IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 September 2023, accepted 13 September 2023, date of publication 22 September 2023,
date of current version 28 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3318014

RESEARCH ARTICLE

IoT-Based Multi-Dimensional Chaos Mapping System for Secure and Fast Transmission of Visual Data in Smart Cities

BHARTI AHUJA[1], RAJESH DORIYA[1], SHARAD SALUNKE[2], MOHAMMAD FARUKH HASHMI[3], (Senior Member, IEEE), AND ADITYA GUPTA[4]

[1]Department of Information Technology, National Institute of Technology Raipur, Raipur, Chhattisgarh 492020, India
[2]Department of Electronics and Communication Engineering, Amity University Gwalior, Gwalior, Madhya Pradesh 474020, India
[3]Department of Electronics and Communication Engineering, NIT Warangal, Warangal, Telangana 506004, India
[4]Department of Information and Communication Technology, University of Agder, 4879 Grimstad, Norway

Corresponding author: Aditya Gupta (aditya.gupta@uia.no)

ABSTRACT A "smart city" sends data from many sensors to a cloud server for local authorities and the public to connect. Smart city residents communicate mostly through images and videos. Many image security algorithms have been proposed to improve locals' lives, but a high-class redundancy method with a small space requirement is still needed to acquire and protect this sensitive data. This paper proposes an IoT-based multi-dimensional chaos mapping system for secure and fast transmission of visual data in smart cities, which uses the five dimensional Gauss Sine Logistic system to generate hyper-chaotic sequences to encrypt images. The proposed method also uses pixel position permutation and Singular Value Decomposition with Discrete fractional cosine transform to compress and protect the sensitive image data. To increase security, we use a chaotic system to construct the chaotic sequences and a diffusion matrix. Furthermore, numerical simulation results and theoretical evaluations validate the suggested scheme's security and efficacy after compression encryption.

INDEX TERMS Smart city, compression, encryption, gauss map, sine map, logistic map, singular value decomposition, fractional cosine transform, image security.

I. INTRODUCTION

A "smart city" is an urban region that employs technological and data-driven solutions to enhance both the living standards of its citizens and the effectiveness and longevity of local services and infrastructure. Smart transportation systems, smart energy management, smart waste management, and smart governance are just a few examples [1]. The purpose of a "smart city" is to improve the quality of life via the use of information and communications technology to make cities safer, more secure, and more sustainable.

Digital images play a critical role in many applications of smart cities, including;

Public safety: Digital images can be used for surveillance and monitoring of public spaces, helping to prevent and investigate crime, and enhance emergency response.

Traffic management: Digital images from cameras and sensors can be used to monitor traffic flow and detect congestion, accidents, or other issues that can affect the safety and efficiency of urban mobility.

Environmental monitoring: Digital images can be used to track and analyze air and water quality, temperature, humidity, and other environmental factors to ensure a healthy and sustainable urban environment.

Infrastructure management: Digital images can be used to monitor and maintain critical infrastructure, such as bridges, tunnels, and buildings, to ensure safety and prevent damage or failures.

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

Tourism and marketing: Digital images can be used to promote and showcase the city's cultural and natural attractions to attract visitors and support the local economy.

In all these applications, digital images can provide valuable insights and data to help improve the quality of life, safety, and sustainability of urban environments. However, as this data is transmitted and stored digitally, it is also vulnerable to cyber attacks and other security threats. To ensure the protection of digital images in smart cities, measures such as encryption, authentication, and access control can be implemented to prevent unauthorized access, tampering, or theft of this data. Additionally, it is important to have policies and procedures in place to manage and secure the storage and transmission of digital images.

In the context of a smart city, the data produced by a variety of sources are saved on a cloud platform, where both concerned government agencies and people may access and change the data. Citizen data includes things like health records, shopping habits, weather reports, and traffic patterns. Images and other visuals play a significant role in the way people in smart cities share information with one another. Trying to gain access to images of people going about their daily lives can seriously damage personal and vital information. By developing strong encryption mechanisms and incorporating them into cloud infrastructure, data can be protected from unauthorized access. To improve residents' quality of life, numerous algorithms for image application must be proposed further. The locals' best interests could be in risk if there is no reliable way to transmit the image. Smart cities rely heavily on the ability to transmit images in a safe and efficient manner.

Chaos map based methods [2], [3], [4], [5], DNA sequence [6], automata [7], Quantum cryptography [8], Optical encryption [9], [10], Affine, Fractional Random Transform, Fast Fourier Transform [11], [12], [13] and Compressive sensing [14], [15] are only a few examples of the many image encryption and compression approaches that make use of cryptographic properties. Nevertheless, encryption alone is not adequate to fulfill the current requirement for safer communication and little storage space. This has led to several academics making important contributions and presenting their own versions of compressed cryptosystems.

The remaining article is structured as follows. In section II, introduces related work. Section III explains the suggested framework. The central ideas are broken down in detail in section IV. Part V provides examples of the recommended algorithm, while Part VI presents the simulation results via simulation experiments. Part VII is the climactic conclusion.

II. RELATED WORK

Image compression and encryption can be a useful technique in smart city applications for several reasons such as:

- Efficient use of storage and bandwidth: Smart city applications often involve the collection and transmission of large amounts of image data. Image compression can help reduce the size of these files, making them easier to store and transmit.
- Improved performance: Compressed images can be processed more quickly than uncompressed images, which can be important for real-time applications such as video surveillance.
- Privacy protection: compression can help to obscure details in images, which can be important for protecting the privacy of individuals captured in the images.
- Security: Encryption can be used in conjunction with compression to help protect sensitive images from unauthorized access or use. However, it's important to note that compression and encryption can both have an effect on the quality of the image information. In some cases, compression can result in a loss of detail or resolution, while encryption can add overhead to processing and transmission. As with any security or privacy technique, it's important to carefully consider the trade-offs and use the most appropriate methods for the specific application and context.

Furthermore, several researchers have successfully applied this in the past. Some recent and notable work is discussed in further depth. Ghaffari et al. [16] proposed sparse recovery and chaotic system for compression encryption. Sparse recovery is a technique used to recover a sparse signal from a small number of measurements. It involves finding a sparse representation of a signal using a linear transformation, such as the discrete cosine transform (DCT) or wavelet transform. The combination of sparse recovery and chaotic systems can be used for both compression and encryption of data. In this approach, the data is first transformed into a sparse representation using a linear transform. The sparse representation is then multiplied with a chaotic sequence generated by a chaotic system. The resulting signal is a chaotic sequence with a sparse representation. This signal can be compressed for efficient storage or transmission, and can be decrypted by first recovering the sparse representation using inverse transformations and then dividing by the chaotic sequence. The sparse encrypted form in two dimensions is compressed using the singular value decomposition. The next step is to use a chaotic confusion-based compression scrambled matrices to minimize the connection between adjacent pixels in the scrambled image. Similar to JPEG compression, the method reported by Chaudhary et al. [17] suggested column-wise scanning and optimization instead of zigzag scanning. JPEG uses arithmetic coding rather than Huffman coding during the entropy phase of compression. XOR encryption, which cannot be cracked, is used primarily to encode one-time pad images in this method.

According to Tang et al. [18] compressing and encrypting data at the same time is essential for the secure transmission of multimedia content, such as digital images. Due to the advancement of compressive sensing, this issue is now moot. Together, these two benefits of compressive sensing-reduced network transmission bandwidth and increased system security-make it an attractive option. To combat this,

compressive sensing cryptosystem requires storing the whole measurement matrix, but when integrated in a chaos, the system's confidentiality could be strengthened by tapping into the sensitivity of the chaos. This new global chaotic architecture creates the arrangement's chaotic map, which not only enhances the functionality of the chaotic map but also expands its chaotic range.

A new encryption scheme for colored images using advanced chaotic maps was also presented previously. While this method introduces innovation in image encryption through the utilization of high-level chaotic maps, there are other points to consider. These might include the possibility of increased computational complexity, leading to slower encryption and decryption processes. Furthermore, the scheme's effectiveness may depend heavily on the choice of chaotic maps, potentially resulting in reduced security if the maps are not chosen carefully [19]. Another work introduces an upgraded approach for image encryption that relies on sophisticated chaotic maps and an improved gravity model. While this method aims to enhance encryption security by utilizing intricate chaotic systems and refining the gravity model [20].

An image encryption algorithm that incorporates dynamic permutation and a large chaotic S-box was also given in the past. This approach [21] aims to enhance image encryption by introducing dynamic elements and chaotic components. The dynamic permutation process might introduce a higher computational overhead, leading to slower encryption and decryption speeds. Additionally, the effectiveness of the encryption could be sensitive to the parameters and initial conditions of the chaotic S-box, potentially impacting the security if not managed precisely.

A hybrid color image encryption method that relies on an extended logistic map was presented in the past. This approach [22] aims to enhance image encryption through the integration of a complex logistic map. Further the extended logistic map might introduce a higher computational load, possibly resulting in slower encryption and decryption processes. Moreover, the security of the encryption could be compromised if the parameters of the logistic map are not chosen carefully, potentially leading to vulnerabilities in the algorithm. These points should be carefully evaluated alongside the benefits when considering the adoption of this hybrid encryption method.

As established by Zhu et al. [23], in the context of image compression encryption, CS with chaotic measurement matrices can be used to compress an image and simultaneously encrypt it using the chaotic matrix as the encryption key. The compressed measurements can be transmitted or stored securely, and the original image can be reconstructed by solving an optimization problem that involves sparsity or compressibility constraints and the chaotic measurement matrix. The reconstruction can only be done by someone who has the correct chaotic matrix or key, which adds an extra layer of security to the image compression process. However, it should be noted that the use of chaotic measurement matrices in CS for image compression encryption is still an active area of research, and there are challenges and limitations that need to be addressed, such as the trade-off between security and compression performance, the sensitivity of chaotic systems to perturbations and noise, and the computational complexity of solving the optimization problem with the chaotic matrix. In parallel, an easy and efficient chaos-based Substitution box creation technique is created.

