IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2024, accepted 21 August 2024, date of publication 2 September 2024, date of current version 12 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3453171

RESEARCH ARTICLE

Z-Crypt: Chirp Z-Transform-Based Image Encryption Leveraging Chaotic Logistic Maps and Substitution Permutation Network

ABDULLAH ALAKLABI[1], (Student Member, IEEE), ARSLAN MUNIR[2], (Senior Member, IEEE), MUHAMMAD ASFAND HAFEEZ[1], (Student Member, IEEE), AND MUAZZAM A. KHAN KHATTAK[3], (Senior Member, IEEE)

[1]Department of Computer Science, Kansas State University, Manhattan, KS 66506, USA
[2]Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA
[3]Department of Computer Sciences, Quaid-i-Azam University, Islamabad 45320, Pakistan

Corresponding author: Arslan Munir (arslanm@fau.edu)

ABSTRACT In the current era of digital communication, image security is a top priority as sending images, which often contain personal or confidential information, over the Internet poses security risks. Many encryption algorithms have been proposed in the literature to address this issue. However, these algorithms have limitations, such as low-key space, high computational overhead, and susceptibility to differential attacks. To address these limitations, this paper proposes Z-Crypt, a novel image encryption approach that combines a substitution-permutation network (SPN) and a chaotic logistic map (CLM) with the Chirp Z-Transform (CZT) to enhance security and resist attacks. The CLM-generated matrices introduce confusion and diffusion, while SPN creates a dissociation of the cipher from the plaintext. Finally, the CZT strengthens the encryption by transforming the image into the frequency domain, creating multiple layers of confusion and diffusion to produce a robust encryption algorithm. We have evaluated the proposed Z-Crypt's security using various metrics, including correlation coefficient, entropy, peak signal-to-noise ratio (PSNR), and key sensitivity analysis. Experimental results verify that the proposed algorithm outperforms existing methods, achieving high security while maintaining computational efficiency.

INDEX TERMS Chaotic logistic map, chirp Z-transform, efficiency, image decryption, image encryption, robustness, security, substitution-permutation network.

I. INTRODUCTION

In recent times, users' privacy has emerged as a crucial security concern, particularly while dealing with data shared over the Internet or other publicly accessible communication channels [1]. These privacy issues hold equal significance for digital images, as they often contain sensitive information that must be protected from leakage, such as military, medical, and private online images. Encryption is key for maintaining information security, and numerous encryption algorithms have been discussed in the literature.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed Elhoseny.

The foundational work by Robert Matthews [2] introduced the chaotic encryption method [3], which revolutionized the approach to secure image transmission. Chaotic systems provide an ideal foundation for cryptosystems designed to protect digital images. Chaotic systems are characterized by their ergodic nature and sensitivity to initial conditions and parameters, contributing to their inherent unpredictability. As a result, chaotic systems are suitable for building cryptographic algorithms [4]. However, if an observer knows the initial conditions of a chaotic system, its behavior becomes predictable, thereby posing a security concern that needs to be addressed. Digital images consist of two-dimensional arrays that contain pixel values and exhibit significant

redundancy because of the correlated nature of adjacent pixels [5], [6], [7]. Depending on whether they are grayscale or color images, they have one, three, or four channels, including a transparency channel. To enhance security, two key techniques are employed in image encryption: confusion and diffusion. Confusion involves substituting pixel values using a cryptographic key. Diffusion, conversely, ensures that modifying a single pixel in the plaintext image affects approximately half of the total pixels in the cipher image, thereby significantly diminishing the correlation between the pixels in the plaintext image and the pixels in the cipher image [8]. Employing confusion and diffusion techniques enhances security significantly and is integral to image encryption algorithms [9].

Among various cryptographic techniques, the substitution-permutation network (SPN) is a widely adopted technique for image encryption across multiple applications, including digital rights management, medical imaging, and military communications [10], [11]. The SPN technique is an integral part of many block cipher algorithms, including the Advanced Encryption Standard (AES), and it uses a combination of substitution and permutation operations. These operations are performed on blocks of pixels, and the encrypted blocks are then reassembled to create the final encrypted image. The security of SPN is further enhanced by using complex and unpredictable S-boxes and P-boxes [4], [12].

In addition to SPN, the Chirp Z-Transform (CZT) is another encryption tool that offers several advantages over the conventional Discrete Fourier Transform (DFT). CZT provides higher resolution frequency analysis and the ability to multiplex image patches [13]. The CZT operates by sampling the Z-plane along a logarithmic spiral contour instead of the equally-spaced points around the unit circle used by the DFT. This makes the CZT well-suited for image processing and encryption applications, where it can be used to achieve high-resolution frequency analysis, interpolation, and dynamic multiplexing of images [14].

Despite significant advancements, existing image encryption methods face challenges such as vulnerability to differential attacks, computational overhead, and limited key space. Addressing these limitations, a novel SPN-CLM technique Z-Crypt based on CZT has been developed in this research. The proposed encryption technique employs SHA-2 256 for hashing a key from the plaintext image and also incorporates a pre-shared secret key. Chaotic matrices are utilized for confusion and diffusion, and the technique employs both SPN for substitution and CZT for high-resolution frequency-domain conversion and encryption. These measures in the proposed algorithm enhance the security and resistance to attacks significantly.

This paper presents the following novel contributions:

1) A highly secure encryption scheme, Z-Crypt, is presented based on CLM and SPN. The CLM algorithm serves as a pseudorandom generator, which helps add a high diffusion level to the encrypted image. Conversely, the SPN algorithm adds non-linearity to the image encryption process, creating a high level of confusion. As a result, the encrypted image exhibits negligible similarity to the original plaintext image (as verified through our experimental results). Through analysis, it has been observed that the proposed image encryption algorithm Z-Crypt satisfies a stringent avalanche condition, wherein even a single bit change in the plaintext image can create an avalanche effect (AE), causing approximately 50% of the bits to change in the cipher image. This makes our proposed algorithm highly resistant to advanced attacks.
2) In addition to CLM and SPN, we introduce CZT, which allows for the conversion of plaintext data into the frequency domain. The conversion to the frequency domain can help in modifying the spectral characteristics of the cipher. Furthermore, the use of CZT increases the complexity of the association between the cipher and the plaintext data, enhancing resistance to differential statistical attacks.
3) The performance of our proposed Z-Crypt has been evaluated by benchmarking it against multiple state-of-the-art schemes available in the literature. The results demonstrate that Z-Crypt is superior in terms of both encryption efficiency and security.

The rest of the paper is structured as follows. In Section II, we introduce image encryption schemes and related work. We then elaborate on the steps involved in our proposed Z-Crypt in Section III, followed by an analysis of its security in Section IV. Results are presented in Section V. Finally, we conclude the paper in Section VI.

II. BACKGROUND AND RELATED WORK

This section first discusses two different image encryption foundational techniques: SPN and CLM. Afterwards, this section introduces CZT. Finally, this section reviews the current state of research on image encryption.

A. THE SUBSTITUTION-PERMUTATION NETWORK

The creation of a round function in a block cipher relies heavily on two key properties: diffusion and confusion. Typically, SPN enables layers of diffusion and confusion to develop secure and efficient block ciphers [15]. The SPN architecture comprises of two fundamental operations: substitution, which entails the effective use of S-boxes, and permutation, represented by P, which rearranges individual bits or groups of bits [16]. The substitution layer is responsible for a non-linearity of the SPN that executes a one-to-one transformation on the Galois field represented as $F_{2^n}^m$. This field F is essentially a Galois field modulo 2, with n representing the number of bits and m the number of pixels to be transformed. This transformation is achieved by using m parallel S-boxes, each of which operates on F_{2^n}. In other words, the input $S(x_0, x_1, \ldots, x_{m-1})$ maps to an output of the form $(s_0(x_0), s_1(x_1), \ldots, s_{m-1}(x_{m-1}))$.

Conversely, the diffusion layer entails a reversible linear transformation P, defined over $F_{2^n}^{m\times m}$, where $m\times m$ represents a square patch of m pixels. This linear operation mixes the connections between sub-blocks, thereby making a single pixel in the plaintext to have a large influence over the entire ciphertext. Mathematically, SPN can be represented as:

$$X_{i+1} = K_i \oplus P(S(X_i)), \tag{1}$$

where X_i represents the current state, $S(X_i)$ is the substitution operation using S-boxes, P represents the permutation operation, and K_i is the key used for the bitwise XOR operation ($\oplus$). This equation describes how the state X_i is transformed to X_{i+1} in the SPN. The flowchart in Figure 1 shows the steps of our SPN diagram.

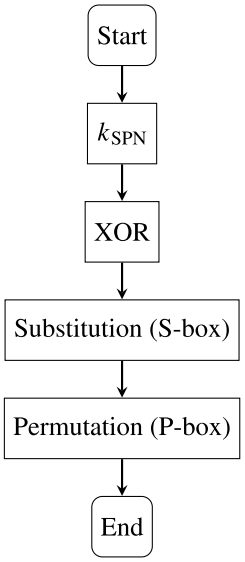


FIGURE 1. The Substitution-Permutation Network (SPN) Flowchart.

B. THE LOGISTIC CHAOTIC MAP

The logistic chaotic map is a mathematical model of population growth that was initially introduced by Robert May in 1975 [4]. It is described as a one-dimensional discrete dynamical system defined by the following equation:

$$x_{n+1} = \alpha x_n(1 - x_n), \tag{2}$$

where x_n represents the population at time step n, and α is a parameter that controls the growth rate of the population. Despite its simplicity, the logistic chaotic map exhibits many complex behaviors, including chaos. Chaos refers to a dynamic behavior known for its extreme sensitivity to initial conditions and unpredictability over the long term behavior.

The logistic chaotic map operates as a discrete dynamical system that describes a system's evolution over time in discrete steps. It can be analyzed using various mathematical tools, including bifurcation theory, chaos theory, and ergodic theory [17]. The Hopf bifurcation diagram, as shown in Figure 2, can be used to analyze the logistic map for the range [3.57, 4] having $x_0 \in [0, 1]$ as the initial condition and $\alpha \in [0, 4]$ control parameter. In this range, the map is observed to be chaotic for α, and little changes in the initial value result in significant variations in the randomly produced values, which follow a sequence that is neither periodic nor converging.

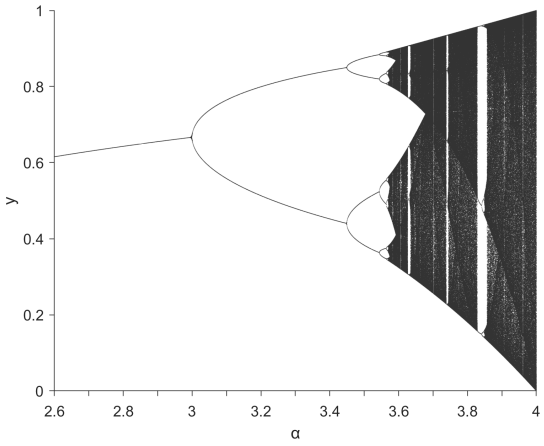


FIGURE 2. Logistic map bifurcation diagram.

C. THE CHIRP Z-TRANSFORM

The CZT is a mathematical technique related to the DFT. Unlike the DFT, which samples the Z-plane along a unit circle, the CZT samples along a logarithmic spiral contour in the Z-plane. This allows for more efficient computation of the Z-transform for signals that are concentrated in a narrow frequency band. The term CZT is named due to its basis in a process similar to the 'chirp' signal used in radar systems, where a frequency-modulated signal sweeps through a range of frequencies. The CZT uses high-speed convolution techniques, making it more flexible than the fast Fourier transform (FFT) and allowing for arbitrary contour shapes and spacing. This flexibility in evaluating the Z-transform at equiangular points is central to its naming and functionality [21].

