Mathematics and Statistics 12(5): 428-442, 2024
DOI: 10.13189/ms.2024.120504 http://www.hrpub.org

Dual Level Strategy Integrating 2-D CA and DCT Technique for Enhanced Data Protection

K. Gaverchand, R. Venkatesan*, A. Yasmin

Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur 603203, Tamil Nadu, India

Received May 3, 2024; Revised July 18, 2024; Accepted August 23, 2024

Cite This Paper in the Following Citation Styles

***(a)**: [1] K. Gaverchand, R. Venkatesan, A. Yasmin, "Dual Level Strategy Integrating 2-D CA and DCT Technique for Enhanced Data Protection," Mathematics and Statistics, Vol.12, No.5, pp. 428-442, 2024. DOI: 10.13189/ms.2024.120504*

***(b)**: K. Gaverchand, R. Venkatesan, A. Yasmin (2024). Dual Level Strategy Integrating 2-D CA and DCT Technique for Enhanced Data Protection, Mathematics and Statistics, 12(5), 428-442. DOI: 10.13189/ms.2024.120504*

Abstract In the contemporary era, safeguarding vast amounts of sensitive data and ensuring secure communication are paramount concerns, especially given the prevalence of insecure networks. The fields of cryptography and steganography have emerged as pivotal tools in addressing these challenges, attracting significant scientific attention due to their established effectiveness. This paper proposes a sophisticated double-layered security system for protecting textual data, initially utilizing 2-D Cellular automata (CA) to establish robust encryption protocols. The incorporation of the Von Neumann neighborhood of 2-D CA enhances the generation of secure cryptographic keys, leveraging its complex and chaotic behavior to improve randomness and unpredictability. Subsequently, to enhance security, the discrete cosine transform (DCT) technique is integrated, facilitating discreet embedding of encrypted data into the least significant bit (LSB) of DCT coefficients. An exhaustive range of tests and evaluations, encompassing analysis of key space, complexity, performance metrics, avalanche effect, security attacks, peak signal-to-noise ratio (PSNR) and mean square error (MSE), substantiates the effectiveness and resilience of this multifaceted approach. The findings reveal a significant average avalanche effect of 76%, indicating resilience against cryptographic attacks. Furthermore, superior preservation of image quality is observed during encryption, as evidenced by improved PSNR and MSE values compared to alternative technique. The overall analyses validate the proposed method as a proficient solution for secure data encryption in cyberspace.

Keywords Encryption, Decryption, Cellular Automata, DCT Technique

1 Introduction

In the dynamic and ever-evolving landscape of secure communication, the integration of cryptography [1] and steganography [2] is crucial amid escalating cyber threats [3]. Cryptography employs advanced encryption and key management to ensure message confidentiality and integrity. Steganography complements cryptography by covertly embedding messages within digital media, enhancing defense against prying eyes in the digital domain.

Steganography [4], originating in ancient Greece, has evolved across centuries with diverse historical methods for message concealment. Techniques like invisible ink and microdots from World War II exemplify its historical significance. Even during the Renaissance, artists such as Leonardo da Vinci used steganography by embedding concealed messages within their artwork, highlighting its enduring relevance in communication. This historical context underscores steganography's role alongside cryptography in today's digital landscape, illustrating its historical and contemporary importance in safeguarding sensitive information.

In modern image steganography, two main categories emerge: conventional techniques like Least Significant Bit (LSB) embedding [5], Discrete Fourier Transform (DFT) [6], and Discrete Cosine Transform (DCT) [7] manipulate image data for covert concealment. Deep learning methods such as Generative Adversarial Network (GAN) [8] and Convolutional Neural Network (CNN) employ neural networks to enhance security in steganographic communication, reflecting an evolving landscape where both traditional and innovative approaches safeguard sensitive data.

CA [9], pioneered by mathematician John von Neumann in the 1940s, are sophisticated mathematical constructs comprising a grid of cells that evolve over discrete time steps based on

predefined rules. 1-D CA [10], such as Wolfram's elementary CA, operate linearly, updating cell states using neighboring cell states. 2-D CA expand this concept to a grid of cells arranged in rows and columns, fostering intricate patterns and behaviors. CA applications span physics, biology, computer science, and artificial intelligence. Integrating Cryptography with 2-D CA leverages spatial complexity for innovative encryption, facilitating dynamic interaction between cryptographic algorithms and CA grid states for resilient methods.

This research endeavor explores the efficacy of integrating 2-D CA and DCT for digital communication security. By seamlessly combining cryptography and steganography with CA, an innovative dual-layered encryption method is introduced. The unpredictable and robust nature of CA significantly fortifies the resilience of the proposed model. Analysis of test results, including performance analysis, key space analysis, complexity analysis and avalanche effect, along with metrics like PSNR and MSE, unequivocally demonstrates the efficiency and adaptability of the dual-layer scheme. This analysis confirms secure encryption of textual data for communication and validates the efficacy of the encryption technique. In case of an unauthorized breach at the primary security level, the secondary layer ensures continued data protection. This research highlights the significance of pioneering strategies in ensuring the security of digital data.

The rest of the sections are organized as follows: Section 2 summarizes prior research pertinent to the proposed framework. Section 3 provides a brief overview of the preliminaries. Section 4 is divided into two parts, each providing a succinct description of the suggested encryption and embedding technique. Section 5 discusses security measures, including test evaluations. Finally, Section 6 summarizes the suggested model and suggests future prospects for the paper.

2 Literature Survey

This section delves into an analysis of various literature concerning cryptography and steganography.

The utilization of CA in cryptographic systems has seen significant advancements over the years. In 1994, Nandi et al. [11] discussed CA-based block and stream ciphers, defining block cipher functions based on CA state transitions and proposing pseudorandom pattern generators for stream cipher key generation. Tomassini and Perrenoud in 2001 [12] introduced a cryptographic system based on non-uniform CA, highlighting its resilience against cryptanalytic attacks and its potential for high-speed hardware implementation. Bouvry et al. leveraged 1-D non-uniform CA for secret key cryptography in 2004 [13], enhancing pseudo-random number generation and system resilience. In 2005 Rey Angel et al. [14] proposed a novel secret sharing scheme using CA memory, ensuring perfection within a (k, n) threshold scheme. In 2017, Farwa et al. [15] developed a secure image encryption method using an algebraic substitution box and the Arnold transform. In the following year, 2019, Shahzad et al. [16] used a coset diagram to construct S-boxes via a modular group acting on a finite field. In 2020, Zhu et al. [17] proposed a combined chaotic system integrating a pseudo-random sequence with a linear congruence random number generator, constructing S-boxes through original S-box scrambling. That same year, Ahmed et al. [18] introduced a dual-layer encryption approach utilizing LSB and XOR operations, suitable for complex data like images, while Gutub et al. [19] analyzed LSB and DWF algorithms for concealing multiple images. Kaur and Butta introduced RCVD in 2021 [20], a steganographic approach combining DCT and chaotic maps for high security, and Banerjee and Kundu [21] proposed a CA-based hardware encryption for wireless networks. Anil and Sharma proposed asymmetric key encryption using image processing and CA in 2022 [22], while Kumar et al. [23] recommended a medical image watermarking technique using DCT and SVD. That same year, Hematpour et al. [24] developed a robust image steganography scheme employing coupled chaotic maps and a new-chaotic S-Box. In 2023, Gaverchand and Venkatesan [25] utilized CA rule-150 for generating secure randomized secret keys, Waleed [26] proposed a robust digital image watermarking technique based on DCT and linear modulation, Maurya et al. [27] introduced a secure steganography scheme using a modified quantum S-Box, and George and Anghelescu [28] expounded on CA-based stream cipher encryption for VLSI hardware. In 2024, Dennunzio et al. [29] proposed an efficient algorithm for determining chaos in LCA, emphasizing polynomial degree limitation for computational efficiency.

The survey highlights the critical roles of steganography and cryptography in digital security, emphasizing their collaborative efforts to safeguard sensitive data in the digital realm. Addressing gaps in current literature, particularly concerning comprehensive evaluations and multilayered security approaches, this study proposes a dual security framework that integrates steganography, cryptography and CA. By leveraging the unique properties of these disciplines, the framework enhances data protection against emerging threats with unparalleled efficacy.

3 Preliminaries

3.1 Cellular Automata

CA [30] are mathematical models wherein cells evolve according to predefined rules. Featuring storage elements and combinational logic, they serve as versatile tools for studying complex phenomena using computer algorithms. CA capture emergent behaviors and patterns, aiding in the understanding of dynamic systems, thereby enriching scientific inquiry. CA comprise a grid of cells organized on a regular lattice grid, with each cell having a finite state of either 0 or 1. The state of each cell evolves based on predefined rules determined by the states of its local neighboring cells. In general, CA can be represented as a quintuple, $\mathcal{C} = (N, \Sigma, \delta, A, \rho)$, where N is a finite set of states, Σ is a finite set of cells in the local neighborhood, δ is the transition function, A is a neighbourhood configuration and ρ is a initial state.

