IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Two-Level Image Encryption Algorithm Based On Key-Image

V. Sridhar[1*], M. Dyna[2]

[1,2]CSE Dep. MVSR Engineering College, Hyderabad India

[*]*Corresponding Author: sridhar_cse@mvsrec.edu.in*

Available online at: www.ijcseonline.org

Accepted: 13/Aug/2018, Published: 31/Aug/2018

***Abstract*—** In the contemporary world images play key role in information interchange. Medical, defence, space and various areas of domain make use of high scale images in several applications. Security becomes the main concern wherein the images are to be protected so that they cannot be seen by any advisory. This can be achieved by image encryption. There are various image encryption methods that are based on textual key and text data which are not efficient for high definition images. In this paper we propose a three step image encryption algorithm which uses another image as a key. In the first step, key image is scaled and tiled, in step2 encryption is achieved using grey-value substitution and in step-3 the output generated from step-2 is scrambled using Fibonacci transformation to add additional security. This multistep encryption provides high security for images. The performance of this algorithm is analyzed using different attack models which results in high security without any loss of input image.

***Keywords*—**Image security, Cryptography, Network security, Image processing

I. INTRODUCTION

In recent trends, high scale images are being processed for various applications. To provide security for the information transmitted over the channel, a transformation methodology called Encryption is used.

Encryption is a process of protecting information from malicious attacks by converting data into a form which is un-recognizable by its attackers. Data encryption makes use of scrambling of the content, such as text, image, audio, video and so forth to make the data inaccessible during the transmission. The goal of encryption is to protect the data against the attackers. On the counter part, of data encryption is data decryption, which recovers the original data.

The security of any encryption system relies on the confidentiality of the encryption/decryption key. The security level of an encryption algorithm is measured by the size of its key space. The larger the size of the key space is, the more time the attacker needs to do the exhaustive search of the key space, which improves the security level.

There are two encryption/decryption key types: the public-key system and the private-key system. The private-key system is also called the symmetric system because the decryption key is the same as the encryption key. Because of its symmetric property, the encryption/ decryption key has to be transmitted prior to the transmission of the cipher text.

The drawback of the private-key system is that a secure communication channel is required for encryption/decryption key transmission. Figure 2 shows a typical private-key system.

The public-key system, which is also called the asymmetric system, has a decryption key that is different from the encryption key. Each person in the group knows the encryption key. This way each member can use the public key to encrypt a message. But the person who has the decryption key can only decrypt the ciphertext. In most of the cases, it is computationally infeasible to derive the decryption key from the encryption key, and this is how the ciphertext can be protected. With the public-key encryption system, there is no need for a secure communication channel for the transmission of the encryption key.

A. Image Encryption

Network technologies and media services provide ubiquitous conveniences for individuals and organizations to collect, share, or distribute images/videos in multimedia networks and wireless or mobile public channels. Image security is a major challenge in storage and transmission applications. For example, video surveillance systems for homeland security purposes are used to monitor many strategic places such as public transportation, commercial and financial centers. Huge collection of videos and images that store private information are generated, transmitted, or restored every day. In addition, clinical images with a patient's records may be

shared among the doctors in different branches of a health service organization over networks for different clinical purposes.

These images and videos may contain confidential information. Providing security for these images and videos becomes a serious issue for individuals, and various group of organizations. Moreover, applications in the automobile, medical, construction and fashion industry require scanned data, and blue-prints to be protected against espionage.

In order to provide security for the images or videos in various fields we can use standard encryption algorithms. Among them, DES, RSA, AES and IDEA are elaborately designed and widely adopted. However, image and video data are usually very large in size that makes these encryption standards computationally demanding. In recent trends, many new algorithms for image/video encryption have been proposed. These algorithms explore the properties inherent to image and video data and thus improve the efficiency of data encryption.

Considering the long lifetime of image in the fore-mentioned domains, it is imperative to develop and employ techniques which protect the content throughout their lifetime. Image encryption is an effective approach to protect images or videos by transforming them into completely different formats.

B. Motivation

Various image encryption methods are developed. Some of them change the value of the pixels of the image, and some do not change the value instead, scramble the image pixel positions. The algorithms developed using these two methods are not secured because they are limited to only either method of encryption, it makes cryptanalysis simpler. The existing algorithms use normal text as the key for the image encryption also.

It is thus vulnerable to the existing attacks. The motivation for this project is to use image as the key for the Encryption /Decryption and to combine two methods of existing image encryption schemes into one.

C. Encryption based on key-images

To achieve higher levels of security, one solution is to change image pixel values while scrambling the positions of image pixels or blocks using different techniques. In paper [1] author introduces new private key lossless image encryption algorithm using a new concept "key-image" which is a given by user of any size. In this paper we propose a new image encryption algorithm which combines encryption based on key-image along with image scrambling. The secret image is encrypted through the combination of a new gray value substitution operation and image scrambling algorithm which makes the encryption system strong. The algorithm accepts key-image from the user defined size and it generates new key-image with size equal to original image by tiling and scaling initial input key-image. The algorithm can use key-image of same size as input image, but in a private key system both encryption and decryption systems must use the same key. The sender must transmit user chosen key-image to the receiver through secure communication channel. So transmission of larger image for the key results in computational overhead and in fact the same channel can be used to transmit original image without involving in encryption process. Hence the image used for the key can be of any size.

