Image Encryption Technology Based on Fractal Image Compression Algorithm

Jinna Yu

School of Computer Engineering, Shangqiu Polytechnic, Shangqiu, 476100, China
E-mail: Jinna_Yu2023@outlook.com

Received 28 June 2023; Accepted 23 September 2023;
Publication 13 February 2024

Abstract

As the most commonly used information transmission method, digital images often store a large amount of personal information. To prevent information leakage, encrypting images is essential. Common image encryption techniques suffer from certain limitations, such as overly simple encryption methods and long encryption times. In response to the above issues, this study proposes the Frobenius canonical form image encryption scheme. It calculates the fractal code through the fractal compression algorithm and to encrypt the image, it adjusts the brightness coefficient in the fractal code. To address unsatisfactory correlation coefficients in encrypted images, the Frobenius canonical form image encryption is improved by introducing the Arnold transformation encryption, which combines the two methods to reduce correlation coefficients. Finally, the knight tour algorithm is put forward. In response to the long image scrambling time in the knight tour algorithm, the Tetragonal theorem is combined with the scheme to encrypt the image. It is then re-encrypted using the Frobenius canonical form. The experimental findings illustrate that when using Frobenius canonical form, Arnold

Journal of Cyber Security and Mobility, Vol. 13_2, 283–304.
doi: 10.13052/jcsm2245-1439.1325

transformation combined with Frobenius canonical form, and the tetragonal algorithm combined with knight tour algorithm to encrypt Lena images, the three decryption methods correspond to image similarity of over 70%, over 80%, and over 90%, respectively. Combining the tetragonal algorithm and the knight tour algorithm can significantly increase the security of image encryption.

Keywords: Knight's tour, fractal image compression, Arnold transformation, Frobenius canonical form, fractal.

1 Introduction

Digital images have played an irreplaceable role in various industries for a long time in the rapidly developing information age due to their intuitive, simple, and convenient transmission methods. However, the information in images often contains privacy, and some even involve company and national security secrets [1]. Therefore, the security and confidentiality of images are particularly important, and the research on image encryption (IE) develops as a hot research direction in this field. The complexity of IE and the quality of the decrypted image are the main indicators for measuring an encryption method [2].

This study focuses on the research of IE technology, especially image scrambling technology. By combining Arnold transformation with Frobenius canonical form, image segmentation and scrambling are performed, followed by encryption and decoding. Then, improvements are made to the tetragonal and knight tour (KT) algorithms, and an improved IE model is proposed. This encryption method aims to provide higher security for encrypted images, ensuring user privacy.

Considering the above, the primary novel contributions of this work are:

- Take advantage of the low matrix requirement and wide applicability of the Frobenius canonical form.
- The introduction of the square algorithm and the KT algorithm improves the model, resulting in better IE performance.

This paper is structured as follows. Section 2 is a brief introduction to other scholars' research related to IE. Then, Section 3 reviews the main methods used in this study. Next, the fourth part is a summary of all the above studies and an outlook for future studies. Finally, the last Section 5 concludes the performed studies and outlines prospects for future research.

2 Related Work

With the advancement of the Internet, information theft methods have emerged one after another. As the most commonly used approach of information transmission, images often contain a large amount of user privacy information. The introduction of IE technology has, to some extent, protected the personal privacy of users.

Peng et al. [3] designed an IE system based on a chaotic hardware encryption system, which used a multi-scroll chaos system and Arnold transformation as the most entropy source. The image was processed through a chaotic sequence, and the Arnold transform was used for scrambling. The laboratory findings showed that the encryption system had low power consumption, fast operation speed, and good encryption performance.

Then, Boussif et al. [4] proposed a new IE method to secure medical digital imaging and communication images. This method first transformed the image into a pixel matrix, then encrypted the image blocks individually, and then modified the key through Arnold transformation. The research outcomes illustrated that this method could succesfully encrypt keys and had a faster computational time compared to traditional encryption algorithms.

Next, Wang et al. [5] made a chaotic IE algorithm with a matrix semi-tensor product and composite key. This method divided the image into four parts, performed Arnold conversion on the pixels in each part, and then combined the four parts to generate an encrypted image. The research outcomes denoted that this algorithm had better security than other encryption algorithms and was suitable for encrypting color images.

Jain et al. [6] addressed the issue of digital images in remote healthcare that often stored a large amount of patient privacy information and required encrypted transmission. The research team has proposed a chaotic IE method that combined Arnold's Cat mapping and 2D Logistic Sine Coupling Map. The research results indicated that this scheme could improve the randomness and security of encrypted images, ensuring sufficient protection of patient privacy information.

In the paper [7], Zarebnia Mde et al. introduced a multi-IE method with a chaotic system. This method scrambled images through Arnold transformation. The research results indicated that the encryption performance of this method was good and could resist different attacks, effectively protecting user image information.

Then, Hu et al. [8] proposed a color IE algorithm based on a 3D chaotic system to promote the security of color images. This algorithm scrambled the original image through Arnold transform, and then scrambled the RGB channel with the chaotic sequence generated by the chaotic system to achieve encryption. The research results indicated that the uniform distribution of pixel values in encrypted images could effectively improve the security of encrypted images.

Next, Huang et al. [9] designed an IE algorithm based on two-dimensional chaotic mapping. This method used two-dimensional chaotic mapping, and Arnold transform to scramble the original image, and then used chaotic sequences to obfuscate and diffuse the encrypted image. The research results indicated that this method had excellent encryption performance for images and could effectively protect user image information.

Later, in [10], Nie et al. put forward an IE algorithm by combining hyperchaotic system with Advanced Encryption Standard. The algorithm eliminated the partial blocking effect through Arnold transform, then compressed the original image through the cosine transform, and finally encrypted it using Advanced Encryption Standard. It was proved that the algorithm had high security and good compression performance.

