

Magic cube puzzle approach for image encryption

Achmad Fanany Onnilita Gaffar ^{a,1}, Rheo Malani ^{a,2}, Arief Bramanto Wicaksono Putra ^{a,3,*}



^a Department of Information Technology, Politeknik Negeri Samarinda, East Kalimantan, Indonesia

¹ onnygaffar212@gmail.com; ² anaogje@gmail.com; ³ ariefbram@gmail.com

* corresponding author

ARTICLE INFO

Article history

Received October 30, 2019

Revised October 29, 2020

Accepted November 3, 2020

Available online November 30, 2020

Keywords

Magic cube puzzle

An 8-bit grayscale image

Image encryption

Transposition orientation

Random permutation

ABSTRACT

In principle, the image encryption algorithm produces an encrypted image. The encrypted image is composed of arbitrary patterns that do not provide any clues about the plain image and its cipher key. Ideally, the encrypted image is entirely independent of its plain image. Many functions can be used to achieve this goal. Based on the functions used, image encryption techniques are categorized into: (1) Block-based; (2) Chaotic-based; (3) Transformation-based; (4) Conventional-based; and (5) Miscellaneous based. This study proposes a magic cube puzzle approach to encrypt an 8-bit grayscale image. This approach transforms a plain image into a particular size magic cube puzzle, which consists of a set of blocks. The magic cube puzzle algorithm will diffuse the pixels of the plain image as in a Rubik's Cube game, by rotating each block in a particular direction called the transposition orientation. The block's transposition orientation is used as the key seed, while the generation of the cipher key uses a random permutation of the key seed with a certain key length. Several performance metrics have been used to assess the goals, and the results have been compared to several standard encryption methods. This study showed that the proposed method was better than the other methods, except for entropy metrics. For further studies, modification of the method will be carried out in such a way as to be able to increase its entropy value to very close to 8 and its application to true color images. In essence, the magic cube puzzle approach has a large space for pixel diffusion that is possibly supposed to get bigger as a series of data has transformed into several magic cubes. Then, each magic cube has transposed with a different technique. This proposed approach is expected to add to a wealth of knowledge in the field of data encryption.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

Cryptography is the science of data encryption. Encryption is the process of encoding data (messages, information, etc.) in such a way that the data becomes messy and difficult to understand its contents. Encrypted data can only be recovered through the decryption process by using a particular key (commonly called a secret key). Data encryption aims to prevent unauthorized parties from accessing the contents of the data. Cryptography is used to achieve several objectives, such as the following: (1). Authentication (ensuring the user's identity is the right party); (2). Confidentiality (encrypted data can only be recovered by those who have a passkey); (3). Data integrity (only authorized parties can access the consistency and correctness of original data); (4). Non-repudiation (preventing part or one of the parties denying the process of sending data); (5). Access control (only groups with correct authentication are eligible to enter the data sent) [1]. Original data is usually called plain data, while encrypted data is usually called cipher data. The encryption process requires a particular algorithm where the algorithm in question must also serve the decryption process as well — similarly, the generation of cipher keys. The

assessment of the encryption goodness depends on how messy (incomprehensible) the encrypted data is. The critical points are in the encryption algorithm and the generation of the cipher key used.

In the case of the cipher key used, there are two types of encryption: (1). Symmetric encryption (sender and receiver have the same cipher key); (2). Asymmetric encryption (each party has a private key pair consisting of a secret key and a public key, where the secret key is used to encrypt the data sent, while the public key is used to decrypt the data received by the other party) [2]. Some encryption algorithms that classified as symmetric encryption are Data Encryption Standard (DES) [3]–[5], Advanced Encryption Standard (AES) [6]–[10], Blowfish [11]–[14], Rivest Cipher 6 (RC6) [15]–[17], RC4 [18]–[20]. Some encryption algorithms that classified as asymmetric encryption are *Rivest–Shamir–Adleman* (RSA) [21], [22], Diffie Hellman [23], Elliptic Curve Cryptographic (ECC) [24], Escrowed Encryption Standard (EES) [25]. Hybrid types of encryption are sometimes also used to combine the advantages of the two types of encryption in question [26][27].

There are two modes of the encryption process: (1) block cipher; and (2) stream cipher. Block ciphers consist of encryption and decryption algorithms, where the decryption algorithm is the inverse of the encryption algorithm. In order to make the block cipher secure, a cipher key is made as random as possible using random permutations of a certain length. In the case of key lengths, block cipher security also depends on determining block size. Stream ciphers are symmetric ciphers where each digit of the plaintext is encrypted one by one with the corresponding keystream digits. A keystream is a pseudo-random stream of alphabetical digits combined with plain/ciphertext [2] [28].

In principle, various kinds of encryption methods are classified into two, namely: (1). Random based; (2). Constant based. Random based encryption algorithm consists of deterministic and heuristic methods. Deterministic methods produce ciphertext uniquely based on its plain text, whereas the heuristic method generates ciphertext by using random values. Included types are machine learning-based encryption methods. A constant-based encryption algorithm emphasizes numerical constant values (especially the hash method and symmetric encryption). These values must be reasonable and not contain back doors [2][29]. Some techniques usually used in deterministic methods are: (1). Transposition; (2). Substitution; (3). They are combined both. The transposition technique is an encryption technique that changes the plaintext's character arrangement without changing the original character [30]–[32]. And vice versa for substitution techniques [24][30][32]–[35].

Image encryption is encoding images into a difficult format to understand using several cryptographic algorithms and a cipher key visually. In contrast, decryption of images is to decode encrypted images into their original format. Not all traditional cryptographic algorithms can be used to encrypt the image because the image has adjacent pixels that are highly correlated [36]. Two essential properties that must be possessed by a suitable image cryptosystem are diffusion and confusion. Diffusion is used to increase the occurrence of zero correlation between plain image and cipher image such that the image cipher cannot be recognized. Confusion is used to create a clueless image cipher. It means that the cipher key is related to plain image through non-simple ways [37]. Many functions can be used to achieve this, such as permutation, scrambling, substitution, chaotic functions, etc. Based on the functions used, image encryption techniques are categorized into: (1). Block-based [38], (2). Chaotic-based [15][39]; (3). transformation-based [40]; (4). Conventional-based; (5). miscellaneous-based [41].