Image Scrambling

Image encryption can also be accomplished by scrambling image pixel positions using different techniques in the spatial domain. One example is the recursive sequence based image scrambling approach. It scrambles images using different recursive sequences such as the Fibonacci sequence, Cellular automata and chaotic maps. Image encryption can also be accomplished by scrambling coefficient matrices/blocks in the transform domain. Nevertheless, these approaches have extremely low security levels due to the lack of security keys or the small key space. Furthermore, the permutation-only based encryption schemes are known to be vulnerable for plaintext attacks.

Another approach for image encryption is to change image pixel values based on combination of transformations and logic operations between input image and key-image. The security level of this method is much lower because the results of its decomposition process and the logical operations are predictable. It is not immune to plaintext attacks.

As a major technique of image encryption, image scrambling algorithms are becoming a research hotspot and one of the effective means to protect knowledge property right.

According to sorting element, image scrambling algorithms can usually be categorized as three types. They are scrambling algorithm based on pixel value sorting, binary bit exchange of pixel value and coefficient reset of discrete transformation. Image scrambling algorithms based on pixel value sorting, usually relies on outside sorting matrix to change the position of the image pixel value. They need to store or transmit additional information to get original image, which raises algorithm safety problems.

Image scrambling algorithms based on pixel value bit, often uses the idea of pixel value sorting with bit operations. The security of this algorithm relies on encrypting algorithm itself. To decrypt the encrypted image, we require a password or a serial number etc. Image scrambling

algorithms based on coefficient reset of discrete transformation, usually causes some reverse discrete transformation value smaller than zero or larger than 255. If the result is stored as an image format file, we cannot get the exact image when discrete transformation is applied to get reverse scrambling image.

Many Image scrambling algorithms are developed, but based on Comparison of Recursive Sequence Based Image Scrambling Algorithms [2], the experimental results demonstrate that the scrambling algorithms based on both P-Fibonacci and P-Lucas sequences have better performance when subjected to attacks and also in terms of algorithm execution analysis in terms of implementation efficiency and low computational requirements. This makes them suitable for real-time applications. Hence we are using Image Scrambling algorithm based on P-Fibonacci in the third phase [3].

The rest of the paper is organized as follows. The Proposed Encryption process is discussed briefly in Section-II. In Section-III key image tiling and scaling process is discussed. Then in Section-IV, P-Fibonacci based image scrambling process is discussed. In Section-V experimental results and attach models are discussed. And the conclusion is in Section VI.

II. NEW IMAGE ENCRYPTION SCHEME USING KEY-IMAGES

To achieve higher levels of security, one approach is to change the image pixel values while scrambling the positions of image pixels or blocks using different techniques. In this project, we introduce new private key lossless image encryption algorithm using a new concept "key-image" which is a given by the user of any size. First, the secret image is encrypted through the combination of a new gray value substitution operation and image scrambling algorithm which makes the encryption system strong. The algorithm accepts a key-image from the user defined size and it generates a new key-image with size equal to the original image by tiling and scaling initial input key-image.

The algorithm is decomposed into three phases. First phase is responsible for generating the key-image whose size is equal to the input user image by accepting initial key-image of any size of user choice. In Second phase it changes the pixel data of the image by applying gray value substitution of pixel matrix. In third phase, it accepts generated resultant image from second phase and implements image scrambling algorithm which changes the positions of image pixels.

This new encryption scheme can used to encrypt color images in different domains like medical images, Satellite images. The underlying foundation of the algorithm is to change image pixel values by performing the XOR operation between each bit of original image with corresponding 3×3 block of key-image. This is followed by an image scrambling process which changes the locations of image pixels or blocks.

Many applications benefit from scrambling technologies; these include pay-TV, confidential video conferencing, facsimile, medical applications, as well as various defence applications. It seems that digital image scrambling is easy. But this is not the case. The main reason is that a scrambled image must endure certain attacks and the original image must be recovered as much as possible, at least by our human vision system.

Since digital image encryption presents a set of issues, aside from security, that are unique in the data cryptography field, a digital image- scrambling scheme should have a relatively simple implementation, amenable to low-cost decoding equipment and low-delay operation for real-time interactive applications. It should have minimum adverse impact on the compressibility of the image. It should preferably be independent of the bit stream compression selected for the image, and allow compression scalability without having to decrypt. It should provide good overall security, although it may also be preferable in some systems to allow no authorized users a level of transparency, both to entice them to pay for full transparency, and to discourage code - breaking.

An Efficient Image Encryption Algorithm

The proposed Algorithm can encrypt gray scale as well as RGB images. Gray scale images are stored as 2D array of pixel values. This 2D array is directly given as parameters to core encrypt/decrypt functions.

RGB images are stored as 3D array with three 2D arrays each for red, green, blue components. To encrypt 3D images, it is divided into three 2D arrays. Each 2D array will be separately encrypted/decrypted. After that they are combined as RGB layers to give the final output.

a) Encryption Process

The proposed Encryption algorithm is divided into three phases. First phase in only for generating final key image and rest of the phases include actual algorithm implementations.