Finally, Li et al. [11] proposed a data encryption algorithm for Internet of Things (IoT) terminals and intermediate nodes to address the security issues of image data transmission on the Internet. The research results indicated that this algorithm could effectively resist image cracking attacks, such as brute force cracking and differential attacks and protect the personal privacy and security of users.

In summary, many scholars have conducted research in the fields of IE and image compression and have achieved significant results. However, most encryption methods are too complex and cannot achieve good encryption results, posing a risk of being violently cracked. This study proposes an IE algorithm based on the Frobenius canonical form and combines it with fractal compression to achieve shorter IE time and higher encryption security.

3 Analysis of Image Information Encryption Technology Based on Fractal Image Compression Theory

Along with the growth of the Internet, more users choose images to transmit information. Therefore, in complex network environments, ensuring users' personal privacy is of utmost importance. This study first discusses the encryption technology of images and then proposes a digital images encryption scheme which uses the convolutional Arnold transform combined with the Frobenius canonical form. Then, the model is improved by introducing the tetragonal algorithm and the KT algorithm.

3.1 Arnold Transform Combined with Frobenius Canonical Form Encryption and Fractal Image Compression

Image fractal compression coding is a method of using the local similarity of the image itself to model the image as a fractal volume to compress the image. The encoding of the fractal image compression encoding algorithm is illustrated in Figure 1.

From Figure 1, the image is divided to be compressed into multiple small blocks, each of which is called a "block". Usually, an image is divided into small blocks of equal size. One of the divided small blocks is selected as a reference block. In general, the first block is selected as the reference block. For each non-reference block, it needs to search for blocks in the image that are similar to the reference block. Similarity can be measured using methods such as mean square error and structural similarity. For similar blocks found, it calculates the transformation parameters between them and the reference block. The commonly used transformation parameters include translation, rotation, and scaling. It needs to quantify the transformation parameters to reduce the number of bits required for encoding. The quantized transformation parameters are encoded usually using lossless compression algorithms such as Huffman encoding. Based on the transformation parameters obtained after encoding, it reconstructs non-reference blocks in the image. The reconstruction method can use methods such as inverse transformation. Repeating the above steps until all blocks are encoded. The output encoded data include reference blocks, transformation parameters, and reconstructed images. Image segmentation generates two subblocks. One is the Range (R) block, and the other is the Domain (D) block. The R block is smaller, with the ith R block being R_i, which is usually in size of 4 × 4.

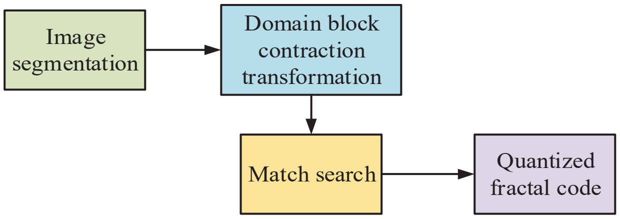


Figure 1 The encoding process of fractal image compression encoding algorithm.

Block D is relatively large and usually has a size of 8 × 8. It sets pixel size to N × N. Uncoded image I is divided into $N_r \bullet N_r$ pixel blocks without overlapping each other and is divided into multiple R blocks to form an R block pool. The same method is used to form a D block pool, but the size of D_i is usually twice the size of R_i, which is $2B \times 2B$, and can overlap as shown in Equation (1) [12].

$$I = \bigcup_{i=1}^{N_r \bullet N_r} R_i, R_i \bigcap R_j = \emptyset, \;\; i \neq j \tag{1}$$

In Equation (1), $N_r = \frac{N}{B}$ and set $\bigcup_i R_i$ mean R block pools. After image segmentation, the D block pool is used as the sample pool, and the R block pool is used as the uncoded set. R_i is matched with the D_i in the D block pool, and the matched D block is recorded as $D_{m(i)}$. The matched D block is changed to the R block, and the fractal change is expressed in Equation (2).

$$W(I) = \sum_{0 \leq i \leq N^{r^2}} (W(I))|R_i = \sum_{0 \leq i \leq N^{r^2}} w_i(I|D_{m(i)}), \quad \forall I \in R^{N \times N} \tag{2}$$

In Equation (2), w_i indicates the transformation from D block to R block, and W represents the set composed of all the transformations [13]. $D_{m(i)}$ denotes the D block that best matches R_i. This equation represents translating $D_{m(i)}$ to the position of R_i, aligning the D block with the R block by contraction, and then sampling the pixel values of $D_{m(i)}$ to match its size with the R block. In fact, the contraction of D block and the matching of R_i are continuous. Finally, the fractal code is quantified, as shown in Equation (3).

$$E(R_i, \hat{D}_{m(i)}) = \frac{1}{B^2} \min_{D \in \Omega} E(R_i, D) \tag{3}$$

In Equation (3), $E(R_i, D)$ denotes the matching block for constrained minimization. Ω represents a codebook. Finally, each data is quantized and the operation is repeated on each R-block to obtain its fractal code. The decoding is relatively simple, formed by applying a change W to any image, as shown in Equation (4) [14].

$$W^N \mu_0 \approx \lim_{n \to \infty} W^n \mu_0 = \mu_{fix} \tag{4}$$

In Equation (4), μ is the iterative image. Decoding is realized through this equation. This study is based on the Frobenius canonical form and improves it as an IE algorithm. Assuming a second-order square matrix A, its characteristic matrix, determinant factor, and standard form of square matrix A can be obtained as shown in Equation (5).