Many image encryption algorithms have developed with the aim of not only protecting all visual information but also enough from attempts at cryptanalyst attacks. In this case, the image encryption algorithm should have the following characteristics: (1). Cipher image must not depend directly on a plain image; (2). Key-streams are built from the concept of the relationship between plain image and cipher image, which is very difficult to guess; (3). The encryption algorithm operates on a long cipher keyspace to prevent brute force attacks; (4). The correlation between a plain image and cipher image is close to zero; (5). Cipher images must not contain any information about the plain image and its passkey; (6). Cipher images must contain high disturbances and are visually unpredictable. In summary, an encrypted image (cipher image) consists of arbitrary patterns that do not provide any clues about the

plain image and its cipher key. The encrypted image is entirely independent of its plain image. Some performance metrics commonly used to assess an image encryption algorithm’s merit are: (1). The correlation coefficient between plain and encrypted images; (2). Bit length metric of the cipher key; (3). Histogram deviation; (4). Information entropy; (5). NPCR (Number of Pixels Changing Rate) and UACI (Unified Averaged Changed Intensity); (6). PSNR (Peak signal-to-noise ratio) between plain and encrypted images [1][36][37][41].

This study proposes a magic cube puzzle approach to encrypt images. This approach transforms a plain image into a particular size magic cube, consisting of a set of blocks. The magic cube puzzle algorithm will diffuse a plain image pixel like in a Rubik’s Cube game by rotating each block in a particular direction. The number of rotations in a particular direction is used as the cipher key.

2. Method

2.1. Composing the magic cube

A magic cube consists of a set of blocks of the same size to form a cube. A magic cube of $S \times S \times S$ -size requires some S -sized $S \times S$ blocks. To build a magic cube of a specific size requires as much data as a block of the magic cube. In general, the proposed method is shown in Fig. 1.

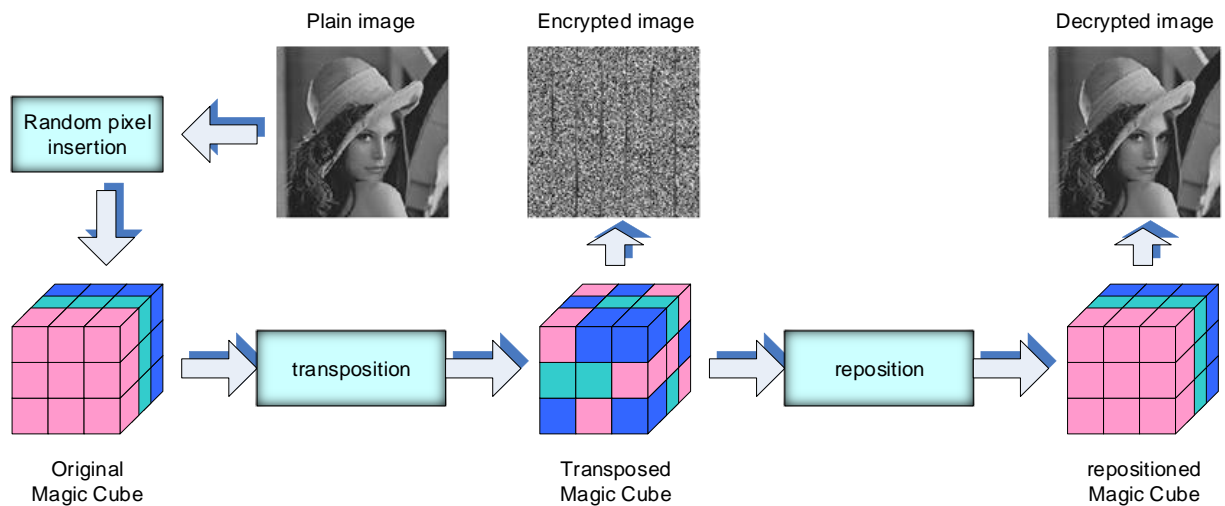


Fig. 1. The proposed method (a magic cube approach)

The transposition stage is considered as the encryption stage, while the reposition stage is the decryption stage.

Suppose an 8-bit grayscale image has a row-column size $M \times N$. Each block is set to $S \times S$ size. The number of magic cubes needed is $ncube = floor((M \times N)/S^3) + 1$. All pixels are placed in each block. The remaining empty blocks of $M \times N - ncube$ filled with random numbers within the range 0..255. Suppose an 8-bit grayscale image stated by:

$$I_{plain} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{bmatrix} \tag{1}$$

Pixel composition in stream form expressed by: $I_{plain(stream)} = \{a_{11}, \dots, a_{M1}, \dots, a_{1N}, \dots, a_{MN}, \dots a_L\}$ where $L = S \times S \times S$. A number $ncube$ of the $S \times S \times S$ -sized magic cube composed from pixel streams in sequence ($I_{plain(stream)}$) after random pixels are inserted. The illustration is shown in Fig. 2. The algorithm for composing the gray image onto a magic cube is shown in Fig. 3.