Phase 1 Key-Image Tiling and Scaling: Here it accepts key-image which can be of any size. The final key-image is generated by tiling the input key-image repeatedly so as to form its size equals to (m+2) by (n+2) where original user input image size is m by n.

Phase 2 Grey-Value Substitution: It accepts the key-image generated from phase-1 and input image from the user, performs XOR operation between each bit of input image

pixel with corresponding pixel bits of 3-by-3 block of Key-Image, followed by it performs shift operation.

Phase 3 Image scrambling: It accepts output image from phase-2 and changes the pixel positions of it to make image encryption method stronger. Here we are using Image scrambling algorithm based on P-Fibonacci to change the pixel positions randomly.

The block diagram of Encryption process is shown in Figure 1. In the complete encryption process the key-space includes key-image, p value and i value. The given input image is used in phase-1 and phase-2 of the encryption algorithm.

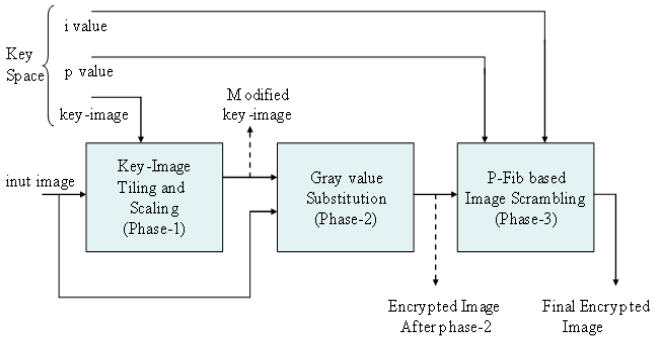


Figure 1: Block diagram of encryption system in the proposed algorithm

b) Decryption Process

The presented algorithm uses private key system. Both sender and receiver will use the same key-image for encryption and decryption respectively. In private key system, decryption process uses same algorithm used by encryption but in the reverse order.

Decryption process is also divided into three phases. The input to decryption includes encrypted image, key image and p, i value. In the decryption process phase-1 is similar to encryption process which is responsible for generating final key-image, where as Second phase and Third phases were differing. The phase-2 and phase-3 f encryption process are becomes phase-3 and phase-2 of decryption process respectively. And in turn the operations in the phases are also reversed.

Phase 1: Key-Image Tiling and Scaling: Here it accepts key-image which size can be of any size. The final key-image is generated by tiling the input key-image repeatedly so as to form its size equals to (m+2) by (n+2) where encrypted image size is m by n.

Phase 2: Image Unscrambling: It accepts encrypted image, p and i values unscrambles it. It uses same Image scrambling algorithm based on P-Fibonacci to unscramble the image. The output image from this phase is decrypted image after one unscrambling.

Phase 3: Grey-Value Substitution: It accepts key-image generated from phase-1 and output from phase 2, implements all operations in reverse order. It implements circular right shift operation followed by XOR operation between each bit of image pixel with corresponding pixel bits of 3-by-3 block of Key-Image. The block diagram of Decryption process is shown in Figure 2.

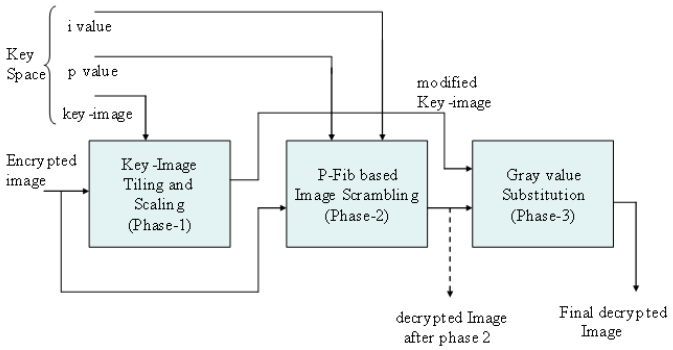


Figure 2: Block diagram of decryption system in proposed algorithm

III. PHASE-1: KEY-IMAGE TILING AND SCALING

This is the first phase of the algorithm. It performs only initial operation on user given key-image to generate final key-image. The original user input image can be of any size and key image can be of same size or smaller when compared to input image.

In order to use small key-image to encrypt large user input image, the key-image must be tiled and then scaled to make its size increased by m rows and r cols with reference to user input image. If the size of input image is m rows and n cols, then the size of the final key-image generated from this phase is (m+2) rows and (m+2) cols, Because in the next phase each bit of encrypted bit depends on a bit of user input image and corresponding 3-by-3 block of final key-image.
The different steps of the algorithm are:

1. Read Input image and Key-image of any size.
   Let L be the user input image with m rows and n cols (m by n), and k be the key-image with a rows and n cols (a by b).

2. Calculate the parameters x and y based on m and n,and a and b values. The x value indicates number of repetitions of initial key-image row wise in the final key-image; similarly y value indicates number of repetitions of initial key-image column wise in the final key-image. The values of x and y are generated based on the following calculation

$$\left.\begin{array}{l} x = (m + 2) / a, \text{ and} \\ y = (n + 2) / b \end{array}\right\} \text{(Equation 3.1)}$$

Based on these values, the given key-image is tiled x times horizontally and y times vertically within final key-image.