The mathematical operation for Z-transform for a finite sequence $x(n)$ $(0 \le n \le N-1)$ can be represented as [19] and [21]:

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}, \tag{3}$$

To compute Z-transform, we can choose a trajectory within the Z-plane and then extract a portion of the spiral within the Z-plane using equally spaced angles for sampling. The sampled points are represented as z_k and can be represented as follows:

$$z_k = AW^k \quad (k = 0, 1, \ldots, M-1), \tag{4}$$

where $A = A_0 e^{j\theta_0}$, $W = W_0 e^{-j\varphi_0}$, and M denotes the total number of sampling points. A establishes the starting position of the sampling trajectory, characterized by its radius A_0 and

phase angle θ_0. Typically, A_0 should be less than or equal to 1 to ensure that z_0 remains within the unit circle. The parameters A and W determine the shape of the logarithmic spiral path and the positioning of samples along it, as shown in Figure 3 [18]. W serves as the spiral parameter, with W_0 indicating the spiral's extension or contraction rate. When W_0 is greater than 1, the spiral contracts inward as k increases, while if W_0 is less than 1, the spiral extends outward as k increases. φ_0 represents the angular spacing between the sampling points. Given the flexibility in picking φ_0, a limited number of parameters can enhance the frequency resolution as the value of φ_0 can directly impact the values of A_0 and W_0. If A_0 and W_0 are properly selected, it can be seen that ω_0 will have a large impact on the contour spacing. This is because, as it is exponential, a large value will significantly reduce the value of W. Raising the power of k, AW^k, will also reduce it further. The spiral contracting inward has the impact of reducing the frequency resolution. Therefore, a large sample point will cause a large error, rendering the ICZT impossible to compute. With the expression $z_k = AW^k$, the Z-transform values at these sampling points are given as follows:

$$X(z_k) = \sum_{n=0}^{N-1} x(n) A^{-n} W^{-nk} \quad (k = 0, 1, \ldots, M-1), \quad (5)$$

Ultimately, $W^{-(k-n)^2/2}$ corresponds to an M-by-N Toeplitz matrix represented as $W^{-(k-)^2/2}$ [18]. Given that $W = P\hat{W}Q$, the CZT algorithm can be seen as an efficient way to compute the following matrix equation:

$$X(z_k) = P\hat{W}QAx, \quad (6)$$

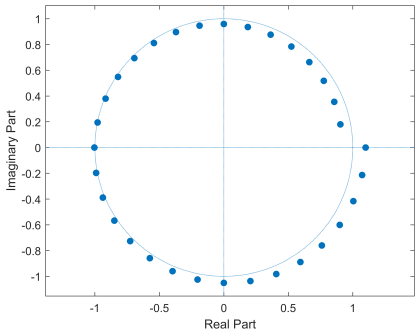
where $\hat{W}$ is an $M \times N$ Toeplitz matrix, and P, Q, and A are diagonal matrices. A formula for the inverse Chirp Z-transform (ICZT) is only applicable when M equals N, and it can be obtained by reversing the matrices in Equation (6), which means:

$$x = A^{-1}Q^{-1}\hat{W}^{-1}P^{-1}X, \quad (7)$$

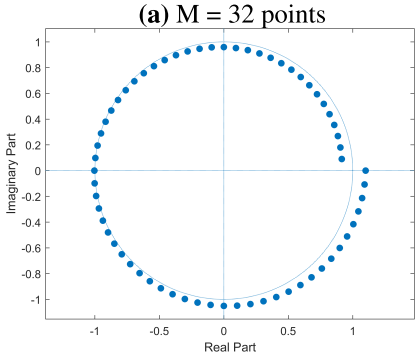
D. RELATED WORK

Several techniques are available in the literature for image encryption based on chaos. However, each design has its own advantages and disadvantages. For instance, Alanezi et al. [20] have introduced an image encryption method based on two logistic maps, a common approach reflecting a standard practice in chaotic cryptography. Nonetheless, this encryption scheme requires enhancements in its security features. Similarly, Arif et al. [22] provide further insights into the effectiveness of various encryption techniques. However, it is still unknown how Josephus's traversing and the chaotic system improve the security and performance of the algorithm and resist more advanced attacks.

Lu et al. [23] have introduced a method utilizing a compound chaotic system and a single S-Box to achieve



(a) M = 32 points



(b) M = 64 points

FIGURE 3. Chirp contour in (a) M = 32 points and (b) M = 64 points.

confusion and diffusion in encryption, which demonstrates resistance to chosen-plaintext attacks and provides notable efficiency advantages. Specifically, the use of the CZT to perform the XOR operation between the chaotic key stream and the plaintext image enhances the encryption process, making it more efficient and faster. XOR operations are inherently fast due to their simple bitwise nature, which contributes to the overall efficiency of the encryption process. Building upon this work, Niu and Zhang [24] have proposed a novel encryption scheme that combines Josephus traversing, a chaotic system, bitwise XOR, crossover, and cipher feedback to achieve confusion and diffusion. This method is key-sensitive and can withstand common attacks. However, it remains unclear how the integration of Josephus traversing and the chaotic system specifically enhances the security and performance of the algorithm, particularly in resisting more advanced attacks. Further investigation is needed to understand the impact of these techniques on the robustness and efficiency of the encryption scheme.

In an effort to enhance speed, Alghamdi et al. [25] have proposed a new image encryption algorithm based on logistic map that exhibited performance (execution time) improvement of up to 15× as compared to other contemporary encryption methods. Similarly, Alenezi et al. [26] have proposed the Z-Transformation Encryption (ZTE) technique based on Z-transform and XOR function. While the scheme exhibits promising throughput compared to other symmetric algorithms. However, the algorithm's complexity and determining the optimal Z-value for the transformation for

different applications still pose a challenge. Recently, Mondal and Singh [27] proposed a lightweight encryption scheme merging chaotic maps with diffusion circuits, allowing for efficient permutation and substitution within a single image scan, though questions about its security remain.

Javeed et al. [28] have further explored and proposed a new approach for securing plaintext images. They used a chaotic oscillator that relies on a second-order differential equation to generate a sequence of random numbers, which introduces confusion and diffusion in the image. They have used this sequence to scramble the pixels of the original image. Muhammad and Özkaynak [29] have further explored and developed a scheme that utilizes chaotic systems and cryptographic primitives to enhance resistance to application-specific attacks, demonstrating notable efficiency in securing digital images.

Despite the advantages of the presented encryption algorithms, including their resilience against some specific attacks, there is a pressing need for improvements to ensure robust security capable of withstanding advanced threats. Additionally, the new encryption schemes should maintain a balance between security and performance. While efficiency is crucial, neglecting security can lead to potential vulnerabilities and data breaches. Therefore, there is an increasing demand for new encryption schemes that strike a balance between performance and security. This can be achieved by utilizing SPNs and the CZT, which are more resistant to advanced attacks while maintaining performance efficiency.

III. METHODOLOGY

This section provides details of the workflow of the proposed CZT-based image encryption algorithm Z-Crypt utilizing SPN-CLM technique based on CZT. It also explains how image encryption and decryption are performed using our proposed algorithm.

A. PROPOSED IMAGE ENCRYPTION ALGORITHM

The proposed Z-Crypt is designed to achieve high security and attack resistance levels by combining SPN, CLM, and CZT. The algorithm requires a plaintext image and a pre-shared secret key as input and produces a cipher image as output. The flowchart of Figure 4 provides an overview of the encryption process of Z-Crypt.

First, Z-Crypt generates a hash from the plaintext image using SHA-256. The cryptographic hash function SHA-256 is utilized in many security applications owing to its efficient computation and resilience against several known attacks. It is a suitable option for protecting sensitive image data [30].

The generated hash from the plaintext image is then XORed with the pre-shared secret key to produce a derived key. A random 256-bit keystream is produced and used as the initialization vector (IV), which is then XORed with the derived key to generate the unique key. The unique key is divided into two halves: $key1$ and $key2$, each containing 128 bits, which are passed through mod 0.9999 to make them suitable for use as initial values in the chaotic map later on. The resulting keys are then used to generate two matrices, $M1$ and $M2$, using the chaotic map given in Equation (2). $M1$ contains $r + c$ elements where r is the number of rows of the image and c is the number of columns of the image.

A chaotic permutation of rows is applied to the plaintext image, afterwards, using the first r elements of $M1$, where r is the number of rows of the plaintext image. The substitution-permutation network (SPN) transformation is then applied to obscure the pixels and add another layer of encryption to further increase the security. The SPN consists of a substitution and permutation box, where the substitution box replaces each pixel value in the image with a new value based on the corresponding entry in the box, and the permutation box rearranges the outputs of the substitution box. The SPN box is used to transform the image C by computing the output of each pixel value in C using the following expression:

$$D = \left\{ S_{\text{SPN}}(c_{ij}) \mid c_{ij} \in C \right\}, \tag{8}$$

where D is the resulting image after applying the SPN box, and C_{ij} is the (i, j)-th pixel value in image C.

After SPN transformation, the chaotic column permutation is applied to the image pixels using the remaining elements of the matrix $M1$ corresponding to the number of columns in the image by iteratively applying the following formula:

$$P_k = \sum_{i=0}^{n-1} M_{ik} P_{k-1}, \tag{9}$$

where P_k is the (k)-th iterate of the image pixels, M_{ik} is the (i, k)-th element of the transition matrix M, and n is the number of pixels in the image. This process improves the obfuscation, confusing the attackers' ability to distinguish the original plaintext image. Following this, an XOR operation subjects the obscured image to a chaotic column permutation using $M2$. The XOR operation is applied to the resulting permuted image D by computing the bitwise XOR of each pixel value in D with a random binary mask A.

$$F = D \oplus A, \tag{10}$$

where F is the resulting image after applying the XOR operation, and A is a random binary mask $M2$ generated. Finally, the CZT is applied to introduce non-linearity and enhance the encryption. The CZT is a complex function that generates a pseudo-random sequence of integers, which are then used to modify the image's pixel values. The CZT is performed by transforming samples of 256×256 pixels at a time. This is done to reduce the transformation error involved in the operation [18]. For color images, the same process is applied; however, CZT is performed on each channel separately for the color images. This transform is difficult to invert without the correct key as a single digit difference will generate a random sequence of data that has no similarity with the plaintext image. The CZT is applied to the image

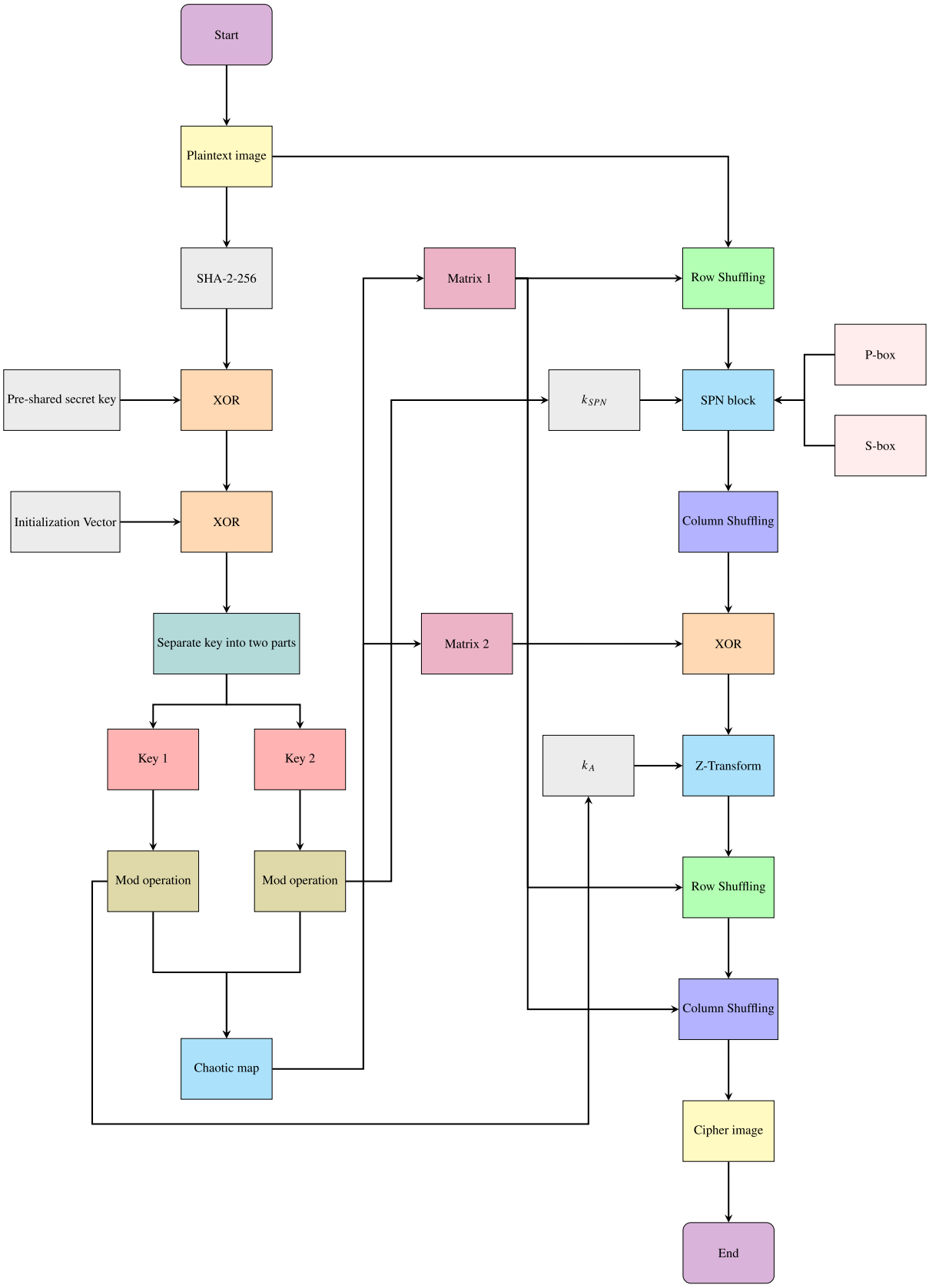


FIGURE 4. Flowchart of the proposed image encryption algorithm Z-Crypt.

F using the formula given in Equation (11) to produce the output image G.

$$E = \{X(d_{ij}) \mid d_{ij} \in G\}, \tag{11}$$

In the cipher image, the CZT creates a new pattern that is completely different from the plaintext image yet still undesirable. To remove this pattern, matrix M_1 scrambles the image again by performing the row and column permutations, which produces the final cipher image C.

To decrypt the cipher image encrypted by Z-Crypt, the first step is to perform reverse column and row permutation. This will yield the Z-transformed image. Then, an inverse Z-transform operation is performed using the inverse CZT given in Equation (7). As in the encryption process, the key initialization is performed using the matrices M_1 and M_2. After the inverse Z-transform, the resulting image is XORed with the matrix M_2. The XORed output is then subjected to the inverse column permutation, and the result is passed through the reverse SPN transformation. Finally, reverse row permutation is performed, and the resulting image is decrypted.