CA can be classified based on dimensions, such as 1-D (linear array), 2-D (grid) and 3-D (cubic lattice), which notably

affect complexity and dynamics, consequently shaping diverse applications and behaviors.

In 1D-CA (ECA), the grid is linearly arranged and state changes depend on adjacent states. Equation 1 specifies that the transition of the i^{th} cell is contingent upon its current state and neighboring cells, resulting in $2^3 = 8$ potential neighborhood configurations. This leads to 255 CA rules for ECA.

$$\mathcal{C}(i)^{(t+1)} = \delta(\mathcal{C}(i-1)^t, \mathcal{C}(i)^t, \mathcal{C}(i+1)^t) \tag{1}$$

3.2 2-D Cellular Automata

2-D CA diverge from 1D-CA by adopting grid-based structures, facilitating interactions across two dimensions [31]. The following section explores the neighborhood structure, transition function, rule numbering and boundary conditions of 2-D CA, elucidating their significance in shaping intricate spatial patterns and behaviors.

3.2.1 Neighborhood structure

The neighborhood configuration of 2-D CA delineates the interaction between neighboring cells and a central cell within a 2-D grid, significantly influencing pattern formation and system behavior. Various configurations, notably the Von Neumann neighborhood (V_N) and Moore neighborhood (M_N), determine the extent to which neighboring cells affect state transitions, thus impacting the CA dynamics and emergent properties.

The V_N, as depicted in Figure 1, comprises four adjacent neighbors, directly affecting state transitions around a central cell. Conversely, the M_N illustrated in Figure 2 encompasses eight neighboring cells, incorporating both adjacent and diagonal cells, thereby broadening the range of interactions within the CA. The neighborhood cells in Figures 1 and 2 can be expanded by incorporating cells indexed according to Equation 2 and 3, where r represents the radius of the neighborhood.

$$V_N(i,j,r) = [(i',j') : |i'-i| + |j'-j| \leq r] \tag{2}$$
$$M_N(i,j,r) = [(i',j') : |i'-i| \leq r, |j'-j| \leq r] \tag{3}$$

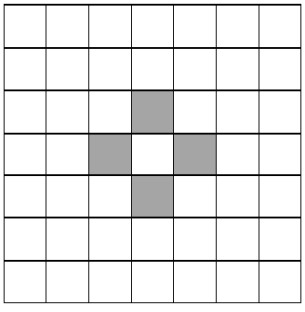

Figure 1. Von neighborhood ($r = 1$)

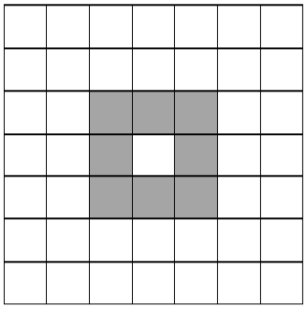

Figure 2. Moore neighborhood ($r = 1$)

3.2.2 Transition function

In 2-D CA, the transition function δ, as signified in Equation 4, governs the computation of the subsequent state $\mathcal{C}(i,j)^{(t+1)}$ of each cell at index (i,j) and time $t+1$, where $\mathcal{C}(i,j)^t$ represents the current state of the cell.

$$\mathcal{C}(i,j)^{t+1} = \delta(\mathcal{C}(i,j)^t, \mathcal{C}(i',j')^t) \tag{4}$$
$$\text{where } \mathcal{C}(i',j') \in [V_N(i,j,r) || M_N(i,j,r)]$$

3.2.3 Rule formulation

In 2-D CA with a nine-neighborhood configuration, the state evolution of a cell is influenced by both its current state and the states of its neighboring cells, as detailed in Table 1. In V_N, considering the four cells within its local neighborhood with $r = 1$, there are $2^{(2\times5)} = 1024$ rules. Contrarily, in M_N, with $r = 1$, considering the nine neighbors yields $2^{(2\times9)} = 262144$ rules.

Example: The derivation of Rule-171 through the integration of 2-D rule box is expressed as

Rule-171 = Rule-1 + Rule-8 + Rule-32 + Rule-128 + Rule-2

Table 1. 2-D Rule box

64	128	256
32	1	2
16	8	4

3.2.4 Boundary conditions

The boundary conditions in 2-D CA regulate grid edge management during evolution. They encompass periodic boundary conditions (PBC), which involve the grid wrapping around, fixed boundary conditions (FBC), where edge cells have static values or remain unaltered and circular boundary conditions (CBC), creating a circular shape for the grid. Typically, under PBC, the 2-D CA lattice adopts a toroidal shape. These conditions significantly influence the behavior and stability of CA.

3.3 Steganography

Steganography, an intricate practice, involves covertly embedding confidential data within seemingly innocuous carrier

media like text, images and audio [32], facilitating clandestine communication. This technique discreetly conceals information within the carrier, ensuring undetected transmission to unintended recipients. By leveraging such covert methods, individuals can securely transmit sensitive data, evading detection by unauthorized parties. Steganography is classified into spatial and frequency domain techniques, both of which are prominent, enabling concealment through direct manipulation of pixel data or alteration of frequency components. These methodologies establish a robust framework for covert communication across diverse contexts, ensuring confidentiality and security in data transmission.

3.3.1 Spatial domain

Spatial domain steganography [5] pertains to concealing information within a carrier medium, typically images, by directly manipulating the pixel values. Various techniques are employed for this purpose, among which LSB substitution stands prominent. LSB substitution involves replacing the least significant bits of pixel values with concealed data. Furthermore, spatial domain steganography encompasses methods such as embedding text within the voids of an image, capitalizing on the intrinsic attributes of the carrier medium to seamlessly integrate secret information.

3.3.2 Frequency domain

Frequency domain steganography [6] operates by manipulating the frequency components of the carrier medium, typically using mathematical transforms such as DCT or DFT. This approach embeds information in the frequency domain representation of the carrier, making it less perceptible to human observers. By exploiting the frequency domain properties of the carrier medium, frequency domain steganography can achieve effective concealment of secret data.

Both spatial and frequency domains have distinct strengths and limitations. Spatial methods are simpler with higher data hiding capacity but are more detectable due to visible alterations, while frequency techniques offer greater detection resistance despite potentially requiring more resources and having lower hiding capacity.

3.4 Substitution Box

The Substitution Box (S-Box) serves as a cornerstone in block cipher algorithms, facilitating resilient encryption by intricately mapping input bit sequences to output bit sequences. Crafted meticulously, these S-Boxes fortify cryptographic strength through sophisticated non-linear transformations, enhancing data security across various applications, from digital image encryption to steganography. Within this realm, the Advanced Encryption Standard (AES) S-Box emerges as a pinnacle, boasting unparalleled cryptographic properties that defy linear and differential cryptanalysis, thus enhancing the overall security posture of encryption algorithms. Recognizing its merits, this proposed model seamlessly integrates the AES S-Box into its encryption and decryption frameworks, harnessing its robustness to ensure the impregnable transmission and safeguarding of sensitive data, spanning from delicate communication networks to indispensable data storage systems. For instance, by utilizing the AES S-Box, the binary value 10100111 is segmented into 4-bit groups, yielding hexadecimal values 10 and 7. Corresponding to row 10 and column 7 in the S-box, the output is 5C. Converting 5C to binary, 5 and C(12) are represented as 0101 and 1100, respectively, resulting in the binary value 01011100. The transformation of the S-box and its inverse is illustrated in Figure 3.

AE	75	F2	22
7E	C1	05	BF
EC	10	DB	54
0A	FF	9A	41

E4	9D	89	93
F3	78	6B	08
CE	CA	B9	20
67	16	B8	83

Figure 3. Sub Bytes Transformation

4 Proposed methodology

The proposed model is designed to elevate the security of confidential data as it navigates through susceptible networks. This is achieved through the implementation of a dual-layer security approach, ensuring robust protection for confidential information. The model adheres to high security standards to effectively mitigate risks and safeguard data integrity through the following idea.

The proposed cryptographic algorithm integrates both confusion and diffusion principles to guarantee resilient encryption. Confusion, synonymous with the non-linear rule, establishes a complex relationship between ciphertext (CT) and the encryption key, deterring attackers through ambiguity. Conversely, diffusion, known as the linear rule, disperses changes in plaintext (PT) across a large portion of the CT, bolstering security by hindering pattern recognition. This combined approach fortifies cryptographic algorithms, safeguarding confidential data effectively. Additionally, traditional steganography, utilizing the DCT technique, enhances security by embedding the CT and secret key K within the RGB image, further reinforcing data protection. DCT partitions images into blocks, convertimg them to frequency domain, efficiently concealing encrypted data while remaining imperceptible. The sample illustration of the proposed model is given in Figure 4

The principal ideas of the proposed scheme are as follows

1. Proposing a pioneering method for dual data security by integrating cryptography and steganography.

2. Exploiting 2-D CA, the encryption method effectively integrates the characteristics of confusion and diffusion, bolstering the resilience of data protection.