3. The tiled image in previous step does not generate final key-image of size (m+2) rows and (n+2) cols because the calculation in previous step may gives fractional values into x and y, but in tiling operations it takes only integer part of x and y. So in order to make the image generated from previous step of size (m+2 rows and (n2) cols it must be scaled to that size. Scale the image generated from step 2 to the size of (m+2) rows and (n+2) cols.

4. The output from this phase is input image of size m-by-n and final key-image of size (m+2)-by-(n+2).

IV. PHASE-2: GRAY VALUE SUBSTITUTION OF PIXELS MATRIX

Any encryption algorithm changes the data of plain text to produce cipher text by performing different operation on it. In the process of image encryption the plain text data itself includes the pixel values of input image. Changing the pixel values of input image generates cipher image which is visually unrecognizable. The operations performed on input image must be reversible so that these operations are performed in reverse at the decryption process to generate input image again at the receiver system.

This phase accepts input image and final key-image generated from phase-1 and generates encrypted image. This phase is based on 3 × 3 neighbourhood bit exclusive-or and bit shift operation. Grey value substitution of pixel matrix changes pixel value by performing combination of bitwise XOR and shift operations. Bit shift factor is introduced in the algorithm. Select-plaintext attacked is better overcome.

A. Encryption Process

1. **Input:** The input is Original image c (m-by-n) and final key image (m+2)-by-(n+2) from phase-1.
2. **Mapping:** We assume the original image is $f(x, y)$, 256 gray levels, its size is $M \times N$ pixels, where $f(x, y)$ represents the pixel gray value. (x, y) is the coordinate of pixel, x=0,1,…,M-1 ; y=0,1,…,N-1

Final key-image has 256 gray levels and (M+2) × (N+2) size, where x=0, 1, M+1; y=0, 1,…, N+1. The positions of $f(x, y)$ and $c(x+1, y+1)$ are overlapped. The location relationship of original image and key-image is shown below.

c(x, y)	c(x,y+1)	c(x,y+2)
c(x+1,y)	(x+1,y+1) f(x, y)	(x+1,y+2)
c(x+2,y)	(x+2,y+1)	c(x+2,y+2)

This arrangement makes the every pixel value of encrypted image will depends on 3 × 3 neighbourhood block of key-image, and hence if any small change in the key-image will change the pixel value of encrypted image completely. It makes cryptanalysis difficult.

3. **Determine bits of encrypted image:** Each bit of the pixel value of encrypted image will be generated from corresponding bit of the pixel value of key-image and input image.

Let $k1$ $k2$ $k3$ $k4$ $k5$ $k6$ $k7$ $k8$ represents 8 bits of encrypted image $g(x, y)$ in the coordinate (x, y). It can be obtained from bits which the original image and key image do bit exclusive-or according to the rule as follows:

$k1$ is equal to the first bit obtained from $f(x, y) \oplus c(x, y)$,

$k2$ is equal to the second bit obtained from $f(x, y) \oplus c(x, y+1)$,

$k3$ is equal to the third bit obtained from $f(x, y) \oplus c(x, y+2)$,

$k4$ is equal to the fourth bit obtained from $f(x, y) \oplus c(x+1, y)$,

$k5$ is equal to the fifth bit obtained from $f(x, y) \oplus c(x+1, y+2)$,

$k6$ is equal to the sixth bit obtained from $f(x, y) \oplus c(x+2, y)$,

$k7$ is equal to the seventh bit obtained from $f(x, y) \oplus c(x+2, y+1)$,

$k8$ is equal to the eighth bit obtained from $f(x, y) \oplus c(x+2, y+2)$.

The symbol $\oplus$ represents the exclusive OR operation bit-by-bit

4. **Determine bit shift factor:** Shift operation adds additional strength to encryption algorithm. Shifting will be done by generating bit shift factor by computing the average (modulo 8) of 3 × 3 neighbourhood values.

Compute the average of pixels of $c(x+1, y+1)$ in 3×3 neighbourhood, the result is stored in $aver$.

$$aver=(c(x,y) + c(x,y+1) + c(x,y+2) + c(x+1,y) + c(x+1,y+1) + c(x+1,y+2) + c(x+2,y) + c(x+2,y+1) + c(x+2,y+2)) / 9 \quad \text{(Equation 4.1)}$$

The bit shift factor of encrypted image $k(x, y)$ in (x, y) is determined by following equation 4.2

$$\varepsilon=\text{mod}(aver, 8) \quad \text{(Equation 4.2)}$$

5. **Modify bit:** The bit shift factor generated in previous step is used to shift the cipher image pixel bit positions.

   Circular bit shift to the left, *k*1 *k*2 *k*3 *k*4 *k*5 *k*6 *k*7 *k*8 shift bit to the left, and then we obtain 8 new bits arrangement *s*1 *s*2 *s*3 *s*4 *s*5 *s*6 *s*7 *s*8

6. **Gray value Substitution:** The gray value of encrypted image *k*(*x*, *y*) is determined by following equation 4.3.

$$k(x,y) = \sum_{i=1}^{8} s_i 2^{8-i} \quad \text{(Equation 4.3)}$$

   Now, we will get the encrypted image based on grey value substitution of pixel matrix.