$$\begin{cases} A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \\ \lambda E - A = \begin{pmatrix} \lambda - a_{11} & -a_{12} \\ -a_{21} & \lambda - a_{22} \end{pmatrix} \\ D_1(\lambda) = 1, D_2(\lambda) = \lambda^2 - (a_{11} + a_{22})\lambda + a_{11}a_{22} - a_{21}a_{12} \\ A_F = \begin{pmatrix} 0 & a_{12}a_{21} - a_{11}a_{22} \\ 1 & a_{11} + a_{22} \end{pmatrix} \end{cases} \tag{5}$$

In Equation (5), A means the matrix; $\lambda E - A$ denotes the characteristic matrix of matrix A; A_F indicates the Frobenius canonical form of matrix A. From this, a second-order square matrix C can be set up and partitioned using the theory of partitioned matrices. The Frobenius canonical form of the partitioned square matrix is obtained, which is the encrypted matrix. The key is shown in Equation (6).

$$\begin{cases} C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1,2n-1} & c_{1,2n} \\ c_{21} & c_{22} & \cdots & c_{2,2n-1} & c_{2,2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{2n-1,1} & c_{2n-1,2} & \cdots & c_{2n-1,2n-1} & c_{2n-1,2n} \\ c_{2n,1} & c_{2n,2} & \cdots & c_{2n,2n-1} & c_{2n,2n} \end{pmatrix} \\ b = \{(c_{11}, c_{21}), \ldots, (c_{1,2n-1}, c_{2,2n-1}), \ldots, (c_{2n-1,2n-1}, c_{2n,2n-1})\} \end{cases} \tag{6}$$

In Equation (6), C indicates the second-order square matrix; b means the key, and the decryption method is to solve the inverse operation of the Frobenius canonical form. It performs inverse operations on all matrices to obtain the original image. Since this algorithm does not encrypt the original image as a whole, but encrypts the brightness adjustment coefficients in the fractal code obtained during the fractal encoding process, even if the encrypted brightness adjustment coefficients are small, the original image can be encrypted through iteration. The specific process is shown in Figure 2.

Figure 2 is the flowchart of IE and decryption. The original graphics are divided into non overlapping R-block pools based on 8 × 8 pixels, and then they are generated into 16 × 16 D-block pools in steps of 8 from top to bottom and left to right. Each D-block is evenly divided into 8 × 8 pixel blocks

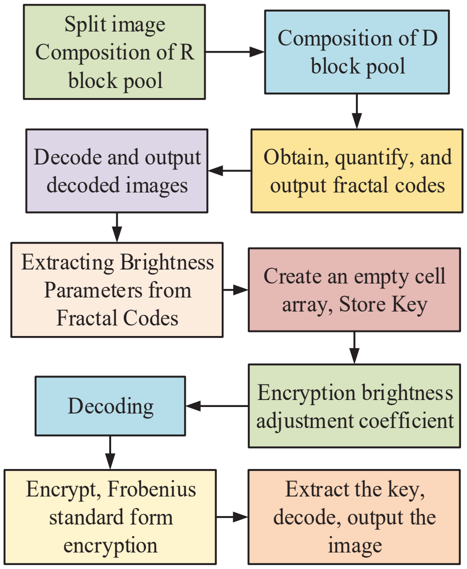

Split image Composition of R block pool
Composition of D block pool
Obtain, quantify, and output fractal codes
Decode and output decoded images
Extracting Brightness Parameters from Fractal Codes
Create an empty cell array, Store Key
Encryption brightness adjustment coefficient
Decoding
Encrypt, Frobenius standard form encryption
Extract the key, decode, output the image

Figure 2 IE process.

to form a codebook, and fractal codes are obtained and output. The image is decoded to obtain the original decoded image. Then it extracts and stores the brightness parameters from the fractal code, encrypts the brightness parameters, decodes the encrypted brightness parameters as a new fractal code, outputs the resulting image, and performs Frobenius canonical form encryption. Finally, it decrypts the stored brightness parameters before fractal decoding, and finally outputs the decoded image.

Arnold transformations usually refers to a two-dimensional Arnold transformation, which has the characteristics of scrambling reversibility and periodicity, making it an important guarantee for IE security. Usually, a single Arnold transformation cannot achieve a good scrambling effect, therefore multiple transformations need to be repeated to fully utilize the performance of the Arnold transformation. However, after a certain number of transformations, the Arnold transformation will restore the image. The expression of Arnold transformation is shown in Equation (7) [15].

$$\begin{cases} x' = (x+y)\text{mod}(N) \\ y' = (x+2y)\text{mod}(N) \end{cases} \tag{7}$$

In Equation (7), x, x, x', y' denote the coordinate points before and after scrambling; N indicates the edge length of the image, and mod represents the modular operation. In fact, pixel shuffling can be seen as a special mapping and is one-to-one correspondence. If the shuffled pixels are not one-to-one correspondence, then one point's pixel value can be retained, and other pixel values will be lost. The narrow Arnold transformation, due to its simple form, cannot fully meet the requirements. Therefore, the generalized Arnold transformation is used, as shown in Equation (8).

$$\begin{cases} x' = (x + ay)\text{mod}(N) \\ y' = (bx + (ab+1)y)\text{mod}(N) \end{cases} \tag{8}$$

In Equation (8), x, y, x', y' express the coordinate points before and after scrambling. To restore the original shape, it performs an inverse transformation on Equation (8) as shown in Equation (9).

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} ab+1 & -a \\ -b & 1 \end{bmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \text{mod}(N) \tag{9}$$

Equation (9) is the expression after inverse transformation. Although Arnold transformation can meet the requirements for encrypted images, its security is relatively low, as shown in Figure 3. Arnold transformation is rarely used alone for IE.