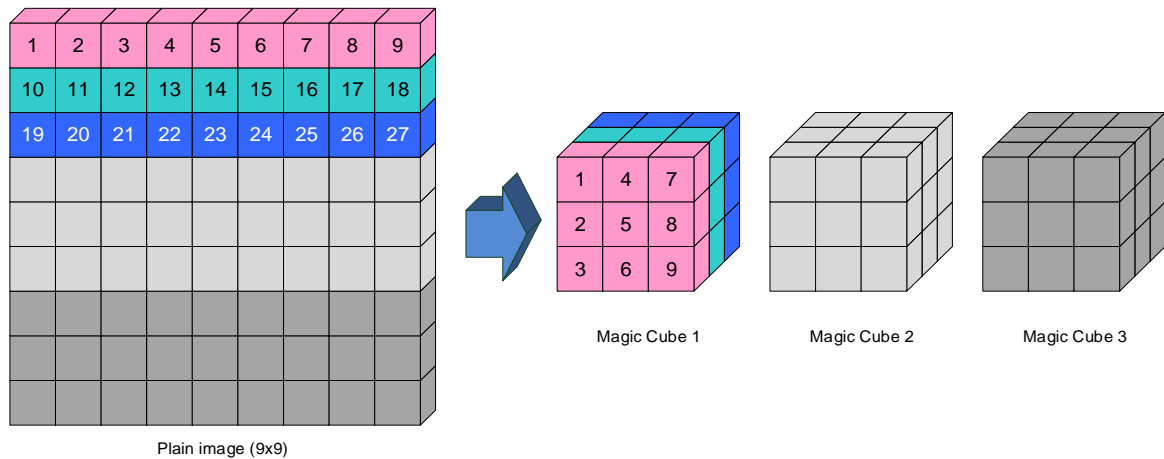


Fig. 2. Illustration of composing the magic cube

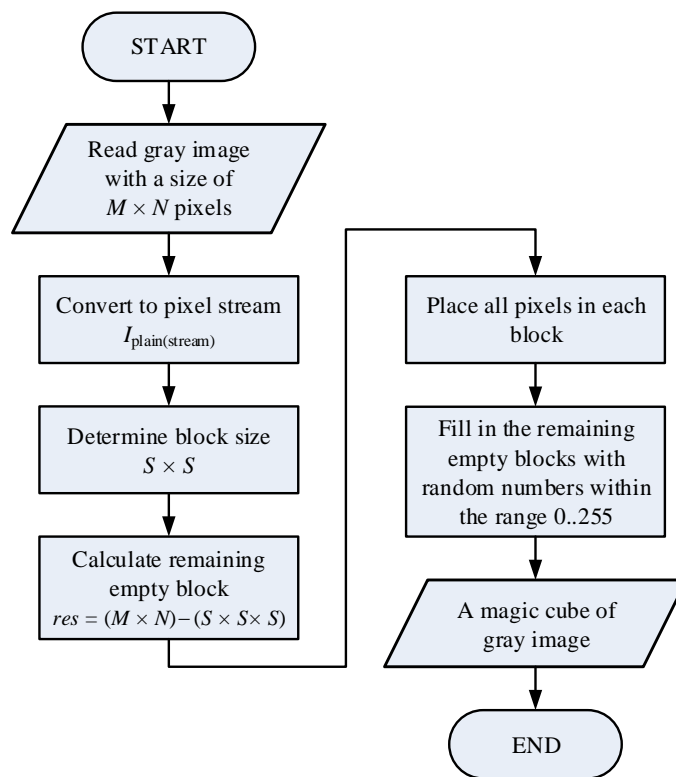


Fig. 3. The algorithm of composing the gray image onto a magic cube

2.2. Transposition the magic cube

This study uses a rotation angle of transposition $\pm 90^\circ$. In principle, the magic cube has three side views (YX, ZY, ZX) as shown in Fig. 4. Each side display has three transposition orientations where each orientation can be positive and negative, so there will be a 3×2 number of transposition orientations. If all transposition orientations are arranged sequentially ((Y → X), (X → Y), (Z → Y), (Y → Z), (Z → X), (X → Z)), then this arrangement is considered as the key seed (1, 2, 3, 4, 5, 6). Each side view has S blocks, so there will be several orientation transpositions $n = 2 \times 3 \times S$. Hence, key seed can be stated by $seed(i, j)$ where i is the index number of transposition orientation ($i = 1 \dots S$) and j is the index number of block ($j = 1 \dots 6$). Generation of the key seed using random permutation. It also indicates that the key length should be more than 6. If the number of k key lengths of each block is specified, the number of permutations of k key lengths of each block expressed by [42]:

$$N_{keys/block} = N_{seed} \times 6! + \frac{6!}{(6 - (k - N_{seed} \times 6))!} \quad N_{seed} = \text{floor}(k/6) \quad (2)$$

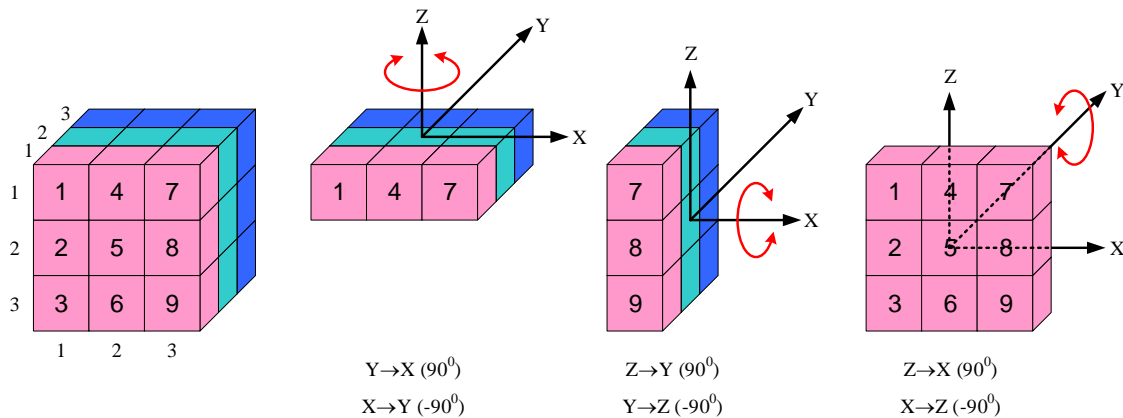


Fig. 4. The side views of the magic cube

Illustration of key structures per block is shown in Fig. 5. Next, the cipher key stated by $K(1 \dots S, [1 \dots k])$. For example, the use of key $K(1, [2,3,6,4])$ shown in Fig. 6.

