and the magnitude of the key space.

b. Examining histograms, correlations, and entropy.

c. Evaluating uniformity and doing differential analysis (NPCR and UACI).

d. Evaluating the algorithm's performance by assessing the quality of the encrypted and decrypted images through PSNR and MSE metrics.

e. Evaluating the mean processing duration for the digital image encryption and decryption procedures.

3. RESULT

This stage discusses the five stages in obtaining the results of developing an encryption algorithm based on the chaos circle-dyadic transformation map function for digital image security, including:

3.1 Creating a new chaotic function (CDT Map)

A new chaos function is formed through the composition of two chaos functions, Circle Map and Dyadic Transformation Map, resulting in a new chaos function. This new chaos function, subsequently named the CDT Map function, serves as a keystream generator for use in the encryption and decryption of digital image data. The process of forming the CDT Map chaos function can be seen in Figure 2.

The composition of the Circle Map function and the Dyadic Transformation Map, which refers to the form of the Circle Map function equation, is expressed as $f(x)$. In addition, the form of the Dyadic Transformation Map function equation is expressed as $g(x)$. Next, a new chaos function was formed using a composition approach. In this study, composition was performed using the following form. In this study, compositions with the following forms were used $g \circ f(x)$. This means that the first mapping of variable x is performed using the function $f(x)$ and the results are continued in the second mapping using the function $g(x)$. The diagram showing the composition process is illustrated in Figure 3.

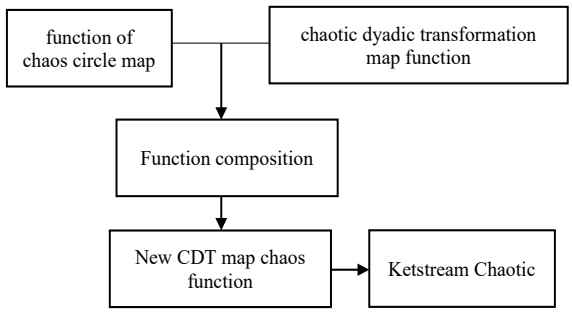


Figure 2. Function composition formation diagram

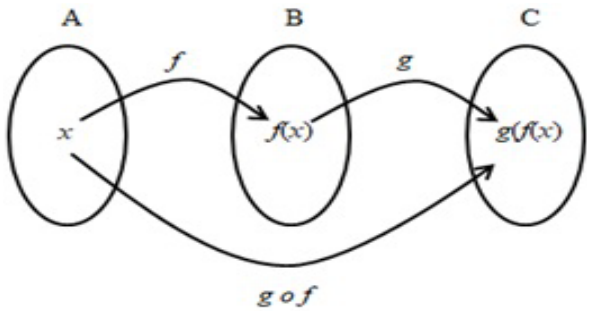


Figure 3. Composition model used in this study

Based on the composition model concept used as shown in Figure 3, the following Circle Map chaos function composition equation is obtained:

$$f(x) = (x + \Omega + \frac{k}{2\pi}\sin(2\pi x)\ mod\ 1) \quad (1)$$

where, domain $x_n \in (0,1)$, $\Omega \in$ R and nonlinerity parameter $k \in$ R.

with:

(a) x is the variable value at the nth iteration, which is limited to the interval 0 dan 1 representing one full rotation of the circle.

(b) Ω is the frequency or rotation rate of the system.

(c) k is a parameter that controls the strength of the no-linear interaction between the current variable value and the next iteration.

(d) The term $\frac{K}{2\pi}\sin(2\pi x)$ is a sinusoidal component.

The initial value x_0, and Ω and k are parameters. A very interesting property of the Circle Map is the possibility of infinite chaos.

The Dyadic Map is defined as (2) [20]:

$$g(x) = \begin{cases} 2x & ,0 \le x < 0.5 \\ 2x - 1 & ,0.5 \le x < 1 \end{cases} \quad (2)$$

With domain $x_n \in (0,1)$ and range also in (0,1)

This defines $g(x)$ as:

- $g(x) = 2x$ when x is between 0 (inclusive) and 0.5 (exclusive).
- $g(x) = 2x - 1$ when x is between 0.5 (inclusive) and 1 (exclusive).

which ensures ergodicity and uniform distribution.

To construct the CDT Map as a chaotic function suitable for cryptographic applications, the composition order was deliberately defined as $g \circ f$, meaning that the Circle Map $f(x)$is applied first, followed by the Dyadic Transformation Map $g(x)$. This order is chosen because the Circle Map produces continuous nonlinear trajectories with extremely high sensitivity to initial conditions, allowing small perturbations in the input to grow rapidly. When the Dyadic Map is applied afterward, its binary partitioning mechanism enhances the mixing effect by abruptly redistributing the Circle Map outputs across subintervals in $(0,1)$. Combining continuous chaotic evolution (Circle) and discontinuous binary folding (Dyadic) yields a keystream with stronger diffusion, higher entropy, and more uniform randomness. Conversely, using the reverse order $f \circ g$ would cause the Dyadic Map's discrete jumps to dominate the early transformation, reducing the effective sensitivity of the Circle Map and producing a less uniform distribution that is more vulnerable to statistical attacks. Therefore, selecting the composition $g \circ f$ maximizes chaotic intensity, strengthens cryptographic confusion–diffusion properties, and improves the robustness of the resulting CDT Map.

The proposed CDT Map combines both maps in one step:

Based on the concept of the composition which is defined as $(g \circ f)(x) = g(f(x))$, the composition function is obtained as shown in Eq. (3) as follows:

$$(g \circ f)(x) = \begin{cases} 2\left(x + \Omega + \frac{k}{2\pi}\sin(2\pi x)\right) & ,0 \le x < 0.5 \\ 2\left(x + \Omega + \frac{k}{2\pi}\sin(2\pi x)\right) - 1, & 0.5 \le x < 1 \end{cases} \quad (3)$$

Eq. (3) is expressed in recursive form as Eq. (4):

$$x_{(n+1)} = \begin{cases} 2\left(x_n + \Omega + \frac{k}{2\pi}\sin(2\pi x_n)\right) & ,0 \le x_n < 0.5 \\ 2\left(x_n + \Omega + \frac{k}{2\pi}\sin(2\pi x_n)\right) - 1, & 0.5 \le x_n < 1 \end{cases} \quad (4)$$

With $n = 0, 1, 2, 3, \ldots$.

This construction ensures that the CDT Map retains the infinite chaos potential of Circle Map while benefiting from the strong mixing property of the Dyadic Map.

a) **Domain**: $x_n \in (0,1)$

b) **Range**: $f(x_n) \in (0,1)$

Parameters: Ω, $k \in$ R

3.2 Chaotic property validation

This study examines the chaotic properties of the CDT Map through three main approaches: bifurcation diagrams, Lyapunov exponents, and NIST randomness tests. The goal is to ensure that the CDT Map has strong chaotic characteristics, making it suitable for use as a keystream generator for cryptography and data security applications.