1) STEPS TO PERFORM ENCRYPTION

The encryption steps of Z-Crypt are discussed below:

1) Start by reading the plaintext image P.
2) Compute a hash of the plaintext image P using the SHA-2 256 algorithm (Algorithm 1, line 2).
3) Generate a 256-bit keystream to act as an initialization vector IV.
4) Create a 256-bit key by XORing the image hash with a pre-shared secret key k_p.
5) XOR the key generated in the previous step with the initialization vector IV to create a unique key (Algorithm 1, line 4).
6) Partition the result obtained in the previous step into two halves, each 128-bit.
7) Map the two halves to the range [0, 0.9999] by using modulus operations to convert to hexadecimal to 0-0.9999. The two values are stored as Key 1 and Key 2, respectively. Also, generate the k_A by taking the first 8-bits of the hexadecimal numbers, applying modulus with 0.2, and then adding 0.9. This process results in a range of values from 0.9 to 1.1.
8) Use $Key1$ and $Key2$ as the initial parameters for the chaotic system's chaotic map to generate Matrix 1 ($M1$) and Matrix 2 ($M2$). The number of rows in $M1$ is r, and the number of columns in $M1$ is $r + c$. Matrix 2 ($M2$), on the other hand, has $(r * c)$ elements (Algorithm 1, line 9-11).
9) Use the first r elements of $M1$ to perform chaotic row permutation on the plaintext image P Algorithm 1, line 12).
10) Run the SPN on the shuffled result from Step 9 using Key 2 (k_{SPN}) as the key to the SPN module (Algorithm 1, line 13).
11) Use the remaining c elements of $M1$ to perform column permutation on the result obtained in Step 10 (Algorithm 1, line 14).
12) XOR Matrix 2 ($M2$) with the resulting permuted image (Algorithm 1, line 15).
13) Use the Z-transform to transform the resulting image to produce the transformed cipher (Algorithm 1, line 16).
14) Perform row and column permutations using Matrix 1 to scramble the transformed image to remove correlation and produce the cipher image C (Algorithm 1, line 17-18).

2) STEPS TO PERFORM DECRYPTION

The decryption process steps for the proposed algorithm are detailed as follows:

1) Receive the cipher image C and the SHA-2 256 hash.
2) Reconstruct $Key1$, $Key2$ and k_A by repeating Steps 3 to 7 of the encryption process.
3) Recover the transformed cipher by performing a reverse row and column shuffle using $M1$.
4) Recover the untransformed cipher by executing an inverse Z-transform.
5) Execute an XOR operation between the matrix $M2$ and the resulting cipher.
6) Perform a chaotic reverse column permutation on the cipher from Step 5.
7) Use the reverse SPN to replace the pixel values in the permuted cipher.
8) Finally, utilize $M1$ for the reverse chaotic row permutation on the resulting substituted cipher to produce the decrypted image I.

B. THE MATRIX GENERATION PROCESS

Matrix ($M1$) values are generated using the logistic map in the proposed algorithm 1. Let's assume that $Key1$ serves as the initial value, which is 0.45678. The generation of the matrix follows a recursive method of using the previous value to compute the new value. The first value of X is computed using $Key1$ as $0.45678 \times 3.99876 \times (1-0.45678) = 0.99222$ where 3.99876 is α as described in Equation (2). In the second iteration, the generated value of X from the previous iteration is used in the logistic map to compute the next value of X as $0.99222 \times 3.99876 \times (1 - 0.99222) = 0.03087$. The next value of X is calculated using the last computed value as $0.03087 \times 3.99876 \times (1-0.03087) = 0.11963$. Similarly, other values of X are calculated following the methodological steps.

To obtain the values of $M1$, the value of X is multiplied by 10^6 and a mod operation is performed. The first value of $M1$ is computed as $0.99222 \times 10^6 \text{mod} 256$, where 256 is assumed to be the number of rows and columns in the plaintext image. Table 1 shows the first 15 iterations of the matrix generation process.

From Table 1, it can be seen that the values of X change rapidly over time because of the chaotic nature of the logistic

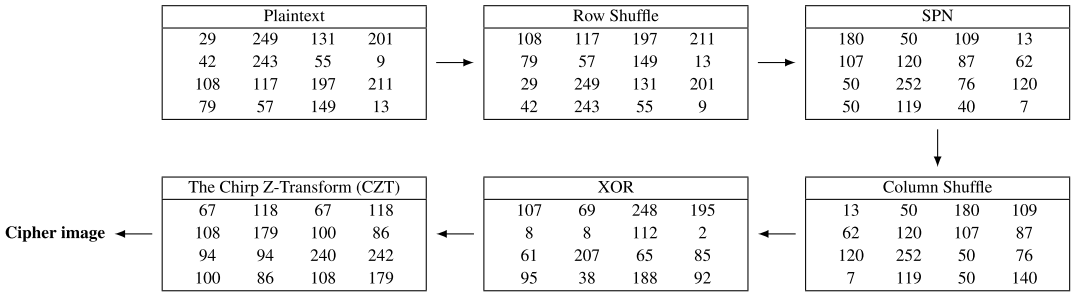


FIGURE 5. 4 × 4 sample is processed using the proposed algorithm.

TABLE 1. The process of matrix generation (M1) from the logistic map.

Iteration	X	M1	Iteration	X	M1	Iteration	X	M1
1	0.99222	221	6	0.09822	174	11	0.48575	115
2	0.03087	147	7	0.35419	139	12	0.99888	222
3	0.11962	67	8	0.91467	239	13	0.00448	132
4	0.42111	245	9	0.31210	34	14	0.01785	183
5	0.97480	211	10	0.85851	137	15	0.07009	205

map. However, the values of $M1$ remain within the range 0–255 because we apply the modulus operation to the values of X before using them to generate $M1$. For instance, Figure 5 shows an example where we apply the proposed encryption process to a 4×4 image.

IV. SECURITY ANALYSIS OF THE PROPOSED ALGORITHM

This section provides an analysis of the security components employed in the algorithm, including key scheduling, row and column permutation, SPN, CLM, XOR, and CZT. The purpose of this analysis is to assess the effectiveness of each component in ensuring the proposed encryption algorithm's security.

1) KEY SCHEDULE

Two main steps are performed in the key schedule to ensure the image is securely encrypted. The plaintext image undergoes a 256-bit hash function operation. An XOR operation is then performed between the bits of the pre-shared key and the hashed image. The resulting output is further XORed with an initialization vector to increase the likelihood of generating a unique final key and also so that the encryption of the same plaintext image multiple times gives a completely different cipher image each time. The goal is to make it extremely challenging for an attacker to guess the key in encrypting the image through brute force or side-channel attack. Since the initial key is generated using an XOR operation, the key space for the proposed algorithm is $2^n = 2^{256}$. This indicates that a key of n bits provides a security level of *256* bits for the proposed image encryption algorithm.

2) CHAOTIC LOGISTIC MAP

The chaotic logistic map has shown high security due to its ability to generate highly pseudorandom numbers. In addition, it generates changes in completely different patterns with little change in its initial condition. Due to these robust security features, it has been used to generate the keys (matrices) for the row and column permutation of the image pixels.

3) PIXEL-BASED PERMUTATION

A robust security system will have some level of confusion that ensures the resulting data bits are highly uncorrelated with the initial values, which can help resist differential attacks and chosen plaintext attacks. These permutation operations work by altering the pixel positions of the input data based on the permutation matrix. In our proposed algorithm, this matrix is generated using the CLM. Due to the randomness of the permutation matrix, there are $(H \times W)!$ possible results for the row and column permutation operations when performed together. Considering that the security level of m bits is impacted by the permutation is $2^m = (H \times W)!$ expresses the impact of the permutation on the affected bits, which shows that $m = \log_2(H \times W)!$ is the security level of the pixel-based permutation [31].

4) THE XOR OPERATION OF THE PLAINTEXT

The algorithm's security is further strengthened by incorporating the bitwise XOR operation with a matrix generated by CLM. A significant advantage of using XOR is that the operation is nonlinear and cannot be expressed as a combination of its input bits. This assists mitigate linear attacks and ensures that the output bit depends on all the input bits, which makes it difficult for an attacker to gain information by observing only parts of the input bits. If the image size is $(H \times W)$ and all pixels are equally impacted, then the security level is $(H \times W \times b)$, where b is the number of bits in a pixel.

5) SUBSTITUTION-PERMUTATION NETWORK (SPN)

The substitution and permutation network helps add confusion and diffusion properties to the system. The pixel-based permutation itself does not guarantee confusion, as the resulting cipher still has some statistical similarities with the input data. As an example, for a 4×4 image, if the only operations performed on it are row and column permutations, it is expected that the pixels that make up the image are the same, but their ordering is different. Adding a

Algorithm 1 Proposed Image Encryption Algorithm

Input: Plaintext image P, pre-shared key k_p, Initialization Vector IV
Output: Cipher image C

$[H, W, n_c] \leftarrow \text{size}(P)$
// Reshape image into $H \times W \times n_c$, n_c is number of channels
$H \leftarrow \text{hash}(P)$
$Key \leftarrow H \oplus k_p$
$Key \leftarrow Key \oplus IV$
$halfLen \leftarrow \text{length}(Key)/2$
$k_{SPN} \leftarrow Key(1 : halfLen)$
$Key1 \leftarrow Key(1 : halfLen) \mod m$
// where m is 0.9999
$Key2 \leftarrow Key(halfLen : \text{end}) \mod m$
$k_A \leftarrow Key(1 : 8) \mod 0.2 + 0.9$
for $n \leftarrow 1$ to H **do**
  $x_1(n+1) \leftarrow r \times x_1(n) \times (1 - x_1(n))$
  // where $x_1(1)$ is $Key1$, r is 3.99879
  $M1(n) \leftarrow x_1(n+1) \times 10^6 \mod H$
end for
for $n \leftarrow H + 1$ to $H + W$ **do**
  // For width
  $x_1(n+1) \leftarrow r \times x_1(n) \times (1 - x_1(n))$
  $M1(n) \leftarrow x_1(n+1) \times 10^6 \mod H$
end for
for $n \leftarrow 1$ to $H \times W$ **do**
  $x_2(n+1) \leftarrow r \times x_2(n) \times (1 - x_2(n))$
  // where $x_2(1)$ is $Key2$
  $M2(n) \leftarrow x_2(n+1) \times 10^6 \mod 256$
end for
$Encrypt \leftarrow \text{rowShuffle}(P, M1(1 : H))$
$Encrypt \leftarrow \text{SPN}(Encrypt, k_{\text{SPN}})$
$Encrypt \leftarrow \text{columnShuffle}(Encrypt, M1(H + 1 : H + W))$
$Encrypt \leftarrow Encrypt \oplus M2$
$Encrypt \leftarrow \text{CZT}(Encrypt, k_A)$
// where k_A is from 0.9 to 1.1
$Encrypt \leftarrow \text{rowShuffle}(Encrypt, M1(1 : H))$
$C \leftarrow \text{columnShuffle}(Encrypt, M1(H + 1 : H + W))$
return C

form of substitution operation between the row and column permutation can help ensure a statistical difference between the input and the output data. Also, the substitution and permutation operations in the SPN network introduce an apparent effect of randomness.

The SPN contains three primary operations applied to the data. These are the substitution, permutation, and XOR operations. The substitution, much like the XOR, operates on the individual bits by changing the value of bits in a pixel of data. All the bits are likely to be changed, and the changes are limited to the number of possible combinations. The security for the substitution cipher is $(H \times W) \times 256^n$, where n is the key size used. The P-box involves replacing the order of the pixels in the image by changing their spatial positions. For a permutation involving all bits, the security level imparted is given as $2^m = (H \times W)!$. For the P-box, $m/2$ bits are impacted such that the security level becomes $2^{m/2} = (H \times W)!$.

Thus, it indicates that $m = 2\log_2[(H \times W)!]$ for the P-box operation. The security level for the XOR operation is as previously described. Hence, the security level of the SPN is estimated as $(H \times W) \times 256^n + 2\log_2[(H \times W)!] + (H \times W \times b)$.

6) THE CHIRP Z-TRANSFORM (CZT)

The use of the CZT transformation helps fully transform the input data from the original time domain into the frequency domain. This makes the resulting cipher become random to any statistical tool as it has no relationship with the plaintext. The transformation to the frequency domain establishes an entirely different relationship between the pixels of the data. This, coupled with the Strict Avalanche Criterion (SAC) established by the previous operations, helps resist differential attacks. The CZT operates on individual pixels in sequential order of $H \times W$. The frequency domain transformation for a 2D image with encryption creates a complexity of $m = \log_2\left[(H \times W) \times e^{(q \cdot B)}\right]$, where q is the number of patches to be transformed and B is the number of bits of transformation key. In our case, each patch has a size of 256×256. Therefore, only one transformation is required for an image of size 256×256. Similarly, for an image of size 1024×1024, four transformations are required. In the case of a color image, the number of transformations required is $3 \times I_s/(256 \times 256)$, where I_s denotes image size.

7) COMPLETE SECURITY ANALYSIS OF OUR PROPOSED IMAGE ENCRYPTION ALGORITHM

To conduct a thorough security assessment of our proposed encryption algorithm, the security evaluations of the Z-Crypt algorithm operations are aggregated. The comprehensive security analysis, represented by $\mathscr{L}$, of the proposed algorithm is detailed as follows:

$$\begin{aligned}\mathscr{L} = {} & n + 256^n \times (H \times W) \\ & + 2\log_2[(H \times W)!] \\ & + 2 \times (H \times W \times b) \\ & + \log_2\left[(H \times W) \times e^{(q \cdot B)}\right] \text{ bits,}\end{aligned} \tag{12}$$

where $n = 256$ is the length of the key, H is the height of the image, W is the width, b is the number of bits in a pixel, q is the number of individual transformations made and l is the number of pixels transformed at a time. In this case, q corresponds to the row of a 2D matrix, and l corresponds to the column.

V. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

In this section, we discuss the experimental results of the proposed Z-Crypt encryption algorithm using a number of grayscale and color images obtained from the USC-SIPI

Image Database [32]. These images have been utilized as plaintext inputs, with dimensions of 256 × 256, 512 × 512, and 1024 × 1024 pixels, and encompassing both grayscale (8-bit) and color formats (24-bit). To create permutation matrices, we have used specific parameters for the logistic maps ($r = 3.99876$, $m = 0.9999$). Figure 6 showcases all the original plaintext images alongside their corresponding encrypted images.