3. Leveraging the 2-D CA Von Neumann neighborhood enhances cryptographic key generation by capitalizing on its complex and chaotic behavior.

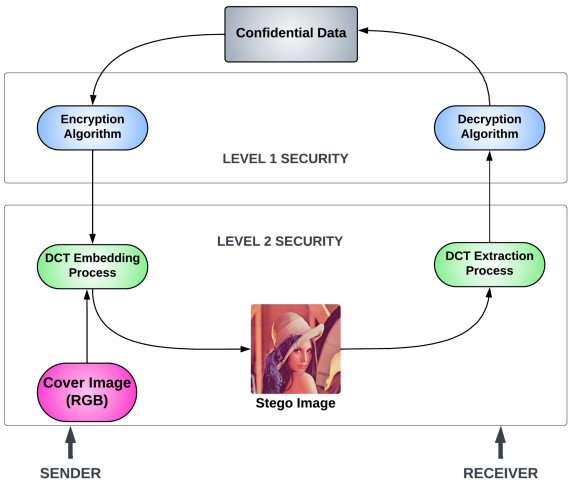

Figure 4. Block diagram of proposed model

4. DCT technique is employed to embed the encrypted data and secret key into RGB images provides enhanced security and imperceptibility.

5. Numerous tests and analysis are conducted to demonstrate the efficacy of the proposed model.

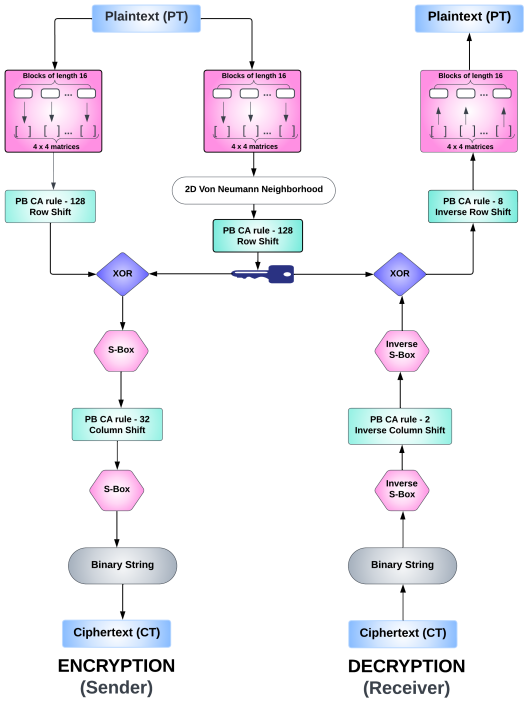

Figure 5. Workflow of the CA cryptosystem

4.1 Phase:1 Proposed CA based cryptosytem

The proposed cryptographic technique employs a sophisticated approach, integrating meticulously fortified layers of security and leveraging 2-D CA for heightened resilience against attacks, enabled by intricate nonlinear transformations and efficient parallel processing. Initially, a secret key is generated by considering PT and integrating diffusion properties within a 2-D Von Neumann neighborhood. Encryption and decryption algorithms are strategically designed to exploit both confusion and diffusion properties, thereby enhancing security. Rigorous analysis, including key space, comparative and complexity analysis, precisely evaluate the resilience and efficacy of the cryptosystem. The aggregate findings unequivocally confirm the proposed model efficiency and robust security. The workflow of the proposed cryptosystem is delineated in Figure 5.

4.1.1 Key Generation Process

The proposed model derives the secret key from the plaintext (PT). Initially, the PT is divided into blocks, denoted as $P = \{p_1, ..., p_n\}$ where each block p_i has a length of 16 characters $|p_i| = 16$. Padding is applied if $|p_n| < 16$, ensuring uniform block sizes across the PT. This proposed key generation process guarantees a distinct secret key for each block of P.

To obtain the secret key for block p_1, the following procedure is employed. Initially, a 4×4 matrix M_1 is constructed from p_1. Each element in M_1 is then represented by its 8-bit binary value, resulting in matrix M_2. Subsequently, the columns of M_2 are concatenated to form a 4×1 matrix, denoted as M_3. Following that, a 2-D CA Von Neumann neighborhood is applied to matrix M_3, updating the next generation M_4. M_4 is then split into a 4×4 matrix, yielding M_5, where each element's size is 8 bits. Finally, a row shift operation using PB CA rule-128 is performed on M_5. By utilizing PB CA rule-128 to every cell within the matrix M_5, a cyclic shifting action is initiated. This involves shifting values from the first row to the second row, those from the second row to the third row and so on, following a repetitive cycle. The shifting progression persists cyclically, with values from the last row being relocated to the first row, all while preserving PBC. This process derives the secret key matrix k_1 for the p_1 block.

This procedure iterates to derive secret keys for all blocks, thereby yielding distinct secret keys $k_1, k_2, ..., k_n$ corresponding to blocks $p_1, ..., p_n$ respectively.

The rules for determining a cell next state based on 2-D Von Neumann CA is as follows

Let $\mathcal{C}(i,j)$ represent the state of the cell at position (i,j) in the grid, where $\mathcal{C}(i,j) = 1$ if the cell is alive and $\mathcal{C}(i,j) = 0$ if the cell is dead, $N(i,j)$ represents the number of live neighbors of the cell at position (i,j) in the grid and $\mathcal{C}(i,j)^{t+1}$ represents the state of the cell (i,j) in the next generation, given the current state $\mathcal{C}(i,j)^t$ at time t.

- $\mathcal{C}(i,j) = 1$ and $N(i,j) < 2 \implies \mathcal{C}(i,j)^{t+1} = 0$ for all (i,j) in the grid.
- $\mathcal{C}(i,j) = 1$ and $N(i,j) = 2||3 \implies \mathcal{C}(i,j)^{t+1} = 1$ for all (i,j) in the grid.
- $\mathcal{C}(i,j) = 1$ and $N(i,j) > 3 \implies \mathcal{C}(i,j)^{t+1} = 0$ for all (i,j) in the grid.
- $\mathcal{C}(i,j) = 0$ and $N(i,j) = 3 \implies \mathcal{C}(i,j)^{t+1} = 1$ for all (i,j) in the grid.

4.1.2 Encryption Process

In the encryption process, the sender begins by inputting the PT. Similar to the key generation process, the PT is divided into blocks $P = \{p_1, ..., p_n\}$ where $|p_i| = 16$. Each block is transformed into a 4×4 matrix M_1 and then to its binary. The PB CA rule-128 is applied to binary matrix M_1 for row shifting, resulting in M_{11}. An XOR operation is performed between M_{11} and secret key k_1, yielding M_{12}. The binary values in M_{12} are substituted with their corresponding hexadecimal values using an S-box, producing M_{13}. A column shift operation based on PB CA rule-32 is applied to M_{13}, updating it to M_{14}. This process initiates cyclic column shifting, sequentially moving values from one column to the next in a repetitive cycle, with values from the last column wrapping around to the first column. After passing M_{14} through the S-box again, M_{15} is obtained. The hexadecimal values of M_{15} are then converted into 8-bit binary, resulting in M_{16}. Subsequently, all binary values within M_{16} are concatenated to form a single binary string, which serves as the CT.

After this encryption process, the sender proceeds to embed both the CT and the secret key K into an RGB image. This embedding is achieved through the utilization of the DCT method.

4.1.3 Decryption Process

In the decryption process, subsequent to retrieving the CT along with the secret key k from the RGB image, the recipient commences by constructing a 4×4 matrix M_1', where each element possesses a size of 8 bits. Following this, the binary values undergo conversion into hexadecimal values through the inverse S-box, resulting in the matrix M_2'. An inverse column shift operation, based on PB CA rule-2, is then applied to M_2', yielding M_3'. Subsequently, the matrix M_3' undergoes an inverse S-box transformation, giving rise to M_4', followed by the conversion of hexadecimal values into binary values within M_5'. An XOR operation is subsequently executed between M_5' and k, where $k \in K$, yielding M_6'. M_6' then undergoes an inverse row shift operation in accordance with PB CA rule-8, resulting in M_7'. Finally, each 1-byte binary value within M_7' is reverted to its corresponding character, thereby unveiling the confidential data PT.

In Figure 6, the program window displays a user-friendly interface that allows users to enter a confidential message into a specified text field. Clicking the "Encrypt" button initiates the encryption process, which displays the results in the encrypted text field. Conversely, the "Decrypt" button is used to decrypt encrypted text, revealing the original message in the decoded text field. This explains how to convert PT into encrypted data and vice versa using the suggested 2-D cryptography approach.

4.2 Algorithms of proposed cryptosystem

4.2.1 Algorithm : 1 Formation of Secret Key

Input: PT is split into $P = \{p_1, ..., p_n\}$ blocks, each with a length of 16 ($|p_i| = 16$).
Output: Secret key matrices $\{K = k_1, k_2, \ldots, k_n\}$.