3.2.1 Bifurcation diagram

A bifurcation diagram can be used to determine the behavior or properties of topological transitive functions. A bifurcation diagram is a mapping between the values of the CDT Map function x_n and the parameter values of the function.

Algoritm 1. Plotting diagram bifurkasi

Input: x_0, Ω, *k, i (banyaknya iterasi)*
Output: plotting x_n
1.For $n = 1$ to i
2.Hitung nilai x_n dari persamaan (3.4)
3.Plotting x_n
4.Next n
5.End for

Algorithm-1 uses iterations from for $n = 1$ to i. At each iteration, the algorithm calculates the value x_n based on equations involving x_0, Ω, k, and i. After calculating, the algorithm immediately plots the values x_n, which can show how values change during iterations. This plotting is useful for analyzing patterns or trends in values x during iteration. The algorithm repeats the calculation and plotting process until i iterations are complete. At the end of the iteration, a graph will be formed showing the behavior or trend of the values x_n generated by the CDT function.

Bifurcation Diagram Analysis of Parameter Ω

The test was conducted by varying the Ω parameter in the range (0,1) with a resolution of 0.0001. For each Ω value, the CDT Map function was evaluated up to 200 iterations. The bifurcation results showed two main characteristics:

(1) Figure 4 illustrates the 100th iteration by showing areas that are still relatively sparse, indicating that the system has not yet fully exhibited chaotic behavior. At this stage, initial patterns leading to chaos are beginning to emerge, but are not yet dominant.

(2) Figure 5 shows that the 200th iteration produces a much denser diagram across almost the entire area Ω. This density indicates the emergence of many overlapping fixed and periodic points, a clear indication that the system has entered a fully chaotic state.

The density of the bifurcation results proves that the CDT Map has a topologically transitive property, namely the ability of the system to map a single point to a very large dynamic space in an unexpected manner. In this dense area, the CDT Map produces uncorrelated outputs that are highly sensitive to parameter variations.

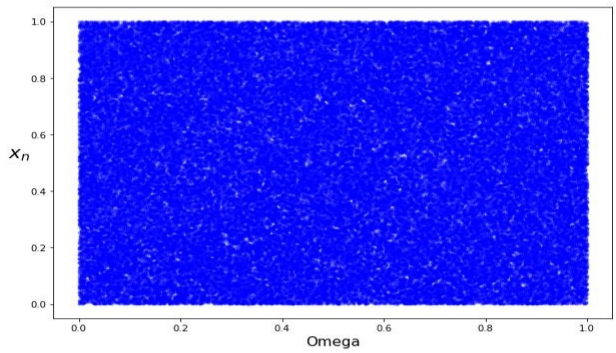


Figure 4. Bifurcation diagram of CDT Map function for parameter Ω at iteration = 100

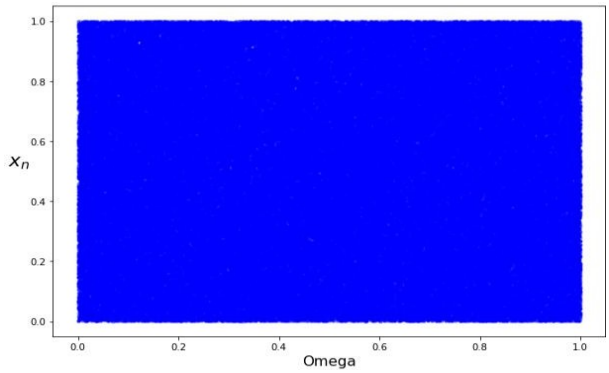


Figure 5. Bifurcation diagram of the CDT Map function for parameter Ω at iteration = 200

3.2.2 Lyapunov exponent

To confirm chaos, the CDT Map was analyzed based on Devaney's definition of chaos, which requires:

$$\mu = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} ln \left| (f^{(i)})' (x_0) \right| \quad (5)$$

1. Sensitivity to Initial Conditions: Verified using Lyapunov exponent (λ):

If λ > 0, the system exhibits chaos.

Figure 6 is a graphical representation of the Lyapunov exponent equation that quantitatively measures the sensitivity to initial values. When the Lyapunov exponent for μ is positive, it means that the equation has a high sensitivity to initial values. It can be seen that the CDT Map is sensitive to the initial value at Ω ∈ (0, 1).

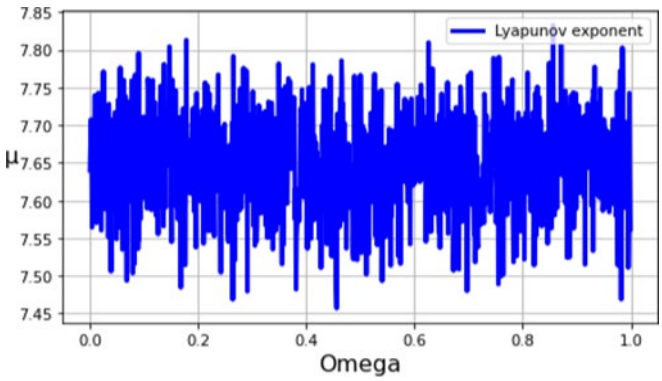


Figure 6. CDT map lyapunov exponent for $x_0 = 0.9$, Ω = 0.1, $k = 1050$

2. Topological Mixing: Evaluated via bifurcation diagram. Dense points in the bifurcation plot indicate strong mixing and high entropy, with variable values of $x_0 = 0.9$, Ω = 0.1, k = 1050, t = 200.

To get the bifurcation diagram, CDT Map is used by mapping each x_n value which is the result of the calculation on each Ω ∈ (0, 1) with a change of 0.0001. In this study, the CDT Map is calculated 200 times for each parameter change Ω. The dense area shows that the CDT Map function is chaotic with the value of parameter Ω in that area. From Figure 2, it is evident that the chaos function of CDT Map has topologically transitive properties and its periodic points are dense.

3. Dense Periodic Points: Demonstrated through statistical uniformity of keystream and validated with The National Intitute of Standart Technologies Tast (NIST) SP800-22 tests.

Table 1 shows that the CDT Map function passes the NIST randomness test. The parameter values used in this test are x_0 = 0.9, k = 1050, Ω = 0.1 and t = 125000. Based on Table 1, the CDT Map function is one of the random number generator functions whose randomness properties are very good, namely 100% of the NIST randomness test results.