B. Decryption Process:

1. **Input:** Let e be the encrypted input image of size m × n, and ke be the modified key-image of size (m+2) × (n+2).
2. **Determine bit shift factor:** For every bit of encrypted image, we are calculating bit shift factor by computing the average (modulo 8) of 3 × 3 neighborhood values of key-mage.

   Compute the average of pixels of *c*(*x*+1, *y*+1) in 3×3 neighborhood, the result is stored in *aver*.

   aver=((*x*,*y*) + ke(*x*,*y*+1) + ke(*x*,*y*+2) + ke(*x*+1,*y*) + ke(*x*+1,*y*+1) + ke(*x*+1,*y*+2) + ke(*x*+2,*y*) + ke(*x*+2,*y*+1) + ke(*x*+2,*y*+2)) / 9 (Equation 4.4)

   The bit shift factor of encrypted image e(*x*, *y*) in (*x*, *y*) is determined by following equation 4.5

   ε=mod (*aver*, 8) (Equation 4.5)

3. **Modify bit:** The bit shift factor generated in previous step is used to shift the encrypted image pixel bit positions. Let s1 s2 s3 s4 s5 s6 s7 s8 represents 8 bits of encrypted image e(*x*, *y*) in the coordinate (*x*, *y*)

   Circular bit shift to the right, s1 s2 s3 s4 s5 s6 s7 *k*8 shift bit to the right, and then we obtain 8 new bits arrangement k1 k2 k3 k4 k5 k6 k7 k8

4. **Determine bits of encrypted image:** Each bit of the pixel value of decrypted image will be generated from corresponding bit of the pixel value of key-image and encrypted image.

   Let *k*1 *k*2 *k*3 *k*4 *k*5 *k*6 *k*7 *k*8 represents 8 bits generated after step 3 in the image e(*x*, *y*) at coordinate (*x*, *y*). It can be obtained from bits which the encrypted image and key image do bit exclusive-or according to the rule as follows:

   *k*1 is equal to the first bit obtained from e(*x*, *y*) ^ ke(*x*, *y*),

   *k*2 is equal to the second bit obtained from e(*x*, *y*) ^ ke(*x*, *y*+1),

   *k*3 is equal to the third bit obtained from e(*x*, *y*) ^ ke(*x*, *y*+2),

   *k*4 is equal to the fourth bit obtained from e(*x*, *y*) ^ ke(*x*+1, y),

   *k*5 is equal to the fifth bit obtained from e(*x*, *y*) ^ ke(*x*+1, *y*+2),

   *k*6 is equal to the sixth bit obtained from e(*x*, *y*) ^ ke(*x*+2, *y*),

   *k*7 is equal to the seventh bit obtained from e(*x*, *y*) ^ ke(*x*+2, *y*+1),

   *k*8 is equal to the eighth bit obtained from e(*x*, *y*) ^ ke(*x*+2, *y*+2).

   The symbol ^ represents the exclusive OR operation bit-by-bit

5. **Gray value Substitution:** The gray value of decrypted image *d*(*x*, *y*) is determined by following equation 4.6.

$$d(x,y) = \sum_{i=1}^{8} s_i 2^{8-i} \quad \text{(Equation 4.6)}$$

V. PHASE-3: P-FIBONACCI BASED IMAGE SCRAMBLING

Image scrambling is used to make images visually unrecognizable such that unauthorized users have difficulty decoding the scrambled image to access the original image. Image scrambling technologies are very useful tools to ensure image security by transforming the image into an unintelligible image. Scrambling makes the image unrecognizable to prevent eavesdroppers from decoding the true form or meaning of the image using the human visual system or a computer system.

Adding Image scrambling algorithm to this algorithm makes it stronger. Many Image Scrambling algorithms are exist, but based on Comparison of Recursive Sequence Based Image Scrambling Algorithms [2], Image scrambling based on P-Fibonacci code gives better results compared to other scrambling methods[3]. So in this encryption scheme we are using Fibonacci P-code based Image scrambling algorithm in this phase.

A parameter, p, is used as a security-key and has many possible choices to guarantee the high security of the scrambled images. The presented algorithms can be implemented for encoding/decoding both in full and partial image scrambling, and can be used in real-time applications, such as image data hiding and encryption. Examples of image scrambling are provided. Computer simulations are shown to demonstrate that the presented methods also have good performance in common image attacks such as cutting (data loss), compression and noise. The new scrambling methods can be implemented on grey level images and 3-color components in color images.

A. Fibonacci p-code and Transform

The Scrambling method is based on Fibonacci p-code and its transform which are defined below.

The Fibonacci p-code:

It is a sequence defined by

$$F_p(n) = \begin{cases} 0 & \\ 1 & n=1 \\ F(n-1)+F(n-p-1) & n>1 \end{cases} \quad \textit{(Equation 5.1)}$$

where p is a nonnegative integer.

From the definition above, Fibonacci p-code sequences will differ based on the p value.

