# PermutEx: Feature-Extraction-Based Permutation – A New Diffusion Scheme for Image Encryption Algorithms

Muhammad Shahbaz Khan<sup>1</sup>, Jawad Ahmad<sup>1</sup>, Ahmed Al-Dubai<sup>1</sup>, Zakwan Jaroucheh<sup>1</sup> Nikolaos Pitropakis<sup>1</sup>, William J. Buchanan<sup>1</sup>,

<sup>1</sup>School of Computing, Engineering and the Built Environment,

Edinburgh Napier University,

Edinburgh, UK,

Emails: {muhammadshahbaz.khan, j.ahmad, a.al-dubai, z.jaroucheh, n.pitropakis, b.buchanan}@ napier.ac.uk

Abstract-Traditional permutation schemes mostly focus on random scrambling of pixels, often neglecting the intrinsic image information that could enhance diffusion in image encryption algorithms. This paper introduces PermutEx, a feature-extractionbased permutation method that utilizes inherent image features to scramble pixels effectively. Unlike random permutation schemes, PermutEx extracts the spatial frequency and local contrast features of the image and ranks each pixel based on this information, identifying which pixels are more important or information-rich based on texture and edge information. In addition, a unique permutation key is generated using the Logistic-Sine Map based on chaotic behavior. The ranked pixels are permuted in conjunction with this unique key, effectively permuting the original image into a scrambled version. Experimental results indicate that the proposed method effectively disrupts the correlation in information-rich areas within the image resulting in a correlation value of 0.000062. The effective scrambling of pixels, resulting in nearly zero correlation, makes this method suitable to be used as diffusion in image encryption algorithms.

*Index Terms*—Diffusion, permutation, feature extraction, spatial frequency, local contrast, Josephus permutation, chaos.

## I. INTRODUCTION

In digital image security, image encryption plays a vital role in ensuring confidentiality and integrity [1]-[3]. Image encryption involves two basic processes, i.e., confusion and diffusion. According to Claude Shannon [4], confusion refers to changing the values of the pixels based on a key and is usually achieved by substituting one value for another. Diffusion, on the other hand, refers to changing the position of the pixels based on a key. This is usually achieved through methods like the permutation. Traditional permutation schemes often focus on the random scrambling of pixels, neglecting intrinsic image information, which could be utilised for a more effective permutation strategy. Such schemes include chaotic row-column shuffling [5]-[7], classic Josephus permutation [8], and improved Josephus permutation [9]. Although these methods may offer a certain level of unpredictability, they often fall short of exploiting the image's inherent features, which could be important for enhancing the quality of diffusion. To achieve effective scrambling, this paper argues that the permutation algorithms should utilize intrinsic image features.

Features like spatial frequency and local contrast both relate to the variation in intensity values of an image, but they focus on different aspects and scales of variation. Spatial frequency refers to the rate at which pixel intensities change in an image [10]–[12]. It can be thought of as the fineness or coarseness of patterns in an image. High Spatial Frequency refers to rapid changes in pixel values over short distances. This is often seen in areas with detailed textures or sharp edges. For example, in an image of a zebra, the black and white stripes represent a high spatial frequency because of the rapid alternation of contrasting colors. Low Spatial Frequency, on the other hand, depicts slow changes in pixel values over larger distances. This would be seen in areas with gradual intensity changes, like a smooth gradient sky during sunset. Spatial frequency can be analyzed using Fourier transforms. Furthermore, local contrast or local dynamic range refers to the difference in intensity between a pixel and its immediate surroundings. It's a measure of how much a pixel stands out from its neighbors-[13], [14]. A high local contrast means that a pixel that has a much different intensity than its neighbors. For instance, if a single white pixel is surrounded by black pixels, this white pixel has a high local contrast because of its stark difference from its immediate environment.

This paper introduces PermutEx, a novel approach to permutation that takes into account the inherent features within the image for effective scrambling of pixels. Unlike traditional methods, our scheme examines the spatial frequency and local contrast of each pixel, ranking them based on this information. This information is used to scramble the pixels, disrupting the correlation in information-rich areas within the image effectively. Fig. 1 depicts the basic workflow of the proposed permutation scheme. The clusters in Fig. 1a represent the information in an image such as edges, shapes, etc. The darker dots represent the information-rich areas, and the lighter dots represent the information-poor areas. Fig. 1b depicts the pixels ranked on the basis of information content. Each dot is annotated with its rank where dots with higher ranks represent