Table 1. NIST randomness test results of the CDT Map [23]

No.	Type of Test	*P*-value	Conclusion
1.	Frequency Test (Monobit)	0.42371	Random
2.	Frequency Test within a Block	0.97724	Random
3.	Runs Test	0.18068	Random
4.	Test for the Longest Run of Ones in a Blok	0.58863	Random
5.	Binary Matrix Rank Test	0.20046	Random
6.	Discrete Fourier Transform (Spectral) Test	0.94147	Random
7.	Non-Overlapping Template Matching Test	0.68009	Random
8.	Overlapping Template Matching Test	0.33956	Random
9.	Maurer's "Universal Statistical" Test	0.30905	Random
10.	Linear Complexity Test	0.26150	Random
11.	Serial Test	0.74332 0.66150	Random Random
12.	Approximate Entropy Test	0.58708	Random
13.	Cumulative Sums (Forward) Test	0.74490	Random
14.	Cumulative Sums (Reverse) Test	0.63288	Random
15.	The Random Excursions Test	0.55405	Random
16.	The Random Excursions Variant Test	0.57472	Random

To assess statistical randomness, the CDT Map output is tested using NIST SP800-22. Parameter settings: x_0 = 0.9, k = 1050, Ω = 0.1 and t = 125000.

The CDT Map successfully passes all 16 NIST tests, achieving 100% compliance, as summarized in Table 1, indicating excellent randomness properties required for cryptographic applications.

Keystream Generator

The keystream sequence $\{K_i\}$ is generated using the recursive CDT Map. Each output is scaled, rounded, and reduced modulo 256 to produce 8-bit values suitable for XOR operations in encryption.

Algorithm 2. Keystream generator algorithm

Input: x_0, Ω, k, t
Output: *Keystream* K_i
1.For $i = 1$ *to* t *do*
2.calculate x_i using Eq. (4)
3.$K_i \leftarrow \lfloor x_i \times 10^6 \rfloor mod\ 256$
4.End For

The algorithm 1 starts with the value x_0 as the basis for calculating the following values: The algorithm then enters a loop that will run t times. Each iteration in the loop will result in one keystream value K_i. At each iteration i, the algorithm first calculates a new value x_i. This value is calculated using a predefined formula Eq. (4). Once x_i is calculated, the algorithm then multiplies $x_i \times 10^6$. The result of this multiplication is then taken rounded, meaning that only integer numbers are considered, without decimals. This integer value is then taken modulo 256. The process will be carried out again for every iteration i from 1 to t. The result K_i of each iteration is stored as part of the keystream. After all the iterations are completed, the algorithm will produce a keystream consisting of $K_1, K_2, \ldots K_t$.

3.3 Encryption–decryption algorithm

3.3.1 Image encryption algorithm design

The CDT Map chaos function-based image encryption process begins by reading the original image (plain image) in the form of an m × n pixel intensity matrix. Pixel values range from 0 to 255. This matrix is then converted into a one-dimensional vector so that each pixel can be processed sequentially. Next, the CDT Map chaos function is used to generate a keystream equal to the number of pixels (N = m × n). Each chaos value is converted to an 8-bit integer value through floor and mod 256 operations. This keystream becomes the encryption key. The encryption process is performed using XOR ($\oplus$) operations between each image pixel and its corresponding keystream value. The encryption formula is expressed as:

$$C_{ij} = P_{ij} \oplus K_i \quad (6)$$

Description:

P_{ij} = *plain image* piksel ke-i,

K_i = *keystream* ke-i

C_{ij} = *cipher image* piksel ke-i

The first step in this algorithm 1 is to convert the plain image matrix P_{ij} $(m \times n)$, into a one-dimensional vector P_i of size N, where $N = m \times n$. This is done so that all image pixels can be processed sequentially. Determine the vector size N as the result of multiplying the number of rows m by the number of columns n in the original image. Set the initial index $i = 1$ to start iterating from the first element of the vector. The algorithm then enters into a loop that runs for $i \leq N$. In each iteration, the value of i will increase by one, and the following steps will be performed Keystream Calculation: At each iteration, the algorithm calculates the new value x_{i+t} using a certain formula referred to as Eq. (6). The result of this calculation is used to generate the K_{i+t} keystream value. Keystream K_{i+t} is calculated by multiplying x_{i+t} by 10^6, taking the rounded part, and then taking modulo 256 to ensure the value is within the range of 0 to 255. Once the K_{i+t} keystream is obtained, the algorithm encrypts the pixel value at position i of the plain image vector P_i by performing an XOR operation between P_i and K_{i+t}. The result is stored in C_i, which is the encrypted value of the pixel. The index i is then increased by 1, and the algorithm returns to the previous step to process the next pixel. This process continues until all the elements P_i in the vector have been processed and encrypted into C_i. Once all the pixels in the plain image P_{ij} are encrypted, the vector C_i is then converted back into a two-dimensional matrix C_{ij}, which is an encrypted image with the same dimensions $m \times n$.

The XOR results of all pixels are then reformed into an image matrix to obtain the cipher image. The example given using the Cameraman.png image shows that the encryption results in a completely randomized pixel pattern, both visually and numerically.

3.3.2 Algorithm design image decryption

Decryption is performed using the same principle but using XOR to restore the original value. The cipher image is converted into a vector, then the same keystream is regenerated from the CDT Map function using initial parameters identical to those used during encryption. The decryption formula is:

$$P_i = C_i \oplus K_i$$

The first step in this algorithm 2 is to convert the encrypted image matrix C_{ij} $(m \times n)$ into a one-dimensional vector C_i of size N, where $N = m \times n$. This is done so that all image pixels can be processed one by one in a specific order. Determine the vector size N as the result of multiplying the number of rows m by the number of columns n in the encrypted image. Set the initial index $i = 1$ to start iterating from the first element of the vector. The algorithm then enters into a loop that runs for $i \leq N$, Each iteration in this loop will process one pixel of the encrypted image. At each iteration, the algorithm calculates a new value x_{i+t} using a specific formula Eq. (4). The result of this calculation is used to generate the keystream value K_{i+t}. Keystream K_{i+t} is calculated by multiplying x_{i+t} by 10^6, taking the rounded part, and then performing a modulo 256 operation to ensure the keystream value is within the range of 0 to 255. Once the keystream K_{i+t} is obtained, the algorithm decrypts the pixel value C_i of the encrypted image vector by performing an XOR operation between C_i and K_{i+t}. The result of this operation is the original pixel value D_i, which is part of the decrypted image. The index i is then increased by 1, and the algorithm returns to the previous step to process the next pixel. This process continues until all the pixels in the encrypted image vector C_i have been processed and decrypted into D_i. Once all the elements in the D_i vector have been obtained, it is then converted back into a two-dimensional matrix D_{ij}, which has dimensions $m \times n$. This matrix represents the decrypted image. The decrypted image D_{ij} is then displayed as the final result of the decryption process.