1. Partition the PT into blocks $P = \{p_1, ..., p_n\}$.
2. If the length of the $p_n < 16$, pad it with zeros.
3. Convert each block into a 4×4 matrix denoted as M_1.
4. Represent the elements of matrix M_1 in 1-byte binary format to obtain M_2.
5. Concatenate the columns of M_2 to form matrix M_3, with size $z_\alpha \times 1$ ($|\alpha| = 32$).
6. Employ a 2-D CA Von Neumann neighborhood to iterate over M_3 and derive M_4.
7. Split the columns of M_4 to generate matrix M_5, with dimensions $z_\beta \times z_\beta$, where β denotes the size of each element ($|z| = 4$, $|\alpha| = 8$).
8. Apply PB CA rule-128 to shift the rows of M_5, resulting in M_6, where $M_6 = k \; \forall \; (k \in K)$.
9. Repeat the steps 3 to 8 for every block within P, ensuring the iterative derivation of the secret key for each block.
10. The matrices $k_1, k_2, \ldots, k_n$ serve as the secret keys for blocks $p_1, p_2, \ldots, p_n$.

4.2.2 Algorithm :2 Encipherment of the confidential data

Input: Plaintext (PT)
Output: Ciphertext (CT)

1. Partition the PT into blocks $P = \{p_1, ..., p_n\}$.
2. If the length of the $p_n < 16$, pad it with zeros.
3. Convert each block into a 4×4 matrix M_1, subsequently transformed into its 1-byte binary M_2.
4. Apply PB CA rule-128 to shift the rows of M_2, resulting in M_{11}
5. Perform XOR operation $M_{12} = M_{11} \oplus k_1$.
6. Transform M_{12} into hexadecimal values using an S-box to obtain M_{13}.
7. Apply PB CA rule-32 column shift to M_{13} to produce M_{14}.
8. Reprocess M_{14} through the S-box to derive M_{15}.
9. Convert hexadecimal values in M_{15} to binary, yielding M_{16}.
10. Concatenate binary values in M_{16} to form CT.

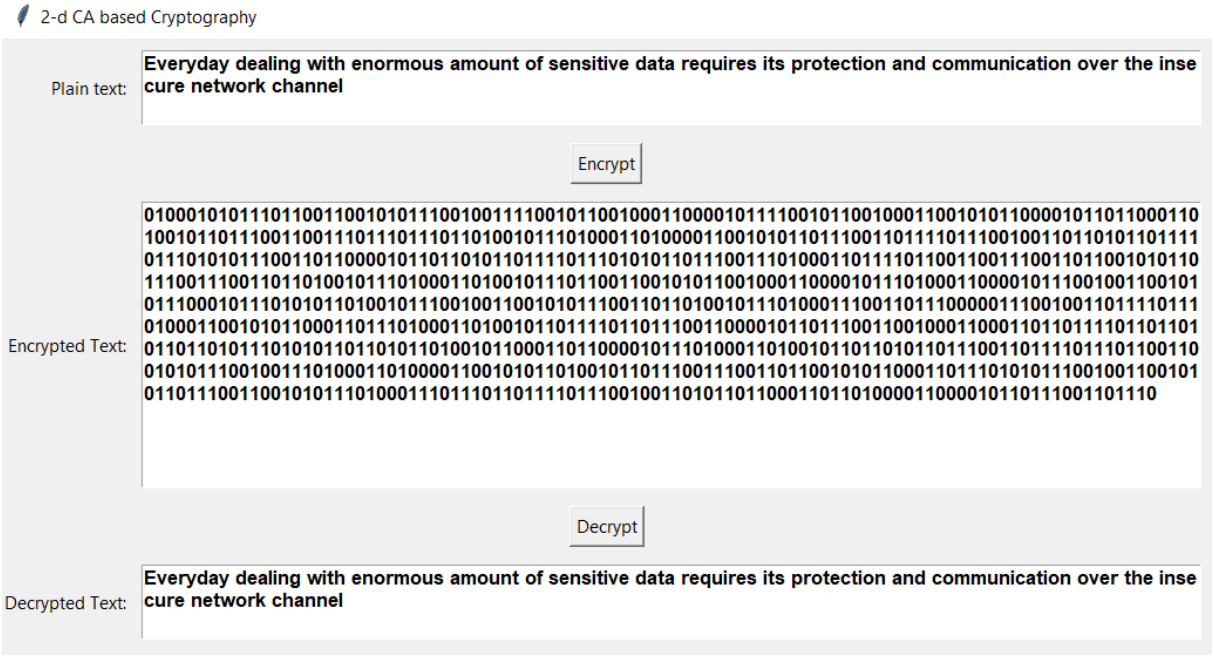

Figure 6. Illustration of the proposed 2-D cryptosystem

4.2.3 Algorithm :3 Decryption of the data

Input: Ciphertext (CT)
Output: Plaintext (PT)

1. Divide the CT into blocks, each with a length of 8 and transform into matrix, denoted as M_1'.
2. Perform inverse S-box transformation on M_1' to convert binary values to hexadecimal, yielding M_2'.
3. Apply inverse column shift on M_2' using PB CA rule-2 to generate M_3'.
4. Execute an inverse S-box transformation on M_3' to obtain M_4'.
5. Transform the hexadecimal values of M_4' into binary to obtain M_5'.
6. Execute XOR between k_1 and M_5' to yield M_6'.
7. Perform an inverse row shift on M_6' using PB CA rule-8 to produce M_7'.
8. Convert binary values in M_7' to characters and concatenate to reveal PT.

4.2.4 Illustration

To demonstrate the proposed cryptographic algorithms, "Cryptoprocessing" has been selected as the PT. The process involves generating the secret key and performing conversions between PT and CT in both directions. As previously stated, the secret key in this model is derived from the confidential data (PT). The PT under consideration has a size of 16.

Let P be "Cryptoprocessing". P is then converted into matrices denoted as M_1, with M_2 representing its binary form.

$$M_1 = \begin{bmatrix} C & r & y & p \\ t & o & p & r \\ o & c & e & s \\ s & i & n & g \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 01000011 & 01110010 & 01111001 & 01110000 \\ 01110100 & 01101111 & 01110000 & 01110010 \\ 01101111 & 01100011 & 01100101 & 01110011 \\ 01110011 & 01101001 & 01101110 & 01100111 \end{bmatrix}$$

Matrix M_3 is generated by concatenating the columns of M_2. It then undergoes iterative processes using a 2D-CA Von Neumann neighborhood to generate the subsequent generation, denoted as M_4.

$$M_3 = \begin{bmatrix} 01000011011100100111100101110000 \\ 01110100011011110111000001110010 \\ 01101111011000110110010101110011 \\ 01110011011010010110111001100111 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 11110111100111111000111110001010 \\ 10011111100110001000110110001111 \\ 10011000100111001001111111000 1100 \\ 10011100100111111001101110011100 \end{bmatrix}$$

The M_5 matrix is derived through column splitting of M_4, and k is obtained from M_4 using PB CA rule-128, forming the crucial secret key matirx

$$M_5 = \begin{bmatrix} 11110111 & 10011111 & 10001111 & 10001010 \\ 10011111 & 10011000 & 10001101 & 10001111 \\ 10011000 & 10011100 & 10011111 & 10001100 \\ 10011100 & 10011111 & 10011011 & 10011100 \end{bmatrix}$$

$$k = \begin{bmatrix} 10011100 & 10011111 & 10011011 & 10011100 \\ 11110111 & 10011111 & 10001111 & 10001010 \\ 10011111 & 10011000 & 10001101 & 10001111 \\ 10011000 & 10011100 & 10011111 & 10001100 \end{bmatrix}$$

The process of deriving CT from PT can be illustrated as follows. Similar to the key generation procedure, PT is initially considered as P, which undergoes transformation into matrix M_1 and subsequently into matrix M_2. Thus, the M_2 matrix is directly obtained from the preceding sequence of operations.

$$M_2 = \begin{bmatrix} 01000011 & 01110010 & 01111001 & 01110000 \\ 01110100 & 01101111 & 01110000 & 01110010 \\ 01101111 & 01100011 & 01100101 & 01110011 \\ 01110011 & 01101001 & 01101110 & 01100111 \end{bmatrix}$$

Applying PBCA rule 128 to M_2 results in M_{11}, and M_{12} signifies the XOR operation between M_{11} and k.

$$M_{11} = \begin{bmatrix} 01110011 & 01101001 & 01101110 & 01100111 \\ 01000011 & 01110010 & 01111001 & 01110000 \\ 01110100 & 01101111 & 01110000 & 01110010 \\ 01101111 & 01100011 & 01100101 & 01110011 \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} 11101111 & 11110110 & 11110101 & 11111011 \\ 10110100 & 11101101 & 11110110 & 11111010 \\ 11101011 & 11110111 & 11111101 & 11111101 \\ 11110111 & 11111111 & 11111010 & 11111111 \end{bmatrix}$$

M_{12} is transformed into hexadecimal M_{13} using an S-box, followed by applying PB CA rule-32 shifts the columns to produce M_{14}.

$$M_{13} = \begin{bmatrix} df & 42 & e6 & 0f \\ 8d & 55 & 42 & 2d \\ e9 & 68 & 54 & 54 \\ 68 & 16 & 2d & 16 \end{bmatrix}$$

$$M_{14} = \begin{bmatrix} 0f & df & 42 & e6 \\ 2d & 8d & 55 & 42 \\ 54 & e9 & 68 & 54 \\ 16 & 69 & 16 & 2d \end{bmatrix}$$

Reapplying the S-box to M_{14} produces M_{15}, which is then converted into binary to yield M_{16}.

$$M_{15} = \begin{bmatrix} 76 & 9e & 2c & 8e \\ d8 & 5d & fc & 2c \\ 20 & 1e & 45 & 20 \\ 47 & 45 & 47 & d8 \end{bmatrix}$$

$$M_{16} = \begin{bmatrix} 01110110 & 10011110 & 00101100 & 10001110 \\ 11011000 & 01011101 & 11111100 & 00101100 \\ 00100000 & 00011110 & 01000101 & 00100000 \\ 01000111 & 01000101 & 01000111 & 11011000 \end{bmatrix}$$

Ultimately, combining the values of M_{16} creates the binary string referred as CT.