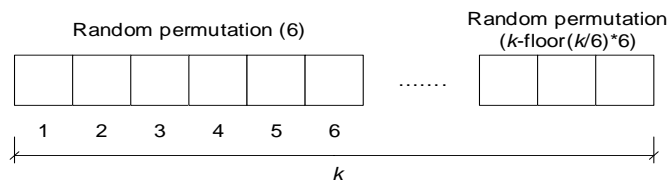


Fig. 5. The illustration of key structures per block

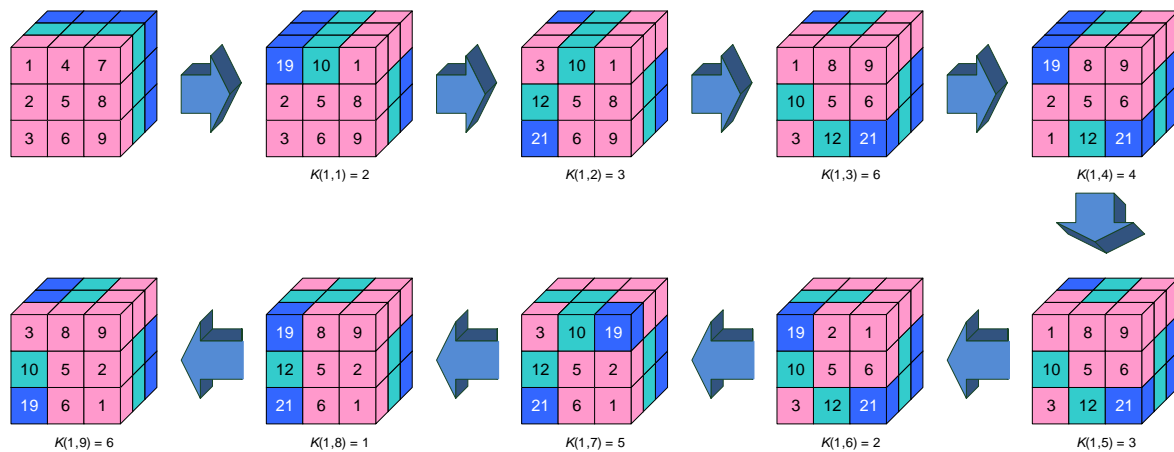


Fig. 6. The illustration of the use of key $K(1, [2,3,6,4,3,2,5,1])$

Because the total number of key length is $k \times S$, then keyspace is stated by $2^{k \times S}$. The cipher key with length k will contain the order of transposition orientation according to the magic cube block number. To achieve good image encryption, it must meet $\text{cipher key space} > 2^{100}$ [41]. Hence, it must meet $2^{k \times S} > 2^{100}$. To fulfill this requirement, determining the cipher key lengths of each block and block size is very important. The magic cube transposition algorithm is shown in Fig. 7.

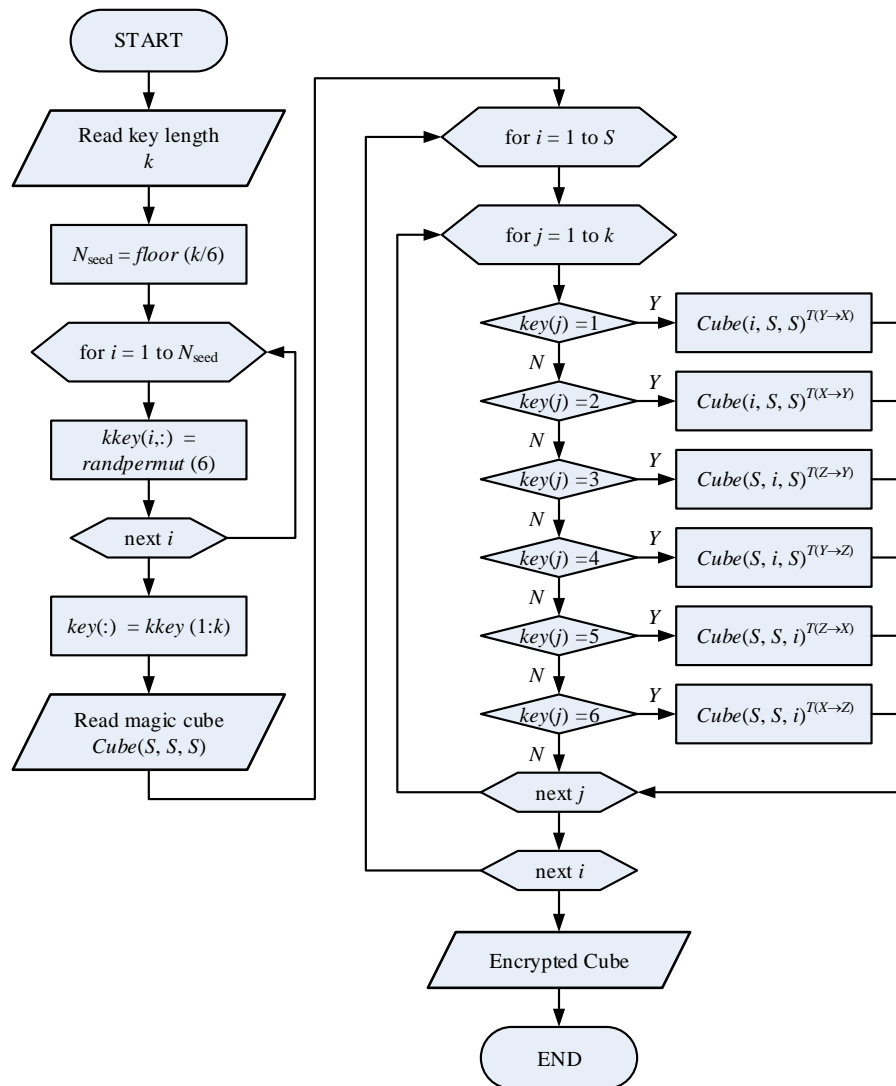


Fig. 7. The magic cube transposition algorithm

2.3. Building an encrypted image

A number $ncube$ of the $S \times S \times S$ -sized transposed magic cube then transformed into a stream of pixels. The encrypted image is built from $M \times N$ pixels of its pixel streams. If the pixel streams of the all transposed magic cubes stated by $I_{cipher}(stream) = \{b_1, \dots, b_L\}$, the encrypted image stated by:

$$I_{cipher} = \begin{bmatrix} b_1 & b_{M+1} & \dots & b_{L-M+1} \\ b_2 & b_{M+2} & \dots & b_{L-M+2} \\ \dots & \dots & \dots & \dots \\ b_M & b_{2*M} & \dots & b_L \end{bmatrix} \tag{3}$$

2.4. Image decryption

The encrypted image was transformed into pixel streams. A number $ncube$ of the $S \times S \times S$ -sized transposed magic cube composed from pixel streams in sequence $(I_{cipher}(stream))$. The reposition stage is carried out for each transposed magic cube using a cipher key descending in order. The all repositioned magic cubes transformed into pixel streams in sequence $(I_{plain}(stream))$. Finally, the decrypted image is built from $M \times N$ pixels of its pixel streams.

2.5. Performance measurement

Evaluation of an image encryption algorithm's performance is related to several characteristics that need to be considered. Various performance metrics commonly used are to assess the independence of an encrypted image against its plain image. Encrypted images do not provide any clues about the plain image and its cipher key. This section outlines some of the performance metrics used in this study.

2.5.1. Keyspace metric

An encrypted image's essential feature is the key's sensitivity and the initial parameters used to generate it. A keyspace is the limit of all possible cipher keys of a certain length generated randomly. Large keyspace can reduce brute force attacks. A keyspace is mathematically expressed by [41][43]:

$$keyspace = 2^{key-length} \quad (4)$$

where key-length is the total number of bits in the key, commonly, cipher key longer than 100 bits prevents brute force attacks.

