Magic cube puzzle approach for image encryption

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

1 Department of Information Technology, Politeknik Negeri Samarinda, East Kalimantan, Indonesia
2 onnygaffar212@gmail.com; 3 anaogie@gmail.com; 4 ariefbram@gmail.com

* corresponding author

ARTICLE INFO

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 is 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 possible 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 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)](image)

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 $ncube = \lfloor (M \times N) / S^3 \rfloor + 1$. 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:

$$I_{\text{plain}} = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1N} \\
a_{21} & a_{22} & \ldots & a_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
a_{M1} & a_{M2} & \ldots & a_{MN}
\end{bmatrix}$$

(1)

Pixel composition in stream form expressed by: $I_{\text{plain(stream)}} = \{a_{11}, \ldots, a_{M1}, \ldots, a_{1N}, \ldots, a_{MN}, \ldots a_{L}\}$ 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 ($I_{\text{plain(stream)}}$) 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.
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 ($XY,ZY,ZX$) 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 $3 \times 2$ number of transposition orientations. If all transposition orientations are arranged sequentially ($Y \rightarrow X, X \rightarrow Y, Z \rightarrow Y, Y \rightarrow Z, Z \rightarrow X, X \rightarrow Z$), 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 ($i = 1 \ldots S$) and $j$ is the index number of block ($j = 1 \ldots 6$). 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 $[42]$. 

![Fig. 2. Illustration of composing the magic cube](image)

![Fig. 3. The algorithm of composing the gray image onto a magic cube](image)
\[ N_{\text{keys/block}} = N_{\text{seed}} \times 6! + \frac{6!}{(6 - (k - N_{\text{seed}} \times 6))!} \]
\[ N_{\text{seed}} = \text{floor}(k/6) \]

(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 \( K(1 \ldots S, [1 \ldots k]) \). 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 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.
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 $I_{cipher(\text{stream})} = \{b_1, \ldots, b_L\}$, the encrypted image stated by:

$$I_{cipher} = \begin{bmatrix}
    b_1 & b_{M+1} & \cdots & b_{L-M+1} \\
    b_2 & b_{M+2} & \cdots & b_{L-M+2} \\
    \vdots & \vdots & \ddots & \vdots \\
    b_M & b_{2M} & \cdots & b_L 
\end{bmatrix}$$

(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 $I_{cipher(\text{stream})}$. 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 $I_{plain(\text{stream})}$. Finally, the decrypted image is built from $M \times N$ pixels of its pixel streams.

Gaffur et al. (Magic cube puzzle approach for image encryption)
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]:

\[ \text{keyspace} = 2^{\text{key-length}} \]  

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 \( X = \{x_1, \ldots, x_n\} \), \( Y = \{y_1, \ldots, y_n\} \), \( n \) is the number of data. Correlation between them stated by:

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

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 \( I_{\text{plain}} \) is a plain image and \( I_{\text{cipher}} \) is an encrypted image with \( M \times N \) size. NPCR stated by:

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

UACI stated by:

\[ \text{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left( \frac{I_{\text{cipher}}(i,j) - I_{\text{plain}}(i,j)}{255} \right) \times 100\% \]  

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 \( \{0 \ldots 256\} \). 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:

\[ H(X) = H(P_1, \ldots, P_n) = -\sum_{i=1}^{n} P_i \log_2(P_i) \quad P_i = P_i(X = x_i) \]
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:

$$PSNR = 10 \cdot \log_{10} \left( \frac{(X_{max})^2}{MSE} \right)$$

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_{cipher}(i,j) - I_{plain}(i,j))^2$$ (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

<table>
<thead>
<tr>
<th>Plain image sample</th>
<th>Encrypted image</th>
<th>Correlation metric</th>
<th>Differential metric (%)</th>
<th>Information entropy metric</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="lena.jpg" alt="Image" /></td>
<td><img src="lena.jpg" alt="Image" /></td>
<td>Horizontal: 0.00015</td>
<td>NPCR : 97.33</td>
<td>7.9568</td>
<td>6.8693</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical: 0.00009</td>
<td>UACI : 36.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diagonal: 0.0003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="lena.jpg" alt="Image" /></td>
<td><img src="lena.jpg" alt="Image" /></td>
<td>Horizontal: 0.00074</td>
<td>NPCR : 99.54</td>
<td>7.9833</td>
<td>6.5249</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical: 0.00023</td>
<td>UACI : 31.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diagonal: 0.00071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="lena.jpg" alt="Image" /></td>
<td><img src="lena.jpg" alt="Image" /></td>
<td>Horizontal: 0.0021</td>
<td>NPCR : 99.62</td>
<td>7.9873</td>
<td>7.5386</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical: 0.00055</td>
<td>UACI : 33.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diagonal: 0.00066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="lena.jpg" alt="Image" /></td>
<td><img src="lena.jpg" alt="Image" /></td>
<td>Horizontal: 0.0006</td>
<td>NPCR : 99.59</td>
<td>7.9837</td>
<td>8.2210</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical: 0.00008</td>
<td>UACI : 28.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diagonal: 0.00008</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The image of *lena.jpg* was used to compare the results against some standard image encryption algorithms. This study has used a 75 x 75 x 75 magic cube. It means that there were 421875-pixel slots. An 8-bit grayscale image with 225 x 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.