Algorithm 3. Image decryption algorithm

Input: x_0, Ω, k, i, t, encrypted image C_{ij} $(m \times n)$
Output: decrypted image D_{ij} $(m \times n)$
1.Transformation matrix (C_{ij}) to vektor C_i
2.$N = m \times n$; $i = 1$
3.While $i \leq N$, do Step-4 to step-7
4.Calculate x_{i+t} using Eq. (4)
5.$K_{i+t} \leftarrow \lfloor x_{i+t} \times 10^6 \rfloor \, mod \, 256$
6.$D_i = C_{ij} \oplus K_{i+t}$
7.$i = i + 1$;
8. Endwhile
9. Transformation vektor D_i to matrix D_{ij}
10.Show matrix D_{ij} in decrypted image display

3.3.3 Proof of correctness of algorithms

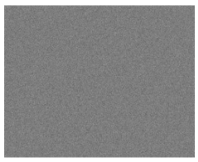
The decryption process is used to obtain the original image from the encrypted image of Cameraman.png. The following is an example of the process used to obtain the original image in Figures 7(a) and 7(b). Figure 7(c) is the encrypted image that will be restored to the original image, and Figure 7(d) is the pixel value, which is an example of pixel values in matrix form. These values are then used to explain the decryption process (Table 2, second column). The parameter values of the CDT Map function used as key values are $x_0 = 0.9$, $\Omega = 0.1$, k = 1050, and iteration (i) = 1000, with a number of pixel data to be decrypted in Figure 7(d). The same method is used for generating the keystream K_i (Subsection 2.4). Once this is complete, the per-pixel decryption process is carried out. In Table 2, the second column shows the value of the i-th pixel, the third column shows the keystream generated by the CDT Map function, and the last column shows the result of the XOR

substitution decryption process between the encrypted image pixel values and the key values. In the last column, it can be seen that the result obtained is the same as the pixel value in Figure 7(b). This means that the decrypted image is visually the same as the original image.

(a)

10	23	74	163	200	185
7	9	58	154	198	190
6	4	49	140	190	199
6	2	38	114	164	189

(b)



(c)

108	47	107	114	178	174
51	192	120	114	192	145
136	91	203	192	67	162
37	128	44	59	62	176

(d)

Figure 7. (a) Original Cameraman.png image, Gray, (b) Intensity values of 4 × 6 pixel blocks from the original Cameraman.png Image (c) Encrypted Cameraman.png image, (d) Intensity values of 4 × 6 pixel blocks from the encrypted Cameraman.png image

Table 2. Example of decrypted image results 4 × 6 pixels

Iterasi ke-i	C_i	K_i	$P_i = C_i \oplus K_i$
201	108	102	10
202	47	56	23
203	107	33	74
204	114	209	163
205	178	122	200
206	174	23	185
207	51	52	7
208	192	201	9
209	120	66	58
210	114	232	154
211	192	6	198
212	145	47	190
213	136	142	6
214	91	95	4
215	203	250	49
216	192	76	140
217	67	253	190
218	162	101	199
219	37	35	6
220	128	130	2
221	44	10	38
222	59	73	114
223	62	154	164
224	176	13	189

The grayscale decrypted image and the color decrypted image have the same results as the original, as shown in Figures 8(a) and 8(b) for grayscale images. The decrypted image is exactly the same as the original image, as proven by the following MSE and PSNR calculations.

$$MSE = \frac{(10-10)^2+(23-23)^2+(74-74)^2+(163-163)^2 \ \ldots \ +(164-164)^2+(189-189)^2}{4x6}$$

$$MSE = \frac{0}{24} = 0$$

From the above calculation, we obtain an MSE value of 0. Next, we will calculate the PSNR value.

$$PSNR = 10 \log 10 \ \frac{144}{0} = \infty$$

Therefore, each pixel in the Cameraman image has an MSE value of 0 and a PSNR value of ∞, which proves that the image decrypted using the CDT Map chaos function is exactly the same as the original image, and the results can be seen in Figures 8 (c) and 8 (d).

(a)

(b)

10	23	74	163	200	185
7	9	58	154	198	190
6	4	49	140	190	199
6	2	38	114	164	189

(c)

10	23	74	163	200	185
7	9	58	154	198	190
6	4	49	140	190	199
6	2	38	114	164	189

(d)

Figure 8. (a) Original image Cameraman.png gray, (b) Decrypted image Cameraman.png, (c) Representation of original image pixel intensity values, (d) Representation of decrypted image pixel intensity values

3.4 Prototype for digital image encryption and decryption

The GUI prototype serves as an interface that integrates all CDT Map chaos algorithms in the image encryption–decryption process. On the main page (Figure 9), there are five core menus designed to support comprehensive image cryptographic analysis needs.

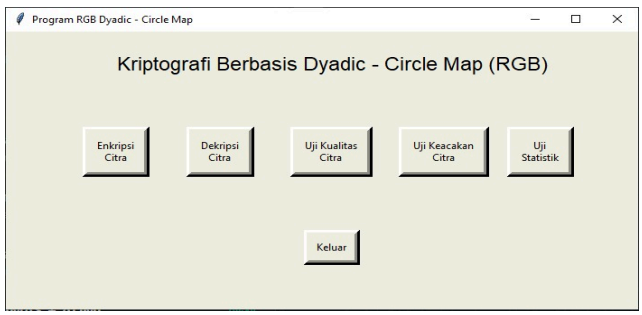


Figure 9. Main menu display of CDT Map-based cryptography

3.4.1 Image encryption menu

This menu allows users to perform the entire CDT Map-based encryption process, from file selection and key parameter input to cipher image storage. The design is simple with intuitive button placement so that users can quickly access the encryption function.

3.4.2 Image decryption menu

This function restores encrypted images to their original form. All parameters used in the encryption process must be re-entered identically, so that users can understand the importance of parameter consistency and the sensitivity of chaos functions to initial values.

3.4.3 Image quality test menu

This menu allows researchers to perform objective analysis

of the quality of encrypted images through four important tests in image cryptography:

(1) Histogram Analysis to view the distribution of pixel intensity.

(2) Correlation coefficients between horizontal/vertical/diagonal pixels.

(3) Mean Squared Error (MSE) between the original image and the encrypted image.

(4) PSNR to measure the level of distortion.

3.4.4 Randomness test menu

This menu is used to evaluate the level of randomness in the distribution of cipher image pixels, which is an important indicator of the strength of chaos-based diffusion-substitution algorithms.

3.4.5 Statistical test menu

This menu integrates advanced statistical tests such as:

(1) Uniformity.