Zhu et al. [24] suggested one compression-encryption technique that combines a random Gauss matrix and sparse transform is the Random Gaussian Fourier Matrix (RGFM) technique. This technique uses a random Gaussian matrix as the measurement matrix and a sparse transform, such as the discrete cosine transform (DCT), as the basis for signal representation. The RGFM technique works by first transforming the image into the sparse domain using the DCT or other sparse transform. Then, the transformed image is compressed by multiplying it with a random Gaussian matrix to obtain a set of compressed measurements. The compressed measurements are then encrypted using a secret key, such as a block cipher or a stream cipher, to provide confidentiality and integrity protection. Finally, the encrypted compressed measurements are transmitted or stored. However, the RGFM technique also has some limitations. First, the security of the technique depends on the strength of the encryption algorithm used to encrypt the compressed measurements. Second, the performance of the technique depends on the choice of the random Gaussian matrix and the sparse transform basis, and different choices may yield different compression and encryption performance. Finally, the technique may not be suitable for images with non-sparse or non-compressible content, as the compression and encryption performance may degrade in such cases.

The 2D logistic sine map used by Ye et al. [25] combines permutation, modulation, and diffusion to create an image encryption technique. Repeatedly switching between rows and columns is a hallmark of this permutation method. The encrypted image is obtained by applying a modulation function to a permuted image and then diffusing the columns independently of one another. It fixes the problem with regular encryption techniques where the pixel position needs to be jumbled around thoroughly before the diffusion procedure can start.

DNA coding and fractional-order hyper-chaotic systems are used together for image compression encryption by Dong et al. [26]. To use both techniques together, the image data can first be compressed using a lossless compression algorithm such as Huffman coding or Lempel-Ziv-Welch (LZW) algorithm. The compressed data can then be converted into a sequence of nucleotides using DNA coding. Next, the fractional-order hyper-chaotic system can be applied to a seed value to generate a key that can be used to encrypt the DNA-coded image data. The encrypted data can then be transmitted over a secure channel or stored in a secure

location. To decrypt the data, the recipient would first apply the fractional-order hyper-chaotic system to the same seed value to generate the same key that was used to encrypt the data. The key can then be used to decrypt the encrypted DNA-coded data, which can then be decoded back into its original compressed image data using DNA decoding. Finally, the original image data can be reconstructed from the compressed data using a lossless decompression algorithm. Overall, using DNA coding and fractional-order hyper-chaotic systems together can provide a secure and efficient method for image compression encryption.

Yu et al. [27] suggested a method using SVD for image compression, encryption, and identity authentication. This approach may be used to establish identity authentication and encryption for cloud-based image data. Here, the SVD is utilized to decompose the picture data into its left and right singular value matrices.

By fusing block compressive sensing (BCS) with singular value decomposition (SVD) embedding, Zhu et al. [28] created an image encryption method. Compressive sensing uses the sparse representation of the original image in the DCT domain to create the coefficient vectors, which are then made confusing using the coefficient random permutation method in order to encrypt the image. One way to beef up security is to encrypt data using the hyper-chaotic Lorenz system.

A triple-image encryption method was proposed by Lidong et al. [29] based on a chaotic system, an S-box, and image compression. To build a new image, the 2D-LSCM (two-dimensional Logistic-Sine-coupling map) method first compresses three simple images by 25% before combining them using a random matrix. These enhancements over previous methods provide the proposed image encryption strategy a leg up in terms of the efficiency with which it transmits encrypted images. Z-scan and the recommended coded lock scrambling approach are then used to randomly jumble the pixel coordinates in the final image, resulting in a low time complexity. Scrambling the image and then using the diffusion process with S-box and chaotic sequences results in a cipher image.

As per the high security needs and light weight transmission for saving data storage space, a compression encryption multidimensional approach is presented in this work. Memory requirements and the appropriate structure of the predicted output from the compression method also affect the overall usefulness of the produced work. The compression algorithm's foundation relies heavily on its performance. In spite of the fact that SVD is a method that requires a significant amount of computational power, it has proven to be an effective factorization algorithm for obtaining specific information about matrices. Data compression is one of the many uses of this technology because it has the potential to lessen the amount of information necessary for encoding an image while preserving the image's quality.

Security algorithms with limited key space are susceptible to various forms of attacks. To address this issue, the study proposes the development of an image encryption algorithm rooted in multi-dimensional chaotic systems based on novel five dimensional Gauss Sine Logistic system to generate hyper-chaotic sequences. The use of low-dimensional chaotic maps offers a more streamlined architecture due to the reduced number of system components. But by employing techniques for estimating chaotic signals, it becomes possible to predict system characteristics and initial values for low-dimensional chaotic systems. Conversely, the proposed multi-dimensional chaotic systems exhibit remarkable chaotic behavior alongside intricate structures.

We draw on the preceding discussion to propose a novel compression and encryption multidimensional scheme in this paper.

A. MOTIVATION

The motivation behind developing an IoT-based multi-dimensional chaos mapping system for secure and fast transmission of visual data in smart cities arises from the need to effectively address the dual challenges of rapid urbanization and data communication. As smart cities increasingly rely on visual data for various applications, including traffic management and environmental monitoring, there is a pressing demand for a transmission system that can ensure both speed and security. Traditional methods often struggle to provide this balance, leading to potential vulnerabilities. By leveraging the inherent complexity of chaos mapping and the connectivity of IoT devices, this innovative system seeks to optimize data transmission efficiency by simultaneously mapping multiple dimensions of visual data, ensuring real-time delivery, and enhancing security through collaborative encryption techniques. Ultimately, the proposed system aims to redefine the way data is exchanged in smart cities, fostering a more efficient, secure, and responsive urban environment.

B. CONTRIBUTION

The major contribution of the IoT-based multi-dimensional Gauss logistic sine chaotic mapping system that uses SVD with DFRCT for image compression encryption can be summarized as follows:

1. Improved Security: The proposed chaotic mapping system provides better security for image encryption due to its highly non-linear and chaotic behavior. The combination of different chaotic maps such as Gauss, logistic, and sine ensures that the encryption key is highly unpredictable and difficult to crack.

2. Efficient Compression: The use of SVD and DFRCT techniques in the proposed system helps to reduce the size of the encrypted image while preserving its quality. This is achieved by compressing the image in the frequency domain and removing redundant information.

3. Multi-Dimensional Mapping: The use of multi-dimensional chaotic maps such as Gauss, logistic, and sine in the proposed system allows for efficient encryption and compression of high-dimensional images. This is particularly

useful in IoT applications where images may be captured from different sources and in various formats.

4. Low Complexity: The proposed system has a low computational complexity and can be easily implemented on resource-constrained IoT devices. This makes it suitable for real-time image encryption and compression in IoT applications.

5. Low correlation coefficients between encrypted images ($-0.0034, -0.0065, -0.0008$) show that the proposed work's key is sensitive to the variables and security of the encryption system used.

6. Comparison of important quality metrics of the encrypted image to the evaluating indicators of the decrypted image demonstrates that the proposed cryptosystem is far more effective and secure than the other image cryptosystems already in use.

Overall, the IoT-based multi-dimensional Gauss logistic sine chaotic mapping system that uses SVD with DFRCT for image compression encryption provides an efficient and secure method for protecting sensitive images in IoT applications.

III. SUGGESTED FRAMEWORK

Public transport, digital administration, environmental services, medical services, and education are just some of the urban infrastructure areas that ''smart cities'' aim to enhance for the benefit of their citizens. The effective use of information and communication technologies enables delivery of services at the required quality. Data is processed further in a smart city setting before being used to inform policy and action. Each day, countless gigabytes of data are produced by smart city applications. Cloud servers are designed to store and manage information of this nature. Now, thanks to cloud computing, we can take advantage of a vast array of useful online tools and resources. Cloud service infrastructure is used by smart city applications. Data transmission and storage are simplified by the built-in security, authenticity, and integrity standards of the cloud platform. It is crucial to protect the privacy of personal information when using digital devices. Our goal is to provide a cryptosystem strong enough to safeguard digital picture data used by smart cities.

Smart city architecture typically employs various measures to protect its image data. Here are some examples:

- Encryption: Smart city systems can encrypt image data as it is transmitted or stored, using algorithms that make it difficult for unauthorized individuals to access or decipher the data.
- Access controls: Access controls are put in place to restrict who can access the image data, and what they can do with it. These controls can include password protection, two-factor authentication, and permissions-based access.
- Data storage security: Smart city systems store image data in secure locations, such as data centers or cloud-based storage facilities that are protected by physical and digital security measures.
- Monitoring and auditing: Smart city systems can use monitoring and auditing tools to track who is accessing image data, when they are accessing it, and what they are doing with it. This can help identify and prevent unauthorized access or misuse.
- Anonymization: Smart city systems may use techniques such as blurring or masking to anonymize image data, so that individuals cannot be identified in the images.
- Data retention policies: Smart city systems can establish data retention policies that dictate how long image data will be stored and when it will be deleted. This can help prevent the unauthorized use or retention of data.

Overall, protecting image data is an important aspect of smart city architecture, and requires a combination of technical, organizational, and policy measures to ensure the security and privacy of individuals and communities.

As for as images are concerned, it could be used to represent a wide range of data, including clinical data of patients, actual traffic incidents, climate patterns, and so on. Sharing and storing encrypted image data in the cloud is a secure option. Once the decoding procedure is applied at the receiving end, the image data can be used. If data is encrypted in the cloud, the opposition cannot access it.