Figure 3 shows the pixel distribution of the original image and the image after Arnold transformation. The pixel distribution of the image after the Arnold transformation is almost identical to the original image, so the Arnold transformation method cannot encrypt pixel features and has poor security. This study combines Arnold transformation with Frobenius canonical form encryption. First the pre-processing of the image using Frobenius canonical form is performed, followed by the Arnold transformation. This can compensate for the shortcomings of Arnold transformation and provide better

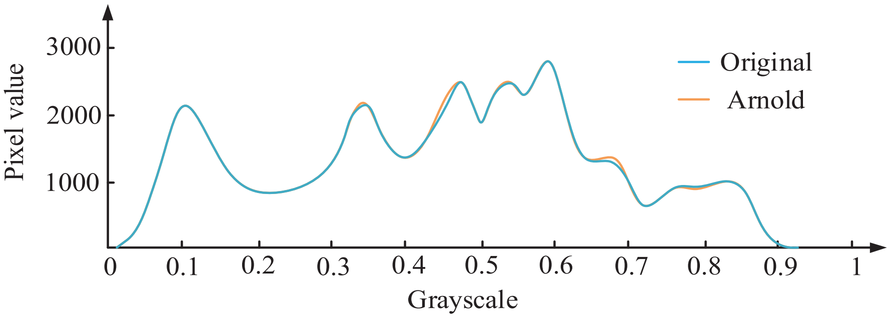


Figure 3 The pixel distribution of the original image and the Arnold transformation image.

security for the encryption processed by the Frobenius canonical form [16]. The specific process is to first divide the pixel matrix into blocks based on a second-order square matrix, use the Frobenius canonical form algorithm to encrypt the pixel matrix, output the processed pixel matrix, and save the key required for decoding. Then it uses Arnold transformation to scramble the pre-processed pixel matrix, and the degree of image scrambling is shown in Equation (10), outputting the encryption matrix. Then it performs the inverse transformation on the output encryption matrix, restores the matrix, and finally obtains the original image through the key.

$$SM(I,\tilde{I}) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M}(i_{ij}-\tilde{i}_{ij})^2}{\sum_{i=1}^{M}\sum_{j=1}^{M}(i_{ij}-r_{ij})^2} \tag{10}$$

In Equation (10), $I = \{i_{ij}\}_{M\times N}$ denotes the image before scrambling; $\tilde{I} = \{\tilde{i}_{ij}\}_{M\times N}$ indicates the image after scrambling; $\{r_{ij}\}_{M\times N}$ denotes the noise distribution image with the same size as the original image. The scrambling degree expresses the degree to which graphics are scrambled, or in other words, the encryption degree. Scrambling degree is one of the indicators that measure algorithm performance. Algorithms with better scrambling degree have higher encryption degree and better security.

3.2 Improving Fractal Image Compression Algorithms and IE

A new image segmentation method is obtained by combining the Tetragonal theorem (TT) with image segmentation. In TT, any natural number can be represented by the sum of the squares of at most four numbers, as shown in Equation (11).

$$N = m^2 + n^2 + h^2 + l^2 \tag{11}$$

In Equation (11), N is any natural number. Using the TT, it divides the graph into four unequal-sized squares, as shown in Figure 4.

Figure 4 shows the block diagram of the TT. From Figure 4, the original image is divided into four cubes with unequal sides. Assuming that the $h \times h$ graph in Figure 4 is a larger block, it needs to be processed again. It processes it using the sum of the tetragonal formula, takes the random number $h_1 \in (0, h-1]$, and decomposes the image block as shown in Equation (12) [17].

$$h^2 = (h_1 + (h-h_1))^2 = h_1^2 + 2h_1(h-h_1) + (h-h_1)^2 \tag{12}$$

In Equation (12), h means the side length of the block; h_1 indicates the side length of a block in the $h \times h$ graph. From Equation (12), the $h \times h$

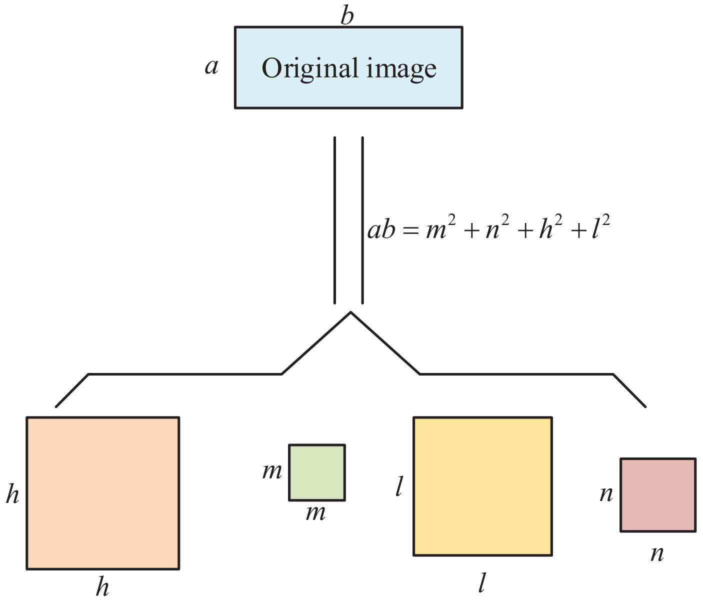


Figure 4 Block diagram of the TT.

graphic block is divided into three parts: h_1^2, $2h_1(h - h_1)$, and $(h - h_1)^2$. If the values of h_1^2 and $2h_1(h - h_1)$ are not greater than the values of the classification conditions, it indicates the completion of the block. If they are greater than the values of the classification conditions, a new value needs to be taken. If $2h_1 = (h - h_1)$, it indicates that $2h_1(h - h_1)$ is also a sum of squares. If the classification criteria are met, it indicates that the classification is complete. If $2h_1 \neq (h-h_1)$, then the part is a rectangular shape. Repeating the TT until all blocks are square [18].