Specially,

(1) Binary sequence: p=0, the sequence is powers of two, 1, 2, 4, 8, 16…;

(2) Classical Fibonacci sequence: p=1, the sequence is 1, 1, 2, 3, 5, 8, 13, 21…;

(3) For the large values of p the sequence starts with consecutive 1's and immediately after that 1, 2, 3, 4….p

P-Fibonacci Transform:

Let $F_p(n)$ and $F_p(n+1)$ are two consecutive Fibonacci p-code elements. The permutation $\{T_1, T_2, T_3, \ldots T_{Fp(n+1)}\}$ of an input sequence {1, 2, 3,..., $F_p(n+1)$-1} is called 1-D P-Fibonacci Transform if $\{T_1, T_2, T_3, \ldots T_{Fp(n+1)}\}$ is defined by

$$T_k = k\,[F_p(n) + i\,] \bmod F_p(n+1) \quad \text{(Equation 5.2)}$$

Where k = 0, 1, 2, 3,..., $F_p(n+1)$-1; I = −3,−2,−1,0,1, 2,3 ; $F_p(n) + i < F_p(n+1)$

For example, for a MxN grayscale image the data is a 2D matrix A,

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{11} & a_{12} & \ldots & a_{1N} \\ \ldots & \ldots & \ldots & \ldots \\ a_{M1} & a_{M2} & \ldots & a_{MN} \end{pmatrix} \quad \text{(Equation 5.3)}$$

The column coefficient matrix can be generated by using equation (5.2) based on different p values. There are N columns in this image. The input sequence is k = 1, 2, 3, ..., N , thus N= $F_p(n+1)$-1.

For a certain p value, the output sequence {T (N)} should be the permutation of the input sequence {1,2,3,..., N} which is defined by

$$T_{pN} = \{T_{p1}, T_{p2}, T_{p3}, \ldots\ldots T_{pN}\} \text{(Equation 5.4)}$$

From T_{pN}, the column coefficient matrix of 2-D P-Fibonacci Transform T_c(N, N) can be generated as

$$T_c(i,j) = \begin{cases} 1 & (T_{pi}, i) \\ 0 & otherwise \end{cases} \quad \text{(Equation 5.5)}$$

As the same procedure, there are M rows in the image data matrix. For the certain p value, the output sequence should be the permutation of the input sequence k = 1, 2, 3... M, after transformation by

$$T_{pM} = \{T_{p1}, T_{p2}, T_{p3}, \ldots\ldots T_{pM}\} \quad \text{(Equation 5.6)}$$

The row coefficient matrix of 2-D P-Fibonacci Transform T_r (N, N) can be generated as

Definition: Let B be the original image matrix, Tr, the row coefficient matrix, and Tc, the column coefficient matrix. The following matrix is called the 2-D P-Fibonacci Transform:

$$S = T_r B T_c \quad \text{(Equation 5.8)}$$

Where S is the scrambled image matrix with dimensions $M \times N$.

Definition: Let S be the scrambled image matrix, T_r the row coefficient matrix, and T_c the column coefficient matrix. The following matrix is called the 2-D Inverse Fibonacci Transform:

$$R = T_r^{-1} S T_c^{-1} \quad \text{(Equation 5.9)}$$

Where R is the reconstructed image matrix.

B. Image scrambling in the spatial domain

The block diagram of presented image scrambling algorithm in the spatial domain is shown in Figure 3. It is designed to change the image pixel position using the 2-D P-Fibonacci Transform. It can scramble both grey-scale images and color images.

Color images have three color components and the scrambling algorithm is applied to each color component individually. Grayscale images are treated as color images with one component. The presented algorithm is a lossless image scrambling method.

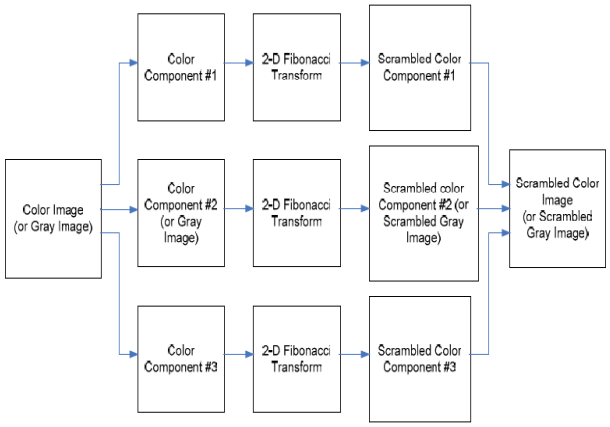


Figure 3: Block diagram of the image scrambling algorithm in the spatial domain

$$T_r(i,j) = \begin{cases} 1 & (i, T_{pi}) \\ 0 & otherwise \end{cases} \quad \text{(Equation 5.7)}$$

Algorithm for Image scrambling:
Input Color image (or Grayscale image) to be scrambled
Step 1: Choose key parameter, P, Calculate the row and column coefficient matrices of 2-D P-Fibonacci Transform.
Step 2: Separate the color image into three color components (for example R G B, Y Cb Cr, etc.). Each component is a 2D matrix (Grayscale image: skip this step.)
Step 3: Apply 2-D P-Fibonacci Transform to each color component to get the scrambled color components. (Grayscale image: Apply the 2-D P-Fibonacci Transform to the grayscale image to get the scrambled grayscale image.)
Step 4: Recombine the three scrambled color components to get the scrambled color image. (Grayscale image: Skip this step.)
Output: Scrambled color image (or scrambled grayscale image)

C. Image unscrambling algorithm in the spatial domain
For authorized users to reconstruct the original image, the following security keys are required: p and i. The inverse row and column coefficient matrices of the 2-D Inverse P-Fibonacci Transform can be generated based on these keys. The original image can be reconstructed by applying the 2-D Inverse P-Fibonacci Transform to the scrambled image.