Figure 1 illustrates the suggested architecture for safe data transfer in cloud-based smart cities. The digital eyes, such CCTV cameras etc., are constantly recording scenes, people, events, and happenings that may be used as inputs for this architecture. All of these pictures are encrypted using the recommended technique for top-tier security and are stored in the cloud. This information is accessible to any citizen or government official who has been granted access and registered to do so.

IV. BACKGROUND OF THE PROPOSED METHODOLOGY

A. SVD

Singular value decomposition (SVD) is a powerful mathematical tool used in linear algebra and data analysis. It is used to break down a matrix into its constituent parts, revealing the structure and information hidden in the data. SVD can be used to reduce the dimensionality of a dataset, discover patterns in the data, and more. SVD can be used to compress images by representing them as a linear combination of rank-one matrices. This reduces the amount of data that needs to be stored and can lead to considerable savings in storage and transmission costs. Additionally, the use of SVD can improve the quality of an image, since the rank-one matrices are able to capture the important features of an image. As can be seen in equation 1, the matrix S is decomposed into three sub matrixes, denoted by the letters P, b, and Q. Matrix P is orthogonal to Matrix Q, and Matrix b is a diagonal matrix whose entries are sorted from highest to lowest singular value.

$$S_{m\times n} = P_{m\times m} b_{m\times n} (Q_{n\times n})^T \tag{1}$$

Following application of the SVD, the vast majority of singular values will be thrown out, with only a small subset being stored. We can draw the conclusion that there is very little

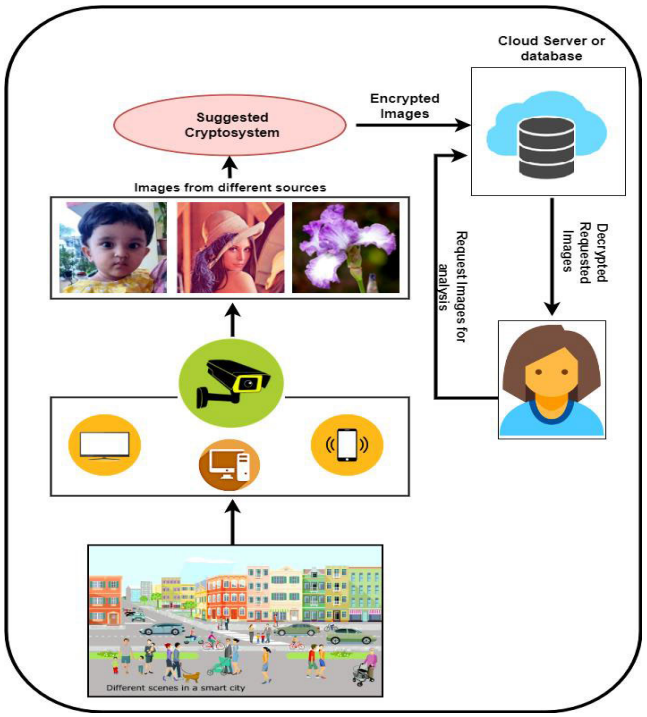

FIGURE 1. Suggested architecture for safe data transfer in cloud-based smart cities.

information stored in the lower singular values. As a result, ignoring these parameters results in a decreased file size with no noticeable distortion.

B. DFRCT

The Discrete Fractional Cosine Transform (DFRCT) is indeed a generalization of the Discrete Cosine Transform (DCT). The DCT is a type of transform that is used to represent a signal as a sum of cosine functions with different frequencies and coefficients. The DFRCT extends the DCT by allowing fractional orders, which provides a more flexible and powerful representation of signals. This makes the DFRCT well-suited for image compression and encryption applications, where the ability to capture certain signal features is important.

In image compression, the DFRCT is used to transform image data into a compact representation, making it easier to store and transmit. In encryption, the DFRCT can be used to scramble image data in a secure and reversible manner, protecting it from unauthorized access or manipulation. The kernel matrix of the DFRCT is defined as follows [30]:

$$R_c^{\alpha} = V_c D_c^{\alpha} V_c^{t} \tag{2}$$

In the equation V_c represents unitary matrix made from the eigen vector and D_c is diagonal matrix of eigen values for R_c. Mathematically it is expressed as:

$$D_c^{\alpha} = diag\left[1, \exp(-4i\pi\alpha/M), \ldots, \exp(-2i\pi(2N-2)\alpha/M)\right] \tag{3}$$

C. CHAOTIC MAP

A chaotic map is a mathematical function that exhibits chaotic behavior, which is characterized by extreme sensitivity to initial conditions and unpredictable long-term behavior. Chaotic maps are used in many applications, including cryptography, random number generation, and data encryption.

In a chaotic map, a small change in the initial conditions or the parameters of the function can lead to a large change in the output values. This sensitivity to initial conditions is what makes chaotic maps useful for generating pseudo-random numbers, which are important in many applications that require secure communication or randomization [31]. One popular chaotic map used in image encryption is the logistic map.

Logistic map is a type of chaotic map that exhibits complex and random behavior. It can be used for image encryption by generating a random key sequence that is used to encrypt and decrypt the image. Mathematically, it is expressed by [32]:

$$z_{n+1} = a * z_n * (1 - z_n) \tag{4}$$

Other examples of chaotic maps include the Sine map and Gauss map that are used in the work. Sine map and gauss map is a one-dimensional chaotic map that is commonly used in dynamical systems and chaos theory. They are defined by the following recurrence relation [33], [34]:

$$y_{n+1} = q * sin\,(\pi y_n) \tag{5}$$

$$x_{n+1} = \exp(-\alpha * x_n * x_n) + \beta \tag{6}$$

where y_n and x_n are the state of the system at time n and q, and α is a parameter that determines the behavior of the map. In this work, because of the superior performance and the same class of chaotic trajectory, t he said three chaotic maps are being used together.

V. PROPOSED METHODOLOGY

A. 5D GAUSS SINE LOGISTIC MAP

With some necessary modifications, we combine the Gauss, Sine and Logistic maps to create a novel hyper chaotic mapping system. The following is a description of the new hybrid mapping system:

$$\left.\begin{aligned} x_{i+1} &= e^{(-r'x_i^2)} + t + qy_i^2 x_i + pz_i^3 \\ y_{i+1} &= r * Sine\pi * y_i + qz_i^2 y_i + px_i^3 \\ z_{i+1} &= rz_i(1 - z_i) + qx_i^2 z_i + py_i^2 \\ w_{i+1} &= rw_i(1 - w_i) + qs_i^2 w_i + pz_i^2 \\ s_{i+1} &= rs_i(1 - s_i) + qx_i^2 s_i + pw_i \end{aligned}\right\} \tag{7}$$

The first two equations describe the Gauss and Sine map respectively, while the last three equations describe the Logistic map. The 5D Gauss Sine Logistic hyper chaotic system is utilized in the proposed encryption algorithm to generate hyper-chaotic sequences. These sequences are extremely delicate and sensitive to initial conditions. Where p, q, r, r', and t are control parameters, selected to achieve the system behavior of hyper chaos, with $p = 0.0135, q = 0.0177, r = 3.75, r' = 4.9$ and $t = -0.58$.

B. PERMUTATION

To create an unstable image, pixel position permutation can be used to shuffle around the pixels. This operation may be performed arbitrarily or with a permutation key. The proposed algorithm employs row and column permutation of the input image using the sequence generated in the equation 1st, 2nd, 3rd, and 4th dimensions of the 5D Gauss Logistic chaotic map. In the following sections, we will discuss the row and column permutation in depth.

1) ROW PERMUTATION

For simplicity, we'll assume that the first-dimensional chaotic values acquired from equation 2 of size 1×4 are x = [1 2 3 4]. After that, a size-k variable is assigned a random selection of values from the 'x' vector. With x = [1 2 3 4], we choose k = [2 3] as an example. The P input matrix is shown in Figure 2. It is 2×2 Matrix. A shift to the right is made if the value of the chaotic sequence k is an even number; otherwise, a shift to the left is made. According to the phenomena R1's values are shifted right 2 times and R2's values are shifted left 3 times. The Q is a row permutated matrix which is shown in Figure 3. In the algorithm two times row permutation process is applied, 1st time for x sequence and 2nd time for z sequence.

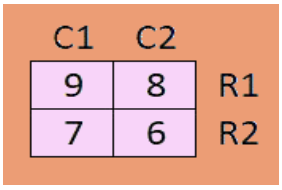

FIGURE 2. Input matrix.

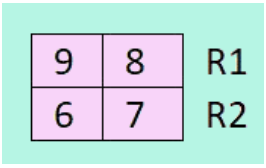

FIGURE 3. Row permuted matrix.

2) COLUMN PERMUTATION

In column permutation, we'll assume that the 2nd-dimensional chaotic values acquired from equation 3 of size 1×4 are y = [4 3 2 1]. After that, a size-l variable is assigned a random selection of values from the 'y' vector. With y = [4 3 2 1], we choose k = [32] as an example. The P input matrix is shown in Figure 1. It is a 2×2 Matrix. A shift to the upward is made if the value of the chaotic sequence l is an even number; otherwise, a shift to the downward is made. According to the phenomena C1's values are shifted downward 3 times and C2's values are shifted upward 2 times. The R is column permutated matrix which is shown in Figure 4. In the algorithm two times column permutation process is applied, 1st time for y sequence and 2nd time for w sequence.

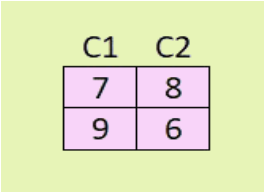

FIGURE 4. Column permuted matrix.