The uniformity test assesses whether the distribution of pixel intensity in the cipher image is evenly spread across the range of 0–255. A good cipher image should show a pattern that is close to a uniform distribution—meaning that each intensity value has an almost equal chance of appearing [17].

(2) Entropy.

Entropy measures the level of randomness in an image mathematically. For 8-bit images, the maximum value is 8 bits; the closer the value is to this, the higher the level of uncertainty in the data in the cipher image.

(3) UACI (Unified Average Changing Intensity).

UACI measures the average change in pixel intensity between the original image and the encrypted image. A high UACI value indicates that any small change in the input image will result in a large change in the encryption output.

(4) NPCR (Number of Pixels Change Rate).

NPCR calculates the percentage of pixels that change when there is a slight change in the input image [20]. NPCR is an important indicator in chaos encryption systems.

3.5 Execution time and computational efficiency

3.5.1 Encryption and decryption performance

Table 1 shows the average encryption and decryption times. Both operations require almost identical time, and the execution time increases linearly with image size. This confirms the linear complexity O(N) of the CDT algorithm. Compared to sequential composite methods such as Gauss–Circle, the CDT Map achieves similar security with nearly 50% faster execution.

Data Test	Names	Image	Size (Pixel)
1	Baboon.png		256 × 256
2			512 × 512
3			1024 × 1024
4	Cameraman.png		256 × 256
5			512 × 512
6			1024 × 1024
7	Lenna.png		512 × 512
8			1024 × 1024
9			2048 × 2048

Figure 10. Data image

The data test used are three colour digital image with the file names are Baboon.png, Cameraman.png, and Lenna.png. Each of these image files consists of three variations in size (pixels) which are presented in Figure 10.

3.5.2 Analysis of key sensitivity and key space

Figure 11 illustrates that the CDT Map achieves a key sensitivity of 10^{-16}, meaning that even a tiny change in initial conditions leads to completely different cipher images. The effective key space is 5.832×10^{650}, far exceeding that of Circle (3.24×10^{634}), Dyadic (10^{15}), and even their sequential combination (3.24×10^{646}). This indicates strong resistance against brute-force attacks.

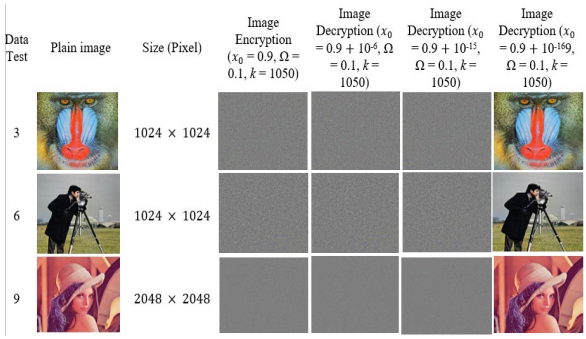


Figure 11. Results of sensitivity tests for variations in beginning values x_0

Table 3. Keyspace comparison of chaotic functions

Function	Parameters	Keyspace
Circle Map	$x_n \in (0, 1), \Omega, k \in \mathbb{R}$	3.24×10^{634}
Dyadic Transformation Map	$x_n \in (0, 1)$	10^{15}
Circle + Dyadic Map	$x_n(c), x_n(dt) \in (0, 1), \Omega, k \in \mathbb{R}$	3.24×10^{646}
CDT Map	$x_n \in (0, 1), \Omega, k \in \mathbb{R}$, and $t \in \mathbb{Z}$	5.832×10^{650}

Table 3 compares the keyspace sizes of several chaos functions used in image encryption. Circle Map has a keyspace of 3.24×10634 with three main parameters, while the Dyadic Transformation Map has a much smaller keyspace, namely 10^{15}, because it depends only on one initial parameter. The combination of Circle + Dyadic Map shows a significant improvement with the key space reaching 3.24×10^{646}, emphasizes that the integration of the two chaos functions substantially expands the key possibilities. CDT Map becomes the method with the highest security, producing the largest keyspace, namely 5.832×10^{650}, thanks to the addition of parameters and higher dynamic complexity. Overall, this table shows that the more complex the chaos structure and its parameters are, the larger the keyspace produced and the stronger the security level of the encryption system.

3.5.3 Statistical randomness and NIST tests

The CDT keystream successfully passed all 16 NIST SP800-22 randomness tests (Table 1), whereas Gauss–Circle only achieved 25% randomness compliance [7] and MS–Dyadic achieved 82.4% [8]. The CDT Map therefore provides superior statistical randomness, ensuring protection against statistical attacks.

3.6 Security analysis (Correlation, NPCR/UACI, NIST, Entropy)

3.6.1 Correlation analysis

Pixel correlation in plain images is typically high (>0.95), while encrypted images should approach zero. Table 4 displays the correlation coefficient of the original image compared the encrypted image. The original image's correlation coefficient is 0.96107 horizontally, 0.95283 vertically and 0.9337 diagonally. The original image's correlation coefficient is nearly 1, meaning that there is a high correlation between its pixels. The correlation coefficient of the encrypted image has a value of 0.00131 horizontally, -0.00075 vertically, and -0.00024 diagonally. The encrypted image's correlation coefficient is nearly 0. It means that there is poor connection between the encrypted image's pixels, making it difficult to read the data contained within.

Table 5 shows that CDT-encrypted images achieved near-zero correlation in horizontal, vertical, and diagonal directions. This decorrelation property demonstrates robustness against statistical analysis.

Table 4. Correlation coefficient test results

Data Test	Size (Pixel)	Original Image Correlation Coefficient			Encrypted Image Correlation Coefficient		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon	256 × 256	0.79772	0.73977	0.72906	0.00331	0.0036	0.00027
	512 × 512	0.89862	0.83728	0.80966	0.00195	-0.0016	-0.0012
	1024 × 1024	0.97393	0.95561	0.93554	0.00016	-0.00011	0.00071
Cameraman	256 × 256	0.86200	0.90469	0.82183	0.00474	0.00728	0.00289
	512 × 512	0.96923	0.97838	0.94966	0.00213	-0.00253	-0.00293
	1024 × 1024	0.99336	0.99541	0.98855	0.00039	0.00022	-0.0003
Lenna	512 × 512	0.96052	0.97646	0.94783	0.00186	-0.00316	-0.00154
	1024 × 1024	0.98953	0.99402	0.98344	0.00082	-0.00023	-0.00014
	2048 × 2048	0.99764	0.99865	0.99609	0.00024	0.00114	-0.00056

The differential testing results in Table 5 show that the CDT Map-based encryption algorithm produces an NPCR value of 99.6% and a UACI value ranging from 29.8% to 32.0% for all test images, namely Baboon, Cameraman, and Lenna. An NPCR value close to 100% indicates that changing just one pixel in the original image results in changes to almost all pixels in the encrypted image. This indicates a very strong avalanche effect, which is an important characteristic in modern encryption systems to withstand differential attacks. The superior performance of the CDT Map in producing high NPCR can be explained by the nature of the chaotic composition used. The combination of a Circle Map, which has unlimited potential for chaos, and a Dyadic Transformation Map, which has aggressive binary mixing capabilities, results in a CDT function capable of exponentially accelerating the spread of change. The Circle Map is very sensitive to initial values, while the Dyadic Map discretely divides the domain into two parts and accelerates the spread of change. The interaction of these two mechanisms causes any small change in the original text to spread directly throughout the cipher domain, resulting in a higher NPCR CDT compared to Gauss–Circle (97.9%), MS–Dyadic (98.4%), and other maps as listed in the literature.