CT=011101101001111000101100100011101101100000101
110111111100001011000010000000011110010001010010
000001000111010001010100011111011000

The preceding CT advances seamlessly into Phase 2 to bolster data security. The decryption process for generating the PT is detailed as follows.

The CT is converted into a matrix M_1', with the S-box binary matrix integrated directly. The inverse S-box then translates the binary values into hexadecimal, yielding M_2'.

$$M_1' = \begin{bmatrix} 01110110 & 10011110 & 00101100 & 10001110 \\ 11011000 & 01011101 & 11111100 & 00101100 \\ 00100000 & 00011110 & 01000101 & 00100000 \\ 01000111 & 01000101 & 01000111 & 11011000 \end{bmatrix}$$

$$M_2' = \begin{bmatrix} 0f & df & 42 & e6 \\ 2d & 8d & 55 & 42 \\ 54 & e9 & 68 & 54 \\ 16 & 68 & 16 & 2d \end{bmatrix}$$

Matrix M_3' is derived by applying column shifts with the inverse PB CA rule-2, while M_4' is produced by executing the inverse S-box transformation on M_3'

$$M_3' = \begin{bmatrix} df & 42 & e6 & 0f \\ 8d & 55 & 42 & 2d \\ e9 & 68 & 54 & 54 \\ 68 & 16 & 2d & 16 \end{bmatrix}$$

$$M_4' = \begin{bmatrix} ef & f6 & f5 & fb \\ b4 & ed & f6 & fa \\ eb & f7 & fd & fd \\ f7 & ff & fa & ff \end{bmatrix}$$

Matrix M_5' is acquired from the conversion of M_4' into binary values, and M_6' is generated by XORing matrices k and M_5'.

$$M_5' = \begin{bmatrix} 11101111 & 11110110 & 11110101 & 11111011 \\ 10110100 & 11101101 & 11110110 & 11111010 \\ 11101011 & 11110111 & 11111101 & 11111101 \\ 11110111 & 11111111 & 11111010 & 11111111 \end{bmatrix}$$

$$M_6' = \begin{bmatrix} 01110011 & 01101001 & 01101110 & 01100111 \\ 01000011 & 01110010 & 01111001 & 01110000 \\ 01110100 & 01101111 & 01110000 & 01110010 \\ 01101111 & 01100011 & 01100101 & 01110011 \end{bmatrix}$$

M_7' is produced by implementing PB CA rule-8 on M_6'. The resultant binary output is converted into characters to form M_8'.

$$M_7' = \begin{bmatrix} 01000011 & 01110010 & 01111001 & 01110000 \\ 01110100 & 01101111 & 01110000 & 01110010 \\ 01101111 & 01100011 & 01100101 & 01110011 \\ 01110011 & 01101001 & 01101110 & 01100111 \end{bmatrix}$$

$$M'_8 = \begin{bmatrix} C & r & y & p \\ t & o & p & r \\ o & c & e & s \\ s & i & n & g \end{bmatrix}$$

Combining all characters from M_8 unevil the confidential data, "PT =Cryptoprocessing". This example demonstrates the process of the proposed CA cryptosystem.

4.3 Phase:2 Steganography technique through DCT

Though the Phase 1 cryptographic model offers sufficient security for confidential data, encrypted data may still allure attackers as a challenge. Therefore, integrating steganography with cryptography effectively addresses this vulnerability, enhancing overall data protection.

As we know, frequency domain techniques offer bolstered security in comparison to spatial domain methods by concealing secret bits behind sub-band frequency coefficients. Thus, in Phase 2, encrypted data derived from the proposed cryptographic model and the generated secret key K are embedded into the image using the DCT technique. This process ensures integration of the secret information into image areas less susceptible to compression and cropping.

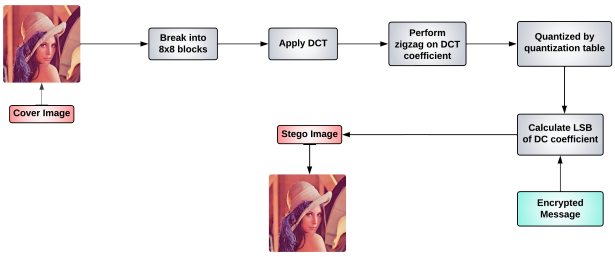

Figure 7. Workflow of the DCT

4.3.1 Discrete Cosine Transform

The DCT [2] is a mathematical technique crucial in signal processing and image compression. It converts spatial domain data into frequency components as illustrated in Figure 8, facilitating efficient compression while preserving essential image information.

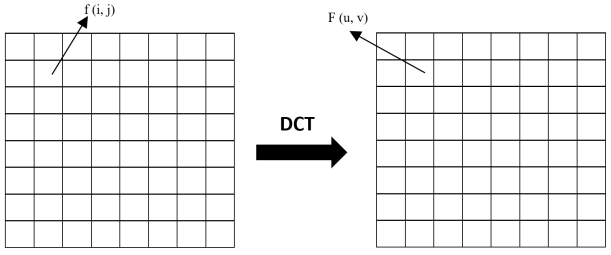

Figure 8. DCT transform

The DCT plays an integral role in steganography, initiating the process by segmenting the image into 8×8 blocks of pixels.

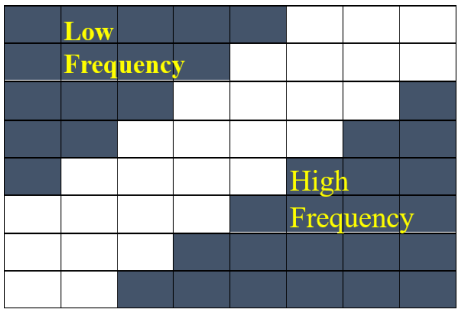

Figure 9. DCT coefficient

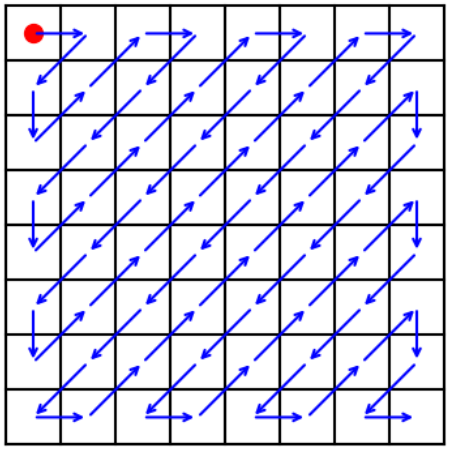

Figure 10. Zigzag compression

This partitioning is essential due to the inherent composition of images, which comprise pixels represented in color channels (R, G, B) organized to construct the visual content. Such meticulous division ensures that the embedding process can be carried out efficiently and effectively, preserving the integrity of the image while concealing sensitive information within its structure.

In Figure 9, the 64 (8×8) DCT basic coefficients are illustrated. The top-left corner represents low frequencies, while the bottom-right corner represents high frequencies. Notably, distortions in the high frequency regions are less perceptible to the human eye. The fundamental expression of 2-D DCT is represented as [27].

$$F(u,v) = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \alpha(u)\alpha(v) cos\left[\frac{(2i+1)u\pi}{2N}\right] cos\left[\frac{(2j+1)v\pi}{2N}\right] \cdot f(i,j) \quad (5)$$

To further bolster the security of both encrypted data and the secret key K, we utilize the DCT technique as a crucial step in the embedding process. Initially, the image is divided into 8x8 pixel blocks, and DCT is applied to each block, generating an 8x8 matrix comprising 64 DCT coefficients. Subsequently, a zigzag scan compression method, as illustrated in Figure 10, is employed to traverse all 64 coefficients efficiently. These coefficients then undergo quantization using the specified Table 2,

Table 2. Quantization table (Q)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

allowing for the extraction of the LSBs of the DC coefficients. These LSBs serve as the carriers for the secret bits, which are seamlessly embedded into the image by substituting the LSBs of the DC coefficients. Through this process, the concealed text is seamlessly integrated into the image, resulting in the creation of a steganographic image. The workflow process of utilizing the DCT for embedding data into an RGB image is depicted in detail in Figure 7, illustrating the in-depth approach taken to ensure robust data embedding within the image while maintaining its visual integrity.