2.5.2. Correlation metric

A good encryption technique must produce an encrypted image with a correlation close to zero between pixels adjacent to each other. Commonly, a correlation test was performed on each pair of horizontal, vertical, and diagonal pixels between an encrypted image and a plain image [41][43]. Suppose there is a series of X and Y , where $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_n\}$, n is the number of data. Correlation between them stated by:

$$R_{XY} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2}} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (5)$$

2.5.3. Differential metric

Sensitivity to pixel changes between an encrypted image and its plain image is measured using NPCR (Number of Pixels Changing Rate) and UACI (Unified Averaging Changed Intensity) [37][43]. Suppose I_{plain} is a plain image and I_{cipher} is an encrypted image with $M \times N$ size. NPCR stated by:

$$D(i, j) = \begin{cases} 1, & \text{if } I_{cipher}(i, j) \neq I_{plain}(i, j) \\ 0, & \text{if } I_{cipher}(i, j) = I_{plain}(i, j) \end{cases} \quad NPCR = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N D(i, j) \times 100\% \quad (6)$$

UACI stated by:

$$UACI = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \left(\frac{|I_{cipher}(i, j) - I_{plain}(i, j)|}{255} \right) \times 100\% \quad (7)$$

2.5.4. Information Entropy metric

Entropy is a measure of the uncertainty of a random variable. A gray image can be interpreted as a sample emitted from a source with an intensity of $\{0 \dots 256\}$. We can model these intensity values using a gray image histogram and produce an estimate of source entropy [43]–[45]. The entropy of a gray image X is expressed by:

$$H(X) = H(P_1, \dots, P_n) = - \sum_{i=1}^n P_i \cdot \log_2(P_i) \quad P_i = P_r(X = x_i) \quad (8)$$

where P_i is the probability of the intensity value of the i index, and $H(X)$ is the entropy value of the gray image X . An entirely random 8-bit grayscale image has an entropy value of 8. This value shows the maximum diffusion of evenly spread pixels with a probability of pixel intensity of $1/256$.

2.5.5. PSNR (Peak Signal to Noise Ratio) metric

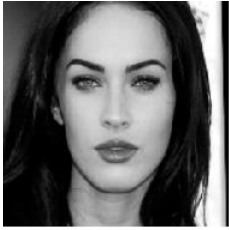
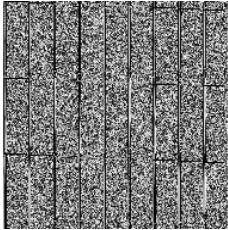

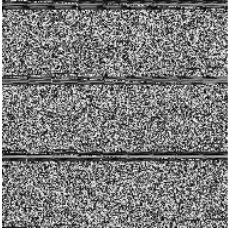

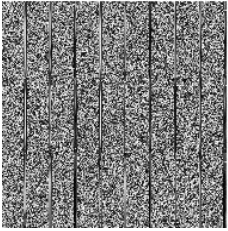

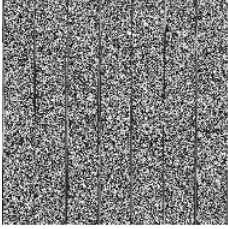
An encrypted image can be considered as a plain image exposed to noise. The amount of noise that is measured using PSNR expressed by:

$$PSNR = 10 \cdot \log_{10} \left(\frac{(X_{max})^2}{MSE} \right) \quad MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (I_{cipher}(i,j) - I_{plain}(i,j))^2 \quad (9)$$

3. Results and Discussion

This study was used a face image dataset obtained from <https://www.kaggle.com/datasets>, and the selected face image data converted into a grayscale image. All 8-bit grayscale image samples are 225x225 in size. The results are shown in Table 1.

Table 1. Encryption performance metrics for all 8-bit grayscale image samples

Plain image sample	Encrypted image	Correlation metric	Differential metric (%)	Information entropy metric	PSNR
		Horizontal: 0.00015 Vertical: 0.00009 Diagonal: 0.0003	NPCR : 97.33 UACI : 36.54	7.9568	6.8693
		Horizontal: 0.00074 Vertical: 0.00023 Diagonal: 0.00071	NPCR : 99.54 UACI : 31.78	7.9833	6.5249
		Horizontal: 0.0021 Vertical: 0.00055 Diagonal: 0.00066	NPCR : 99.62 UACI : 33.97	7.9873	7.5386
		Horizontal: 0.00006 Vertical: 0.00008 Diagonal: 0.00008	NPCR : 99.59 UACI : 28.47	7.9837	8.2210

The image of *lena.jpg* was used to compare the results against some standard image encryption algorithms. This study has used a $75 \times 75 \times 75$ magic cube. It means that there were 421875-pixel slots. An 8-bit grayscale image with 225×225 size has a total pixel of 50625. To complete the magic cube slot requires $421875 - 50625 = 371250$ pixels generated randomly. The cipher key length for

each block that has used was 11, with the number of blocks of the magic cube was 75. Hence, the cipher key length was $11 \times 75 = 825$, and the keyspace of 2^{825} . The study results were compared with several standard encryption methods (*Vigenere*, *RC4*, *DES*, *3DES*, *AES*) with the results obtained from [37]. A comparison of image encryption results between the proposed method with some standard encryption algorithms has shown in Fig. 8. The results of performance comparisons successively have shown in Table 2 to Table 6.