KT refers to the movement of a knight on a chessboard, with a trajectory of "ri". Assuming the knight's position on the chessboard (x, y), according to the traditional KT algorithm, the knight may appear at eight similar point paths, such as points (x + 1, y + 2), as shown in Figure 5.

Figure 5 shows the path of the KT. From Figure 5 it is visible that the path points are all in a "ri" shape, and each parade path does not overlap until all the chessboard points are completed. The encryption of the KT is presented in Figure 6.

From Figure 6, the size of the original image is first calculated, and then it is divided into several blocks. The corresponding chessboard matrix is generated using a wandering matrix. The number of transformations is input

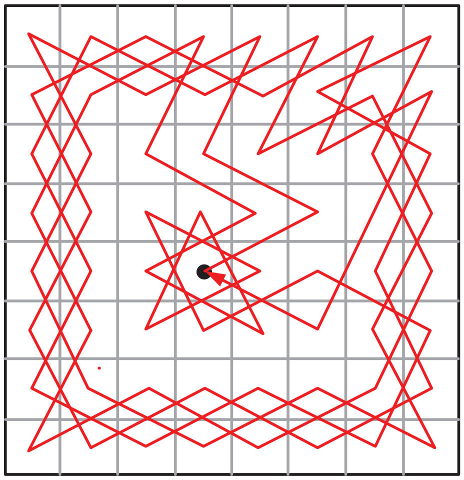
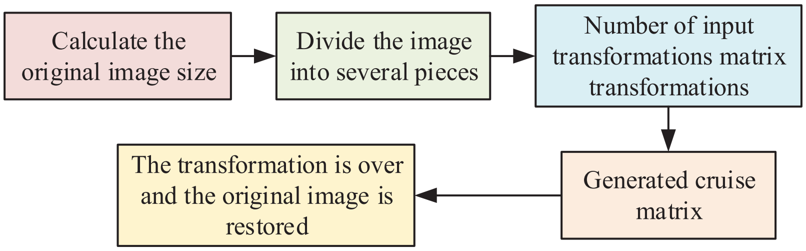


Figure 5 The path of KT.



Calculate the original image size → Divide the image into several pieces → Number of input transformations matrix transformations → Generated cruise matrix → The transformation is over and the original image is restored

Figure 6 KT algorithm encryption process.

to transform the image. After the transformation is completed, the image is restored to its original size. Due to the high time cost, the similarity between the encrypted and the original images, and the low encryption security issues of the KT, this study improves the algorithm by using blocks as a chessboard, labeling each block one by one, using KT to shuffle the numbers, and then using Frobenius canonical form for encryption. The encrypted blocks are transformed into Arnold, which can fully utilize the performance of the KT algorithm. It can also save time and ensure encryption security [19]. Firstly, it pre-processes the image, sets the expected final block edge length to z, and uses the TT to segment the image so that the decomposed shapes are all cubes and less than the maximum expected edge length z, until all shapes comply with the segmentation rules. Then it encrypts the image and the number all the blocks one by one. The optimized KT is used to scramble the blocks and output the scrambled coordinates according to the numbers. After scrambling the coordinates, it performs Arnold transformation to scramble the graphics. After the image is scrambled, encryption is completed. The corresponding decryption is the inverse operation of the encryption.

In fact, scrambling graphics include two types: the first is to scramble local images after image segmentation, and the other is to scramble the entire image. The first method can make the scrambling sufficient and have a higher safety factor, but due to different scrambling rules, blocky effects are prone to occur. The second method can alleviate the blocky effect, therefore, after the encrypted image is completed, an overall scrambling is performed to eliminate the blocky effect. Objectively evaluating an image compression method will use the evaluation indicators shown in Equation (13).

$$\delta^2 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M}[f(x,y) - f'(x,y)]^2}{\sum_{i=1}^{M}\sum_{j=1}^{M}[f(x,y)]^2} \tag{13}$$

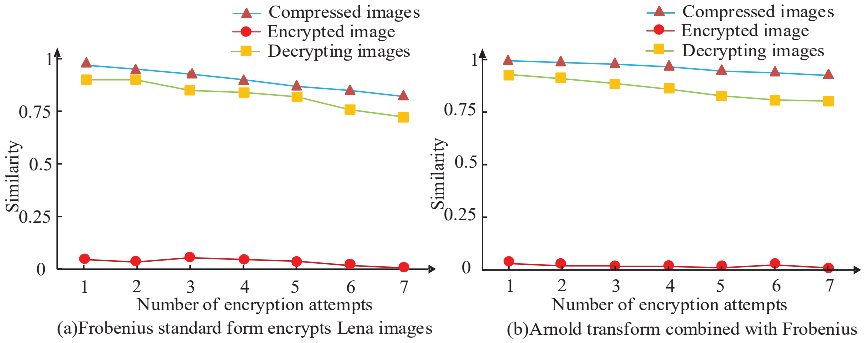
Equation (13) shows the normalized mean square error, where f stands for the pre compressed image; f' indicates the compressed image, and M and N stand for the size of the image. The degree of IE can also be measured by the peak signal-to-noise ratio (SNR), as shown in Equation (14) [20].

$$PSNR = 101g\frac{f_{\max}^2 \cdot MN}{\sum_{i=1}^{M}\sum_{j=1}^{M}[f(x,y) - f'(x,y)]^2}(dB) \tag{14}$$

In Equation (14), $f_{\max}$ denotes the maximum grayscale value. Generally speaking, the higher the peak SNR, the better the image quality. When the peak SNR is higher than 40 dB, it indicates that it is close to the original image. When the peak SNR is between 40 dB and 30 dB, it indicates good image quality. When the peak SNR is between 30 dB and 20 dB, it indicates poor image quality. When the peak SNR is higher than 20 dB, it indicates that it is basically unrelated to the original image.