Moreover, we have evaluated and compared our proposed Z-Crypt algorithm with methods found in literature using different metrics. Furthermore, we have assessed the randomness characteristics of Z-Crypt using the NIST SP 800-22 test suite. We have conducted experiments on a computer with an Intel(R) Xeon(R) CPU E5-1620 v4 running at 3.50 GHz outfitted with 32.0 GB of random access memory (RAM) and running Windows 10 Pro, Version 22H2. We have used MATLAB R2023b for obtaining and analyzing results.

A. ENCRYPTION PERFORMANCE

The computational efficiency of Z-Crypt depends on the operations involved in the CZT and the SPN. For an image of size $H \times W$, the CZT requires $O(H \times W \times \log(H \times W))$ operations, primarily due to the frequency domain transformation. Similarly, the SPN adds an additional $O(H \times W)$ operations for each encryption round.

The space and computational complexity of the algorithm is proportional to the size of the input image, with added overhead for storing intermediate transformation results and encryption keys. As an image resolution increases, both time and space complexities scale accordingly, potentially presenting challenges in resource-constrained environments. For instance, encrypting a 1024 × 1024 image will demand significantly more computational power and memory compared to a 256 × 256 image. As detailed in Section III-A, if the image size is 1024 × 1024, our solution involves partitioning the image into 4 patches of 256 × 256 size to efficiently manage memory and enhance scalability while ensuring security.

Table 2 depicts Z-Crypt's encryption time. We have analyzed the algorithm using images of various sizes, such as 256 × 256, 512 × 512, and 1024 × 1024. Table 3 compares the performance of Z-Crypt with other existing methods. However, as the code of the existing methods are not publicly available, we used the scale method to predict the estimated speed of their algorithms on our machine [20]. The results show that Z-Crypt performs better than most of the other encryption algorithms in the literature. Furthermore, the performance (execution time) of Z-Crypt is very close to lightweight encryption schemes.

B. KEY SPACE AND SENSITIVITY ANALYSES

The security of a cryptographic system depends on the size of its key space. A large key space, with a minimum threshold of 2^{100} is needed to prevent brute force attacks [33]. Z-Crypt uses a 256-bit hash function (SHA-2 256) to process the input image, along with a 256-bit pre-shared secret key. These two components are combined using a bitwise XOR operation, resulting in a 256-bit derived security key that serves as an input for the logistic map. The key space of Z-Crypt spans 2^{256} possibilities, providing a breadth sufficient to withstand brute force attacks. Table 4 shows the key space of Z-Crypt compared to those of other methods.

TABLE 2. Z-Crypt's average encryption time, measured in seconds, is assessed across various image dimensions.

Image Size	Type	Time (second)
512×512	Color	0.7075
256×256	Gray	0.0597
1024×1024	Gray	0.9609
512×512	Color	0.7237
512×512	Gray	0.2479
256×256	Color	0.1868

TABLE 3. Comparison of average encryption time in (seconds) for a 256 × 256 grayscale image using our proposed Z-Crypt algorithm and other methods.

Methods	Time (sec)
Proposed	0.0597
[25]	0.0235
[20]	0.0494
[23]	0.3820
[56]	0.7091
[46]	0.7360
[24]	1.2680

TABLE 4. Z-Crypt's key space comparison with other methods.

Cryptosystem	Key space
Z-Crypt	2^{256}
[25]	2^{256}
[23]	2^{124}
[44]	2^{94}
[42]	2^{219}
[43]	2^{187}

Z-Crypt is highly sensitive to even the slightest change in the secret key. As shown in Figure 7, the algorithm generates distinct cipher images even if the key differences are only one bit. To assess key sensitivity, we use two keys, K_1 and K_2, that differ by just one bit to encrypt the 256 × 256 image 'Female'. This produces two wholly dissimilar cipher images, C_1 and C_2. Deciphering C_1 with K_2 or C_2 with K_1 did not yield the plaintext image. These results demonstrate that it is impossible to reverse the cipher image to obtain the plaintext image using a key that has been altered by a single bit, thus ensuring that Z-Crypt generates unique cipher images when a plaintext image is encrypted with two keys that differ by only one bit.

C. STATISTICAL ANALYSIS – HISTOGRAM ANALYSIS

A histogram displays the distribution of pixel values in an image by dividing the range into bins and counting the number of pixels in each bin. By analyzing the histogram,

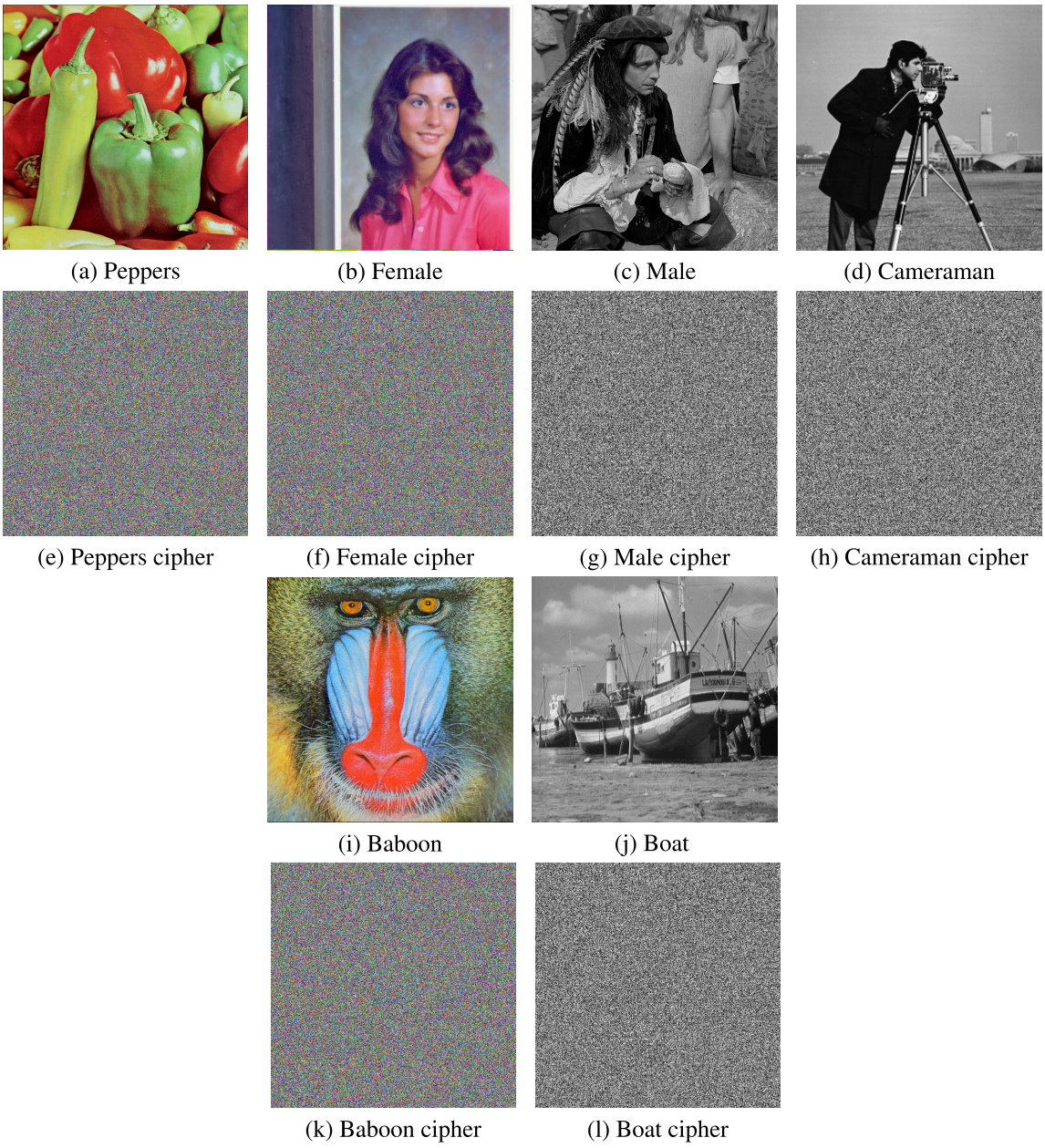


FIGURE 6. The plaintext images and their corresponding encrypted ciphers: (a) Peppers (b) Female (c) Male (d) Cameraman (e) Peppers cipher (f) Female cipher (g) Male cipher (h) Cameraman cipher (i) Baboon (j) Boat (k) Baboon cipher (l) Boat cipher.

we can gain insights into the image's characteristics, such as the range of pixel values, light and dark areas, and anomalies. The chi-square χ^2 test quantitatively assesses the homogeneity of histograms. The mathematical expression of χ^2 is given in Equation (13).

$$\chi^2 = \frac{1}{\mu}\sum_{i=0}^{255}(f_i - \mu)^2, \tag{13}$$

$$\mu = \frac{H \times W}{256}, \tag{14}$$

where f denotes the histogram of the cipher image and f_i represents the specific histogram value corresponding to index i. The symbol μ denotes the anticipated mean value of the cipher image, with H and W denoting the image's height and width, respectively.

The distribution of an encrypted image that closely approximates a uniform distribution is desirable as it implies higher security against statistical attacks. A smaller value of χ^2 corresponds to a distribution that approximates a uniform distribution. A perfectly uniform histogram would result in a value of 0 for χ^2. In Figure 8, histograms of the plaintext images (Baboon, Boat, and Female) and corresponding

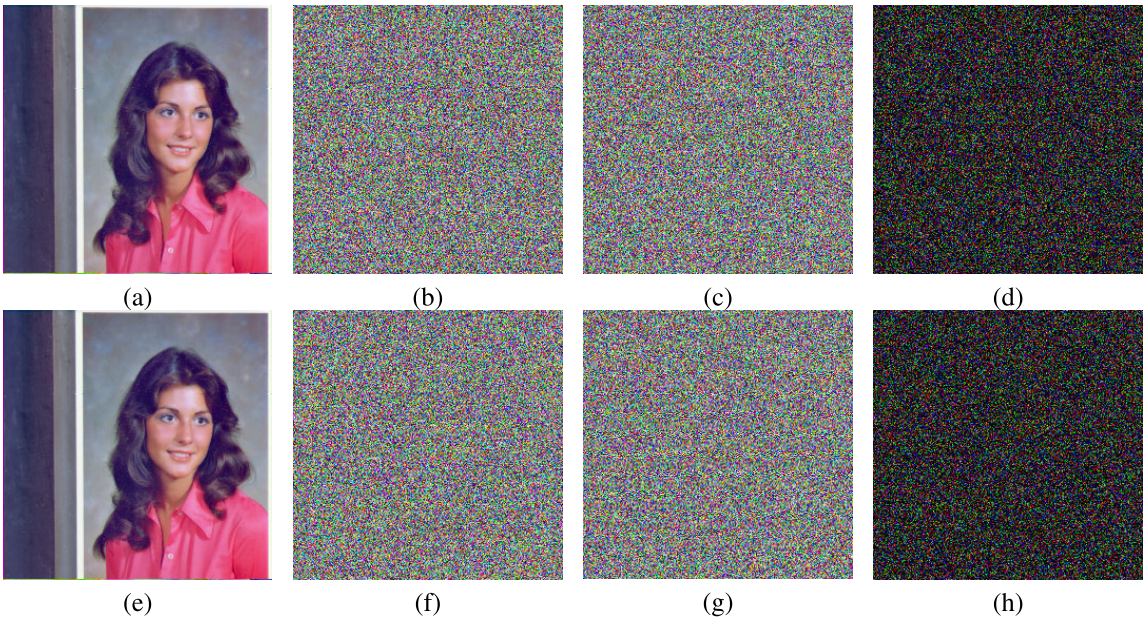


FIGURE 7. Results from the key sensitivity test conducted on a 256 × 256 color image Female: (a) plaintext image, (b) C1 encrypted with Key $K1$, (c) C2 encrypted with Key $K2$, (d) images difference $|(b) - (c)|$, (e) C1 decrypted with Key $K1$, (f) C1 decrypted with Key $K2$, (g) C2 decrypted with Key $K1$, and (h) images difference $|(f) - (g)|$..

cipher images using Z-Crypt are showcased. The cipher images exhibit histograms that follow a uniform distribution pattern, indicating the effectiveness of the proposed algorithm in concealing the information within the plaintext images and its robustness against analytical attacks based on histograms.

The results of the chi-square analysis for the plaintext and the ciphertext generated by Z-Crypt are presented in Table 5. The cipher images produced by Z-Crypt appear to be similar to uniformly distributed data. This, therefore, shows that Z-Crypt is capable of concealing the relationship information present in the plaintext image. This also shows its robustness against histogram attacks.

TABLE 5. Result of a chi-square test for plaintext and cipher images using Z-Crypt.