C. CONFUSION WITH XOR

Combining the shuffled image with a key image and using the XOR logical operation allows for a more robust encryption and more randomness in the image's pixels. The fifth 's' axis of the Gauss Logistic hyper chaotic map is used to construct the key image. The 's' dimensionally generated chaotic sequence is recast as a matrix with dimensions [M, N]. The input image size is M × N. The shuffled image is obtained from the resultant of the permutation process. The key image and the permutation (shuffled) image are now bit-XOR logically combined.

D. ALGORITHM

The proposed cryptosystem is depicted in a flow chart format in Figure 5 and 6. The steps of the encryption and decryption algorithm are explained here.

Step 1: To get the sparse representation, we first factorized the input image matrix into multiple matrices using SVD and apply approximation on diagonal matrix.

Step 2: Regenerate the matrix and remove singularity and get compressed image A.

Step 3: Compressed image A is transformed with DFRCT two times using different keys successively by rows and by columns.

Step 4: Secret keys and other necessary parameters should be used to initialize the Gauss Sine Logistic hyper chaotic map (x, y, z, w and s).

Step 5: After the initial values have been set, the map is iterated for a large enough number of times (T), where T is greater than or equal to 10,000, to produce highly chaotic results. In order to normalize the resulting x, y, z, w and s vectors, we use the equalization formula in Equation 7. Here S = 10000 and M = 256 if image size is 256 × 256.

$$\left.\begin{aligned} x &= ceil((x * S) \bmod M) \\ y &= ceil((y * S) \bmod M) \\ z &= ceil((z * S) \bmod M) \\ w &= ceil((w * S) \bmod M) \\ s &= ceil((s * S) \bmod M) \end{aligned}\right\} \tag{8}$$

Step 6: Select P, Q, and R, random numbers (as explained in section V-B) and using x, y, z, w and s generate five sequences k, l, g, f and m.

Step 7: Record the compressed image A, apply permutation operation using k, l, g and f sequences in row and column wise.

Step 8: After that XOR the shuffle image with m sequence or key image and get the compressed encrypted image T.

Step 9: At the receiver end, repeat the step 4, 5 and 6. And apply permutation operation using k, l, g and f sequences in row and column wise in image T and get image U.

Step 10: After that XOR the image U with m sequence or key image and get image V.

Step 11: To decrypt the image, the inverse process is followed by applying the inverse DFRCT to obtain the frequency coefficients, and then applying the inverse SVD to obtain the decrypted image.

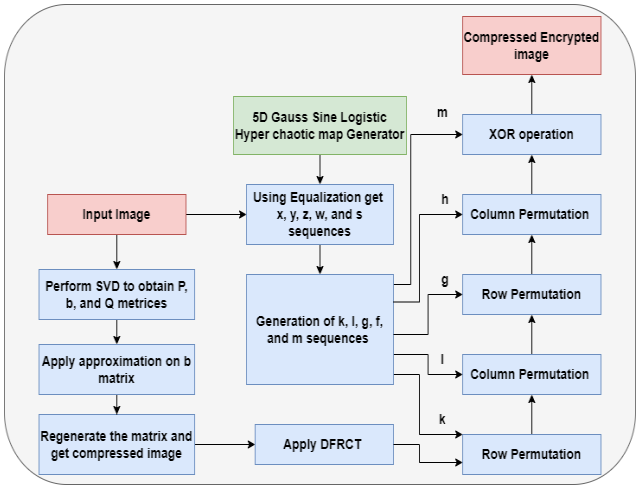

FIGURE 5. Encryption process.

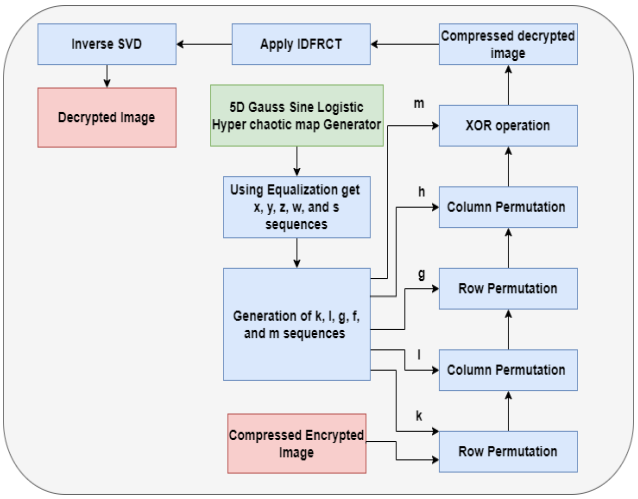

FIGURE 6. Decryption process.

FIGURE 7. Simulation results of the proposed algorithm: original, encrypted, and decrypted colored test images.

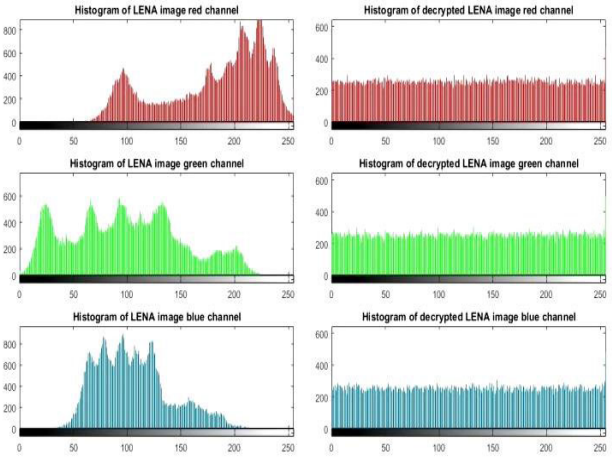

FIGURE 8. Histogram analysis.

VI. RESULT ANALYSIS

The simulation tests are performed using Matlab R2016a on a system with a 2.30 GHz CPU and 8 GB RAM to confirm the validity of the suggested scheme. Flower, Lena, Pepper, Aeroplane, Baboon, Tree, House etc. are among the 256×256 and 512×512 sized images chosen as the test images. The secret keys are set *to* $x_0 = 0.3250, y_0 = 0.4250, z_0 = 0.5250, w_0 = 0.4350$ and $s_0 = 0.5350$ respectively. For the demonstration, the USC-SIPI image collection and a well-known standard test image are used as sources [35], [36]. A Lena color image and flower image with 256×256 pixels are given as examples in this research. Figure 7 shows the results of the Lena and flower images in both plain-text and cipher-text modes.

A. HISTOGRAM ANALYSIS

The histogram of a perfect image encryption technique would be flat, as the likelihood of each pixel value would be evenly distributed across the image. The encrypted images are uncrackable by statistical methods if the histogram remains flat. Histograms for each channel's original Lena image and encrypted version are shown in Figure 8. Our approach is strong because the encrypted image's pixel distribution is more even than the original's, and it can withstand attacks in some instances.

Furthermore, the Chi-square test is used as quantitative metric to demonstrate the attained uniformity. To perform the chi-square test, the encrypted image data is divided into a

number of bins or intervals, and the number of pixels falling into each bin is counted. Then, the expected number of pixels in each bin is calculated assuming a uniform distribution, and the observed and expected counts are compared using a statistical test. The chi-square test statistic is calculated from the differences between the observed and expected counts, and a p-value is calculated from the chi-square distribution [37]. If the p-value is below a certain significance level, typically 0.05, then it indicates that the distribution of pixel values in the encrypted image is significantly different from a uniform distribution, suggesting that the encryption algorithm may not be sufficiently random or secure. Table 1 illustrated the chi square analysis.

TABLE 1. Chi square test for the data used in the study.

Images	p values	Decision
Kia	0.32812	Accept
Flower	0.23459	Accept
Lena	0.65432	Accept
Pepper	0.26543	Accept
Aeroplane	0.43268	Accept
Baboon	0.11235	Accept
Tree	0.65432	Accept
House	0.23684	Accept

B. CORRELATION COEFFICIENT

Correlation coefficient is a statistical measure that describes the strength and direction of the linear relationship between two variables. In the context of image encryption, correlation coefficient is often used as a metric to evaluate the quality of encryption algorithms [38].

In image encryption, the correlation coefficient between the original image and the encrypted image should be as low as possible to ensure that the encrypted image does not reveal any information about the original image. A high correlation coefficient between the original and encrypted images indicates that the encryption algorithm may not be secure and could potentially be vulnerable to attacks.

In the original image, the correlation value between any two neighboring pixels is extremely close to 1 (Horizontal, Vertical, and Diagonal). Therefore, the correlation values between the original and encrypted versions of an image should be small. Table 2, 3, and 4 display the study's findings, demonstrating that the correlation values of cipher images in Red, Green and Blue channels are extremely low. Figure no. 9, 10, and 11 shows the red, green, and blue channels of the original and encrypted Lena images, respectively, with their correlation distributions in all directions before and after encryption.

C. SECURITY ANALYSIS

The level of security provided by an algorithm is directly linked to the size of its encryption key. The IEEE floating-point norm is generally useful for quantifying the keyspace. The following cryptographic keys are used in the proposed system: p, q, r, r', t, x_0, y_0, z_0, w_0, s_0 (key used in the Gauss Sine logistic system) and a, b (key used in the DFRCT). In this context, the overall key size can be expressed as;

$$\begin{aligned} keyspace &= 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \\ &\quad \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \\ &= 10^{180} \approx 2^{597} \end{aligned} \tag{9}$$

With such a substantial key size, the likelihood of successfully guessing the key is exceedingly low. This assertion is supported by the analysis of the key space, which demonstrates that employing a brute force technique to decipher the encrypted information is also a formidable challenge.

However, a potential weak point emerges when the key itself needs to somehow be transmitted to the receiver. To address this concern, asymmetric key encryption methods such as RSA or Diffie Hellman can be employed. These techniques ensure the legitimacy of the key for both the sender and the receiver, thereby bolstering the security of the key exchange process.

TABLE 2. Correlation coefficients in red channel for the data used in the study.