Information Clusters Ranked Information Scrambled Information 54 53 52 34 27 26 49 48 46 47 59 50 57 45 20 22 70 69 67 68 51 63 64 24 25 44 71 65 72 66 
 58
 56
 55
 37
 36
 35
 82
 83
 74
 75
 62 61 60 33 23 43 78 84 80 81 15 16 17 18 19 21 76 79 77 73 94 95 91 96 97 1 2 3 4 5 85 90 98 92 93 38 39 40 41 42 99 86 87 88 89 28 29 30 31 32 (a) (b) (c)

Fig. 1: Basic workflow of information-based scrambling

higher information content and in Fig. 1c, the dots have been scrambled based on the information ranking.

The main contributions of this paper are:

- A novel feature-extraction or image-information-based permutation scheme, PermutEx is proposed that incorporates feature extraction to enhance pixel scrambling.
- The intrinsic image features, such as spatial frequency and local contrast are employed for effective pixel rearrangement. Pixels are ranked on the basis of the maximum and minimum information and are scrambled accordingly.
- 3) A unique permutation key is generated using hybrid Logistic-Sine Map having improved chaotic behaviour and large chaotic range. This key is used in conjunction with the pixel ranking criteria to scramble the pixels and break the correlation effectively.

### II. THE PROPOSED PERMUTATION SCHEME—PERMUTEX

The complete steps involved in the proposed PermutEx scheme, depicted in Fig.2, are as follows:

• Step 1. Read Plaintext Image: Read the grayscale  $256 \times 256$  baboon image as the plaintext image matrix *I* represented as (1).

$$I = \{I(x,y)\}_{(x=1,y=1)}^{256 \times 256}$$
(1)

• Step2. Calculate Spatial Frequencies: To extract the information on how the pixel intensity values in the image vary over space, the spatial frequency information has been calculated using Fast Fourier Transform (FFT). High frequencies represent the regions with more information/complexity, such as the edges, textures, and other fine-grained details in the image. The low frequencies, on the other hand, represent the basic structures and shapes in the image. In Fig.3b, the brighter spots indicate where the maximum information is, while the low frequency components depicting basic shape/structure information

are visualised in Fig. 3c. The FFT of the plaintext image I(x, y) is calculated by:

$$F(u,v) = \sum_{x=0}^{256-1} \sum_{y=0}^{256-1} I(x,y) \cdot e^{-j2\pi \left(\frac{ux}{256} + \frac{vy}{256}\right)}$$
(2)

$$F(u,v) = [F_1, F_2, F_3, \dots, F_{256}]$$
(3)

Where F(u, v) is the Fourier transform of I(x, y), 256 × 256 shows the dimension of the image, j is the imaginary unit, and  $[F_1, F_2, F_3, \ldots, F_{256}]$  are complex numbers.

The zero-frequency component is moved to the centre of the array as:

$$F_{\text{shifted}}(u,v) = F\left(u - \frac{256}{2}, v - \frac{256}{2}\right)$$
 (4)

The magnitude of the complex numbers in  $F_{\text{shifted}}$  is calculated as:

$$F_{\rm mag}(u,v) = \sqrt{{\rm Re}(F_{\rm shifted})^2 + {\rm Im}(F_{\rm shifted})^2} \qquad (5)$$

To enhance the visibility of the components, the logarithmic scaling is applied:

$$F_{\text{mag}}(u,v) = \log(F_{\text{mag}}(u,v) + 1) \tag{6}$$

The final step is to normalize the logarithmically scaled magnitude to fit within the grayscale range [0, 1]:

$$F_{\text{norm}}(u,v) = \frac{F_{\text{mag_log}}(u,v) - \min(F_{\text{mag_log}})}{\max(F_{\text{mag_log}}) - \min(F_{\text{mag_log}})}$$
(7)

• Step 3. Calculate Local Contrast: The local contrast is calculated by using a local window around each pixel and calculating the contrast within that window. The contrast is often defined as the standard deviation of the pixel values within the window and essentially



Fig. 2: PermutEx –The proposed feature-extraction-based Permutation Scheme.

captures the local variability or range of pixel values, which can be thought of as a very rudimentary form of "texture" or "detail" at that pixel location. Visually, high contrast values represent the areas where the image changes dramatically over a small distance, like the edges of objects or textures as shown in Fig. 3d and the low contrast values represent smooth, slowly varying areas, which can also be visualised in Fig. 3d.