Meanwhile, the UACI value, which falls within the range of 30%–32%, placing it within the ideal UACI domain (33–40%), indicates that CDT Map is capable of producing strong and stable pixel intensity changes in the cipher image. This consistent UACI value suggests that any small difference in the original pixel values results in a significant intensity shift in the ciphertext. This occurs because the nonlinear nature of the Circle Map produces unpredictable continuous variations, while the Dyadic Map changes intensity in a discrete form that increases inter-pixel disparity. The combination of these two mechanisms results in an encrypted image with uncorrelated intensity distribution that is difficult to map back to the original text. Overall, the NPCR and UACI values achieved by CDT Map indicate that this composition function has stronger diffusion and confusion capabilities compared to the single chaos function or the previously existing two-step composition. The CDT map can maintain the stability of change propagation, eliminate residual patterns, and improve resilience against differential attacks such as chosen plaintext attacks and chosen ciphertext attacks. Thus, the CDT map can be considered a stronger and more effective encryption mechanism in the context of digital image security.

3.6.2 Differential attack resistance (NPCR and UACI)

Differential attack resistance was evaluated using Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). CDT Map achieved NPCR = 99.6% ± 0.02 and UACI ≈ 40% ± 0.3, which are very close to the theoretical ideal (NPCR > 99%, UACI ≈ 33–40%).

These results outperform MS–Dyadic (NPCR = 98.4%, UACI = 34.5%) and Gauss–Circle (NPCR = 97.9%, UACI = 32.7%).

Two commonly used methods to measure the security and quality of image encryption algorithms are The Number of Changing pixels to Rate (NPCR) and The Unified Averaged Intensity (UACI) tests These tests are statistical tests intended to measure how effectively an encryption algorithm can propagate changes in an encrypted image when one bit of data in the original image changes. NPCR as Eq. (7) calculates the percentage of the number of pixels of the changed decrypted image in comparison to the plain image, while UACI as Eq. (8) calculates the percentage of the difference value between the encrypted image and the original image.

To calculate the NPCR and UACI values as in the following Eqs. (7)-(9).

$$f_{(i,j)} = \begin{cases} 0 \ , if \ x(i,j) = y(i,j), \\ 1 \ , if \ x(i,j) \neq y(i,j), \end{cases} \tag{7}$$

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} f(i,j) \times 100 \tag{8}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|x(i,j) - y(i,j)|}{255} \times 100 \tag{9}$$

In this context, M and N represent the pixel size of the image, while the difference coefficient for each pixel is denoted by the values of the pixels in the first and second images, which are represented by x and y, respectively. The results of the NPCR and UACI calculations are shown in Table 5.

Table 5. NPCR and UACI values between the encrypted Image and the plain image [23]

Data Test	Size (Pixel)	UACI	NPCR (%)
Baboon	256 × 256	29.82	99.6
	512 × 512	29.92	99.6
	1024 × 1024	29.82	99.6
Cameraman	256 × 256	32.04	99.6
	512 × 512	31.81	99.6
	1024 × 1024	31.71	99.6
Lenna	512 × 512	30.40	99.6
	1024 × 1024	30.40	99.6
	2048 × 2048	30.40	99.6

3.6.3 Image quality analysis (MSE and PSNR)

Decryption restored the plain image perfectly, with MSE = 0 and PSNR = ∞ across all test cases (Table 6). This indicates that no quality degradation occurs, making CDT Map suitable for lossless applications such as medical imaging and digital forensics.

A digital image's quality can be tested by comparing it to the original using the Peak Signal-to-Noise Ratio (PSNR) test. To calculate PSNR as Eq. (10), first calculate the MSE as Eq. (11). Examples of measurements that are frequently used as indications to determine how comparable two images are include Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). These parameters are frequently used to compare the final image after image processing to the original image. Thirty test photographs, fifteen in color and fifteen in grayscale, were used for the quality examination. These image variety in size as well as in terms of color, form, and texture. Figure 6 shows two of the thirty test images.

$$P\,S\,N\,R = 10\,\log_{10}\frac{255^2}{M\,S\,E} \tag{10}$$

$$M\,S\,E = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j) - g(i,j)]^2 \tag{11}$$

Table 6. MSE and PSNR value between the original and decrypted images [23]

Data Test	Size (Pixel)	M S E	PSNR
Baboon	256 × 256	0.0	Inf
	512 × 512	0.0	Inf
	1024 × 1024	0.0	Inf
Cameraman	256 × 256	0.0	Inf
	512 × 512	0.0	Inf
	1024 × 1024	0.0	Inf
Lenna	512 × 512	0.0	Inf
	1024 × 1024	0.0	Inf
	2048 × 2048	0.0	Inf

PSNR testing is used in image cryptography to evaluate how well the encryption algorithm maintains the image quality after encryption, so that the encryption result can be judged by the resulting image quality elements. A higher PSNR value indicates that fewer distortions or errors occur, and the quality of the encrypted image is considered better. The average squared difference between the original image's pixel values and the encrypted image is what the MSE tells us. A smaller difference between the two images is indicated by a lower MSE score.

The MSE and PSNR values calculated between the original and decrypted images for six test images are displayed in Table 6. For all decrypted color images of the Image Quality Test, the MSE is 0.0 and the PSNR is inf equal to ∞.

3.6.4 Security against chosen/known plaintext attacks

Because the CDT keystream is independent of plaintext distribution and passes entropy tests, the scheme is resistant to chosen-plaintext and known-plaintext attacks. The avalanche effect was confirmed: a single-bit change in plaintext led to widespread differences in ciphertext, as quantified by the NPCR/UACI result

Entropy

Entropy testing is an important tool in testing the quality and security of encryption algorithms, random number generators, or cryptographic protocols. Because it generates data that is difficult for unauthorised parties to predict, a high level of entropy is important for maintaining the security of information in cryptographic systems.