Images	Horizontal	Vertical	Diagonal
Kia	-0.0006	-0.0053	-0.0148
Flower	-0.0066	-0.0294	-0.0027
Lena	-0.0065	-0.0021	-0.0087
Pepper	0.0051	-0.0084	-0.0207
Aeroplane	-0.0020	0.0122	-0.0014
Baboon	-0.0098	-0.0070	-0.0009
Tree	-0.0079	-0.0308	-0.0305
House	-0.0027	-0.0106	-0.0092

TABLE 3. Correlation coefficients in green channel for the data used in the study.

Images	Horizontal	Vertical	Diagonal
Kia	-0.0008	-0.0012	0.0123
Flower	-0.0066	-0.0294	-0.0027
Lena	-0.0145	-0.0098	-0.0054
Pepper	0.0076	-0.0098	-0.0376
Aeroplane	-0.0065	0.0987	-0.0043
Baboon	-0.0018	-0.0098	-0.0004
Tree	-0.0087	-0.0784	-0.0089
House	-0.0045	-0.0043	-0.0028

TABLE 4. Correlation coefficients in blue channel for the data used in the study.

Images	Horizontal	Vertical	Diagonal
Kia	-0.0003	-0.0067	-0.0764
Flower	-0.0033	-0.0245	-0.0076
Lena	-0.0876	-0.0052	-0.0065
Pepper	0.0023	-0.0076	-0.0342
Aeroplane	-0.0052	0.0234	-0.0043
Baboon	-0.0087	-0.0065	-0.0003
Tree	-0.0065	-0.0432	-0.0876
House	-0.0034	-0.0236	-0.0075

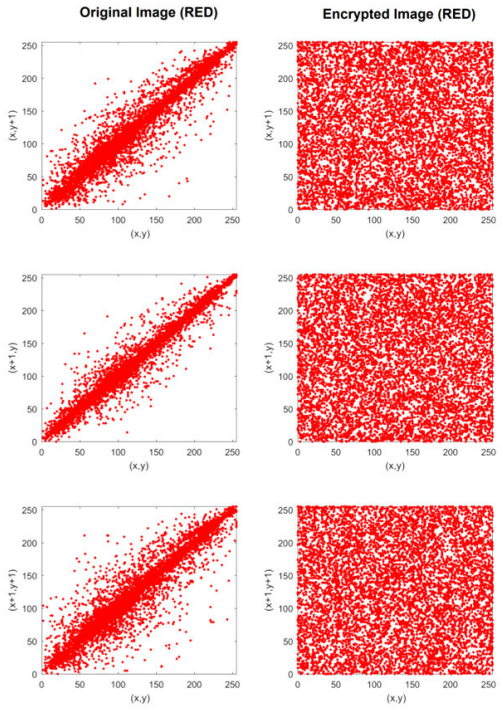

FIGURE 9. Correlation analysis for lena image with red component.

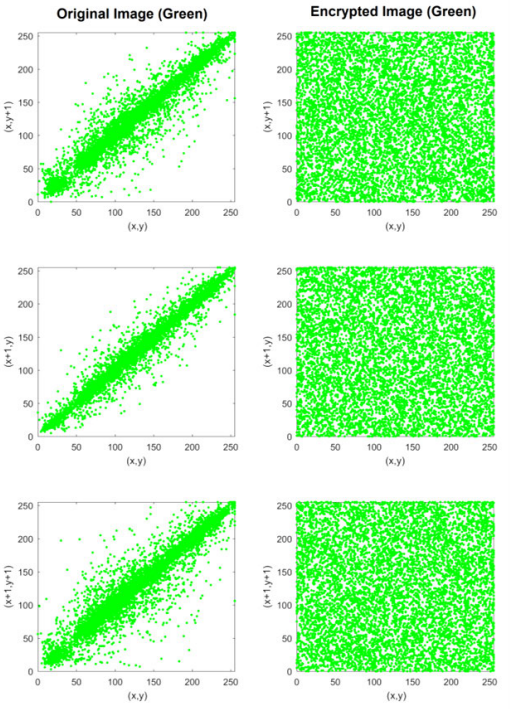

FIGURE 10. Correlation analysis for lena image with blue component.

D. KEY SENSITIVITY ANALYSIS

For a cryptosystem to be really safe, it must be extremely sensitive to even the smallest of modifications made to the secret keys. While encrypting the same plain image, a competent cryptosystem, for instance, should produce entirely different cipher images for each of the possible secret keys. Moreover, any shift in the decryption keys should prevent the plain image from being retrieved.

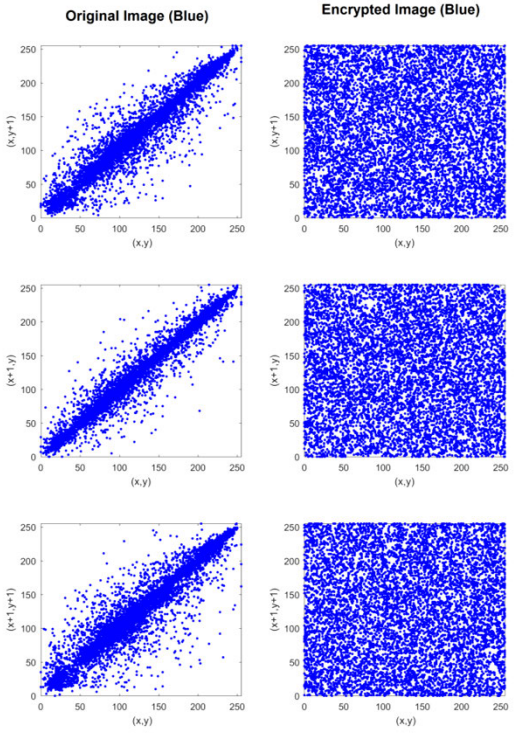

FIGURE 11. Correlation analysis for lena image with blue component.

TABLE 5. CDR calculation for altered keys.

Altered secret key	Slight change in	CDR(%)
x_0	x_0^1	99.65
	x_0^2	99.64
y_0	y_0^1	99.61
	y_0^2	99.63
z_0	z_0^1	99.62
	z_0^2	99.61
w_0	w_0^1	99.61
	w_0^2	99.63
s_0	s_0^1	99.62
	s_0^2	99.61

To evaluate the key sensitivity of an encryption algorithm, the cipher-text difference rate (CDR) can be calculated for different variations of the encryption key. For example, small changes can be made to the key, such as flipping a single bit, and the resulting encrypted image data can be compared to the original encrypted data using the cipher-text difference rate.

If the cipher-text difference rate is high, then the encrypted data is significantly different from the original, suggesting that the encryption algorithm is sensitive to changes in the key. Table 5 shows the CDRs generated for encryptions as a result of changing secret keys as a percentage. In general, a CDR of more than 99% is considered adequate key sensitivity for an encryption scheme. Considering the data in Table 5,

we can infer that the proposed cryptosystem has sufficient key sensitivity to fulfill the aforementioned condition. Figure 12 also displays the results of encrypting a Lena image using a secret key that has been slightly modified. The subfigures a, b, c, d, e and f show the original test image, its related encrypted image AC, its related encrypted image with a slight difference in one of the secret keys AC', absolute intensity differences (|AC - AC'|) of corresponding pixels of encrypted images, and the histogram of AC' image, and the histogram of intensity difference (|AC - AC'|).

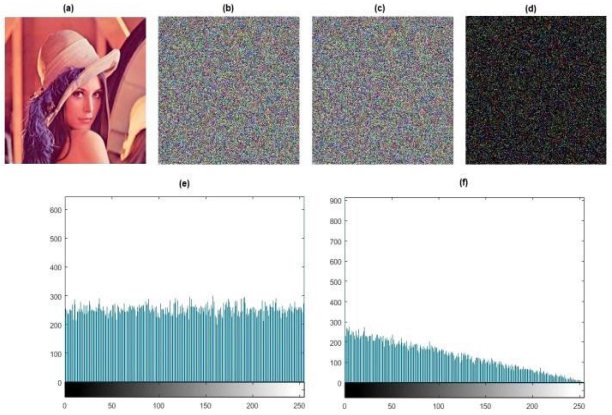

FIGURE 12. Key sensitivity analysis.

E. PSNR

PSNR (Peak Signal-to-Noise Ratio) is a commonly used metric to evaluate the quality of an image or video signal after it has been processed or compressed. It measures the ratio between the maximum possible power of a signal and the power of the noise that affects the fidelity of its representation. In image and video processing, PSNR is commonly used to measure the quality of a compressed image or video signal relative to the original uncompressed signal. Higher PSNR values indicate better quality, as the compressed signal has less noise or distortion relative to the original signal.

In the equation 10, it is defined mathematically where $f'(i,j)$ is the decrypted image and $f(i,j)$ is the original image. MSE is the mean squared error between the original image and the processed or compressed version of it. The image's size is represented by M and N, respectively [39].

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[f'(i,j) - f(i,j)\right]^2 \tag{10}$$

$$PSNR = 10\log_{10}\left[\frac{256\times 256}{MSE}\right] \tag{11}$$

The PSNR values of several images at varying Compression Ratio (CR) are shown in Table 6.

According to the data in the table, the proposed method's image reconstruction quality is better than competing methods' at similar compression ratios. The suggested technique still achieves high quality image reconstructions even with limited image sample data.

TABLE 6. PSNR values of different images under different CR.