Image	Size	Type	Chi-Square	
			Plaintext Image	Cipher Image
Baboon	512×512	Color	142,808.039	487.020
Cameraman	256×256	Gray	110,973.305	532.882
Male	1024×1024	Gray	709,340.680	554.816
Peppers	512×512	Color	318,382.930	504.416
Boat	512×512	Gray	383,969.688	461.396
Female	256×256	Color	64,434.570	574.947
Average				519.246

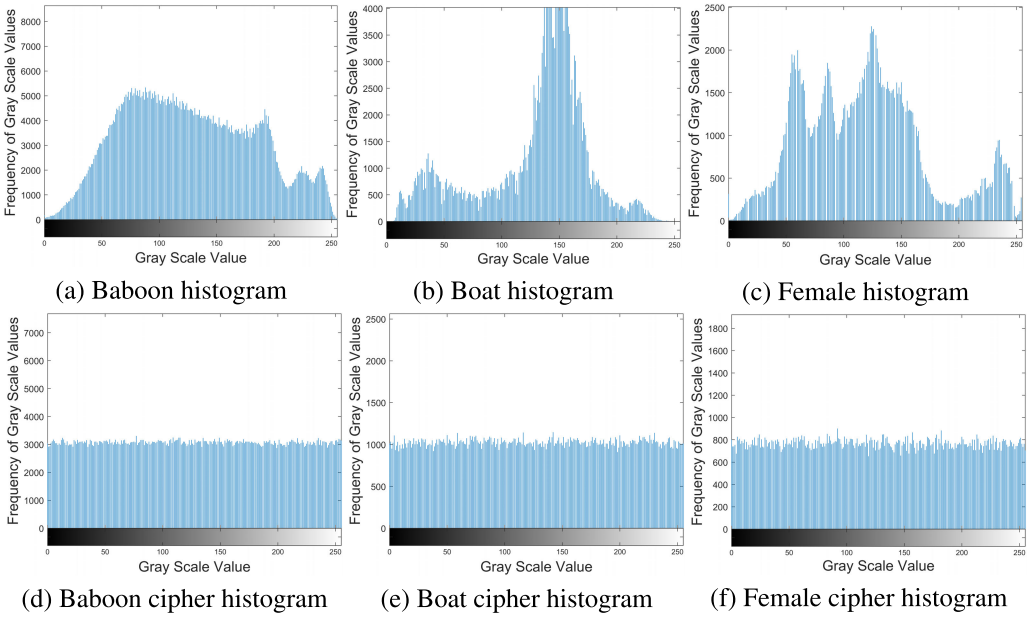
D. CORRELATION COEFFICIENT ANALYSIS

In an image, the correlation coefficient measures the extent to which two neighboring pixels are related to each other. The value of correlation ranges from −1 to 1 of which −1 indicated the perfect negative correlation, 0 indicates there is no correlation and 1 represents the perfect positive coefficient. The aim of encryption is to remove any correlation that may be present in a data so that the ciphertext is uncorrelated. This means that a perfect encryption algorithm will have 0 correlation coefficient. Two methods are used to analyze the correlation coefficient, which provides insights into the security of encryption algorithms [34], [35]. The correlation coefficient is measured in three-pixel directions, which are the horizontal, vertical, and diagonal directions.

1) CORRELATION COEFFICIENT BETWEEN ADJACENT PIXELS OF THE CIPHER IMAGE

The correlation coefficient gives a measure of how much adjacent pixels in an image are correlated. Pixels can be correlated between the adjacent, horizontal, and diagonal pixels, which can be computed using a correlation coefficient. It determines how much an encryption algorithm obfuscates information in an image. This leads to the determination of the effectiveness of the encryption algorithm. The vertical and horizontal correlation can be calculated by using the Equations (15) and (16), respectively.

$$CC_v = \frac{\sum_{i=1}^{H-1}\sum_{j=1}^{W}(C_{i,j}-\bar{C})(C_{i+1,j}-\bar{C})}{\sqrt{\sum_{i=1}^{H-1}\sum_{j=1}^{W}(C_{i,j}-\bar{C})^2 \sum_{i=1}^{H-1}\sum_{j=1}^{W}(C_{i+1,j}-\bar{C})^2}}, \quad (15)$$



(a) Baboon histogram (b) Boat histogram (c) Female histogram

(d) Baboon cipher histogram (e) Boat cipher histogram (f) Female cipher histogram

FIGURE 8. Histograms of images and their corresponding cipher images using Z-Crypt: (a) Baboon histogram, (b) Boat histogram, (c) Female histogram, (d) Baboon cipher histogram, (e) Boat cipher histogram, and (f) Female cipher histogram.

$$CC_h = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W-1}(C_{i,j}-\bar{C})(C_{i,j+1}-\bar{C})}{\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W-1}(C_{i,j}-\bar{C})^2}\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W-1}(C_{i,j+1}-\bar{C})^2}}, \tag{16}$$

where H, W, $C_{(i,j)}$ and $C_{(i+1,j)}$ respectively are the height, width, pixel at position (i, j) and the adjacent pixel at position $C_{(i+1,j)}$ of the image of interest. $\bar{C}$ is the mean pixel value of the image. Table 6 presents the results for the horizontal, vertical, and diagonal correlation between the adjacent pixels in the cipher images produced by our proposed algorithm. Table 7 compares the results of horizontal, vertical, and diagonal correlation for the proposed encryption algorithm with that of other encryption techniques present in the literature. The results indicate that Z-Crypt generates lower correlation coefficient values in comparison to other algorithms.

2) PIXEL CORRELATION COEFFICIENT BETWEEN PLAINTEXT AND CIPHER IMAGES

The pixel correlation coefficient between the plaintext image and the cipher image is another method used to assess the effectiveness of an encryption algorithm in protecting confidential information. It is computed by calculating the correlation between each pixel in the plaintext image and its corresponding pixel in the cipher image. Equation 17 is used for computing the correlation coefficient between the plaintext image and the cipher image to evaluate the degree of correlation between the two images.

$$\mathrm{CC}_{P,C} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(P_{i,j}-\bar{P})\cdot(C_{i,j}-\bar{C})}{\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W}(P_{i,j}-\bar{P})^2}\cdot\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W}(C_{i,j}-\bar{C})^2}}, \tag{17}$$

where H and W represent the height and width of the images, respectively. $P_{(i,j)}$ and $C_{(i,j)}$ stand for the pixel values at position i, j in the plaintext image and cipher image, respectively. $\bar{P}$ and $\bar{C}$ denote the means of the pixel values within the plaintext and cipher images, respectively. The correlation coefficients obtained from the proposed algorithm are presented in Table 6. The obtained results indicate that Z-Crypt engenders low correlation values between the plaintext and cipher images.

E. ENTROPY ANALYSIS

Entropy [36] is a statistical measure used to determine the level of unpredictability and randomness in a system. It is used to calculate the uncertainty in the cipher image, which is a measure of how well the plaintext is obscured. Mathematically, entropy is represented as:

$$H(x) = -\sum_{j=0}^{2^n-1} q(x_j)\log_2 q(x_j), \tag{18}$$

where n signifies the number of bits allocated to represent the symbol x_j, and $q(x_j)$ denotes the probability associated with symbol x_j. This probability corresponds to the likelihood of intensity j occurring within a pixel in the image.

TABLE 6. Correlation coefficients of adjacent pixels in the vertical, horizontal, and diagonal directions for encrypted images along with correlation coefficients for plaintext and cipher images.

Image	Size	Type	Corr. Coeff of adjacent pixels			Corr. Coeff between plaintext and cipher images
			Horizontal	Vertical	Diagonal	
Baboon	512×512	Color	-0.0001	-0.00032	-0.0018	-0.0001
Cameraman	256×256	Gray	0.0029	-0.0090	-0.0024	-0.0032
Male	1024×1024	Gray	0.0014	-0.0006	-0.0002	0.0010
Peppers	512×512	Color	0.0016	0.0021	-0.0017	-0.0006
Boat	512×512	Gray	0.0001	0.0007	-0.0001	-0.0008
Female	256×256	Color	-0.0043	0.0041	-0.0045	-0.0048
Average			-0.0009	0.0002	-0.0017	-0.0014

TABLE 7. Comparison of correlation coefficients (horizontal, vertical, and diagonal) using Z-Crypt with other methods for Boat image.

Methods	Correlation Coefficient		
	Horizontal	Vertical	Diagonal
Proposed	0.0001	0.0007	-0.0001
[20]	-0.0011	0.0004	-0.0005
[24]	-0.0038	0.0035	0.0023
[49]	0.0043	0.0031	0.0017
[57]	0.0021	0.0045	-0.0015

When it comes to image encryption, a cipher image with high entropy is more effective at hiding the plaintext than a cipher image with low entropy. This is because a high-entropy cipher image has a more random and unpredictable distribution of pixel values, making it harder to guess the plaintext. Conversely, a low-entropy cipher image has a more predictable distribution of pixel values, which attackers can exploit to reveal the plaintext.

The entropy of an image is calculated based on Equation (18), which uses $q(x_j)$ to represent the normalized histogram counts for each intensity value within the image. With 256 possible intensity values for an 8-bit pixel representation, it is possible to calculate the ideal entropy for an image using Equation (19).

$$H_{\max} = -\sum_{i=0}^{255} \left(\frac{1}{256}\right) \times \log_2 \left(\frac{1}{256}\right) = 8, \tag{19}$$

The entropy of a cipher image that is generated through an encryption algorithm should ideally approach a value of 8. The entropy in Equation (19) is referred to as Shannon entropy. The Shannon entropy is categorized as global entropy since it assesses the pixel information across the entire image. However, local entropy is determined by calculating the mean Shannon entropy within randomly selected, non-overlapping blocks. According to previous studies, local entropy provides superior accuracy, consistency, and efficiency compared to global Shannon entropy. The calculation of local entropy can be expressed as follows:

$$H_{k,T_B}(X) = \sum_{j=1}^{N} \frac{H(X_j)}{N}, \tag{20}$$

where $H_{k,T_B}(X)$ represents the local entropy of the signal X in non-overlapping blocks of size T_B, N is the total number of random blocks, and $H(X_j)$ signifies the entropy of the j-th block X_j within the signal X.

The results of the entropy assessment for Z-Crypt are presented in Table 8. From the analysis, it can be seen that both the global and local entropy of Z-Crypt averages close to 8. Furthermore, Table 9 provides a comparative analysis where the global entropy of Z-Crypt is measured against other encryption algorithms. The computed entropy values obtained from Z-Crypt are closely aligned with the ideal entropy and outperform the state-of-the-art methods.

TABLE 8. A global and local analysis of Shannon entropy values for images encrypted with Z-Crypt.

Image	Size	Type	Entropy	
			Global	Local
Baboon	512×512	Color	7.99955	7.94736
Cameraman	256×256	Gray	7.99415	7.93094
Male	1024×1024	Gray	7.99961	7.95151
Peppers	512×512	Color	7.99953	7.94748
Boat	512×512	Gray	7.99873	7.94375
Female	256×256	Color	7.99788	7.93802
Average			7.99824	7.94318

TABLE 9. Analysis of global entropy in the Baboon image using Z-Crypt and other methods.

Entropy	Methods						
	Proposed	[20]	[22]	[23]	[24]	[48]	[54]
Global	7.99955	7.99918	7.99923	7.99710	7.99690	7.99520	7.70033

F. ENCRYPTION QUALITY

Image encryption quality is critical in protecting the confidentiality and integrity of encrypted images. It must resist attacks like brute-force, side-channel, and differential attacks. Several tests are conducted on plaintext and encrypted images to assess encryption quality and effectiveness. These tests include MSE, PSNR, and different deviation measurements.

In the following discussion, we will delve into the specifics of each test and its respective results.

1) MEAN SQUARE ERROR (MSE)

The MSE calculates the average of the squared differences between the pixels in the plaintext and cipher images. The MSE can be calculated using the following formula:

$$\text{MSE} = \sum_{i=1}^{H}\sum_{j=1}^{W}[P(i,j) - C(i,j)]^2 \times \frac{1}{H \times W}, \tag{21}$$

where H and W represent the image height and width, respectively; $P(i,j)$ and $C(i,j)$ denote the pixel values of the plaintext image and the encrypted cipher image, respectively, at the position (i,j).

When assessing the security of an algorithm, it is ideal for the algorithm to produce high MSE values, typically ≥ 30 as given in [37]. Table 10 depicts Z-Crypt's MSE values calculated between the plaintext and cipher images. Our method produces an average MSE value of 46.4768, which is significantly higher than the required minimum value. Furthermore, Table 11 provides a comparison of Z-Crypt with other state-of-the-art methods, demonstrating that Z-Crypt outperforms others in terms of quality, as measured by the MSE metric.

2) MEAN ABSOLUTE ERROR (MAE)

The MAE quantifies the dissimilarity existing within the pixel values of two images. The mathematical formulation for computing the MAE between the plaintext image and its corresponding cipher image is calculated by using Equation (22).

$$\text{MAE} = \sum_{i=1}^{H}\sum_{j=1}^{W}\frac{1}{H \times W} \times |P(i,j) - C(i,j)|, \tag{22}$$

where H and W represent the height and width of the image, respectively. $P(i,j)$ and $C(i,j)$ refer to the pixel values found in the plaintext image and the encrypted cipher image at the (i,j) position.

Table 10 presents the MAE values computed for Z-Crypt using different images and attain an average of 78.1802. Table 11 conducts a comparative analysis assessing our proposed algorithm alongside other image encryption algorithms. From Table 11, it is observed that our MAE value is average compared to the other methods. A deeper study shows that this may have been caused by the application of Chirp Z-Transform (CZT) as choices for the value A of the CZT, which specifies the contour location, is limited. To further evaluate the security of the proposed algorithm, an experiment is conducted to evaluate the similarity of pixels between the plaintext image and the ciphertext image. A value of 0% shows that there is 0 similarity between them. The pixel similarity equation is given below:

$$S_p = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(P_{ij} == C_{ij})}{\text{numel}(P)} \times 100\%, \tag{23}$$

where S_p is pixel similarity and P, and C are the plaintext and ciphertext, respectively. The result obtained for the pixel similarity S_p is 0.048%, which indicates no similarity between the pixels of the plaintext image and the ciphertext image. The result shows that although the MAE is low, it does not imply any weakness in the proposed encryption algorithm.

TABLE 10. Performance evaluation of images encrypted by Z-Crypt using MSE, MAE, and PSNR metrics.