Algorithm for Image Unscrambling:
Input: Scrambled color image (or scrambled grayscale image)
Step 1: Calculate the inverse row and column coefficient matrices of 2-D Inverse P-Fibonacci Transform using security keys: *p and i*.
Step 2: Separate the scrambled color image to three color components. (Grayscale image: Skip this step.)
Step 3: Apply the 2-D Inverse Fibonacci Transform to the each color component of the scrambled image separately. (Grayscale image: Apply the 2-D Inverse Fibonacci Transform directly to get the reconstructed image.)
Step 4: Recombined the three color components together to get the reconstructed color image. (Grayscale image: skip this step.)
Output: Reconstructed color image (or grayscale image)

VI. EXPERIMENTAL RESULTS AND COMPARISION

The algorithm is implemented using Matlab and Java. To verify the performance of the presented algorithms, we implemented the algorithm on several (color and grayscale) images and the results are presented in this section. This algorithm shows good performance when image is subjected to common image attacks.

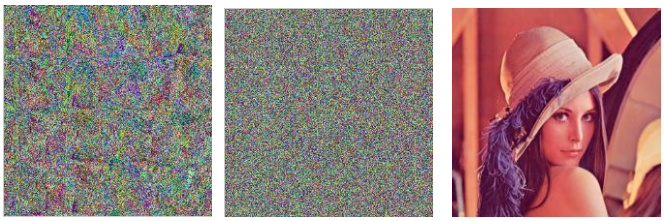
A. Encryption and Decryption of Color images
The algorithm accepts key-image and input image from the user and generates final encrypted image. Here the given input image is encrypted in two levels which make cryptanalysis difficult. The experimental result shows the output images from all 3 phases.

A color image is a digital image that includes color information for each pixel. Color images are also called RGB images. Each pixel is represented with combination of three values, for red, green and blue component. The each color components are accessed with 2D array. Complete image can be accessed by 3D array which includes three 2D arrays for each color component.

The presented encryption scheme processes 2D array at a time. In order to encrypt 3D color image, it is divided into three 2D arrays each for red, green and blue components. Each 2D component array will be encrypted/decrypted separately, finally the results be combined to form color image.

A color image shown Figure 6(a) of size 512 × 512 is encrypted by the presented algorithm using the key-image given in Figure 6(b) of size 128 × 128. The security keys for phase 3 is p=2 and i=-2. After completing phase-1 it gives the modified key-image shown in Figure 6(c) of size 514 × 514. The image after encryption in phase-2 is shown in Figure 6(d). The image after completing last phase is shown in Figure 6(e). The Decrypted Image is shown in Figure 6(f).

The histogram for Input image is shown in Figure 6(g), for Encrypted Image is shown in Figure 6(h), Histogram of deference between decrypted image and original input image is shown in Figure 6(i). The reconstructed image in Figure 6(f) is the same as the original image according to the histogram of the difference between them. The result also verifies that our presented algorithm is a lossless encryption scheme.



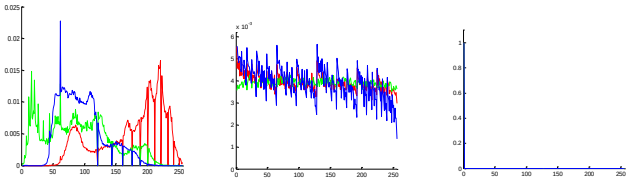


Figure 6: a) Input image of size 512 by 512; b) The key-image of size 128 by 128; c) The Output key-image from Phase-1 of size 514 by 514; d) Output Image of size 512 by 512 after completing phase-2; e) The Output Image of size 512 by 512 from phase-3 when p=2, i=-2; f) The image after complete Decryption process; g) Histogram of input Image shown in figure (a); h) Histogram of Encrypted Image shown figure (e); i) Histogram of Difference between original image shown in figure (a) and decrypted image shown in figure (f).

If the changed key-image shown in figure 7(a) is used in decryption process then the output image is completely different than expected as shown in figure 7(b).

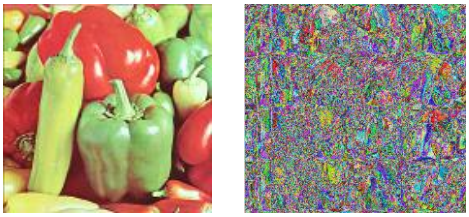


Figure 7: a) Changed key-image; b) Decrypted image when key-image changed

The changes in p value and i value makes significant difference in the decrypted image. The p value is used to generate random sequence using Fibonacci p-code. If the value of p is changed the random sequence generated by it also changes completely. The i value is used as offset to generate different random sequences each time for the same pairs of Fibonacci numbers. The change in i value will also affect the decrypted image. The output image from decryption process will be completely different when the values of either p or i changes with in image scrambling algorithm.