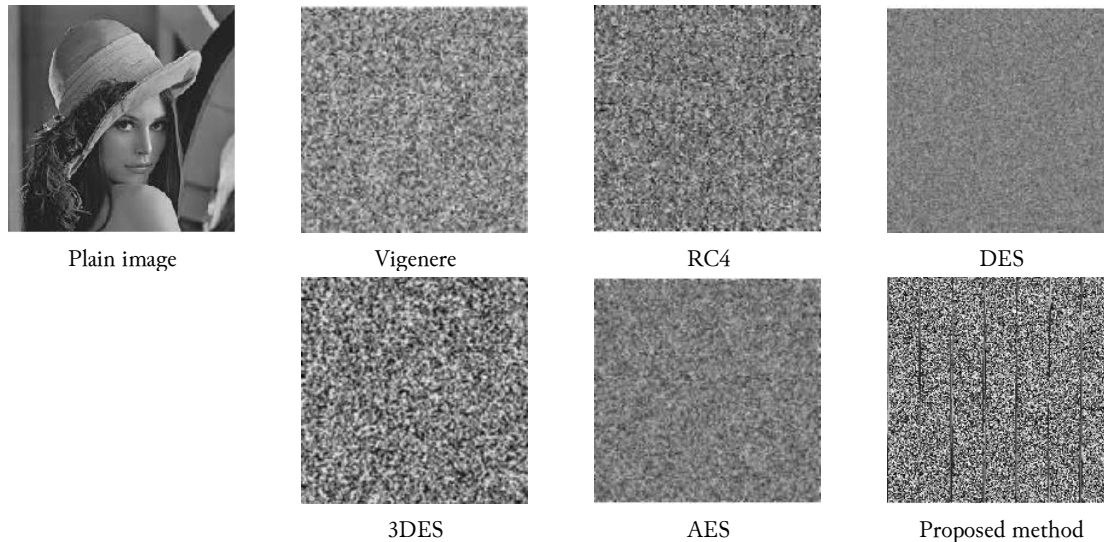


Fig. 8. A comparison of the image encryption results

To achieve good image encryption, it must meet *cipher key space* $> 2^{100}$. Table 2 shows the comparison of the key length and keyspace of the proposed method and other methods. The proposed method has a much larger key length and keyspace than the other methods (key length = 825, keyspace = 2825). This means that the proposed method has the best resistance to brute force attacks compared to other methods. Table 3 has shown that the proposed method has the least correlation of all adjacent pixels (close to zero between pixels adjacent to each other on each pair of horizontal, vertical, and diagonal pixels). It showed the highest level of independence compared to other methods. Table 4 has shown that the proposed method has the highest NPCR and UACI values (NPCR = 99.59%, UACI = 28.47%). It showed that the proposed method is more sensitive to pixel changes than other methods. An entirely random 8-bit grayscale image has an entropy value of 8. This value shows the maximum diffusion of evenly spread pixels with a probability of pixel intensity of 1/256.

Table 5 has shown that the RC4 method has the highest entropy value. It showed that the RC4 method had produced an encrypted image that is more random than other methods. The proposed method was the lowest. This is because the proposed method focuses more on the principle of transposition in 3D space. That is, changes occur only in the pixel position in the image without any substitution. The random addition of pixels is only to fulfill the needs of the magic cube pixel slot. From the gray image sample used, 371250 random pixels were added (about 88% of the total pixel sample image). The plain image's entropy value was 7.2547, while the encrypted image has an entropy value of 7.9837. Around 88% of corresponding pixels turned out to only increase the value of entropy by about 10%. PSNR shows how noisy an encrypted image is compared to its plain image. Good encryption should have a reasonably low PSNR. Table 6 has shown that the proposed method has a lower PSNR compared to other methods.

Table 2. Keyspace metric analysis

Method	Key length	Keyspace	Method	Key length	Key space
Vigenere	128	2^{128}	3DES	168	2^{168}
RC4	256	2^{256}	AES	128	2^{128}
DES	56	2^{56}	Proposed method	825	2^{825}

Table 3. Correlation metric analysis

Method	Correlation between pixels of encrypted and plain image		
	Horizontal	Vertical	Diagonal
Vigenere	0.08920	-0.09070	-0.07580
RC4	-0.00200	-0.00370	0.00390
DES	0.01590	0.05580	0.00690
3DES	0.03320	0.04550	-0.00490
AES	-0.00450	0.00390	-0.00420
Proposed method	0.00006	0.00008	0.00008

Table 4. Differential metric analysis

Method	NPCR (%)	UACI (%)	Method	NPCR (%)	UACI (%)
Vigenere	0.0015	0.00006	3DES	0.0217	0.0063
RC4	99.5172	23.5834	AES	0.0354	0.0137
DES	0.0216	0.0040	Proposed method	99.5900	28.4700

Table 5. Information entropy metric analysis

Method	Entropy	Method	Entropy
Vigenere	7.7943	3DES	7.9966
RC4	7.9968	AES	7.9965
DES	7.9959	Proposed method	7.9837

Table 6. PSNR metric analysis

Method	Entropy	Method	Entropy
Vigenere	9.7637	3DES	8.4548
RC4	8.3781	AES	8.4021
DES	8.4698	Proposed method	8.2210

4. Conclusion

This study has proposed a magic cube approach method to encrypt an 8-bit grayscale image. This method applies the concept of transposition in 3D space from all pixels of the image placed in a magic cube. Changes only occur in the pixel position when it has returned to the image format without substitution. Applying the proposed method to some selected 8-bit grayscale image samples has shown different performance metrics. The performance metrics comparison (keyspace, correlation, differential, information entropy, and PSNR metrics) against several standard encryption methods have demonstrated that the proposed method was better than the other methods, except for entropy metric (lower than other methods). It was due to the proposed method of not substituting pixel values but transposing their positions. The increase in entropy value that is small enough from the plain image to the encrypted image was due to random pixels' presence to fulfill the pixel slots of the magic cube used. For further studies, modification of the method will be carried out in such a way as to be able to increase its entropy value to very close to 8 and its application to true color images.

Acknowledgment

The authors wish to acknowledge the financial support from the Research and Community Service Unit (*Unit Penelitian & Pengabdian Masyarakat*) Politeknik Negeri Samarinda with contract no 1211/PL7/LK/2019 - DIPA Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jendral Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi dan Pendidikan Tinggi - with contract no

151/SP2H/LT/DRPM/2019. Authors would like to express their heartfelt thanks to The Modern Computing Research Center, Department of Information Technology, Politeknik Negeri Samarinda.

Declarations

Author contribution. The first author contributes as the principal researcher who has the idea to propose the method and compiles the research method. The second author contributes as a supporting researcher in charge of testing algorithms to be applied in encryption. The third author contributes as a correspondence author who is in charge of testing the research's performance.