Image	Size	Type	MSE	MAE	PSNR
Baboon	512×512	Color	47.5361	76.3138	6.0647
Cameraman	256×256	Gray	46.8897	79.5484	7.4039
Male	1024×1024	Gray	43.7506	82.8449	10.1295
Peppers	512×512	Color	46.1059	82.0431	5.2476
Boat	512×512	Gray	47.7826	72.4137	6.6273
Female	256×256	Color	46.7958	75.9171	6.5953
Average			46.4768	78.1802	7.0114

TABLE 11. Comparison of Z-Crypt with other methods using MSE, MAE, and PSNR values.

Metric	Methods								
	Z-Crypt	[25]	[22]	[37]	[50]	[51]	[52]	[46]	[20]
MSE	46.476	45.399	39.679	40.340	-	-	-	-	-
MAE	78.180	91.580	-	-	73.530	78.100	90.000	-	-
PSNR	7.011	8.513	8.989	-	8.136	-	-	7.124	8.214

3) PEAK SIGNAL-TO-NOISE RATIO (PSNR)

The PNSR metric is used to measure the similarity between two images, typically the plaintext image and the cipher image. This metric quantifies the signal-to-noise ratio existing between the plaintext and encrypted images. The mathematical formulation for calculating the PSNR between the plaintext image and its corresponding cipher image is given by Equation (24):

$$\text{PSNR} = 40\log_{10}\left(\sqrt{\frac{\text{MAX}_p}{\text{MSE}}}\right), \tag{24}$$

where MAX_p represents the highest possible pixel value, typically 255 in the case of 8-bit pixels, while MSE corresponds to the mean squared error value calculated using Equation (21). A higher PSNR value signifies enhanced image quality and a greater resemblance of the cipher image to the plaintext image. Conversely, a lower PSNR value indicates superior encryption quality.

Table 10 provides an analysis of PSNR values for encrypted images generated by Z-Crypt. Table 11 depicts a comparative analysis of PSNR values of Z-Crypt against other image encryption algorithms. The comparative results reveal that images encrypted using Z-Crypt exhibit lower PSNR values, indicative of superior encryption quality compared to other encryption methods.

TABLE 12. Comparison of maximum deviation, irregular deviation, and uniform histogram deviation measurements of Z-Crypt compared to related methods.

Methods	Images	[25]	[22]	[54]	[4]	[47]	[53]	Z-Crypt
Maximum Deviation	Baboon	363,121	199,158	232,790	-	-	-	393,216
	Cameraman	64,382	64,998	77,074	18,007	24,978	-	50,152
Irregular Deviation	Baboon	59,921	80,203	-	-	-	-	156,672
	Cameraman	32,165	32,706	-	39,244	45,031	-	39,438
Uniform Histogram Deviation	Baboon	0.02560	0.99902	0.99904	-	-	0.04960	0.02013
	Cameraman	0.03050	0.99609	0.99610	0.09420	-	0.05020	0.07040
	Peppers	0.03150	0.99902	0.99904	0.09170	-	-	0.02100

4) DEVIATION MEASUREMENTS

1) **Maximum Deviation:** The maximum deviation metric is used to measure the maximum deviation between the histograms of plaintext and cipher images. It is calculated using the following equations:

$$h = \text{histogram}(|(P - C)|), \tag{25}$$

$$D_{\text{M}} = \sum_{i=1}^{254} h_i + \frac{(h_0 + h_{255})}{2}, \tag{26}$$

where D_M denotes the maximum deviation metric, h represents the histogram accounting for the absolute differences between the plaintext and cipher images, h_i signifies the histogram value of h at position i, and h_0 and h_{255} stand for the histogram values at positions 0 and 255, respectively.
The maximum deviation values of the encrypted images using Z-Crypt have been benchmarked against those of other encryption methods, and the results have been tabulated in Table 12. The findings indicate that Z-Crypt produces cipher images that demonstrate significant deviations from the original plaintext images compared to the other methods.

2) **Irregular Deviation:** The irregular deviation metric is used to measure the deviation of each pixel of the cipher image from the corresponding pixel of the plaintext image. The irregular deviation (D_I) metric is calculated as follows:

$$D_I = \sum_{i=0}^{255} \left| h_i - D_{\text{avg}} \right|, \tag{27}$$

where:

$$D_{\text{avg}} = \sum_{i=0}^{255} \left(\frac{h_i}{256} \right), \tag{28}$$

In Equations (27) and (28), h_i denotes the histogram value of h at position i where the histogram h is calculated using Equation (25), and D_{avg} signifies the mean value of pixels that exhibit deviations at each deviation level.
Table 12 presents the outcomes of ID for Baboon and Cameraman images after undergoing the encryption process with Z-Crypt. Z-Crypt is observed to deliver remarkable results, especially in the case of the Baboon image, surpassing other techniques.

3) **Uniform Histogram Deviation Measurement:** The uniform histogram deviation (D_{uh}) metric measures the difference between the histogram of the cipher image and a uniform histogram. A lower deviation indicates better encryption quality. It is calculated by comparing the cipher image's histogram with a uniform histogram given in Equation (29).

$$D_{uh} = \frac{1}{H \times W} \sum_{i=0}^{255} |h_{ci} - h_{ui}|, \tag{29}$$

where:

$$h_{ui} = \begin{cases} \dfrac{M \times N}{256} & \text{if } 0 \le i \le 255, \\ 0 & \text{if } 0 > i > 255, \end{cases}$$

where M and N represent the image's height and width, respectively. The uniform histogram value associated with the i-th index is denoted by h_{ui}. These h_{ui} values are calculated exclusively for pixel values within the range of 0 to 255, and any values outside this range are considered zero. The histogram value of the cipher image at the i-th index is represented by h_{c_i}.
Table 12 illustrates the deviations from the uniform histogram of a range of images, including Baboon, Cameraman, and Peppers. A comparative analysis of Z-Crypt against other encryption methods reveals that Z-Crypt achieves superior deviations from the uniform histogram in comparison to all other methods for the Baboon and Peppers images, while also achieving decent results for the Cameraman image.

G. RESISTANCE TO DIFFERENT SECURITY ATTACKS

1) DIFFERENTIAL ATTACKS

In order to ensure the security of an image encryption algorithm, it is essential that it is made completely resistant to differential attacks. These attacks involve the attacker attempting to predict the relationship between the plaintext and its cipher image [38] in order to break the encryption algorithm. To ensure that the algorithm is strong enough to withstand such attacks, it is imperative to conduct a series of widely accepted assessments, including examining the AE,

pixel change rate, and the unified and averaged intensity of changes (UACI). These tests are crucial to determine the algorithm's resistance against differential attacks and the outcomes of these tests for our proposed encryption algorithm are given in Table 13. In the following, we elaborate on these tests.

1) **Avalanche Effect** is a property of cryptographic algorithms where a small input change can cause a significant change in the output. The MSE metric can be used to measure this effect. A small change in the plaintext image results in a corresponding change in the ciphertext image, and if the latter changes significantly, the avalanche effect is present. The MSE is calculated by finding the average squared difference between pixel values of two cipher images encrypted from the same plaintext image but with a 1-bit alteration. The avalanche effect MSE can be calculated as follows:

$$MSE_{av} = \sum_{i=1}^{H}\sum_{j=1}^{W} \frac{1}{H \times W} \times [C_1(i,j) - C_2(i,j)]^2, \tag{30}$$

where H and W correspond to the height and width, respectively, of the images under consideration, while MSE_{av} represents the avalanche effect MSE. C_1 and C_2 represent the two cipher images produced from the same plaintext image with one bit altered.
Figure 9 displays the Baboon plaintext image alongside the two cipher images, where one cipher image is produced by encrypting the original plaintext image and the other cipher image is obtained by encrypting the plaintext image altered in 1-bit only from the original plaintext image. Secure encryption algorithms should produce cipher image changes exceeding 50% when modifying a single bit in the plaintext image [39]. Table 13 shows computed MSE_{av} values for various images encrypted using Z-Crypt.

2) **Number of Pixels Change Rate (NPCR)** metric is also used to evaluate encryption algorithm security against differential attacks. It calculates the difference between two cipher images created from the same plaintext image but with a change of 1 bit in the plaintext. If the NPCR value is high, the encryption algorithm is resilient against differential attacks. Conversely, if the NPCR value is low, it's vulnerable to differential attacks. The NPCR metric can be calculated using the Equation (31):

$$\text{NPCR} = \sum_{i=1}^{H}\sum_{j=1}^{W} D(i,j), \tag{31}$$

where:

$$D(i,j) = \begin{cases} 0, & \text{for identical pixels } C_1(i,j) \text{ and } C_2(i,j), \\ 1, & \text{for differing pixels } C_1(i,j) \text{ and } C_2(i,j). \end{cases}$$

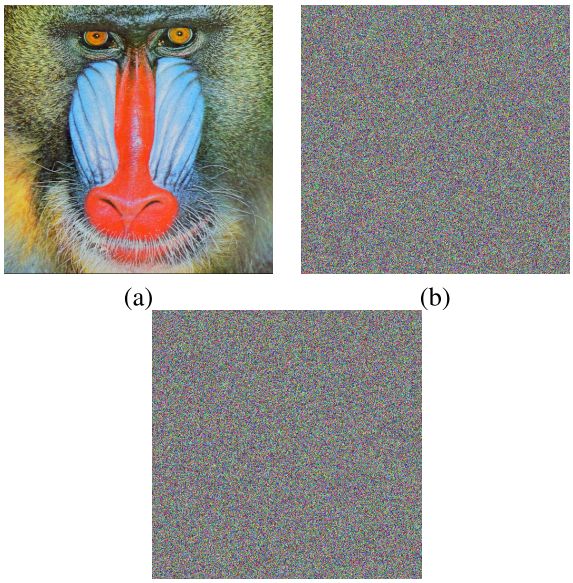


FIGURE 9. Visualization of the avalanche effect through two images, both encrypted from the identical plaintext image, with a 1-bit alteration: (a) Baboon plaintext, (b) Baboon cipher, (c) Baboon cipher image with a 1-bit change in the original plaintext image.

where H and W correspond to the image's height and width, respectively, while C_1 and C_2 represent two cipher images, both derived from the same plaintext image but with a 1-bit modification.
Table 13 provides the NPCR values for different images using Z-Crypt. A higher NPCR value implies resilience against differential attacks with the ideal average NPCR value being ≥ 99.609 [40]. In Table 14, we compare the average NPCR values achieved by Z-Crypt with other state-of-the-art methods. The results indicate that Z-Crypt has a high average NPCR value, which is similar to other algorithms, and very close to the ideal NPCR value.

3) **Unified Average Changing Intensity (UACI)** measures the average intensity difference between two encrypted images generated from the same plaintext image with a 1-bit change. A higher UACI indicates greater robustness, as even a small change in the plaintext image results in a significant change in the ciphertext image. Conversely, a lower UACI indicates less robustness, as a small change in the plaintext image results in a relatively small change in the ciphertext image. It is calculated using the Equation (32).

$$\text{UACI} = \left[\frac{\sum_{i=1}^{H}\sum_{j=1}^{W} |C_1(i,j) - C_2(i,j)|}{2^b - 1}\right] \times \frac{1}{H \times W} \times 100\%, \tag{32}$$

where, H and W denote the height and width of the image, respectively. Furthermore, $C_1(i,j)$ and $C_2(i,j)$ represent the pixel values present in the first and second

cipher images at position (i, j), and b signifies the number of bits used for pixel representation. Table 13 provides an overview of the UACI values computed for various images, such as Baboon, Cameraman, Male, Peppers, Boat, and Female, encrypted using our proposed algorithm. The UACI values indicate how responsive an algorithm is to changes in the plaintext. It is important to note that a higher UACI value is favorable, with the ideal average UACI value set at $\geq$ 33.4635 [40]. Table 14 presents comparative analysis, contrasting the UACI values obtained from Z-Crypt with those derived from other encryption algorithms. The results reveal that Z-Crypt outperforms most of the others encryption algorithm in terms of UACI value.

TABLE 13. Avalanche effect, NPCR, and UACI values for images that have undergone encryption using Z-Crypt.

Image	Size	Type	Aval. effect	NPCR	UACI
Baboon	512×512	Color	117.0093	99.6086	33.4570
Cameraman	256×256	Gray	117.4398	99.6201	33.4710
Male	1024×1024	Gray	117.3657	99.6114	33.4823
Peppers	512×512	Color	117.2148	99.6197	33.4629
Boat	512×512	Gray	117.1865	99.6025	33.4403
Female	256×256	Color	117.2616	99.5859	33.3558
Average			117.2463	99.6080	33.4449

TABLE 14. Comparison of NPCR and UACI values of Z-Crypt with other encryption methods for Baboon image.

Methods	NPCR	UACI
Z-Crypt	99.6086	33.4570
[57]	99.6000	33.6500
[22]	99.6059	33.4375
[46]	99.6100	33.5200
[58]	99.5995	33.5101
[48]	99.1608	33.1975
[24]	99.6277	33.3148

2) NOISE AND DATA LOSS ATTACKS

During the transmission of digital images over a network, there are commonly two types of attacks affecting encrypted images, which are noise attacks and data loss attacks, also known as occlusion attacks. Noise attacks refer to the introduction of random or intentional errors/noise in the cipher image, which can make it difficult or impossible to decrypt the image. On the other hand, data loss attacks involve the intentional or unintentional removal of parts of the cipher image. Different tests can be performed to evaluate the robustness of an image encryption algorithm against noise and data loss attacks. However, the resistance of Z-Crypt can be evaluated for noise attacks by adding salt and pepper noise to the cipher image and assessing the decryption accuracy. Data loss attacks are evaluated by cropping parts of the cipher image and measuring the impact on decryption accuracy.