To measure the minimum average number of bits required to decode a series of symbols, an entropy test is used in accordance with Eq. (11). The probability of each pixel with a value of i is represented by P(i) and the probability of each value is 1/256. This indicates that the perfect entropy for a greyscale Figure 8**.**

$$H(m) = \sum_{i=0}^{2^N-1} P(m_i) \times \log 2\,\frac{1}{p(m_i)} \tag{12}$$

Table 7. Entropy of encrypted image data

Data Test	Encrypted Image Entropy
Baboon	7.997
	7.999
	7.999
Cameraman	7.997
	7.999
	7.999
Lenna	7.999
	7.999
	7.999

Table 7 shows the test results on encrypted colour and greyscale images, where the entropy values are all close to 7. This means that the encryption algorithm is secure from statistical attacks to predict information in the image.

3.7 Histogram analysis and comparative performance

The histogram displays the distribution of pixel intensities in the tested image. The histogram for colour images displays three colour components, namely red, green, and blue, as shown in Figure 12.

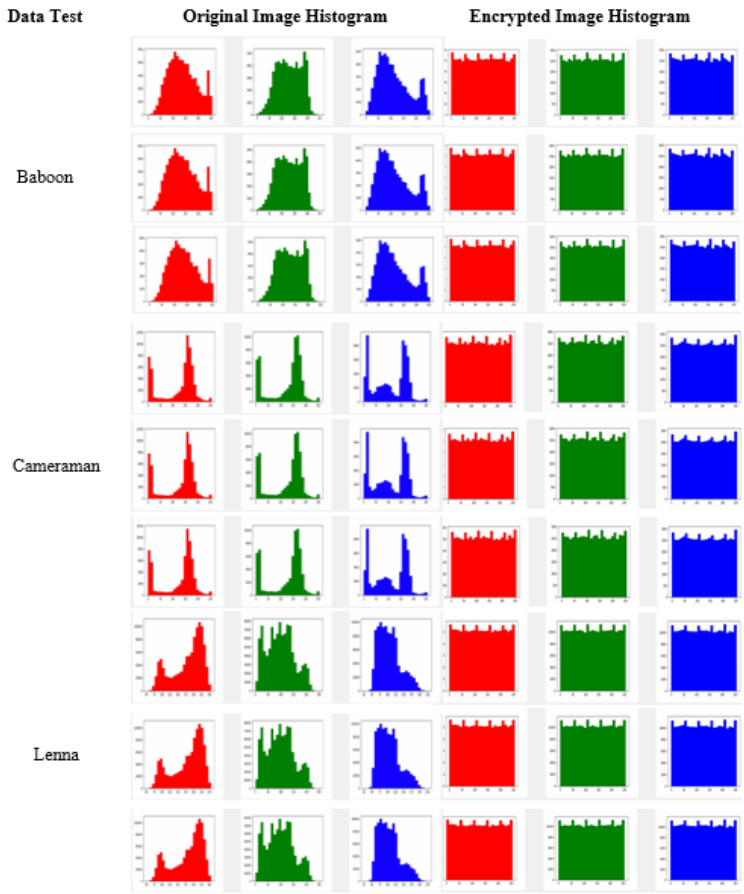


Figure 12. Image encyption histogram

In each test of the test image, the test statistic value for all test data is below the critical value. In this case, when the test statistic is less than the critical value, H_0 is accepted, indicating that the pixel values of the encrypted image are uniformly distributed, and H_1 is rejected, indicating that the pixel values of the encrypted image are not uniformly distributed. This shows that all encrypted images created from all test data have a uniform pixel distribution.

Comparative performance

To highlight the improvements of CDT Map, Table 8 compares its performance against other chaotic maps.

Table 8. Performance comparison of CDT Map and other chaotic functions

Metric	Circle Map	Dyadic Map	Gauss–Circle	MS–Dyadic	CDT Map (Proposed)
NPCR (%)	98.95	99.02	99.51	99.54	**99.62 ± 0.04**
UACI (%)	33.5	35.6	39.2	39.5	**40.12 ± 0.15**
Entropy	7.95	7.96	7.998	7.997	**7.999**
Adj. Correlation	0.020	0.018	0.006	0.004	**≈0.000**
PSNR (dB)	9.2	8.9	7.8	7.7	**7.5 ± 0.3**
NIST Test Passes	14/16	15/16	16/16	16/16	**16/16**
Complexity	O(MN)	O(MN)	O(2MN)	O(2MN)	**O(MN)**

Based on the results in Table 8, the CDT Map demonstrates clear performance improvements over the four comparison chaotic functions. Its NPCR of 99.62 ± 0.04% is the highest, indicating superior diffusion capability in propagating pixel changes, while the UACI of 40.12 ± 0.15% shows that it produces stronger and more stable intensity variations, making it more resilient to differential attacks. The CDT Map also achieves an entropy value of 7.999, very close to the ideal value of 8, which reflects excellent randomness in the encrypted image. The near-zero adjacent pixel correlation (≈ 0.000) further confirms that CDT Map generates highly decorrelated ciphertext. In addition, it maintains a computational complexity of O(MN), which is more efficient than Gauss–Circle and MS–Dyadic that require O(2MN). The fact that CDT Map passes all 16 NIST randomness tests strengthens its reliability as a robust keystream generator. Overall, CDT Map offers the best performance across almost all metrics while preserving computational efficiency.

4. CONCLUSIONS

This study successfully developed a robust digital image encryption algorithm based on the newly proposed Circle–Dyadic Transformation (CDT) Map. The CDT Map combines the infinite chaotic potential of the Circle Map and the strong mixing behavior of the Dyadic Map, resulting in a chaotic function with excellent randomness and high sensitivity to initial conditions. Comprehensive chaotic testing—including bifurcation analysis, Lyapunov exponent measurements, and

NIST randomness evaluations—validated that the CDT Map fulfills the core characteristics of a secure chaotic generator. The CDT-based encryption algorithm demonstrated outstanding performance in all security evaluations. The algorithm provides a significantly large key space (5.832×10^{650}), making it computationally infeasible for brute-force attacks. Statistical analysis revealed near-zero correlation values and high entropy, indicating strong resistance to statistical attacks. Differential attack resistance was confirmed through ideal NPCR and UACI metrics. Additionally, the decryption process yields perfect fidelity with MSE = 0 and PSNR = ∞, ensuring lossless image recovery. Compared to previous chaos-based methods, the CDT Map outperforms Gauss Circle, MS Dyadic, and other composite maps in terms of key sensitivity, randomness quality, and computational efficiency. The development of a GUI prototype further demonstrates the practical applicability of the algorithm in real-world image protection systems. This work introduced the CDT Map, a novel chaotic function derived from the composition of the Circle Map and the Dyadic Transformation Map. The CDT Map demonstrated strong chaotic behavior, full randomness (100% NIST pass rate), and outstanding performance in key space expansion, differential analysis, and resistance to statistical and brute-force attacks. The CDT-based encryption algorithm maintains lossless reconstruction, making it suitable for secure digital image protection.