4.4 DCT based steganography algorithms

4.4.1 Algorithm : 4 Embedding process through DCT

Input: Encrypted data (CT) and secret key (K) with Cover image
Output: Stegno image

1. Import the cover image of size 256×256
2. Import the CT and K then convert it to binary value
3. Divide the cover image into 8 × 8 blocks of pixels (p_i)
4. Apply DCT to the each block of pixel $p_i^{'} = \text{DCT}(p_i)$
5. Perform the zigzag scan compression of the DCT $p_i^{''} = zigzag(p_i^{'})$
6. Compress each of the blocks using the quantization table $p_i^{'''} = compress(p_i^{''})$
7. Extract the DC coefficients $DC_i = ExtractDC(p_i^{'''})$
8. Embed each bits of the encrypted message into the LSB of DC coefficients
9. Stegno image

4.4.2 Algorithm :5 Extracting process

Input: Stegno image
Output: Encrypted data (CT) and secret key (K)

1. Load the stegno image
2. Divide the stegno image into 8×8 pixel blocks p_i
3. Extract the zigzag-scanned DCT coefficients from the block $p_i'' = zigzag^{-1}(p_i^{'''})$
4. Inverse compress the coefficients using the predefined quantization table $p_i^{'} = \text{inverse compress}\ (p_i^{''})$
5. Apply inverse DCT (IDCT) on the coefficients $p_i = IDCT(p_i^{'})$
6. Extract the DC coefficients (DC_i)
7. Decode the LSB of the DC coefficient to retrieve each bit of the secret message
8. Convert the binary string of the secret message back to the text format
9. Encrypted text

5 Results and Discussion

5.1 Experimental setup

The proposed scheme efficacy has been assessed through Python, recognized for its versatility across disciplines like web development and data analysis. Python's intutive design and vast library of frameworks enable users to efficiently tackle diverse computational challanges with precision and versatility. The evaluation is conducted on a Dell laptop equipped with an Intel Core i5 processor clocked at 1.8GHz, running Windows 10, 8GB of RAM and a 128GB SSD. The datasets used to assess performance and avalanche capabilities in Phase 1 are obtained from the 20 newsgroups dataset [33], available on the UCI Machine Learning repository. Following that, in Phase 2, RGB color images are extracted from the USC-SIPI Image Database [34]. Both datasets are well-established resources regularly used in academic research and industrial applications.

5.2 Analysis of Proposed Model

Analyzing cryptographic and steganographic algorithms is crucial for ensuring reliable data protection, efficiency, and resilience against attacks. In the proposed model, the secret message undergoes a dual-stage security process. Initially, the message is encrypted using the proposed CA cryptosystem. Subsequently, this encrypted message is embedded into an image through the steganographic DCT technique. To affirm the model's effectiveness, a comprehensive set of metrics is meticulously assessed. The reliability and robustness of the CA cryptosystem are validated through various analyses, including performance, key space, complexity, avalanche effect, and security assessments. These critical parameters are essential for evaluating any cryptographic algorithm. For the secondary stage, the DCT model's PSNR and MSE metrics are compared with other steganographic methods. This rigorous

evaluation process confirms the proposed model's resilience, efficiency, and security, underscoring its suitability for robust cryptographic applications.

5.3 Evaluation of the CA cryptosystem

The robustness of the proposed scheme is emphasized by a thorough examination encompassing performance analysis, avalanche effect, key space analysis, complexity analysis, and security analysis. Evaluating key management ensures robust security by assessing the total possible keys, while performance analysis enhances resource utilization and scalability. Complexity analysis aids in algorithm selection and optimizes resource utilization, while assessing the avalanche effect verifies algorithm effectiveness and strengthens security, ultimately fostering trust in the cryptosystem's reliability and efficiency. Analyzing security attacks in cryptosystems identifies vulnerabilities, ensuring robust data protection. This extensive analysis aids in ensuring the robust development of the proposed scheme.

5.3.1 Performance analysis

Performance analysis of a cryptosystem emphasizes speed, focusing on encryption and decryption efficiency. By assessing processing times and throughput rates, it evaluates the reliability of cryptographic operations, ensuring effective data security during transmission and retrieval while maintaining strong cryptographic protection.

In this investigation, we evaluate and compare the encryption and decryption time complexities of our proposed model with standard cryptographic algorithms such as Rivest-Shamir-Adleman (RSA) and ElGamal. The analysis involves examining 20 distinct data points extracted from the dataset [33]. Each data point undergoes rigorous testing through 50 iterations in both encryption and decryption programs, producing precise averages detailed in Table 3. These averages provide a comprehensive view of the average encryption and decryption times.

Figure 11 illustrates that proposed model achieves superior encryption time efficiency compared to RSA and ElGamal, reducing times by 31.14% and 24.68% respectively compared to RSA. Similarly, in Figure 12, the proposed model demonstrates superior decryption efficiency, reducing times by 56.97% compared to RSA and by 49.62% compared to ElGamal.These substantial improvements not only underscore the efficacy and reliability of our proposed approach compared to standard models but also underscore its potential advantages in practical cryptographic applications.

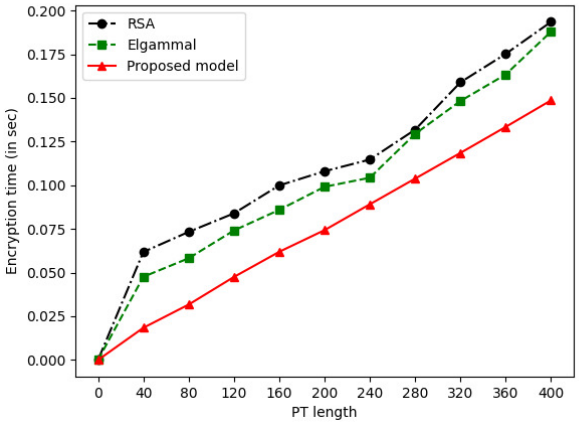

Figure 11. Performance analysis for encryption

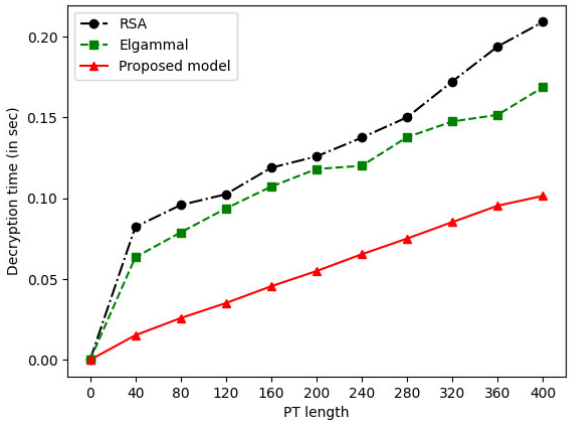

Figure 12. Performance analysis for decryption

5.3.2 Complexity analysis

Complexity analysis in cryptography is essential for evaluating algorithm efficiency in terms of time and space requirements, crucial for assessing performance across different contexts. It aids in identifying vulnerabilities, optimizing resources, and ensuring robust security measures are in place. Understanding the relationship between execution time and input size helps predict suitable algorithms.

Here, we examine the time complexity of key generation, encryption, and decryption in proposed algorithms, comparing them with established standard cryptographic algorithms such as RSA and ElGamal, as outlined in Table 4. The proposed system incorporates several confusion and diffusion properties, including XOR operation, row shifting, column shifting, and S-box transformation. Each of these properties exhibits a complexity of $O(n)$. Overall, the proposed model achieves a linear time complexity of $O(n)$, where algorithm execution time scales linearly with input size. This contrasts with existing models, where some algorithms exhibit quadratic complexity $O(n^2)$, cubic complexity $O(n^3)$, and logarithmic complexities $O(\log n)$ and $O(\log^3 n)$, indicating quadratic, cubic, or logarithmic growth in execution time with input size. Thus, the proposed model offers improved efficiency and optimization, making it well-suited for diverse cryptographic applications.