Figure 10 and 11 demonstrate the resilience of Z-Crypt against data loss (occlusion) and noise attacks. Figure 11 shows cipher images subjected to salt and pepper noise with different densities (0.15, 0.25, and 0.5). It is important to note that a noise density of 0.15 affects approximately 15% of the pixels, a noise density of 0.25 affects around 25% of the pixels, and a noise density of 0.5 impacts approximately 50% of the pixels within the image. Z-Crypt successfully decrypts cipher images even when affected by salt and pepper noise densities as high as 0.5. Similarly, Figure 10 presents the impact of data loss, occurring in varying degrees of severity (10% cropping, 25% cropping, and 50% cropping), on a cipher image. Figure 10 demonstrates the successful recovery of the plaintext image encrypted by our proposed encryption algorithm and then decrypting the resulting cipher image by our proposed decryption algorithm even when the cipher image is cropped by up to 50%. These results show the tremendous resilience of our proposed encryption/decryption Z-Crypt algorithm against data loss (noise) attacks.

H. HOMOGENEITY, CONTRAST AND ENERGY ANALYSIS

1) HOMOGENEITY ANALYSIS

Homogeneity analysis is a method of measuring the similarity between elements in a gray-level co-occurrence matrix (GLCM) to its diagonal distribution. The GLCM calculates the frequency of a pixel with a gray-level value of i appearing horizontally adjacent to a pixel with a gray-level value of j. The homogeneity value ranges from 0 to 1, where a lower value indicates a more efficient encryption algorithm. Equation (33) can be used to calculate the homogeneity.

$$\mathrm{H} = \sum_{i,j} \frac{p(i,j)}{|i-j|+1}, \tag{33}$$

where adjacent gray-level values are represented by i and j, while term $p(i, j)$ denotes the value of the element located at position (i, j) within the normalized GLCM.

The homogeneity analysis results from both the plaintext and cipher images are presented in Table 15. The results indicate that Z-Crypt generates homogeneity values on the lower end (0.38943) of the spectrum, which implies that it is a more efficient encryption/decryption algorithm.

2) CONTRAST ANALYSIS

Contrast analysis is another method used to assess the security of an encrypted image. It involves measuring the variation in intensity between a pixel and its adjacent pixels throughout the entire image. An encrypted image that has high contrast values is considered more secure because it indicates a higher degree of randomness. This makes it challenging for attackers to identify patterns or features in the image. Contrast can be determined by using Equation (34).

$$\text{Contrast} = \sum_{i,j} |i-j|^2 \times p(i,j), \tag{34}$$

Table 15 shows the results of the contrast analysis for both plaintext and cipher images. The average contrast value of the cipher image produced by our proposed encryption

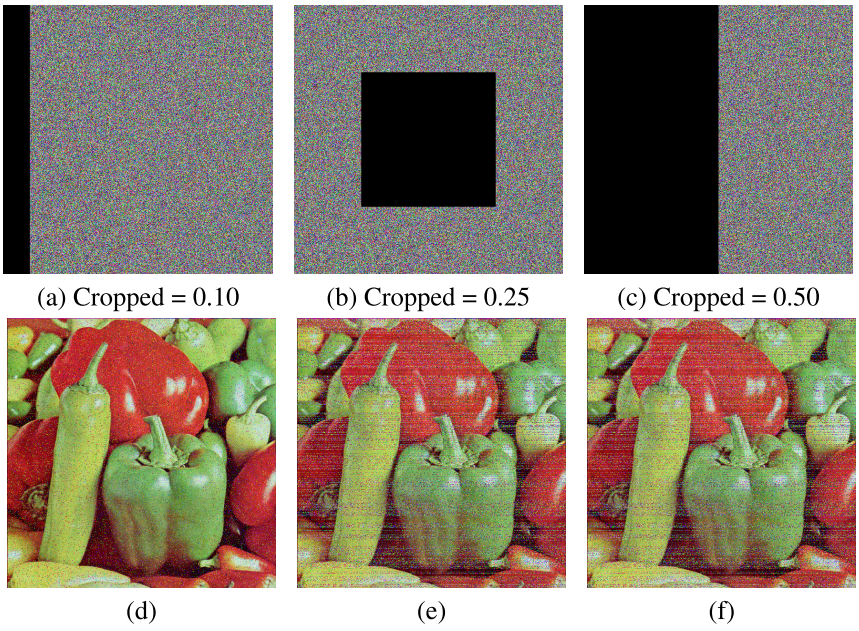


FIGURE 10. Results of data loss (occlusion) attacks on the color image Peppers 512 × 512: (a-c) depict various degrees of occlusion attacks, (d-f) show decrypted images after recovery.

TABLE 15. Homogeneity, contrast, and energy analysis of the plaintext and cipher images.

Image	Size	Type	Homogeneity		Contrast		Energy	
			Plaintext Image	Cipher Image	Plaintext Image	Cipher Image	Plaintext Image	Cipher Image
Baboon	512x512	Color	0.70402	0.38939	1.34026	10.47462	0.05634	0.01563
Cameraman	256x256	Gray	0.87142	0.39011	0.77939	10.50794	0.17055	0.01564
Male	1024x1024	Gray	0.88054	0.38925	0.32397	10.47735	0.11318	0.01562
Peppers	512x512	Color	0.88886	0.38991	0.42372	10.48983	0.11753	0.01562
Boat	512x512	Gray	0.85451	0.38918	0.45439	10.46794	0.17878	0.01562
Female	256x256	Color	0.88972	0.38876	0.36759	10.56728	0.12783	0.01564
Average			0.84818	0.38943	0.61489	10.49749	0.12737	0.01563

TABLE 16. Comparative assessment of homogeneity, contrast, and energy analysis for the cipher image produced by Z-Crypt and other relevant methods.

Methods	Size	Type	Homogeneity	Contrast	Energy
Z-Crypt	256 × 256	Color	0.3887	10.5672	0.0156
[25]	256 × 256	Color	0.3896	10.4893	0.0156
[50]	256 × 256	Color	0.3901	10.4934	0.0156
[55]	-	Color	0.4110	8.6448	0.0156

algorithm is 10.463, which is 17× higher than the average plaintext value. This implies that the algorithm has effectively increased the randomness of the image, leading to an increase in its overall security.

3) ENERGY ANALYSIS

Energy analysis is a technique that evaluates the evenness of the gray-level distribution in an image. It is computed by calculating the total sum of squared elements within the GLCM. When the energy values of cipher images are lower, it indicates that the encryption algorithm has successfully randomized the gray levels. The calculation of energy can be done using Equation (35):

$$\text{Energy} = \sum_{i,j} p(i,j)^2, \tag{35}$$

The energy values of an image are represented by the adjacent gray-level values i and j, with values ranging from 0 to 1. Table 15 presents the results of the energy evaluation for plaintext and cipher images generated by Z-Crypt. The average value for the energy of ciphertext images is 0.01563, which is closer to zero compared to the energy level of the plaintext image, which is 0.12737.

4) COMPARISON WITH STATE OF THE ART IMPLEMENTATIONS

Table 16 presents a comparative analysis of Z-Crypt with other available methods in terms of homogeneity, contrast, and energy. The results demonstrate that Z-Crypt outperforms other encryption algorithms in all three metrics, namely,

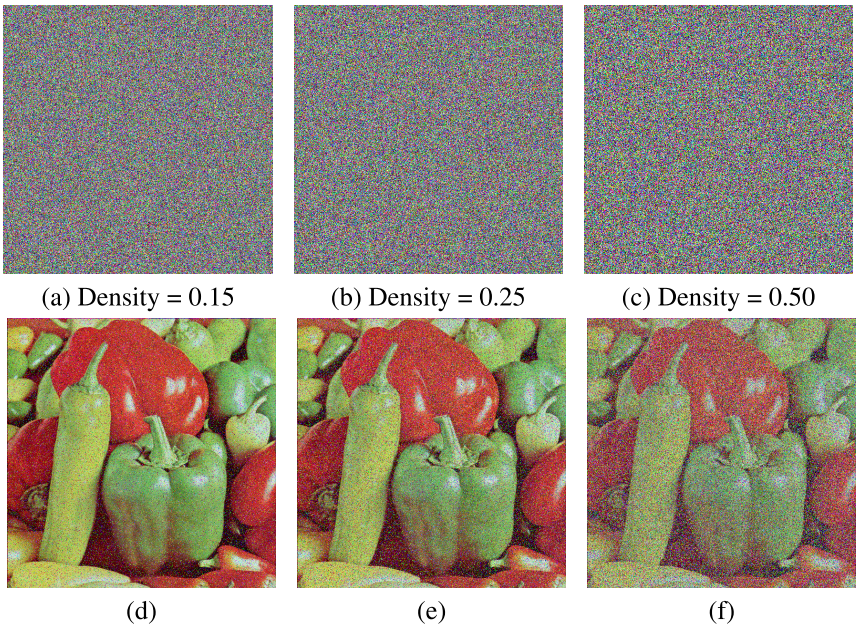


FIGURE 11. Results from salt and pepper noise attacks on the color image Peppers 512 × 512: (a-c) show noise densities, and (d-f) show decrypted images after the attack.

homogeneity, contrast, and energy. In comparison to alternative techniques, Z-Crypt consistently achieves the lowest homogeneity (0.3887), highest contrast (10.5672), and lowest energy (0.0156) values. This provides compelling evidence of Z-Crypt's effectiveness and efficiency in securing image data through encryption.

TABLE 17. NIST SP 800-22 assessments results for the Pepper's image.

Test	*p*-Value	Results
Frequency (Monobits)	0.030773	Pass
Frequency within a Block	0.617438	Pass
Cumulative Sums [Reverse]	0.045940	Pass
Cumulative Sums [Forward]	0.051806	Pass
Runs	0.980902	Pass
Longest Run of Ones in a Block	0.511035	Pass
Binary Matrix Rank	0.670290	Pass
Discrete Fourier Transform (Spectral)	0.531266	Pass
Non-overlapping Template Matching	0.444085	Pass
Overlapping Template Matching	0.103212	Pass
Maurer's (Universal Statistical)	0.548070	Pass
Approximate Entropy	0.896548	Pass
Random Excursions Variant	0.944444	Pass
Random Excursions	1.000000	Pass
Serial [Test1]	0.097007	Pass
Serial [Test2]	0.984573	Pass
Linear Complexity	0.625889	Pass

I. NIST SP 800-22 TEST

We have employed the NIST SP 800-22 test suite [41], version 2.1.2 to thoroughly evaluate the randomness properties and unpredictability of a given random sequence. This test suite comprises a comprehensive collection of statistical assessments and guidelines that are essential for assessing the quality and randomness of cryptographic keys and pseudorandom number generators (PRNGs). These assessments are crucial in determining the security of cryptographic algorithms and generators used in various cryptographic applications such as encryption, digital signatures, and secure communication.

The test suite consists of a collection of 15 statistical tests. Each test generates its own p-value ranging from 0 to 1. To pass the test, a sequence must have a p-value greater than $\mu = 0.01$ threshold. We evaluate the NIST tests by performing the tests on encrypted Peppers image using Z-Crypt, which is presented in Table 17. The table shows that the sequence generated by Peppers cipher image has passed all the NIST tests with flying colors, verifying its randomness and robustness.

VI. CONCLUSION AND FUTURE WORK

This paper presents a novel image encryption algorithm Z-Crypt that combines the strengths of SPN and CLM based on CZT to achieve robust and secure encryption. The proposed Z-Crypt excels in various security metrics, such as key sensitivity, histogram uniformity, correlation coefficients, entropy, MSE, MAE, and PSNR, demonstrating exceptional performance. With a key space of length 2^{256} bits, Z-Crypt renders brute-force attacks futile, generating distinct cipher images for minor key variations, which further strengthens security.

The synergistic action of SPN and CZT introduces multiple diffusion layers, resulting in a statistically uniform distribution of cipher image histograms and low correlation coefficients between adjacent pixels, significantly hindering attacker analysis and strengthening resistance to statistical attacks. Z-Crypt excels in preserving image quality, showcasing comparatively better MSE, MAE, and PSNR values

compared to existing methods. Furthermore, it demonstrates exceptional resilience against noise, successfully decrypting cipher images even with significant noise densities. The encrypted images using the proposed algorithm outperform other contemporary encryption algorithms in terms of homogeneity, contrast, and energy metrics.

In the future, it could be interesting to explore Z-Crypt in combination with watermarking techniques for copyright protection without compromising confidentiality, adapting it for specific image types, optimizing performance, and preserving critical features. Exploring the integration of machine learning techniques for automated key generation and encryption could also offer avenues for further strengthening of an encryption algorithm's robustness to security attacks.

REFERENCES

[1] P. Ping, F. Xu, Y. Mao, and Z. Wang, "Designing permutation—Substitution image encryption networks with Henon map," *Neurocomputing*, vol. 283, pp. 53–63, Mar. 2018.

[2] R. Matthews, "On the derivation of a 'chaotic' encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, Jan. 1989.

[3] S. Zhou, X. Wang, and Y. Zhang, "Novel image encryption scheme based on chaotic signals with finite-precision error," *Inf. Sci.*, vol. 621, pp. 782–798, Apr. 2023.

[4] A. Belazi, A. A. A. El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Process.*, vol. 128, pp. 155–170, Nov. 2016.

[5] M. K. Hasan, S. Islam, R. Sulaiman, S. Khan, A. A. Hashim, S. Habib, M. Islam, S. Alyahya, M. M. Ahmed, S. Kamil, and M. A. Hassan, "Lightweight encryption technique to enhance medical image security on Internet of Medical Things applications," *IEEE Access*, vol. 9, pp. 47731–47742, 2021.

[6] J. Zhang, B. Zheng, A. Gao, X. Feng, D. Liang, and X. Long, "A 3D densely connected convolution neural network with connection-wise attention mechanism for Alzheimer's disease classification," *Magn. Reson. Imag.*, vol. 78, pp. 119–126, May 2021.