![Plain image](image1)

![Vigenere](image2)

![RC4](image3)

![DES](image4)

![3DES](image5)

![AES](image6)

![Proposed method](image7)

**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 = $2^{825}$). 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>
<thead>
<tr>
<th>Method</th>
<th>Key length</th>
<th>Keyspace</th>
<th>Method</th>
<th>Key length</th>
<th>Keyspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vigenere</td>
<td>128</td>
<td>$2^{128}$</td>
<td>3DES</td>
<td>168</td>
<td>$2^{168}$</td>
</tr>
<tr>
<td>RC4</td>
<td>256</td>
<td>$2^{256}$</td>
<td>AES</td>
<td>128</td>
<td>$2^{128}$</td>
</tr>
<tr>
<td>DES</td>
<td>56</td>
<td>$2^{56}$</td>
<td>Proposed method</td>
<td>825</td>
<td>$2^{825}$</td>
</tr>
</tbody>
</table>

**Table 2.** Keyspace metric analysis

Gaffar et al. (Magic cube puzzle approach for image encryption)
Table 3. Correlation metric analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>Horizontal</th>
<th>Vertical</th>
<th>Diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vigene</td>
<td>0.08920</td>
<td>-0.09070</td>
<td>-0.07580</td>
</tr>
<tr>
<td>RC4</td>
<td>-0.00200</td>
<td>-0.00370</td>
<td>0.00390</td>
</tr>
<tr>
<td>DES</td>
<td>0.01590</td>
<td>0.05580</td>
<td>0.00690</td>
</tr>
<tr>
<td>3DES</td>
<td>0.03320</td>
<td>0.04550</td>
<td>-0.00490</td>
</tr>
<tr>
<td>AES</td>
<td>-0.00450</td>
<td>0.00390</td>
<td>-0.00420</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.00006</td>
<td>0.00008</td>
<td>0.00008</td>
</tr>
</tbody>
</table>

Table 4. Differential metric analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>NPCR (%)</th>
<th>UACI (%)</th>
<th>Method</th>
<th>NPCR (%)</th>
<th>UACI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vigene</td>
<td>0.0015</td>
<td>0.00006</td>
<td>3DES</td>
<td>0.0217</td>
<td>0.0063</td>
</tr>
<tr>
<td>RC4</td>
<td>99.5172</td>
<td>23.5834</td>
<td>AES</td>
<td>0.0354</td>
<td>0.0137</td>
</tr>
<tr>
<td>DES</td>
<td>0.0216</td>
<td>0.0040</td>
<td>Proposed method</td>
<td>99.5900</td>
<td>28.4700</td>
</tr>
</tbody>
</table>

Table 5. Information entropy metric analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>Entropy</th>
<th>Method</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vigene</td>
<td>7.7943</td>
<td>3DES</td>
<td>7.9966</td>
</tr>
<tr>
<td>RC4</td>
<td>7.9968</td>
<td>AES</td>
<td>7.9965</td>
</tr>
<tr>
<td>DES</td>
<td>7.9959</td>
<td>Proposed method</td>
<td>7.9837</td>
</tr>
</tbody>
</table>

Table 6. PSNR metric analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>Entropy</th>
<th>Method</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vigene</td>
<td>9.7637</td>
<td>3DES</td>
<td>8.4548</td>
</tr>
<tr>
<td>RC4</td>
<td>8.3781</td>
<td>AES</td>
<td>8.4021</td>
</tr>
<tr>
<td>DES</td>
<td>8.4698</td>
<td>Proposed method</td>
<td>8.2210</td>
</tr>
</tbody>
</table>

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
Conflict of interest

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References


International Journal of Advances in Intelligent Informatics Vol. 6, No. 3, November 2020, pp. 290–302