4 The Results of Image Information Encryption Technology Based on Fractal Image Compression Theory

It analyzed the encryption effect of Frobenius canonical form encryption algorithm and Arnold transformation combined with Frobenius canonical



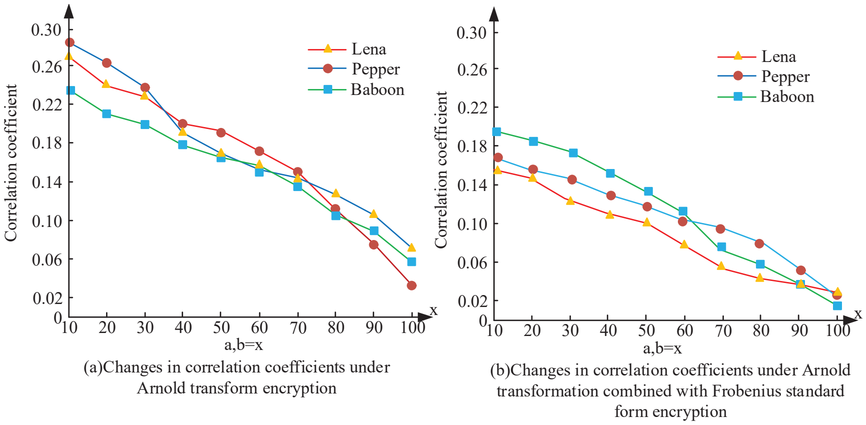
(a)Frobenius standard form encrypts Lena images

(b)Arnold transform combined with Frobenius standard form to encrypt Lena images

Figure 7 Similarity of Lena images encrypted by different algorithms.

form encryption algorithm, and selected Lena image as the encrypted image for this experiment. The experimental environment was a Windows 8 system with Core i7-3320M and 8GB of memory. It encrypted the image, as shown in Figure 7.

Figure 7(a) shows the similarity between the Lena and the original images at different stages of encryption using the Frobenius canonical form. As the number of encryption times increased, the similarity between the images in each stage and the original image gradually decreased. Among them, the fractal compressed image had the highest similarity with the original image, reaching over 80%. Due to IE, the similarity between the fractal compressed encrypted and the original images was extremely low without decryption. The similarity not reaching 0 might be due to a small number of black pixels in the original image fitting with the encrypted black pixels, which were within an acceptable error range. The similarity between the decrypted and the original images was over 70%. Figure 7(b) shows the similarity between Lena images and the original image at different stages of encryption using Arnold transform combined with Frobenius canonical form. The fractal compressed image had the highest similarity with the original image, reaching over 90%. The similarity between the decrypted and the original images was over 80%, while the similarity between the fractal compressed encrypted and the original images was still at a relatively low level, within an acceptable range. We selected three images as Lena, Pepper, and Baboon images, and performed Frobenius canonical form encryption and this Arnold transform combined with Frobenius canonical form encryption on the three images under different initial parameters. The results are presented in Figure 8.



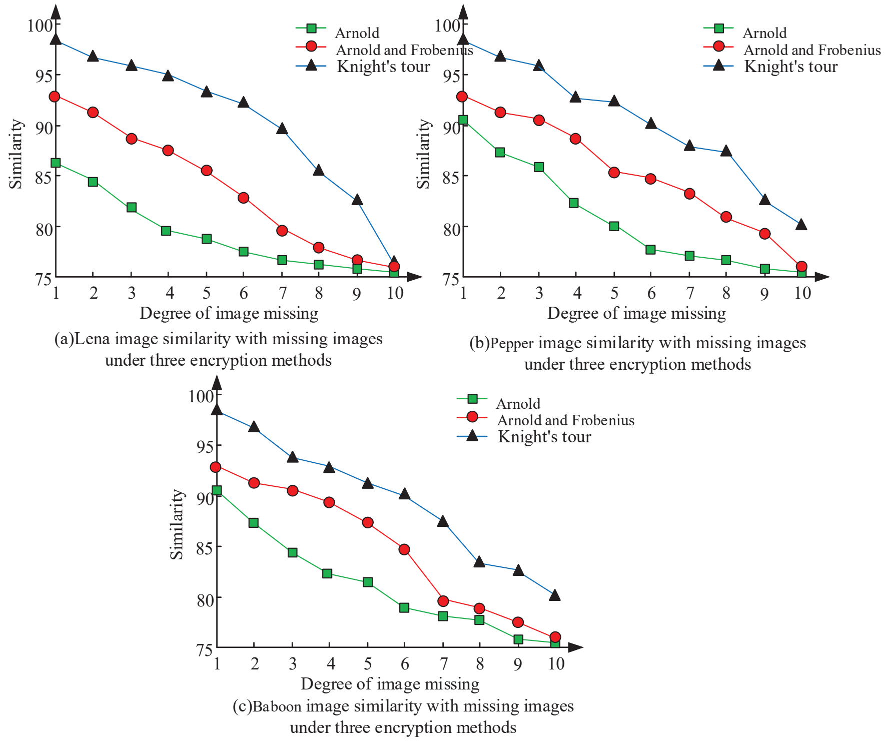
(a)Changes in correlation coefficients under Arnold transform encryption

(b)Changes in correlation coefficients under Arnold transformation combined with Frobenius standard form encryption

Figure 8 Changes in CCs under different encryption algorithms.