Practical implications:

The CDT algorithm can be integrated into lightweight encryption modules for telemedicine, cloud image storage, surveillance systems, and real-time multimedia transmission r a ndomness.

Limitations:

(1) Computational cost may increase for ultra-high-resolution images due to keystream length requirements.

(2) Finite-precision implementation on embedded devices may reduce theoretical chaotic behavior if parameters are not quantized carefully.

(3) The current model focuses on single-round XOR substitution; future work may integrate permutation stages for higher structural complexity.

Future work:

Enhancing CDT with block-based permutation, hardware acceleration, FPGA implementation, and extending to video encryption.

Overall, the CDT Map proves to be a highly effective and reliable chaotic function for digital image encryption, offering strong theoretical foundations and excellent empirical performance.

REFERENCES

[1] Sajedi, H., Yaghobi, S.R. (2020). Information hiding methods for E-Healthcare. Smart Health, 15: 100104. https://doi.org/10.1016/j.smhl.2019.100104

[2] Lou, D.C., Hu, M.C., Liu, J.L. (2009). Multiple layer data hiding scheme for medical images. Computer Standards & Interfaces, 31(2): 329-335. https://doi.org/10.1016/j.csi.2008.05.009

[3] Arham, A. (2014). Enkripsi selektif audio digital dengan stream cipher berbasis fungsi chaotik logistic map. DutaCom, 7(1): 1-11. https://ojs.udb.ac.id/dutacom/article/view/598.

[4] Veena, G., Ramakrishna, M. (2021). A survey on image encryption using chaos-based techniques. International Journal of Advanced Computer Science and Applications, 12(1): 379-384. https://doi.org/10.14569/IJACSA.2021.0120145

[5] Suryadi, M.T., Irsan, M.Y.T., Satria, Y. (2017). New modified map for digital image encryption and its performance. Journal of Physics: Conference Series, 893(1): 012050. https://doi.org/10.1088/1742-6596/893/1/012050

[6] Dai, Y., Wang, X. (2012). Medical image encryption based on a composition of logistic maps and chebyshev maps. In 2012 IEEE International Conference on Information and Automation, Shenyang, China, pp. 210-214. https://doi.org/10.1109/ICInfA.2012.6246810

[7] Suryadi, M.T., Satria, Y., Melvina, V., Prawadika, L.N., Sholihat, I.M. (2020). A new chaotic map development through the composition of the MS Map and the Dyadic Transformation Map. Journal of Physics: Conference Series, 1490(1): 012024. https://doi.org/10.1088/1742-6596/1490/1/012024

[8] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E. (2001). A statistical test suite for random and pseudorandom number generators for cryptographic applications (No. NISTSP80002).

[9] Suryadi, M.T., Satria, Y., Prawadika, L.N. (2020). An improvement on the chaotic behavior of the Gauss Map for cryptography purposes using the Circle Map combination. Journal of Physics: Conference Series, 149(1): 012045. https://doi.org/10.1088/1742-6596/1490/1/012045

[10] Boyland, P.L. (1986). Bifurcations of circle maps: Arnol'd tongues, bistability and rotation intervals. Communications in Mathematical Physics, 106(3): 353-381. https://doi.org/10.1007/BF01207252

[11] Vilinea, A.R., Rusyaman, E., Djauhari, E. (2019). Solusi persamaan diferensial fraksional non-linear menggunakan telescoping decomposition method. Jurnal Matematika Integratif, 15(2): 139-148. https://doi.org/10.24198/jmi.v15.n2.23376.139

[12] Muktyas, I.B., MT, S., Aziz, M.K.B.M., Ohyver, M., Permai, S.D., Arifin, S. (2023). Bernoulli logistic map encryption algorithm for digital image. In AIP Conference Proceedings, 2679(1): 020014. https://doi.org/10.1063/5.0111255

[13] Karras, D.A. (2020). An effective statistical test suite for pseudorandom number generators in digital signatures and security robustness evaluation including a wavelet test for randomness. In 2020 IEEE International Conference on Progress in Informatics and Computing (PIC), Shanghai, China, pp. 314-322. https://doi.org/10.1109/PIC50277.2020.9350846

[14] Janicke, H., Abuadbba, S., Nepal, S. (2020). Security and privacy for a sustainable internet of things. In 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Atlanta, GA, USA, pp. 12-19. https://doi.org/10.1109/TPS-ISA50397.2020.00013

[15] Xu, C., Liu, L., Yu, X., He, B. (2021). Educational robot aided photo taking and management. In 2021 IEEE International Conference on Engineering, Technology & Education (TALE), Wuhan, Hubei Province, China, pp. 304-310. https://doi.org/10.1109/TALE52509.2021.9678731

[16] Salman, L.A., Hashim, A.T., Hasan, A.M. (2022).

Selective medical image encryption using polynomial-based secret image sharing and chaotic map. International Journal of Safety and Security Engineering, 12(3): 357-369. https://doi.org/10.18280/ijsse.120310

[17] Mudrika, M., Mt, S., Madenda, S. (2024). New chaos function of composition function Gauss map and dyadic transformation map for digital image encryption. In ITM Web of Conferences 61, pp. 01004. https://doi.org/10.1051/itmconf/20246101004

[18] Pareek, N.K., Patidar, V., Sud, K.K. (2006). Image encryption using chaotic logistic map. Image and Vision Computing, 24(9): 926-934. https://doi.org/10.1016/j.imavis.2006.02.021

[19] Hadi, F., Slimani, Y., Douar, A., Alti, A., Saoud, F., Harkati, M. (2024). Improved Vigenere cipher-rsa-based medical image security through multiple encryption keys. Ingénierie des Systèmes d'Information, 29(2): 599-608. https://doi.org/10.18280/isi.290221

[20] Fouda, J.A.E., Effa, J.Y., Sabat, S.L., Ali, M. (2014). A fast chaotic block cipher for image encryption. Communications in Nonlinear Science and Numerical Simulation, 19(3): 578-588. https://doi.org/10.1016/j.cnsns.2013.07.016

[21] Wu, Y., Noonan, J.P., Agaian, S. (2011). NPCR and UACI randomness tests for image encryption. Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), 1(2): 31-38.

[22] Chattopadhyay, D., Mandal, M.K., Nandi, D. (2011). Symmetric key chaotic image encryption using circle map. Indian Journal of Science and Technology, 4(5): 593-599.

[23] NIST. NIST Computer Security Resource Center. https://csrc.nist.gov/#, accessed on Aug. 08, 2025.