For our  $256 \times 256$  input image *I*, where I(x, y) represents the intensity of the pixel at location (x, y), the local contrast is calculated by choosing a local window *W* of size  $m \times m$  centered at location (x, y). We selected a  $3 \times 3$  window and it is represented as:

$$W(x,y) = \{I(i,j) \mid i \in [x - (m-1)/2, x + (m-1)/2], j \in [y - (m-1)/2, y + (m-1)/2]\}$$
(8)

Then calculate the mean  $\mu$  of the pixel intensities within W:

$$\mu = \frac{1}{m^2} \sum_{(i,j) \in W} I(i,j)$$
(9)

Then calculate the standard deviation  $\sigma$  of the pixel intensities within W, which serves as the local contrast C(x, y) expressed as (10) and visually depicted as shown in Fig. 3d.

$$C(x,y) = \sigma = \sqrt{\frac{1}{m^2} \sum_{(i,j) \in W} (I(i,j) - \mu)^2}$$
(10)

• Step 4. Calculate Importance Probabilities: Information probabilities are calculated by taking the average of corresponding values from the Spatial Frequency array F(u, v) having frequency values for each pixel as  $F_{x,y}$ and the Local Contrast Array C(x, y) having contrast values for each pixel as  $C_{x,y}$ . This new array captures a blended form of both spatial frequency and local contrast information. The probability P of a pixel at location (x, y) is calculated as:

$$P_{x,y} = \frac{F_{x,y} + C_{x,y}}{2}$$
(11)

This metric provides a measure of "importance" or "information richness" at each pixel location. Pixels with higher values in the probabilities array would mean that both their spatial frequency and local contrast are high, making them significant in terms of edges,



Fig. 3: Visual representation of extracted features: (a) original image, (b) spatial frequency, (c) frequency components with shape/structure info, and (d) local contrast.

texture, and overall complexity. Whereas, the pixels with lower values are less significant in terms of both spatial frequency and local contrast. These are probably smoother areas with less detail.

• Step 5. Normalize Importance Probabilities: Normalizing the "probabilities" array to a range of 0 to 1 is done using the Min-Max normalization method. First, calculate the minimum minProb and maximum maxProb values of the "probabilities" array.

$$\min \operatorname{Prob} = \min_{x,y} (P_{x,y}) \tag{12}$$

$$\max \operatorname{Prob} = \max_{x,y}(P_{x,y}) \tag{13}$$

The probabilities are then normalized to the range [0, 1] using the formula:

$$P_{x,y}^{\text{norm}} = \frac{P_{x,y} - \min \text{Prob}}{\max \text{Prob} - \min \text{Prob}}$$
(14)

- Step 6. Rank Pixels by Importance: In this step, the grayscale pixels are ranked by their importance based on their normalized probabilities. The sorting is done in descending order, meaning the most important (highest probability) pixels come first. The pixels ranked on the basis of importance are visualized in Fig. 4(b), where yellow edges represent high values of probabilities depicting the areas having a lot of information according to the probability metric (a combination of spatial frequency and local contrast). These are the areas of the image with higher texture or edge information. On the other hand, the blue background represents the areas having less information according to the probability metric. These are likely to be smoother, less textured areas of the image. The ranking is done as follows:
  - 1) The normalized probability matrix  $P^{\text{norm}}(256 \times 256)$  is flattened into a vector p' of length  $256 \times 256$ .

$$p' = \text{Flatten}(P^{\text{norm}}) \tag{15}$$

2) The vector p' is then sorted in descending order as follows:

$$p_{\text{sorted}} = \text{Sort}(p', \text{'descend'})$$
 (16)

3) The original indices corresponding to the sorted vector  $p_{\text{sorted}}$  are stored in  $R^{\text{imp}}$ . These indices indicate the positions of the pixels in the sorted vector p'.

$$R^{\rm imp} =$$
Indices of  $p_{\rm sorted}$  in  $p'$  (17)

The  $R^{\text{imp}}$  array contains the indices of the pixels in the sorted normalized probability matrix, sorted by their importance (highest probability first). We use these indices to prioritize or select pixels for further



(a)



Fig. 4: Visualization of ranked pixels based on importance: (a) Information acquisition from spatial frequencies and local contrast, (b) Visualization of significant and less important pixels.

processing, based on their calculated importance.