Figure 8(a) shows the changes in correlation coefficients (CCs) using Arnold transform encryption alone. As the initial parameters increased, the CCs continued to decrease, and the downward trend of the three images was basically consistent. Figure 8(b) is the variation of the CC under the combination of Arnold transformation and Frobenius canonical form encryption. The CC continuously decreased with the increase of initial parameters, and the decreasing trend was significantly greater than that under Arnold transformation encryption. The smaller the CC, the higher the encryption level and the safer the image. The research outcomes expressed that the Arnold transformation combined with the Frobenius canonical form method significantly reduced the CC, and the encryption performance was better than that of the Arnold transformation method. When there was some loss of information in the image, the Lena, Pepper, and Baboon images were encrypted using the Frobenius canonical form, Arnold transformation, Frobenius canonical form, improved TT, and KT, respectively. The image results are restored as illustrated in Figure 9.

Figure 9 shows the similarity of restored images of Lena Pepper Baboon images under three encryption methods. Due to the small number of pixels in the Lena image, the similarity between the encrypted restored and the original images was higher than the other two images. Moreover, the improved TT-KT method showed higher similarity for the restored image after encryption of the same image. The improved TT and KT method had better performance in restoring images and stronger anti-interference ability than the other two



(a)Lena image similarity with missing images under three encryption methods

(b)Pepper image similarity with missing images under three encryption methods

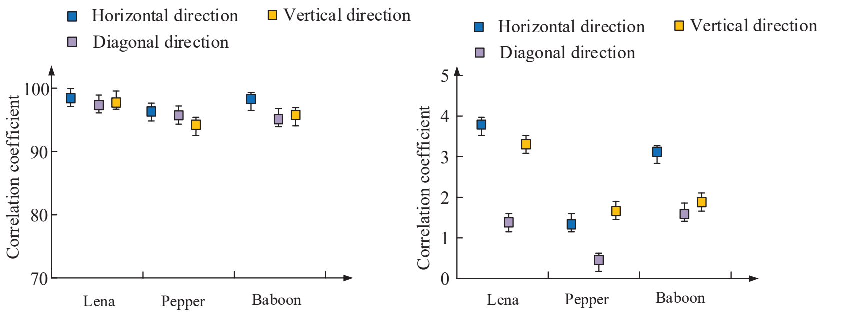
(c)Baboon image similarity with missing images under three encryption methods

Figure 9 Image similarity with missing images in three encryption methods.

methods. The research on improving the encryption performance of the TT and KT method is shown in Figure 10.

Figure 10(a) shows the CCs of the original images in different directions under three different images, and Figure 10(b) shows the CCs of encrypted images in different directions under three different images. From Figure 10, the CCs of the three encrypted images in all three directions were low, indicating that the improved TT and the KT method had better encryption performance. Compared to the previous two methods, the encryption performance of this method was significantly better than the first two methods. The scrambling degree comparison of the three methods is presented in Figure 11.

From Figure 11, the scrambling degree of the three methods increased with the increase of scrambling times. Among them, the improved TT and KT method had the highest scrambling degree, while the other



(a)Original image correlation coefficient (b)Knight's tour Method image correlation coefficient

Figure 10 CCs between original and encrypted images in different directions of three types of images.

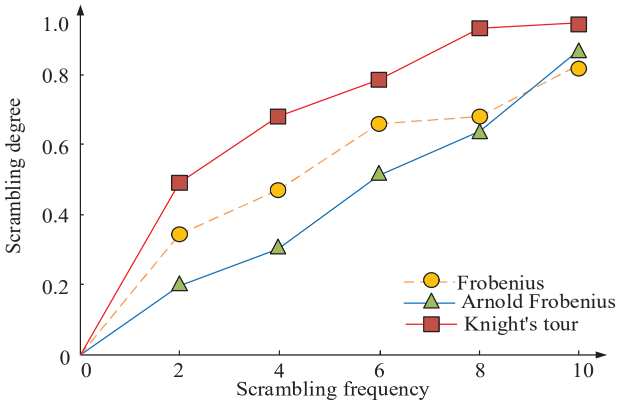


Figure 11 Scrambling degree under three methods.

methods had lower scrambling degree. This indicated that the improved TT and KT method had the best encryption performance and higher security. The comparison of methods involved comparing their encryption performance and decryption image quality and evaluating their encryption and decryption time as shown in Table 1.

From Table 1, the encryption and decryption time of the improved TT and KT method has slightly increased compared to the other two algorithms. However, overall, the time consumption was still relatively short, within an acceptable range, and the encryption and decryption performance of this method was greater than that of the other two algorithms.

Table 1 Encryption and decryption time

Image Types	Number of Experiments	Method Type	Average Encryption Time(s)	Average Decryption Time(s)
Lena	First experiment	Frobenius	6.2	7.3
		Arnold Frobenius	6.7	7.8
		TT KT	7.5	8.2
	Second experiment	Frobenius	6.1	7.6
		Arnold Frobenius	6.4	7.2
		TT KT	7.8	8.4
Baboon	First experiment	Frobenius	5.8	6.9
		Arnold Frobenius	6.6	7.4
		TT KT	7.1	7.8
	Second experiment	Frobenius	6.1	7.3
		Arnold Frobenius	6.4	7.7
		TT KT	7.9	8.5
Pepper	First experiment	Frobenius	6.0	7.1
		Arnold Frobenius	6.8	8.0
		TT KT	7.7	8.5
	Second experiment	Frobenius	5.9	7.0
		Arnold Frobenius	6.3	7.5
		TT KT	7.3	8.1

5 Conclusion

Today, along with the rapid growth and widespread popularity of the Internet, the leakage of personal information is common. As an essential means of information transmission, images contain a lot of user privacy data. This study used Arnold transformation combined with Frobenius canonical form and the TT algorithm combined with the KT algorithm to encrypt images.