Funding statement. The financial is supported by DIPA Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jendral Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi dan Pendidikan Tinggi (RistekDikti) of The Republic of the Indonesia - with contract no 151/SP2H/LT/DRPM/2019, <http://simlitabmas.ristekdikti.go.id/>. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] O. G. Abood and S. K. Guirguis, "A Survey on Cryptography Algorithms," *Int. J. Sci. Res. Publ.*, vol. 8, no. 7, Jul. 2018, doi: [10.29322/IJSRP.8.7.2018.p7978](https://doi.org/10.29322/IJSRP.8.7.2018.p7978).
- [2] B. Megyesi, "Proceedings of the 1st International Conference on Historical Cryptology: HistoCrypt 2018," in *1st International Conference on Historical Cryptology: HistoCrypt 2018, Uppsala, June 18-20, 2018.*, 2018, Available at: [Google Scholar](#)
- [3] S. N. Habib, R. Awan, and W. Haider, "A Modified Simplified Data Encryption Standard Algorithm," *Int. J. Comput. Sci. Softw. Eng.*, vol. 6, no. 7, p. 152, 2017, Available at: [Google Scholar](#)
- [4] Z. Mihret and M. W. Ahmad, "The Reverse Engineering of Reverse Encryption Algorithm and a Systematic Comparison to DES," *Procedia Comput. Sci.*, vol. 85, pp. 558-570, 2016, doi: [10.1016/j.procs.2016.05.221](https://doi.org/10.1016/j.procs.2016.05.221).
- [5] Ratnadewi, R. P. Adhie, Y. Hutama, A. Saleh Ahmar, and M. I. Setiawan, "Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC)," *J. Phys. Conf. Ser.*, vol. 954, p. 012009, Jan. 2018, doi: [10.1088/1742-6596/954/1/012009](https://doi.org/10.1088/1742-6596/954/1/012009).
- [6] B.K.S.Rajaram and K. P. N, "Secure MQTT using AES for Smart Homes in IoT Network," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, 2019.
- [7] S. Fahd, M. Afzal, H. Abbas, W. Iqbal, and S. Waheed, "Correlation power analysis of modes of encryption in AES and its countermeasures," *Futur. Gener. Comput. Syst.*, vol. 83, pp. 496-509, Jun. 2018, doi: [10.1016/j.future.2017.06.004](https://doi.org/10.1016/j.future.2017.06.004).
- [8] T. B. I. Guy-Cedric and S. R., "A Comparative Study on AES 128 BIT AND AES 256 BIT," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 6, no. 4, pp. 30-33, Aug. 2018, doi: [10.26438/ijsrcse/v6i4.3033](https://doi.org/10.26438/ijsrcse/v6i4.3033).
- [9] S. D. Unni and N. M. John, "A Secure MSSS Scheme and AES Encryption over Cloud Data," *Int. J. Comput. Appl. Technol. Res.*, vol. 07, no. 04, pp. 171-174, Apr. 2018, doi: [10.7753/IJCATR0704.1003](https://doi.org/10.7753/IJCATR0704.1003).
- [10] P. Wang, Y. Zhang, and J. Yang, "Research and Design of AES Security Processor Model Based on FPGA," *Procedia Comput. Sci.*, vol. 131, pp. 249-254, 2018, doi: [10.1016/j.procs.2018.04.210](https://doi.org/10.1016/j.procs.2018.04.210).
- [11] S. Ghosh and V. Karar, "Blowfish Hybridized Weighted Attribute-Based Encryption for Secure and Efficient Data Collaboration in Cloud Computing," *Appl. Sci.*, vol. 8, no. 7, p. 1119, Jul. 2018, doi: [10.3390/app8071119](https://doi.org/10.3390/app8071119).
- [12] P. Patel, R. Patel, and N. Patel, "Integrated ECC and Blowfish for Smartphone Security," *Procedia Comput. Sci.*, vol. 78, pp. 210-216, 2016, doi: [10.1016/j.procs.2016.02.035](https://doi.org/10.1016/j.procs.2016.02.035).
- [13] A. R. L. Reyes, E. D. Festijo, and R. P. Medina, "Blowfish-128: a modified blowfish algorithm that supports 128-bit block size," in *8th International Workshop on Computer Science and Engineering, Bangkok, Thailand, 2018*, pp. 578-584, Available at: [Google Scholar](#)

- [14] M. Suresh and M. Neema, "Hardware Implementation of Blowfish Algorithm for the Secure Data Transmission in Internet of Things," *Procedia Technol.*, vol. 25, pp. 248–255, 2016, doi: [10.1016/j.protcy.2016.08.104](https://doi.org/10.1016/j.protcy.2016.08.104).
- [15] M. Baz, "Digital Image Encryption using Logistic Chaotic Key-based RC6," *Int. J. Comput. Appl.*, vol. 182, no. 2, pp. 17–23, Jul. 2018, doi: [10.5120/ijca2018917453](https://doi.org/10.5120/ijca2018917453).
- [16] P. M. B. H. A. S. A. P. U. Emmoh, A. A. Dauda, "Smart Grid Security Solution Model Using RC6 Cryptographic Algorithm," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 6, 2017.
- [17] H. Handschuh, "RC6," in *Encyclopedia of Cryptography and Security*, Springer US, pp. 516–516, doi: [10.1007/0-387-23483-7_346](https://doi.org/10.1007/0-387-23483-7_346)
- [18] C. Fontaine, "RC4," in *Encyclopedia of Cryptography and Security*, Springer US, pp. 515–515, doi: [10.1007/0-387-23483-7_344](https://doi.org/10.1007/0-387-23483-7_344)
- [19] P. Jindal and B. Singh, "RC4 Encryption-A Literature Survey," *Procedia Comput. Sci.*, vol. 46, pp. 697–705, 2015, doi: [10.1016/j.procs.2015.02.129](https://doi.org/10.1016/j.procs.2015.02.129).
- [20] A. Khalid, G. Paul, and A. Chattopadhyay, "RC4-AccSuite: A Hardware Acceleration Suite for RC4-Like Stream Ciphers," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 3, pp. 1072–1084, Mar. 2017, doi: [10.1109/TVLSI.2016.2606554](https://doi.org/10.1109/TVLSI.2016.2606554).
- [21] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An optical image compression and encryption scheme based on compressive sensing and RSA algorithm," *Opt. Lasers Eng.*, vol. 121, pp. 169–180, Oct. 2019, doi: [10.1016/j.optlaseng.2019.03.006](https://doi.org/10.1016/j.optlaseng.2019.03.006).
- [22] J. H. Seo, "Efficient digital signatures from RSA without random oracles," *Inf. Sci. (Njy)*, vol. 512, pp. 471–480, Feb. 2020, doi: [10.1016/j.ins.2019.09.084](https://doi.org/10.1016/j.ins.2019.09.084).
- [23] D. Adrian *et al.*, "Imperfect forward secrecy," *Commun. ACM*, vol. 62, no. 1, pp. 106–114, Dec. 2018, doi: [10.1145/3292035](https://doi.org/10.1145/3292035).
- [24] X. Hu, X. Zheng, S. Zhang, W. Li, S. Cai, and X. Xiong, "A High-Performance Elliptic Curve Cryptographic Processor of SM2 over GF(p)," *Electronics*, vol. 8, no. 4, p. 431, Apr. 2019, doi: [10.3390/electronics8040431](https://doi.org/10.3390/electronics8040431).
- [25] E. Agrawal and P. R. Pal, "A Secure and Fast Approach for Encryption and Decryption of Message Communication," *Int. J. Eng. Sci.*, vol. 11481, 2017, Available at: [Google Scholar](https://scholar.google.com/)
- [26] L. Ma and W. Jin, "Symmetric and asymmetric hybrid cryptosystem based on compressive sensing and computer generated holography," *Opt. Commun.*, vol. 407, pp. 51–56, Jan. 2018, doi: [10.1016/j.optcom.2017.08.047](https://doi.org/10.1016/j.optcom.2017.08.047).
- [27] D. Rachmawati, A. Sharif, Jaysilen, and M. A. Budiman, "Hybrid Cryptosystem Using Tiny Encryption Algorithm and LUC Algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 300, p. 012042, Jan. 2018, doi: [10.1088/1757-899X/300/1/012042](https://doi.org/10.1088/1757-899X/300/1/012042).
- [28] J.-P. Aumasson, *Serious cryptography: a practical introduction to modern encryption*. No Starch Press, 2017, Available at: [Google Scholar](https://scholar.google.com/)
- [29] Supriadi, A. Wajiansyah, H. Purwadi, R. Malani, A. Yunianta, and A. Pratomo, "Secured Data Transmission using Metadata Logger Manipulation Approach," in *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, 2018, pp. 340–344, doi: [10.1109/EIConCIT.2018.8878601](https://doi.org/10.1109/EIConCIT.2018.8878601).
- [30] O. A. Dawood, O. I. Hammadi, and F. M. Mohammed, "Secure Symmetric Block Cipher Design for Encrypting the Bitcoin Wallets in Cryptocurrencies Applications," *J. Comput. Sci.*, vol. 15, no. 5, pp. 758–768, May 2019, doi: [10.3844/jcssp.2019.758.768](https://doi.org/10.3844/jcssp.2019.758.768).
- [31] A. Jawahir and H. Haviluddin, "An audio encryption using transposition method," *Int. J. Adv. Intell. Informatics*, vol. 1, no. 2, p. 98, Jul. 2015, doi: [10.26555/ijain.v1i2.24](https://doi.org/10.26555/ijain.v1i2.24).
- [32] and H. K. E.-J. Lee, B. Omo-Ekpadi, "A Two-Phase Symmetric Key Block Cipher," *J. Comput. Sci. Appl. Inf. Technol.*, vol. 2019.
- [33] T. M. Aung, H. H. Naing, and N. N. Hla, "A Complex Transformation of Monoalphabetic Cipher to Polyalphabetic Cipher: (Vigenère-Affine Cipher)," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 3, pp. 296–303,