Images/ CR	0.95	0.75	0.5	0.25
Kia	52.63	51.345	50.45	49.45
Flower	57.06	56.23	55.67	54.56
Lena	48.90	47.23	45.87	44.90
Pepper	42.26	41.34	40.23	39.76
Aeroplane	44.47	43.12	42.02	41.34
Baboon	48.59	46.45	45.67	44.23
Tree	42.26	41.23	40.34	39.87
House	49.34	48.25	47.25	45.34

F. SSIM

SSIM is a metric that compares two images' brightness, contrast, and structure to find their amount of similarities. Mathematically, it is defined as [40]:

$$SSIM\ (p,q) = l\ (p,q)\ c\ (p,q)\ s\ (p,q) \tag{12}$$

where p is the original image, q is the decrypted image. The values of SSIM might range from 0 to 1. When the SSIM values are higher, it indicates that the two images are more similar to one another. The SSIM values of the various images shown in Table 7.

TABLE 7. SSIM of test images.

Images	p values
Kia	0.9994
Flower	0.9997
Lena	0.9997
Pepper	0.9987
Aeroplane	0.9875
Baboon	0.9992
Tree	0.9930
House	0.9992

G. GLOBAL ENTROPY

Global entropy refers to the overall randomness or uncertainty in the image data, and is usually measured using a metric like Shannon entropy. This measures the average amount of information per pixel in the entire image. High global entropy indicates that the image data is highly randomized and difficult to predict, which is desirable for encryption. The entropy E(m) of a data source m is defined by the Shannon or Global information theory as [41]:

$$E\ (m) = \sum_{i=0}^{L-1} p(m_i)\log_2\frac{1}{p(m_i)} \tag{13}$$

If one wants to generate a completely random 256-level grayscale image, the entropy of the data is 8. Table 8 displays the measured Global entropies of encrypted image data. The encrypted images' information entropies are very near to the theoretical value of 8, demonstrating that the cryptosystem is secure against entropy attacks and disregarding any data breaches that might occur during the encryption process.

TABLE 8. Global entropy of test images.

Images	Encrypted Image Entropy
Kia	7.9993
Flower	7.9993
Lena	7.9993
Pepper	7.9995
Aeroplane	7.9997
Baboon	7.9994
Tree	7.9993
House	7.9996

H. LOCAL ENTROPY

Local entropy, on the other hand, refers to the randomness or uncertainty in specific regions or blocks of the image. This is useful for detecting patterns or correlations that may exist within the image data, which can be exploited by attackers attempting to break the encryption. Table 9 displays the measured information entropies of encrypted image data.

TABLE 9. Local entropy of test images.

Images	Encrypted Image Entropy
Kia	7.9043
Flower	7.9056
Lena	7.9032
Pepper	7.9012
Aeroplane	7.9064
Baboon	7.9021
Tree	7.9064
House	7.9023

In image encryption, both global and local entropy are important for ensuring that the encrypted image data is secure and difficult to reverse-engineer. A good encryption algorithm should produce high global entropy while minimizing correlations and patterns within the image that could be exploited by attackers.

I. DIFFERENTIAL ATTACKS

If you change just one pixel in the original image, the cipher you get back will be completely different, as secure image encryption methods are known to be extremely sensitive to such modifications. NPCR stands for ''Number of Pixel Changes Rate'', which is a measure used to evaluate the sensitivity of an image encryption algorithm to changes in the plaintext image [42]. Specifically, NPCR measures the percentage of pixels that change in the cipher text image when one bit of the plaintext image is modified. Higher NPCR values indicate that small changes in the plaintext image will result in large changes in the cipher text image, which is desirable for encryption algorithms.

UACI stands for ''Unified Average Change Intensity'', which is another measure used to evaluate the performance of image encryption algorithms. UACI measures the average difference in pixel values between the original plaintext image and the corresponding decrypted image after encryption and decryption. Lower UACI values indicate better performance, as it means that the decrypted image is closer to the original plaintext image. The general expression for NPCR and UACI are shown in equation 14 and15.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} K(i,j) \times 100\% \tag{14}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|a_1(i,j) - a_2(i,j)|}{255} \times 100\% \tag{15}$$

where M and N represent the number of rows and columns in the image, and a_1 and a_2 represent the encrypted versions of the image before and after a pixel change in the plain image. The perfect NPCR for an image is 99.6094%, and the perfect UACI is 33.4635%.

TABLE 10. NPCR and UACI values of test images.

Images	NPCR	UACI
Kia	99.56	33.58
Flower	99.60	33.60
Lena	99.62	33.59
Pepper	99.64	33.60
Aeroplane	99.60	33.60
Baboon	99.61	33.58
Tree	99.63	33.49
House	99.59	33.58

In this study, one pixel in the original image was changed at random. Table 10 summarizes the NPCR and UACI performance findings for the suggested technique. NPCR and UACI values are quite close to the theoretical values, demonstrating the cryptosystem's robustness against differential assaults.

J. COMPUTATIONAL COMPLEXITY

Image encryption algorithms can have different levels of computational complexity depending on the specific algorithm used. Computational complexity refers to the amount of time and resources required to execute an algorithm on a computer. Some image encryption algorithms are designed to be computationally efficient, meaning they can encrypt or decrypt images quickly and with low resource requirements. Others may have higher computational complexity, meaning they take longer to execute or require more resources such as memory or processing power.

The choice of algorithm used for image encryption will depend on the specific application and the desired level of security. Generally, more complex algorithms are considered more secure because they are more difficult to break. However, this comes at the cost of increased computational requirements.

The time complexity of an image encryption algorithm depends on the specific algorithm used and the size of the image being encrypted. Generally, the time complexity of an algorithm is expressed in terms of the number of operations required to perform the encryption or decryption operation as a function of the input size.

TABLE 11. PSNR during noise attack.

Images	Noise level 0.05	Noise level 0.005
Kia	21.058980	29.423551
Flower	19.580526	27.836950
Lena	21.454256	29.602246
Pepper	20.787867	28.873604
Aeroplane	20.739511	28.663009
Baboon	21.714945	29.839683
Tree	20.874242	28.715964
House	21.703823	29.843485

TABLE 12. PSNR during cropping attack.

Images	25% Data loss	6.25% Data loss
Kia	19.093962	25.042906
Flower	18.488860	24.485924
Lena	18.580791	24.537514
Pepper	19.931532	25.934694
Aeroplane	18.945061	24.856851
Baboon	19.678318	25.682688
Tree	19.480499	25.413036
House	20.553619	26.603751

For example, some simple image encryption algorithms may have a time complexity of O(n), where n is the number of pixels in the image. This means that the time required encrypting or decrypting the image increases linearly with the size of the image.

It's important to note that the time complexity of an algorithm is only one factor to consider when evaluating its performance. Other factors such as memory usage, security, and ease of implementation also play important roles in determining the overall suitability of an image encryption algorithm for a particular application. Time complexity of the proposed algorithm is calculated as $\ominus(2MN)$.

K. SPEED PERFORMANCE

All tests are performed using MATLAB 2016b with 8 GB RAM and an Intel(R) Core(TM) i5 8265U CPU at 1.6 GHz to determine the running speed. The image was encrypted by our proposed algorithm in 1.6723 sec.

L. ATTACKS

There are many factors that cannot be avoided that will affect the image while it is being transmitted. Noise, for example, can have unfavorable effects like distortion, deterioration, and pollution in the communication system. Therefore, any method used to encrypt images must be robust enough to withstand the attacks. We employ the 256 × 256 Lena color image in our simulations to experiment with various intensities of cropping and noise attacks.

Various levels of salt and pepper noises may populate the encrypted Lena image. As demonstrated in Table 11 for various test images, noise levels of 0.005 and 0.05 have little influence on the decryption outcome, while Table 12 demonstrates PSNR during cropping attacks or data losses.

TABLE 13. NIST testing.

Tests	values	Result
Frequency	0.5421	Pass
Block Frequency	0.6543	Pass
Cumulative Sums Forward & Reverse	0.5567	Pass
Runs	0.6654	Pass
Longest Run	0.7765	Pass
Rank	0.4653	Pass
The discrete Fourier transform test	0.5416	Pass
Overlapping Template	0.4654	Pass
No Overlapping Template	0.8614	Pass
Approximate Entropy	0.2421	Pass
Linear Complexity	0.1398	Pass
Serial	0.2612	Pass

TABLE 14. Chi square test for the data used in the study.

Test U01	Proposed algorithm values in %
Rabbit	85.80
Alphabit	86.45
Block Alphabit	82.32

FIGURE 13. Noise and cropping attacks analysis.

Figure 13 displays encrypted and decrypted images of missing data and Noise, respectively. Even after a data loss attack, the decrypted image may be identified according to the testing results, this shows that the technique is exceptionally robust to noise and data loss attacks.

M. NIST TEST

The NIST (National Institute of Standards and Technology) test suite for image encryption is a standardized set of tests used to evaluate the security and performance of image encryption algorithms. The NIST test suite consists of a collection of statistical tests designed to assess the quality of randomness and the level of security provided by an encryption algorithm. The NIST test suite includes a battery of

TABLE 15. Comparative analysis with state of art.

Algorithms	Horizontal	Vertical	Diagonal	Keyspace	Entropy	PSNR	NPCR	UACI
Ours	-0.0034	-0.0065	-0.0008	2^{597}	7.9993	45.87	99.62	33.59
[4]	0.0008	0.0014	0.0012	2^{224}	7.98	29.9332	99.9	33.13
[26]	0.0071	-0.0012	-0.0015	2^{445}	7.9970	35.56	99.60	33.47
[15]	-0.00074	0.0012	-0.0092	2^{168}	7.9983	34.1119	99.62	-
[12]	0.0016	0.0010	-0.0014	-	-	42.74	-	-
[14]	-0.0138	-0.0027	-0.0026	2^{210}	7.9969	31.5328	-	-
[5]	-	-	-	2^{234}	7.9924	39.64	-	-
[21]	-0.0025	-0.0008	0.0029	2^{128}	7.9991	-	99.6094	33.4734
[43]	-0.0024	0.0052	-0.0003	2^{299}	7.9984	-	99.6084	33.4635

tests, including the Frequency Test, the Block Frequency Test, the Cumulative Sums Test, the Runs Test, the Longest Runs Test, the Rank Test, the Discrete Fourier Transform Test, the Non-overlapping Template Matching Test, the Overlapping Template Matching Test, Approximate Entropy, Linear Complexity and Serial test. These tests evaluate different aspects of an encryption algorithm, including randomness, bias, and resistance to different types of attacks.