- Step 7. Generate the Permutation Key: A permutation key K is generated using the Logistic Sine Map. This is a chaotic map used for generating a sequence of values that appear highly random. The highly chaotic permutation key is depicted in Fig. 5(c) and is generated using the following steps:
  - 1) The number of iterations is set twice the number of pixels in the input image *I*.

$$Iterations = 2 \times numel(I)$$
(18)

2) The hybrid logistic sine map is defined by:

$$X_{N+1} = (L(r, X_N) + S((4-r), X_N)) \mod 1$$
(19)

Where  $L(r, X_N)$  is the logistic map function and  $S((4 - r), X_N)$  is the Sine Map function. Specifically, these are defined as:

The Logistic Map:  $L(r, X_N) = r \times X_N \times (1 - X_N)$ . The Sine Map:  $S((4-r), X_N) = \frac{(4-r) \times \sin(\pi \times X_N)}{4}$ .

By combining these two, the hybrid map becomes:

$$X_{N+1} = \left(r \times X_N \times (1 - X_N) + \frac{(4 - r) \times \sin(\pi \times X_N)}{4}\right) \mod 1 \quad (20)$$

3) This generates a chaotic sequence S. To avoid the initial transients that may not represent the chaotic behavior well, only the latter half of the sequence is taken and the new chaotic sequence S' is obtained.

$$S' = S[\text{end} - \text{numel}(I) + 1 : \text{end}]$$
(21)

4) The values are scaled and rounded to integers to create a finite digital precision format sequence D.

$$D = \operatorname{round}(S' \times 10^3) \tag{22}$$

5) The integer values from D are sorted and their original indices are stored in permuteSequence. Then, the indices of this sorted array are again sorted to create the unique permutation key K.

> (23)permuteSequence = SortIndices(D)

$$K =$$
SortIndices(permuteSequence) (24)

This generates a unique permutation key K which is used to map and permute the pixels in the next step.

- Step 8. Map and Permute to Get Scrambled Image: In this step, we map the ranked list of original indices stored in  $R^{imp}$  to the generated permutation key K. We then use this mapped list to rearrange the pixels in the original image, effectively scrambling it. From the previous steps, we have two sequences:
  - $R^{imp} = [r_1, r_2, ..., r_{256 \times 256}]$  This is the sequence of original pixel indices sorted by their ranked importance.
  - $K = [k_1, k_2, \dots, k_{256 \times 256}]$  This is the unique Permutation Key.

To create the final permuted indices to get the permuted image, we rearrange  $R^{imp}$  according to the key K. The equation for this operation is:

$$permutedIndices[z] = R^{imp}[r_z]$$
(25)

For  $z = 1, 2, \dots, 256 \times 256$ .

Here, permutedIndices[z] is the  $z^{th}$  element in the final permuted sequence, and  $R^{imp}[r_z]$  is the  $r_z^{th}$  element in the ranked sequence.

This is how the ranked pixels are scrambled according to the unique permutation key, resulting in a permuted or scrambled image.

## **III. RESULTS AND ANALYSIS**

The proposed permutation scheme uses the intrinsic information of the plaintext image to scramble the images in an effective manner. The proposed scheme has been evaluated for image permutation analysis and correlation analysis as follows.

#### A. Permutation Analysis

The ranking-based permutation ensures that the most "important" or "information-rich" pixels are treated differently, making it more secure. Fig. 5 presents the permutation results of the proposed scheme depicting the the pixels ranked on the basis of extracted information, the unique chaotic permutation key, and the final permuted image. Moreover, the comparison of the proposed permutation scheme with traditional permutation schemes is given in Fig.6. It also displays the comparison of PermutEx with and without applying the feature-based pixel ranking.

### B. Correlation Analysis

Extensive correlation analysis of the proposed scheme has been carried out. Results of the correlation analysis with comparison are given in Table I showing an ideal value of 0.000062 for the PermutEx-permuted images. The correlation between adjacent pixels is displayed in Fig. 7. The correlation plots are uniformly distributed, which is an excellent indicator of a good permutation scheme as it shows that the pixel dependencies have been well diffused.





important feature-information

(a) The Original Image



(c) The Permutation Key (d) The Final Permuted Image

Fig. 5: Permutation Analysis of PermutEx-the proposed diffusion scheme.