[7] M. H. Khan, S. M. T. Zaidi, S. Khan, and F. Khan, "Mode-guided feature augmentation for domain generalization," in *Proc. BMVC*, 2022, p. 176.

[8] B. Ran, T. Zhang, L. Wang, S. Liu, and X. Zhou, "Image security based on three-dimensional chaotic system and random dynamic selection," *Entropy*, vol. 24, no. 7, p. 958, Jul. 2022.

[9] F. S. Abas and R. Arulmurugan, "Radix trie improved nahrain chaotic map-based image encryption model for effective image encryption process," *Int. J. Intell. Netw.*, vol. 3, pp. 102–108, Jan. 2022.

[10] Priyanka and A. K. Singh, "A survey of image encryption for healthcare applications," *Evol. Intell.*, vol. 16, no. 3, pp. 801–818, Jun. 2023.

[11] J. Zheng and Q. Zeng, "An image encryption algorithm using a dynamic S-box and chaotic maps," *Int. J. Speech Technol.*, vol. 52, no. 13, pp. 15703–15717, Oct. 2022.

[12] D. Arroyo, J. Diaz, and F. B. Rodriguez, "Cryptanalysis of a one round chaos-based substitution permutation network," *Signal Process.*, vol. 93, no. 5, pp. 1358–1364, May 2013.

[13] M. Agoyi, E. Çelebi, and G. Anbarjafari, "A watermarking algorithm based on chirp Z-transform, discrete wavelet transform, and singular value decomposition," *Signal, Image Video Process.*, vol. 9, no. 3, pp. 735–745, Mar. 2015.

[14] E. Mosso and N. Bolognini, "Dynamic multiple-image encryption based on chirp Z-transform," *J. Opt.*, vol. 21, no. 3, Mar. 2019, Art. no. 035704.

[15] T. Cui, C. Jin, and Z. Kong, "On compact Cauchy matrices for substitution-permutation networks," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 2098–2102, Jul. 2015.

[16] R. Girija and H. Singh, "A new substitution-permutation network cipher using Walsh Hadamard transform," in *Proc. Int. Conf. Comput. Commun. Technol. Smart Nation (IC3TSN)*, Oct. 2017, pp. 168–172.

[17] S. H. Strogatz, *Nonlinear Dynamics and Chaos With Solutions Manual: With Applications to Physics, Biology, Chemistry, and Engineering*. Boca Raton, FL, USA: CRC Press, 2018.

[18] V. Sukhoy and A. Stoytchev, "Generalizing the inverse FFT off the unit circle," *Sci. Rep.*, vol. 9, no. 1, Oct. 2019, Art. no. 14443.

[19] Q.-Y. Zhang, Z.-X. Ge, Y.-J. Hu, J. Bai, and Y.-B. Huang, "An encrypted speech retrieval algorithm based on chirp-Z transform and perceptual hashing second feature extraction," *Multimedia Tools Appl.*, vol. 79, nos. 9–10, pp. 6337–6361, Mar. 2020.

[20] A. Alanezi, B. Abd-El-Atty, H. Kolivand, A. A. A. El-Latif, B. A. El-Rahiem, S. Sankar, and H. S. Khalifa, "Securing digital images through simple permutation-substitution mechanism in cloud-based smart city environment," *Secur. Commun. Netw.*, vol. 2021, pp. 1–17, Feb. 2021.

[21] L. Rabiner, R. Schafer, and C. Rader, "The chirp Z-transform algorithm," *IEEE Trans. Audio Electroacoust.*, vol. AE-17, no. 2, pp. 86–92, Jun. 1969, doi: 10.1109/TAU.1969.1162034.

[22] J. Arif, M. A. Khan, B. Ghaleb, J. Ahmad, A. Munir, U. Rashid, and A. Y. Al-Dubai, "A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution," *IEEE Access*, vol. 10, pp. 12966–12982, 2022.

[23] Q. Lu, C. Zhu, and X. Deng, "An efficient image encryption scheme based on the LSS chaotic map and single S-box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020.

[24] Y. Niu and X. Zhang, "A novel plaintext-related image encryption scheme based on chaotic system and pixel permutation," *IEEE Access*, vol. 8, pp. 22082–22093, 2020.

[25] Y. Alghamdi, A. Munir, and J. Ahmad, "A lightweight image encryption algorithm based on chaotic map and random substitution," *Entropy*, vol. 24, no. 10, p. 1344, Sep. 2022.

[26] M. N. Alenezi and F. S. Al-Anzi, "A study of Z-transform based encryption algorithm," *Int. J. Commun. Netw. Inf. Secur. (IJCNIS)*, vol. 13, no. 2, pp. 302–309, Apr. 2022.

[27] B. Mondal and J. P. Singh, "A lightweight image encryption scheme based on chaos and diffusion circuit," *Multimedia Tools Appl.*, vol. 81, no. 24, pp. 34547–34571, Oct. 2022.

[28] A. Javeed, T. Shah, and A., "Lightweight secure image encryption scheme based on chaotic differential equation," *Chin. J. Phys.*, vol. 66, pp. 645–659, Aug. 2020.

[29] Z. M. Z. Muhammad and F. Özkaynak, "An image encryption algorithm based on chaotic selection of robust cryptographic primitives," *IEEE Access*, vol. 8, pp. 56581–56589, 2020.

[30] E. Barker and A. Roginsky, "Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths," NIST, Gaithersburg, MD, USA, Tech. Rep. 800-131A Rev. 2, Jan. 2011.

[31] Y. Alghamdi and A. Munir, "An image encryption algorithm based on trivium cipher and random substitution," *Social Netw. Comput. Sci.*, vol. 4, no. 6, Sep. 2023, Art. no. 713.

[32] J. Brownlee. (1977). *USC-SIPI Image Database*. Accessed: Apr. 20, 2022. [Online]. Available: https://sipi.usc.edu/database/database.php

[33] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, Aug. 2006.

[34] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, no. 3, pp. 749–761, Jul. 2004.

[35] L. Abraham and N. Daniel, "Secure image encryption algorithms: A review," *Int. J. Sci. Technol. Res.*, vol. 2, no. 4, pp. 186–189, 2013.

[36] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[37] J. Ahmad and F. Ahmed, "Efficiency analysis and security evaluation of image encryption schemes," *Computing*, vol. 23, no. 4, p. 25, 2010.

[38] K. Mali, S. Chakraborty, and M. Roy, "A study on statistical analysis and security evaluation parameters in image encryption," *Entropy*, vol. 34, p. 36, Jan. 2015.

[39] S. D. Sanap and V. More, "Performance analysis of encryption techniques based on avalanche effect and strict avalanche criterion," in *Proc. 3rd Int. Conf. Signal Process. Commun. (ICPSC)*, May 2021, pp. 676–679.

[40] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Sel. Areas Telecommun. (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[41] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, and A. Heckert, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," U.S. Dept. Commerce, Technol. Admin., Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-22 Rev 1a, 2001, vol. 22.

[42] X. Hu, L. Wei, W. Chen, Q. Chen, and Y. Guo, "Color image encryption algorithm based on dynamic chaos and matrix convolution," *IEEE Access*, vol. 8, pp. 12452–12466, 2020.
[43] Y. Wang, C. Wu, S. Kang, Q. Wang, and V. I. Mikulovich, "Multi-channel chaotic encryption algorithm for color image based on DNA coding," *Multimedia Tools Appl.*, vol. 79, nos. 25–26, pp. 18317–18342, Jul. 2020.
[44] A. U. Rehman, X. Liao, R. Ashraf, S. Ullah, and H. Wang, "A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2," *Optik*, vol. 159, pp. 348–367, Apr. 2018.
[45] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D sine logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.
[46] A. Qayyum, J. Ahmad, W. Boulila, S. Rubaiee, Arshad, F. Masood, F. Khan, and W. J. Buchanan, "Chaos-based confusion and diffusion of image pixels using dynamic substitution," *IEEE Access*, vol. 8, pp. 140876–140895, 2020.
[47] A. Shafique, M. U. Rehman, K. H. Khan, S. S. Jamal, A. Mehmood, and S. A. Chaudhry, "Securing high-resolution images from unmanned aerial vehicles with DNA encoding and bit-plane extraction method," *IEEE Access*, vol. 11, pp. 44559–44577, 2023.
[48] S. Agarwal, "Secure image transmission using fractal and 2D-chaotic map," *J. Imag.*, vol. 4, no. 1, p. 17, Jan. 2018.
[49] P. He, K. Sun, and C. Zhu, "A novel image encryption algorithm based on the delayed maps and permutation-confusion-diffusion architecture," *Secur. Commun. Netw.*, vol. 2021, pp. 1–16, Mar. 2021.
[50] M. Samiullah, W. Aslam, H. Nazir, M. I. Lali, B. Shahzad, M. R. Mufti, and H. Afzal, "An image encryption scheme based on DNA computing and multiple chaotic systems," *IEEE Access*, vol. 8, pp. 25650–25663, 2020.
[51] W. Alexan, M. ElBeltagy, and A. Aboshousha, "RGB image encryption through cellular automata, S-box and the Lorenz system," *Symmetry*, vol. 14, no. 3, p. 443, Feb. 2022.
[52] M. Khan, L. Khan, M. M. Hazzazi, S. S. Jamal, and I. Hussain, "Image encryption scheme for multi-focus images for visual sensors network," *Multimedia Tools Appl.*, vol. 81, no. 12, pp. 16353–16370, May 2022.
[53] C. Lakshmi, K. Thenmozhi, J. B. B. Rayappan, and R. Amirtharajan, "Hopfield attractor-trusted neural network: An attack-resistant image encryption," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11477–11489, Aug. 2020.
[54] A. Anees, A. M. Siddiqui, and F. Ahmed, "Chaotic substitution for highly autocorrelated data in encryption algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 9, pp. 3106–3118, Sep. 2014.
[55] F. A. Khan, J. Ahmed, J. S. Khan, J. Ahmad, and M. A. Khan, "A novel image encryption based on Lorenz equation, gingerbreadman chaotic map and S8 permutation," *J. Intell. Fuzzy Syst.*, vol. 33, no. 6, pp. 3753–3765, Nov. 2017.
[56] X. Wang and L. Liu, "Image encryption based on hash table scrambling and DNA substitution," *IEEE Access*, vol. 8, pp. 68533–68547, 2020.
[57] J. Ahmad and S. O. Hwang, "A secure image encryption scheme based on chaotic maps and affine transformation," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13951–13976, Nov. 2016.
[58] A.-V. Diaconu, "KenKen puzzle-based image encryption algorithm," *Proc. Romanian Acad., Ser. A*, vol. 16, pp. 313–320, Jan. 2015.

ABDULLAH ALAKLABI (Student Member, IEEE) received the B.S. degree in software engineering (SWE) from the University of Ha'il, Saudi Arabia, in 2011, and the M.S. degree in software engineering and information technology from the University of St. Thomas, Saint Paul, MN, USA, in 2020. He is currently pursuing the Ph.D. degree in computer science with Kansas State University, Manhattan, KS, USA. He is also a Researcher with the Intelligent Systems, Computer Architecture, Analytics, and Security (ISCAAS) Laboratory, Kansas State University, and a Lecturer with the Department of Computer Science, Shaqra University, Saudi Arabia. His research interests include image encryption, data encryption, cryptography, steganography, and information security.

ARSLAN MUNIR (Senior Member, IEEE) received the M.A.Sc. degree in electrical and computer engineering (ECE) from the University of British Columbia, Vancouver, Canada, in 2007, and the Ph.D. degree in ECE from the University of Florida, Gainesville, FL, USA, in 2012. From 2007 to 2008, he worked as a Software Development Engineer at the Embedded Systems Division, Mentor Graphics Corporation. He was a Postdoctoral Research Associate with the ECE Department, Rice University, Houston, TX, USA, from May 2012 to June 2014. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Florida Atlantic University. His current research interests include embedded and cyber-physical systems, secure and trustworthy systems, parallel computing, artificial intelligence, and computer vision. He has received many academic awards, including the Doctoral Fellowship from the Natural Sciences and Engineering Research Council (NSERC) of Canada. He earned gold medals for best performance in electrical engineering and gold medals and academic roll of honor for securing rank one in pre-engineering provincial examinations (out of approximately 300 000 candidates).

MUHAMMAD ASFAND HAFEEZ (Student Member, IEEE) received the B.S. degree in electrical engineering from the University of Management and Technology, in 2021, and the master's degree in IT convergence engineering from Gachon University, South Korea, in 2024. He is currently pursuing the Ph.D. degree in computer science with Kansas State University, Manhattan, KS, USA. He is also a Researcher with the Intelligent Systems, Computer Architecture, Analytics, and Security (ISCAAS) Laboratory, Kansas State University. His research interests include post-quantum cryptography, cryptography, parallel computing, and hardware implementations.

MUAZZAM A. KHAN KHATTAK (Senior Member, IEEE) received the Ph.D. degree from IIUI, in 2011, and the postdoctoral degree from the University of Missouri–Kansas City (UMKC), USA, in 2016. In 2013, he joined the National University of Sciences and Technology (NUST), Islamabad, Pakistan, and was promoted to the Associate Dean, in 2017. He has been with the School of Computer Science, University of Ulm, Germany, and the Networking and Multimedia Laboratory, School of Computer and the Electrical Engineering, UMKC, as a Research Fellow. He is currently an Associate Professor (Tenured) and the Head of the ICESCO Chair for data analytics and edge computing at Quaid-i-Azam University, Islamabad. He has published more than 150 publications and book chapters. His research interests include the Internet of Things, next-generation intelligent networks, blockchain, information and network security, vehicular ad-hoc networks, and acoustic networks. He is a member of Pakistan Academy of Sciences.