Table 3. Performance analysis varies with data length

Data Set	PT length (char)	RSA		ElGammal		Proposed	
		Encryption (in seconds)	Decryption (in seconds)	Encryption (in seconds)	Decryption (in seconds)	Encryption (in seconds)	Decryption (in seconds)
1	40	0.0618	0.0823	0.0476	0.0635	0.0184	0.0153
2	80	0.0733	0.0958	0.0582	0.0789	0.0317	0.0259
3	120	0.0839	0.1025	0.0741	0.0937	0.0475	0.0352
4	160	0.0999	0.1189	0.0859	0.1072	0.0619	0.0456
5	200	0.1080	0.1259	0.0991	0.1181	0.0743	0.0550
6	240	0.1147	0.1375	0.1043	0.1201	0.0891	0.0654
7	280	0.1318	0.1501	0.1294	0.1378	0.1038	0.0751
8	320	0.1589	0.1723	0.1482	0.1475	0.1185	0.0853
9	360	0.1752	0.1939	0.1634	0.1516	0.1334	0.0954
10	400	0.1935	0.2091	0.1879	0.1687	0.1486	0.1015
Average	(in sec)	0.1201	0.1390	0.1098	0.1187	0.0827	0.0598

Table 4. Comparitive analysis of Time complexity

Algorithm	RSA	Elgammal	Proposed
Key Generation	$O(log^3 n)$	$O(log^3 n)$	$O(n)$
Encryption	$O(log^3 n)$	$O(n^2)$	$O(n)$
Decryption	$O(log^3 n)$	$O(n^2)$	$O(n)$

5.3.3 Key space analysis

Key space analysis is pivotal in cryptography, providing a critical evaluation of a cryptographic algorithm, resilience against brute-force attacks. Following Kerckhoff's principle, which prioritizes safeguarding the secrecy of the key, underscores the imperative for expanding the key space. By amplifying the key space, the likelihood of adversaries successfully guessing the accurate key diminishes significantly, thereby fortifying the security of the cryptographic model as a whole. In our proposed scheme, we adhere rigorously to this principle by setting the key size to be eight times larger than the PT, resulting in a calculated key space (KS) expressed as

$$KS = L^{|P|\times 8} = 2^{|P|\times 8}$$

where L represents the binary value 2 and $|P|$ denotes the size of the PT. This expansive key space substantially heightens the complexity of exhaustive key search attempts, thereby enhancing security. This strategic approach not only enhances the security of proposed cryptographic system but also adheres to industry standards, ensuring steadfast protection against unauthorized access attempts while aligning with the best practices for data security.

5.3.4 Security analysis

In a cryptosystem, analyzing security involves assessing vulnerabilities, resilience, and strengths to attacks. This evaluation measures how effectively encryption algorithms, key management, and system design protect data from unauthorized access or alteration, emphasizing the proposed model's ability to withstand broad security threats.

Brute force attack

The intruders systematically test encryption key combinations to gain access. The proposed model reinforces defenses through matrix manipulation and iterative 2D-CA techniques, significantly increasing the complexity of key space analysis while successfully resisting brute force attacks.

Denial of Service

A DoS attack overwhelms a system with excessive traffic, disrupting its normal operations. Through effective block management, transformations, and integrated security measures, the proposed scheme effectively manages high traffic volumes and ensures system availability, providing strong resilience against DoS attacks.

Chosen Plaintext attack

A chosen plaintext attack seeks to obtain CT for specific PT in order to deduce the encryption key. The proposed scheme thwarts this by employing XOR operations, S-box substitutions, and cyclic column shifts, obscuring the PT-CT relationship effectively.

Ciphertext-only attack

A ciphertext-only attack tries to deduce PT from encrypted data without the decryption key. The proposed encryption algorithm counters this through various processes in the encryption technique, making decryption without the key computationally infeasible.

Phishing attack

A phishing attack involves deceptive methods where attackers impersonate trusted entities to mislead individuals into disclosing sensitive information like passwords or financial data. The proposed strategy mitigates phishing attacks by safeguarding sensitive information with robust encryption methods, making it exceedingly difficult for attackers to exploit deceptive tactics to steal valuable data.

5.3.5 Avalanche effect

The avalanche effect $(A.E)$ in a cryptosystem describes the phenomenon where a slight modification in input data or cryp-

tographic key results in a drastically different output, bolstering security by making it arduous for adversaries to deduce sensitive information. This property is crucial in cryptographic design as it amplifies the impact of alterations, enhancing confidentiality and thwarting potential attacks.

The avalanche effect of the proposed model is calculated using Equation 6, utilizing 10 distinct datasets sourced from the dataset [33], with each undergoing a 1-bit change in the PT. The resulting Avalanche effect is detailed in Table 5, while Figure 13 graphically illustrates the robustness of our scheme. Leveraging numerous confusion and diffusion properties, the proposed model achieves an average Avalanche effect of 72%. This analysis highlights the proposed model resilience against cryptographic attacks and its adeptness in preserving data security at an elevated standard.

$$\text{A.E} = \frac{\text{No. of characters changed in the CT}}{\text{Total no. of characters in the CT}} \times 100 \qquad (6)$$

Table 5. Avalanche effect using 1-bit PT modification

Data set	PT length	Alteration in CT
1	50	69%
2	100	76%
3	150	82%
4	200	72%
5	250	66%
6	300	76%
7	350	71%
8	400	80%
9	450	67%
10	500	81%

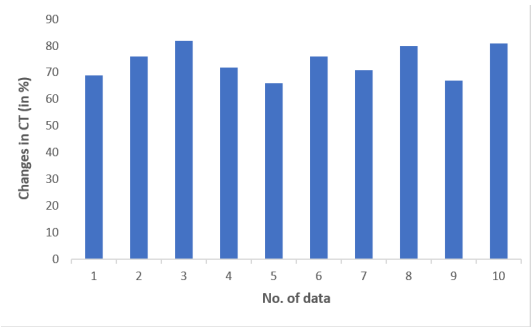

Figure 13. Analysis of avalanche effect

5.4 Evaluation of the DCT technique

In this section, we assessed the performance of the Phase 2 DCT technique on several color images, each with dimensions of 256 × 256 pixels. The chosen digital image set comprises House, Airplance, Female and Splash as shown in Figure 14. Metrics such as MSC and PSNR were utilized to assess stegno image quality compared to the LSB technique, aiding in evaluating steganographic robustness. In this proposed model, C and S represent cover and stegno images, respectively, with varied capacities for computing MSE and PSNR, enabling comparison with LSB. Overall, the results affirm the DCT technique has high fidelity and efficacy in embedding confidential data while preserving image integrity.

Figure 14. Test images of size 256×256 dimensions. a. House, b. Airplane, c. Female and d. Splash

5.4.1 Mean Squared Error

MSE [35] serves as a vital measure to quantify the difference between cover and stegno images, as determined by Equation 7, with X and Y representing the pixel rows and columns, respectively. Functioning as a quantitative gauge, MSE captures the average squared discrepancies between corresponding pixel values, enabling an impartial evaluation of steganographic fidelity. The experimental MSE values, showcased in Table 6, offer unparalleled insights into the efficacy of the DCT technique.

$$MSE = \frac{1}{XY} \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} [C(i,j) - S(i,j)]^2 \qquad (7)$$

Table 6. Test results for MSE of the traditional LSB and DCT technique

Cover Image	Capacity Bits	MSE	
		LSB Technique	DCT Technique
House	156730	0.813	0.759
Airplane	135695	0.735	0.615
Female	113896	0.662	0.581
Splash	107875	0.591	0.436

5.4.2 Peak Signal to Noise Ratio

PSNR [35], assumes a critical role in the realm of steganography as it evaluates the quality of images using Equation 8. It gauges the equilibrium between hiding data and maintaining visual accuracy, where elevated values indicate minimal noticeable decline in image quality. It's essential to highlight that the highest attainable pixel value is 255. Widely embraced in the field, this metric serves as a standard for evaluating the effectiveness of steganographic methodologies, revealing the nuanced compromises between secrecy and maintaining visual integrity. The findings regarding PSNR values are meticulously presented in Table 7.

$$\begin{aligned} PSNR &= 10log_{10}\left[\frac{Max_I^2}{MSE}\right] \\ &= 20log_{10}\frac{Max_I}{\sqrt{MSE}} \\ &= 20log_{10}\left[\frac{\sqrt{XY}\times 255}{\sqrt{\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}\left[C\left(i,j\right)-S\left(i,j\right)\right]^2}}\right] \end{aligned} \tag{8}$$

Table 7. Test results for PSNR of the traditional LSB and DCT technique

		PSNR	
Cover Image	Capacity Bits	LSB Technique	DCT Technique
House	156730	47.284	54.803
Airplane	135695	49.938	56.568
Female	113896	52.836	59.463
Splash	107875	55.579	64.089

6 Conclusions

In this paper, we present a novel dual data encryption approach that incorporates principles of CA across two distinct phases. Phase 1 employs a 2D Von Neumann neighborhood CA to intricately generate a secret key, bolstering encryption through a blend of confusion and diffusion properties. Phase 2 strategically extends the framework by discreetly embedding encrypted data and secret key within RGB images using the DCT technique, enhancing the proposed scheme resilience against sophisticated decryption attempts and adversarial breaches.

Thorough evaluation in both phases, covering key aspects like key space, performance, avalanche effect, complexity, security and metrics such as PSNR and MSE, demonstrates superior performance, confirming the proposed model efficacy in safeguarding textual data against potential threats. This study highlights significant advancements in encryption techniques, advocating for the practical implementation of the framework to enhance the protection of sensitive information in the contemporary world.