The NIST test suite is widely used in the field of cryptography and is considered a benchmark for evaluating the security and performance of encryption algorithms, including those used for image encryption. Table 13 below shows the different NIST tests for the proposed algorithm.

N. TEST U01

The U01 test, also known as the Monobit test, is a statistical test used to evaluate the randomness of a binary sequence. It counts the number of 1's and 0's in the sequence and compares the counts to determine if they are approximately equal. In image encryption, the U01 test can be used to assess the randomness of the encrypted image by converting the pixel values to binary and running the test on the resulting binary sequence. A secure image encryption algorithm should produce an encrypted image that passes the U01 test, indicating that the pixel values are random and not biased towards either 1 or 0. Table 14 below shows the Test U01 for the proposed algorithm.

O. COMPARISON WITH OTHER ALGORITHMS

Comparison results between the proposed scheme and other algorithms are displayed in Table 15. Extensive experimental results show that the suggested technique is highly resistant to common attacks like brute-force attacks, statistical analyses, and differential attacks etc.

VII. DISCUSSION

An IoT-based Multi-Dimensional Chaotic Mapping System and SVD (Singular Value Decomposition) offer several advantages in secure and fast communication of smart city image data, such as:

- Security: The chaotic mapping system provides strong encryption for image data, making it difficult for unauthorized users to access the data. The SVD technique also enhances security by breaking down the image into multiple components, making it harder to reconstruct the original image without the correct key.
- Speed: The multi-dimensional chaotic mapping system and SVD technique can handle a large amount of image data at high speeds, which is crucial for smart city applications that require real-time processing of data.
- Efficiency: The SVD technique helps to reduce the size of the image data by extracting only the most important components, which in turn reduces the amount of data that needs to be transmitted over the network. This makes the communication process more efficient and faster.
- Robustness: The chaotic mapping system and SVD technique provide a high level of robustness against noise and other forms of interference, ensuring that the image data can be transmitted and received without errors.
- Scalability: The IoT-based architecture allows the system to be easily scaled up or down depending on the size and complexity of the smart city network. This makes it easy to add new devices and sensors to the network without disrupting the existing communication system.

Overall, the IoT-based Multi-Dimensional Chaotic Mapping System and SVD technique offer a highly secure, efficient, and scalable solution for transmitting and processing image data in smart cities.

VIII. CONCLUSION

This work offers an IoT-based multi-dimensional chaotic mapping system for secure and fast-communication in smart cities that encrypts images using hyper-chaotic sequences generated by the 5D Gauss-Sine-Logistic system. In addition to Singular Value Decomposition and DFRCT to compress and secure sensitive image information, the suggested technique additionally employs pixel position permutation. The system can be readily scaled up or down depending on the size and complexity of the smart city network, owing to

the IoT-based design. This facilitates the addition of additional devices and sensors to the network without disturbing the current communication infrastructure. There are several potential advantages of using high dimensional chaotic maps in image encryption, such as the following: for security, high-dimensional chaotic maps can provide strong security in image encryption by generating complex and unpredictable sequences of values that can be used as encryption keys or to scramble image pixels. This can make it more difficult for an attacker to decipher the original image. For processing speed considerations, high-dimensional chaotic maps can be computationally efficient and provide fast encryption and decryption of images. This can be especially useful in real-time applications where speed is critical. Also, high-dimensional chaotic maps can be more robust to attacks such as noise, data loss, and compression as they can distribute the information more uniformly throughout the image. Further, it can be easily modified to suit different image encryption requirements. This makes them more versatile and adaptable than other encryption methods. Importantly, high-dimensional chaotic maps are inherently non-linear, which can enhance the randomness and security of the encryption algorithm. This non-linearity can also help to resist attacks based on statistical analysis or linear algebraic methods. Overall, high-dimensional chaotic maps have the potential to provide a highly secure and efficient method for image encryption. However, as with any encryption method, their effectiveness depends on their specific implementation and the quality of the encryption key used. In future this algorithm can also be implemented for real time image and video encryption.

REFERENCES

[1] X. Jin, H. Zhang, X. Li, H. Yu, B. Liu, S. Xie, A. K. Singh, and Y. Li, "Confused-modulo-projection-based somewhat homomorphic encryption—Cryptosystem, library, and applications on secure smart cities," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6324–6336, Apr. 2021, doi: 10.1109/JIOT.2020.3015032.

[2] N. A. Mohamed, M. A. El-Azeim, A. Zaghloul, and A. A. Abd El-Latif, "Image encryption scheme for secure digital images based on 3D cat map and Turing machine," in *Proc. 7th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Nov. 2015, pp. 230–234, doi: 10.1109/SOCPAR.2015.7492812.

[3] W. Shao, M. Cheng, C. Luo, L. Deng, M. Zhang, S. Fu, M. Tang, and D. Liu, "An image encryption scheme based on hybrid electro-optic chaotic sources and compressive sensing," *IEEE Access*, vol. 7, pp. 156582–156591, 2019, doi: 10.1109/ACCESS.2019.2949704.

[4] H. Huang and D. Cheng, "A secure image compression-encryption algorithm using DCT and hyperchaotic system," *Multimedia Tools Appl.*, vol. 81, no. 22, pp. 31329–31347, Sep. 2022, doi: 10.1007/s11042-021-11796-x.

[5] H. Wen, Y. Huang, and Y. Lin, "High-quality color image compression-encryption using chaos and block permutation," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 8, Sep. 2023, Art. no. 101660.

[6] K. A. K. Patro, B. Acharya, and V. Nath, "Secure, lossless, and noise-resistive image encryption using chaos, hyper-chaos, and DNA sequence operation," *IETE Tech. Rev.*, vol. 37, no. 3, pp. 223–245, May 2020, doi: 10.1080/02564602.2019.1595751.

[7] H. Zhang, X.-Q. Wang, Y.-J. Sun, and X.-Y. Wang, "A novel method for lossless image compression and encryption based on LWT, SPIHT and cellular automata," *Signal Process., Image Commun.*, vol. 84, May 2020, Art. no. 115829, doi: 10.1016/j.image.2020.115829.

[8] M. Khan and A. Rasheed, "Permutation-based special linear transforms with application in quantum image encryption algorithm," *Quantum Inf. Process.*, vol. 18, no. 10, p. 298, Oct. 2019, doi: 10.1007/s11128-019-2410-7.

[9] S. Jiao, J. Feng, Y. Gao, T. Lei, and X. Yuan, "Visual cryptography in single-pixel imaging," *Opt. Exp.*, vol. 28, no. 5, p. 7301, 2020, doi: 10.1364/oe.383240.

[10] P. Zheng, J. Li, Z. Li, M. Ge, S. Zhang, G. Zheng, and H. Liu, "Compressive imaging encryption with secret sharing metasurfaces," *Adv. Opt. Mater.*, vol. 10, no. 15, Aug. 2022, doi: 10.1002/adom.202200257.

[11] B. Panna, S. Kumar, and R. K. Jha, "Image encryption based on block-wise fractional Fourier transform with wavelet transform," *IETE Tech. Rev.*, vol. 36, no. 6, pp. 600–613, Nov. 2019, doi: 10.1080/02564602.2018.1533892.

[12] X.-D. Chen, Y. Wang, J. Wang, and Q.-H. Wang, "Asymmetric color cryptosystem based on compressed sensing and equal modulus decomposition in discrete fractional random transform domain," *Opt. Lasers Eng.*, vol. 121, pp. 143–149, Oct. 2019, doi: 10.1016/j.optlaseng.2019.04.004.

[13] M. Zhang, X.-J. Tong, J. Liu, Z. Wang, J. Liu, B. Liu, and J. Ma, "Image compression and encryption scheme based on compressive sensing and Fourier transform," *IEEE Access*, vol. 8, pp. 40838–40849, 2020, doi: 10.1109/ACCESS.2020.2976798.

[14] Z. Gan, X. Chai, J. Bi, and X. Chen, "Content-adaptive image compression and encryption via optimized compressive sensing with double random phase encoding driven by chaos," *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2291–2309, Jun. 2022, doi: 10.1007/s40747-022-00644-6.

[15] X. Chai, J. Bi, Z. Gan, X. Liu, Y. Zhang, and Y. Chen, "Color image compression and encryption scheme based on compressive sensing and double random encryption strategy," *Signal Process.*, vol. 176, Nov. 2020, Art. no. 107684, doi: 10.1016/j.sigpro.2020.107684.

[16] A. Ghaffari, "Image compression-encryption method based on two-dimensional sparse recovery and chaotic system," *Sci. Rep.*, vol. 11, no. 1, p. 369, Jan. 2021, doi: 10.1038/s41598-020-79747-4.

[17] P. Chaudhary, R. Gupta, A. Singh, P. Majumder, and A. Pandey, "Joint image compression and encryption using a novel column-wise scanning and optimization algorithm," *Proc. Comput. Sci.*, vol. 167, pp. 244–253, Jan. 2020, doi: 10.1016/j.procs.2020.03.218.

[18] Y. Tang, M. Zhao, and L. Li, "Secure and efficient image compression-encryption scheme using new chaotic structure and compressive sensing," *Secur. Commun. Netw.*, vol. 2020, Dec. 2020, Art. no. 6665702, doi: 10.1155/2020/6665702.

[19] M. Mollaeefar, A. Sharif, and M. Nazari, "A novel encryption scheme for colored image based on high level chaotic maps," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 607–629, Jan. 2017, doi: 10.1007/s11042-015-3064-9.