- Jun. 2019, doi: [10.18178/ijmlc.2019.9.3.801](https://doi.org/10.18178/ijmlc.2019.9.3.801).
- [34] P. Li and Y. Zhao, "A Simple Encryption Algorithm for Quantum Color Image," *Int. J. Theor. Phys.*, vol. 56, no. 6, pp. 1961–1982, Jun. 2017, doi: [10.1007/s10773-017-3341-7](https://doi.org/10.1007/s10773-017-3341-7).
- [35] A. Madaan, M. Bhatia, and M. Hooda, "Implementation of Image Compression and Cryptography on Fractal Images," 2018, pp. 49–61, Available at: [Google Scholar](#)
- [36] J. Shah and J. Dhobi, "REVIEW OF IMAGE ENCRYPTION AND DECRYPTION TECHNIQUES FOR 2D IMAGES," *Int. J. Eng. Technol. Manag. Res.*, vol. 5, no. 1, pp. 81–84, Feb. 2020, doi: [10.29121/ijetmr.v5.i1.2018.49](https://doi.org/10.29121/ijetmr.v5.i1.2018.49).
- [37] O. F. Mohammad, M. S. M. Rahim, S. R. M. Zeebaree, and F. Y. H. Ahmed, "A survey and analysis of the image encryption methods," *Int. J. Appl. Eng. Res.*, vol. 12, no. 23, pp. 13265–13280, 2017, Available at: [Google Scholar](#)
- [38] S. Imaizumi, T. Ogasawara, and H. Kiya, "Block-Permutation-Based Encryption Scheme with Enhanced Color Scrambling," 2017, pp. 562–573, doi: [10.1007/978-3-319-59126-1_47](https://doi.org/10.1007/978-3-319-59126-1_47)
- [39] M. Jiang and G. Sun, "A Chaotic Searchable Image Encryption Scheme Integrating with Block Truncation Coding," 2018, pp. 349–358, doi: [10.1007/978-3-030-00012-7_32](https://doi.org/10.1007/978-3-030-00012-7_32)
- [40] S. M. Pan, R. H. Wen, Z. H. Zhou, and N. R. Zhou, "Optical multi-image encryption scheme based on discrete cosine transform and nonlinear fractional Mellin transform," *Multimed. Tools Appl.*, vol. 76, no. 2, pp. 2933–2953, Jan. 2017, doi: [10.1007/s11042-015-3209-x](https://doi.org/10.1007/s11042-015-3209-x).
- [41] S. Geetha, P. Punithavathi, A. M. Infanteena, and S. S. S. Sindhu, "A Literature Review on Image Encryption Techniques," *Int. J. Inf. Secur. Priv.*, vol. 12, no. 3, pp. 42–83, Jul. 2018, doi: [10.4018/IJISP.2018070104](https://doi.org/10.4018/IJISP.2018070104).
- [42] S. Janson, "Patterns in random permutations avoiding the pattern 321," *Random Struct. Algorithms*, vol. 55, no. 2, pp. 249–270, Sep. 2019, doi: [10.1002/rsa.20806](https://doi.org/10.1002/rsa.20806).
- [43] X. Hu, L. Wei, W. Chen, Q. Chen, and Y. Guo, "Color Image Encryption Algorithm Based on Dynamic Chaos and Matrix Convolution," *IEEE Access*, vol. 8, pp. 12452–12466, 2020, doi: [10.1109/ACCESS.2020.2965740](https://doi.org/10.1109/ACCESS.2020.2965740).
- [44] Y. Wu, Y. Zhou, G. Saveriades, S. Aгаian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Inf. Sci. (Nijl.)*, vol. 222, pp. 323–342, Feb. 2013, doi: [10.1016/j.ins.2012.07.049](https://doi.org/10.1016/j.ins.2012.07.049).
- [45] C. L. Chowdhary, P. V. Patel, K. J. Kathrotia, M. Attique, K. Perumal, and M. F. Ijaz, "Analytical Study of Hybrid Techniques for Image Encryption and Decryption," *Sensors*, vol. 20, no. 18, p. 5162, Sep. 2020, doi: [10.3390/s20185162](https://doi.org/10.3390/s20185162).