The experimental results showed that when using Frobenius canonical form and Arnold transformation combined with Frobenius canonical form to encrypt Lena images, the decrypted Frobenius canonical form algorithm corresponded to an image similarity of over 70%, while the Arnold transformation combined with Frobenius canonical form algorithm reached an image similarity of over 80%. When some information in the image was missing, the image encrypted using the Frobenius canonical form had a high similarity to the original image when restored to the input image, indicating that the Frobenius canonical form had strong anti-interference ability. When processing Lena images, the encryption time corresponding to Frobenius canonical form algorithm, Arnold transformation combined with Frobenius canonical form algorithm, and TT algorithm combined with KT algorithm were 6.2 s, 6.7 s, and 7.5 s, respectively, and the decryption time was 7.3 s, 7.8 s, and 8.2 s. When processing Baboon images, the encryption times corresponding to the three algorithms were 5.8 s, 6.6 s, and 7.1 s, respectively. The decryption time was 6.9, 7.4, and 7.8 seconds. When processing Pepper images, the encryption times corresponding to the three algorithms were 6.0 s, 6.8 s, and 7.7 s, respectively. The decryption time was 7.1, 8.0, and 8.5 seconds. The combination of TT and KT algorithms had slightly increased computational time, but the overall performance was still stronger than the first two methods. However, there are still shortcomings in this study. The complexity and number of pixels in the sample images used are much smaller than those in practical applications. If images with higher recheck levels and more pixels are used, it can make the model in this study more practical.

References

[1] Kaur G, Agarwal R, Patidar V. Color image encryption scheme based on fractional Hartley transform and chaotic substitution-permutation. The visual computer, 2022, 38(3):1027–1050.

[2] Guo L, Du H, Huang D. A quantum image encryption algorithm based on the Feistel structure. Quantum Information Processing, 2021, 21(1):20–37.

[3] Peng X, Zeng Y. Image encryption application in a system for compounding self-excited and hidden attractors. Chaos Solitons & Fractals, 2020, 139(6):1144–1159.

[4] Mohamed Boussif, Noureddine Aloui, Adnene Cherif, Boussif M, Aloui N, Chrif A. Securing DICOM Images by a New Encryption Algorithm

Using Arnold Transform and Vigenère Cipher. IET Image Processing, 2020, 14(6):1209–1216.

[5] Wang X, Gao S. Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Information Sciences, 2020, 539(9):195–214.

[6] Jain K, Aji A, Krishnan P. Medical Image Encryption Scheme Using Multiple Chaotic Maps. Pattern recognition letters, 2021, 152(12): 356–364.

[7] Zarebnia M, Pakmanesh H, Parvaz R. A fast multiple-image encryption algorithm based on hybrid chaotic systems for gray scale images – ScienceDirect. Optik, 2019, 179(3):761–773.

[8] Hu W, Dong Y. Quantum color image encryption based on a novel 3D chaotic system. Journal of Applied Physics, 2022, 11(131):1142–1155.

[9] Huang H, Yang S, Ye R. An efficient symmetric image encryption by using a novel 2D chaotic system. IET Image Processing, 2020, 14(6):1157–1163.

[10] Nie Z, Liu Z X, He X T, Gong L H. Image compression and encryption algorithm based on advanced encryption standard and hyper-chaotic system. Optica Applicata, 2019, 49(4):545–558.

[11] Li B, Feng Y, Xiong Z, Yang W, Liu G. Research on AI Security Enhanced Encryption Algorithm of Autonomous IoT Systems. Information Sciences, 2021, 575(3):379–398.

[12] Hanif R, Mustafa S, Iqbal S, Piracha S. A study of time series forecasting enrollments using fuzzy interval partitioning method. Journal of Computational and Cognitive Engineering, 2023, 2(2):143–149.

[13] Kaur G, Agarwal R, Patidar V. Color image encryption scheme based on fractional Hartley transform and chaotic substitution-permutation. The visual computer, 2022, 38(3):1027–1050.

[14] Wang X, Gao S. Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Information Sciences, 2020, 539(9):195–214.

[15] Khan S, Han L, Qian Y, Lu H, Jiao SM. Security of multimedia communication with game trick based fast, efficient, and robust color-/gray-scale image encryption algorithm. Transactions on Emerging Telecommunications Technologies, 2020, 32(2):12–41.

[16] Singh P, Acharya B, Chaurasiya R K. Low-area and high-speed hardware architectures of Lblock cipher for Internet of Things image encryption. Journal of electronic imaging, 2022, 5(3):33012–33040.

[17] Erkan U, Toktas A, Toktas F, Alenezi F. 2D eπ-map for image encryption. Information Sciences, 2022, 589(5560):770–789.

[18] Wang X, Yang J. A Privacy Image Encryption Algorithm based on Piecewise Coupled Map Lattice with Multi Dynamic coupling coefficient. Information Sciences, 2021, 569(5):217–240.

[19] Liu C, Wang X, Jiang D. A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT. Information Sciences, 2021, 574(1):505–527.

[20] Ratan R, Yadav A. Security Analysis of Bit plane Level Image Encryption Schemes. Defence Science Journal, 2021, 71(2):209–221.

Biography

Jinna Yu, a professional course teacher at the School of Computer Engineering at Shangqiu Vocational and Technical College, obtained a Master's degree in Software Engineering from Northwestern Polytechnical University in 2016. Her areas of interest include image processing, information security, big data analysis, etc. She have served as an internship instructor in multiple companies and have led students to participate in provincial and national competitions multiple times.