[20] M. Majid, S. Amir, H. Moein, and N. Mahboubeh, "An improved method for image encryption based on high level chaotic maps and improved gravity model," in *Proc. Int. Congr. Technol., Commun. Knowl. (ICTCK)*, Nov. 2015, pp. 253–259, doi: 10.1109/ICTCK.2015.7582679.

[21] Y. Qobbi, A. Jarjar, M. Essaid, and A. Benazzi, "Image encryption algorithm using dynamic permutation and large chaotic S-box," *Multimedia Tools Appl.*, vol. 82, no. 12, pp. 18545–18564, May 2023, doi: 10.1007/s11042-022-14175-2.

[22] H. Çelik and N. Doğan, "A hybrid color image encryption method based on extended logistic map," *Multimedia Tools Appl.*, Jul. 2023, doi: 10.1007/s11042-023-16215-x.

[23] Z. Zhu, Y. Song, W. Zhang, H. Yu, and Y. Zhao, "A novel compressive sensing-based framework for image compression-encryption with S-box," *Multimedia Tools Appl.*, vol. 79, nos. 35–36, pp. 25497–25533, Sep. 2020, doi: 10.1007/s11042-020-09193-x.

[24] S. Zhu and C. Zhu, "A new image compression-encryption scheme based on compressive sensing and cyclic shift," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 20855–20875, Aug. 2019, doi: 10.1007/s11042-019-7405-y.

[25] G. Ye, C. Pan, X. Huang, Z. Zhao, and J. He, "A chaotic image encryption algorithm based on information entropy," *Int. J. Bifurcation Chaos*, vol. 28, no. 1, Jan. 2018, Art. no. 1850010, doi: 10.1142/S0218127418500104.

[26] H. Dong, E. Bai, X.-Q. Jiang, and Y. Wu, "Color image compression-encryption using fractional-order hyperchaotic system and DNA coding," *IEEE Access*, vol. 8, pp. 163524–163540, 2020, doi: 10.1109/ACCESS.2020.3022398.

[27] C. Yu, H. Li, and X. Wang, "SVD-based image compression, encryption, and identity authentication algorithm on cloud," *IET Image Process.*, vol. 13, no. 12, pp. 2224–2232, Oct. 2019, doi: 10.1049/iet-ipr.2018.5912.

[28] L. Zhu, H. Song, X. Zhang, M. Yan, T. Zhang, X. Wang, and J. Xu, "A robust meaningful image encryption scheme based on block compressive sensing and SVD embedding," *Signal Process.*, vol. 175, Oct. 2020, Art. no. 107629, doi: 10.1016/j.sigpro.2020.107629.
[29] L. Lidong, D. Jiang, X. Wang, L. Zhang, and X. Rong, "A dynamic triple-image encryption scheme based on chaos, S-box and image compressing," *IEEE Access*, vol. 8, pp. 210382–210399, 2020, doi: 10.1109/ACCESS.2020.3039891.
[30] B. A. Salunke and S. Salunke, "Analysis of encrypted images using discrete fractional transforms viz. DFrFT, DFrST and DFrCT," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 1425–1429, doi: 10.1109/ICCSP.2016.7754390.
[31] K. M. Hosny, S. T. Kamal, and M. M. Darwish, "A color image encryption technique using block scrambling and chaos," *Multimedia Tools Appl.*, vol. 81, no. 1, pp. 505–525, Jan. 2022, doi: 10.1007/s11042-021-11384-z.
[32] M. Es-Sabry, N. El Akkad, M. Merras, A. Saaidi, and K. Satori, "A new color image encryption algorithm using multiple chaotic maps with the intersecting planes method," *Sci. Afr.*, vol. 16, Jul. 2022, Art. no. e01217, doi: 10.1016/j.sciaf.2022.e01217.
[33] S. Li, L. Zhao, and N. Yang, "Medical image encryption based on 2D zigzag confusion and dynamic diffusion," *Secur. Commun. Netw.*, vol. 2021, May 2021, Art. no. 6624809, doi: 10.1155/2021/6624809.
[34] S. Salunke, B. Ahuja, M. F. Hashmi, V. Marriboyina, and N. D. Bokde, "5D Gauss map perspective to image encryption with transfer learning validation," *Appl. Sci.*, vol. 12, no. 11, p. 5321, May 2022, doi: 10.3390/app12115321.
[35] *The USC-SIPI Image Database*. Accessed: Jan. 11, 2022. [Online]. Available: http://sipi.usc.edu/database/database.php
[36] M.-E. Nilsback and A. Zisserman, "17 category flower dataset," Visual Geometry Group, Inf. Eng. Building (IEB), Dept. Eng. Sci., Univ. Oxford, Oxford, U.K., Tech. Rep. Accessed: Mar. 4, 2022. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/flowers/17/
[37] W. S. Sayed, A. G. Radwan, H. A. H. Fahmy, and A. Elsedeek, "Trajectory control and image encryption using affine transformation of Lorenz system," *Egyptian Informat. J.*, vol. 22, no. 2, pp. 155–166, Jul. 2021, doi: 10.1016/j.eij.2020.07.002.
[38] Y. Liu, G. Cen, B. Xu, and X. Wang, "Color image encryption based on deep learning and block embedding," *Secur. Commun. Netw.*, vol. 2022, Oct. 2022, Art. no. 6047349, doi: 10.1155/2022/6047349.
[39] P. Chakraborty and C. Tharini, "An efficient parallel block compressive sensing scheme for medical signals and image compression," *Wireless Pers. Commun.*, vol. 123, no. 4, pp. 2959–2970, Apr. 2022, doi: 10.1007/s11277-021-09270-w.
[40] B. Ahuja and R. Doriya, "Bifold-crypto-chaotic steganography for visual data security," *Int. J. Inf. Technol.*, vol. 14, no. 2, pp. 637–648, Mar. 2022, doi: 10.1007/s41870-022-00861-9.
[41] X. Song, M. Shi, Y. Zhou, and E. Wang, "An image compression encryption algorithm based on chaos and ZUC stream cipher," *Entropy*, vol. 24, no. 5, p. 742, May 2022.
[42] O. El Ogri, H. Karmouni, M. Sayyouri, and H. Qjidaa, "A new image/video encryption scheme based on fractional discrete Tchebichef transform and singular value decomposition," *Multimedia Tools Appl.*, vol. 82, no. 22, pp. 33465–33497, Sep. 2023.
[43] T. S. Ali and R. Ali, "A new chaos based color image encryption algorithm using permutation substitution and Boolean operation," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19853–19873, Jul. 2020, doi: 10.1007/s11042-020-08850-5.

BHARTI AHUJA received the master's degree from RGPV, Bhopal. She is currently a Ph.D. Research Scholar with the Department of Information Technology, National Institute of Technology Raipur. Her research interests include image encryption, bio medical image security, and applications of chaos theory in cryptography. She is also a Life Member of IE, CSI, ISTE, and IAENG.

RAJESH DORIYA received the master's and Ph.D. degrees from IIIT, Allahabad. He is currently an Assistant Professor with the Department of Information Technology, National Institute of Technology Raipur. His research interests include distributed computing, cloud computing, artificial intelligence, robotics, soft computing techniques, and network security.

SHARAD SALUNKE is currently a Ph.D. Scholar with the Department of Electronics and Communication Engineering, Amity School of Engineering and Technology, Amity University Gwalior, Gwalior, Madhya Pradesh. His research interests include digital signal processing, image processing, and bio medical image processing. He is also a member of IE, ISTE, and IAENG.

MOHAMMAD FARUKH HASHMI (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering from MIT Mandsaur/RGPV Bhopal University, in 2007, the M.E. degree in digital techniques and instrumentation from the Shri Govindram Seksaria Institute of Technology and Science (SGSITS) (Autonomous State Government), Indore/RGPV Bhopal University, in 2010, and the Ph.D. degree from the Visvesvaraya National Institute of Technology (VNIT), Nagpur, in 2015, under the supervision of Dr. Avinash G. Keskar. He was a Principal Investigator of one research project of worth five lakhs funded by Institute Seed grant through TEQIP III. He has a Teaching and Research experience of 15 years. He has supervised two Ph.D. scholars. He has supervised four Ph.D. scholars until now. He is also guiding six Ph.D. scholars. He is currently working as an Assistant Professor Grade-I in the Department of Electronics and Communication Engineering, National Institute of Technology (NIT), Warangal. He has published up to 100 articles; including 35 SCI indexed research papers in international/national journals/conferences of publishers like IEEE, *Elsevier, and Springer*. He has also published three patent to his credit. His current research interests include computer vision, machine vision, machine learning, deep learning, embedded systems, the Internet of Things, digital signal processing, image processing, and digital IC design. He is a Life Member of IETE, ISTE, IE, and IAENG societies, and Fellow Member of IETE. He is also serving as an active and potential Technical Reviewer for IEEE Access, IET Image 2195 Processing, IET Computer Vision, Wireless Personal Communication, IEEE Systems Journal, Sensors (MDPI), Electronics (MDPI), Diagnostic (MDPI), *the Visual Computer, Applied Soft Computing (Elsevier)*, Color Research and Application, *the Journal of Super Computing*, and various other journals like Elsevier/Springer/IEEE Transactions publishers of repute.

ADITYA GUPTA received the B.E. degree in electronics and telecommunication engineering from CSVTU University, in 2012, the M.Tech. degree in signal processing from the College of Engineering Pune, India, in 2014, and the Ph.D. degree from the Visvesvaraya National Institute of Technology (VNIT), Nagpur, India, in 2019, under the supervision of Dr. K. D. Kulat. He is currently a Postdoctoral Fellow with the Department of Information and Communication Technology, University of Agder, Norway.