TABLE I: Correlation ana	ysis of the PermutEx	permuted image and	comparison with	other techniques
--------------------------	----------------------	--------------------	-----------------	------------------

Image	Horizontal Coeff	Vertical Coeff	Diagonal Coeff	GLCM Correlation	Correlation (Orig,
_			-		Permuted) – Corr2
Original Baboon Image	0.8824	0.8397	0.7990	0.7915	1
Random row-column	0.4493	0.4533	0.2009	0.0412	0.003721
shuffling					
Chaotic row-column shuf-	0.4405	0.4578	0.1970	0.0321	0.028485
fling					
PermutEx without feature-	0.3265	0.3617	0.1102	0.0223	0.037976
based pixel ranking					
PermutEx with feature-	0.0021	-0.0029	0.0022	0.0011	0.000062
based pixel ranking					



(a) Random row-column shuffling (Correlation=0.0412)



(c) PermutEx without featurebased pixel ranking (Correlation=0.0223)





(d) PermutEx with featurebased pixel ranking (Correlation=0.0011)

Fig. 6: Comparison of PermutEx with traditional permutation schemes and PermutEX with and without applying featureextraction-based pixel ranking.



Fig. 7: Correlation analysis of the PermutEx-permuted image

## **IV. CONCLUSION**

This paper introduced PermutEx, a new technique to scramble pixels effectively. Unlike traditional methods, PermutEx uses the intrinsic image-information-based features, i.e., spatial frequency and local contrast, to scramble the pixels of the plaintext image. This feature-driven permutation leads to effective disruption in correlation of the information-rich areas within the image. This feature-based ranking makes PermutEx more secure and effective in term of diffusing the pixels. PermutEx achieved a very low correlation value of 0.000062 making PermutEx extremely efficient, effective, and secure method of image permutation and is recommended to be used in image encryption algorithms as a diffusion technique.

#### REFERENCES

- [1] F. Ahmed, M. U. Rehman, J. Ahmad, M. S. Khan, W. Boulila, G. Srivastava, J. C.-W. Lin, and W. J. Buchanan, "A dna based colour image encryption scheme using a convolutional autoencoder," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 19, 11 2022
- [2] A. K. Sahu, K. Umachandran, V. D. Biradar, O. Comfort, V. Sri Vigna Hema, F. Odimegwu, and S. M. A, "A study on content tampering in multimedia watermarking," SN Computer Science, vol. 4, 02 2023.
- SRSS: A New Chaos-Based Single-Round Single S-Box Image Encryp-[3] tion Scheme for Highly Auto-Correlated Data, ICEET 2023. IEEE, 08 2023.
- [4] C. E. Shannon, "Communication theory of secrecy systems," Bell System Technical Journal, vol. 28, pp. 656-715, 10 1949.
- S. Sun, Y. Guo, and R. Wu, "A novel image encryption scheme based on [5] 7d hyperchaotic system and row-column simultaneous swapping," IEEE Access, vol. 7, pp. 28539-28547, 2019.
- [6] M. U. Rehman, A. Shafique, Y. Y. Ghadi, W. Boulila, S. U. Jan, T. R. Gadekallu, M. Driss, and J. Ahmad, "A novel chaos-based privacy-preserving deep learning model for cancer diagnosis," IEEE Transactions on Network Science and Engineering, vol. 9, no. 6, pp. 4322-4337. 2022.
- [7] S. F. Abbasi, J. Ahmad, J. S. Khan, M. A. Khan, and S. A. Sheikh, "Visual meaningful encryption scheme using intertwinning logistic map," in Intelligent Computing. Springer International Publishing, 2019, pp. 764-773
- [8] Z. Hua, B. Xu, F. Jin, and H. Huang, "Image encryption using josephus problem and filtering diffusion," IEEE Access, vol. 7, pp. 8660-8674, 2019
- [9] H. Gao, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption," Chaos, Solitons I& Fractals, vol. 29, pp. 393-399, 07 2006
- [10] A. De Cesarei, S. Mastria, and M. Codispoti, "Early spatial frequency processing of natural images: An erp study," PLoS ONE, vol. 8, p. e65103. 05 2013.
- [11] D. Marr, Vision : a computational investigation into the human representation and processing of visual information. Mit Press, 1982.
- [12] R. L. De Valois and K. K. De Valois, "Spatial vision," Annual Review of Psychology, vol. 31, pp. 309-341, 01 1980.

(b) Chaotic row-column shuffling (Correlation=0.0321)

- [13] Z. Zhou, N. Sang, and X. Hu, "Global brightness and local contrast adaptive enhancement for low illumination color image," *Optik*, vol. 125, pp. 1795–1799, 03 2014.
  [14] E. Peli, "Contrast in complex images," *Journal of the Optical Society of America A*, vol. 7, p. 2032, 10 1990.