In the future, this study aims to refine the scalability and adaptability of the dual data encryption approach using 2-D CA to accommodate expansive datasets and diverse data formats such as audio and video files. Additionally, investigating the feasibility of the proposed work in real-world scenarios, such as healthcare or finance, will provide valuable insights into its practical effectiveness and potential for widespread adoption. Parallel computing techniques will be explored to enhance performance, and alongside expanding applications into emerging technologies like IoT and blockchain security.

REFERENCES

[1] William Stallings, "Cryptography and Netwrok Security," 3rd Edition, Pearson, 2002, pp. 1–681.

[2] Furht B., "Discrete Cosine Transform," in Encyclopedia of Multimedia, Springer, 2006, pp. 203–205. DOI: 10.1007/0-387-30038-4_61

[3] Hassaballah Mahmoud, Hameed Mohamed, Awad Ali Ismail, Muhammad Khan, "A Novel Image Steganography Method for Industrial Internet of Things Security," IEEE Transactions on Industrial Informatics, vol. 17, pp. 7743–7751, 2021. DOI: 10.1109/TII.2021.3053595.

[4] Johnson N.F., Jajodia S., "Exploring steganography: Seeing the unseen," Computer, vol. 31, no. 2, pp. 26–34, 1998. DOI: 10.1109/MC.1998.4655281.

[5] Luo W., Huang F., Huang J., "Edge Adaptive Image Steganography Based on LSB Matching Revisited," IEEE Transactions on Information Forensics and Security, vol. 5, no. 2, pp. 201–214, 2010. DOI: 10.1109/TIFS.2010.2041812.

[6] Mandal J.K., "Discrete Fourier transform-based steganography," in Reversible Steganography and Authentication via Transform Encoding, 1st ed, Springer Singapore, 2020, pp. 63–98.

[7] Zhang X., Peng F., Long M., "Robust Coverless Image Steganography Based on DCT and LDA Topic Classification", IEEE Transactions on Multimedia, vol. 20, no. 12, pp. 3223–3238, 2018. DOI: 10.1109/TMM.2018.2838334.

[8] Goodfellow, I., Pouget-Adadie, M., Mirza M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., "Generative adversarial nets", Proceedings of Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.

[9] Von Neumann J., "Theory of Self-Reproducing Automata," Essays on Cellular Automata, 1970.

[10] Stephen Wolfram, "Statistical mechanics of cellular automata," Reviews of Modern Physics, vol. 55, no. 3, pp. 601–644, 1983. DOI: 10.1103/RevModPhys.55.601.

[11] Nandi S, Kar B. K., Pal Chaudhuri P., "Theory and applications of cellular automata in cryptography," In IEEE Transactions on Computers, vol. 43, no. 12, pp. 1346–1357, 1994. DOI: 10.1109/12.338094

[12] Marco Tomassini, Mathieu Perrenoud, "Cryptography with cellular automata," Applied Soft Computing, vol. 1, no. 2, pp. 151–160, 2001. DOI: 10.1016/S1568-4946(01)00015-1.

[13] Bouvry P., Seredynski F., Zomaya A.Y., "Application of Cellular Automata for Cryptography," in Parallel Processing and Applied Mathematics. Lecture Notes in Computer Science, Springer, 2004, vol. 3019. DOI: 10.1007/978-3-540-24669-5-58

[14] Rey Angel, Mateus Joaquim, Sanchez Gerardo, "A secret sharing scheme based on cellular automata," Applied Mathematics and Computation, vol. 170, pp. 1356–1364, 2005. DOI: 10.1016/j.amc.2005.01.026.

[15] Farwa S., Muhammad N., Khan S., "A novel image encryption based on algebraic s-box and Arnold transform," 3D Research, vol. 8, no. 26, 2017. DOI: 10.1007/s13319-017-0135-x

[16] Shahzad I., Mushtaq Q., Razaq A., "Construction of new s-box using action of quotient of the modular group for multimedia security," Security Communication and Network, 2019. DOI: 10.1155/2019/2847801

[17] Zhu D., Tong X., Zhang M., Wang Z., "A new s-box generation method and advanced design based on combined chaotic system," Symmetry, vol. 12, no. 12, p. 2087, 2020. DOI: 10.3390/sym12122087

[18] Ahmed A., Ahmed A., "A secure image steganography using LSB and double XOR operations," International Journal of Computer Science and network Security, vol. 20, no. 5, pp. 139–144, 2020.

[19] Gutub A., Al-Shaarani F., "Efficient Implementation of Multi-image Secret Hiding Based on LSB and DWT Steganography Comparisons," Arabian Journal for Science Engineering, vol. 45, no. 4, pp. 2631–2644, 2020. DOI: 10.1007/s13369-020-04413-w.

[20] Kaur R., Singh B., "A novel approach for data hiding based on combined application of discrete cosine transform and coupled chaotic map," Multimedia Tools and Applications, vol. 80, no. 10, pp. 14665–14691, 2021. DOI: 10.1007/s11042-021-10528-5

[21] Ayan Banerjee, Anirban Kundu, "Cellular Automata based Cryptography Model for Reliable Encryption Using State Transition in Wireless Network Optimizing Data Security," An International Journal Wireless Personal Communications, vol. 119, no. 1, pp. 877–918, 2021. DOI: 10.1007/s11277-021-08243-3

[22] Anil Kumar, Sandeep Kumar Sharma, "Information cryptography using cellular automata and digital image processing," Journal of Discrete Mathematical Sciences and Cryptography, vol. 25, no. 4, pp. 1105–1111, 2022. DOI: 10.1080/09720529.2022.2072437

[23] Kumar S., Srivastava A., Chaurasiya P. K., Kushawaha A., Vishal V., "DCT and SVD-based Watermarking Technique for Imperceptibility and Robustness of Medical Images. 4th International Conference on Advances in Computing, Communication Control and Networking, pp. 2335–2339, 2022. DOI: 10.1109/ICAC3N56670.2022.10074157.

[24] Sourkhani I.G. Hematpour N., Sodeif A., Sani R.H., "A new steganographic algorithm based on coupled chaotic maps and a new chaotic s-box," Multimedia Tools and Applications, vol. 81, 2022. DOI: 10.1007/s11042-022-12828-w

[25] Gaverchand K., Venkatesan R., "A novel approach of 1-D cellular automata in cryptosystem," Mathematical Modelling of Engineering Problems, vol. 10, no. 6, pp. 2121–2126, 2023. DOI: 10.18280/mmep.100623

[26] Waleed Alomoush, Osama A. Khashan, Ayat Alrosan, Hani H. Attar, Ammar Almomani, Fuad Alhosban, Sharif Naser Makhadmeh, "Digital image watermarking using discrete cosine transformation based linear modulation," Journal of Cloud Computing, vol. 12, no. 96, 2023. DOI: 10.1186/s13677-023-00468-w

[27] Sonam Maurya, Nainil Nandu, Tanay Patel, Dinesh Reddy, Sushil Tiwari, Mahesh Kumar Morampudi, "A discrete cosine transform-based intelligent image steganography scheme using quantum substitution box," Quantum Information Processing, vol. 22, 2023. DOI: 10.1007/s11128-023-03914-5

[28] George Cosmin Stanica, Petre Anghelescu, "Cryptographic algorithm based on hybrid one-dimensional cellular automata," Mathematics, vol. 11, no. 6, p. 1481, 2023. DOI: 10.3390/math11061481

[29] Alberto Dennunzio, Enrico Formenti, Luciano Margara, "An efficient algorithm deciding chaos for linear cellular automata over $(Z/mz)^n$ with applications to data encryption," Information Sciences, vol. 657, p. 119942, 2024. DOI: 10.1016/j.ins.2023.119942.

[30] Manisha Ghosh, Rajeev Kumar, Mousumi Saha, Biplab K Sikdar, "Cellular Automata and Its Applications," IEEE International Conference on Automatic Control and Intelligent Systems, pp. 52–56, 2018. DOI: 10.1109/I2CACIS.2018.8603689.

[31] Norman H. Packard, Stephen Wolfram, "Two-dimensional cellular automata," Journal of Statistical Physics, vol. 38, pp. 901–946, 1985.

[32] Subramanian Nandhini, Elharrouss Omar, Al-maadeed Somaya, Bouridane Ahmed, "Image Steganography: A Review of the Recent Advances," IEEE Access, 2021. DOI: 10.1109/ACCESS.2021.3053998.

[33] Mitchell Tom, "Twenty Newsgroups," UCI Machine Learning Repository, https://doi.org/10.24432/C5C323 (Accessed on 25rd March 2024).

[34] USC-SIPI Image Database. https://sipi.usc.edu/database (Accessed on 25rd March 2024).

[35] Bhardwaj R., Sharma V. Image Steganography Based on Complemented Message and Inverted Bit LSB Substitution, Procedia Computer Science, vol. 93, pp. 832–838, 2016. DOI: 10.5120/ijca2018917743