The output image is shown in Figure 8(a) after decryption when p value is changed from 2 to 3, and i value remains same i.e. i=-2. The output image is show in Figure 8(b) after decryption when i value is changed from -2 to -3 while p value remains same.

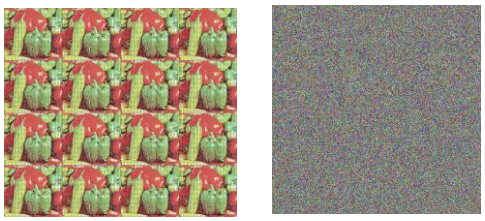


Figure 8: (a) Decrypted Image after changing p value from 2 to 3 while i value remains same; b) Decrypted Image after changing i value from -2 to -3 while p value remains same

B. Encryption and Decryption of grayscale Images and its Histograms

A grayscale (or graylevel) image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In fact a `gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image.

Grayscale input image is given in Figure 9(a), and grayscale key-image is shown in figure 9(b). The key-image generated after phase-1 is shown in figure 9(c). The given input image is encrypted and is shown in figure 9(e). The histograms of input image and encrypted images are shown sin figure 9(d) and 9(e) respectively. By observing the histogram of encrypted image, all the pixel values are randomly distributed and hence it is difficult for attacks.

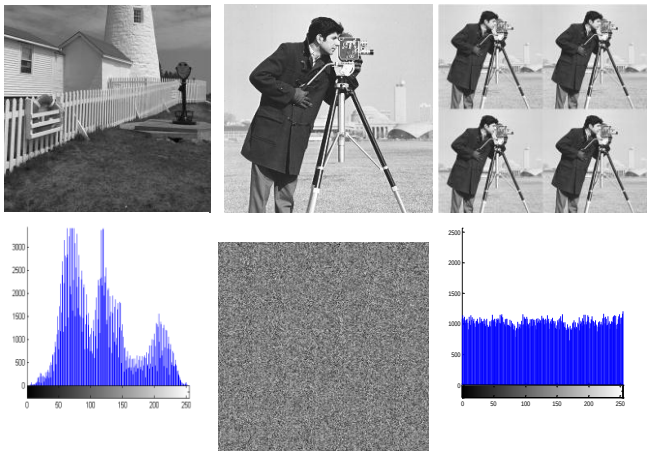


Figure 9: a) Grayscale input image of size 512 by 512; b) The initial Key-Image of size 256 by 256; c) The Final key-Image of size 512 by 512; d) The Histogram of the input image shown in figure (a); e) The Encrypted Image after completing (p = 2, I = - 2) all the phases; f) The Histogram of Encrypted Image is shown if figure (e).

VII. FUTURE WORK

The proposed image encryption process can be combined with image compression so that it can be transferred over the network with less transmission delay. Different existing image compression methods can be applied in the process of encryption.

REFERENCES

[1] Y. Zhou, K. Panetta and S. Agaian, "Image encryption using binary key-images," 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, 2009, pp. 4569-4574. doi: 10.1109/ICSMC.2009.5346780

[2] Y. Zhou, K. Panetta and S. Agaian, "Comparison of recursive sequence based image scrambling algorithms," 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, 2008, pp. 697-701. doi:10.1109/ICSMC.2008.4811359

[3] Yicong Zhou, Sos Agaian, Valencia M. Joyner, Karen Panetta, "Two Fibonacci P-code based image scrambling algorithms", Proc. SPIE 6812, Image Processing: Algorithms and Systems VI, 681215 (3 March 2008); doi:10.1117/12.766591

[4] Jiancheng Zou, R. K. Ward and Dongxu Qi, "A new digital image scrambling method based on Fibonacci numbers," 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512), 2004, pp.III-965-8Vol.3. doi:10.1109/ISCAS.2004.1328909

[5] K. C. Iyer and A. Subramanya, "Image Encryption by Pixel Property Separation," Cryptology ePrint Archive, 2009.

[6] M. Yang, N. Bourbakis, and S. Li, "Data-image-video encryption," Potentials, IEEE, vol. 23, no. 3, pp. 28-34,2004.

[7] J. Zou, R. K. Ward, and D. Qi, "A new digital image Scrambling method based on Fibonacci numbers," in Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on, 2004 pp. III-965-8 Vol.3.

[8] A. Sharma, RS Thakur, S. Jaloree, "Investigation of Efficient Cryptic Algorithm for image files Encryption in Cloud", International Journal of Scientific Research in Computer Science and Engineering, Vol.4, Issue.5, pp.5-11, 2016.

[9] Rashmi P., Bharathi R.K., Shruthi Prabhakar, Reshma Banu, Rachana C.R., "Performance Analysis of Self Adaptive Image Encryption Technique", International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.3, pp.44-58, 2017.

[10] Kodge B. G., "Information Security: A Review on Steganography with Cryptography for Secured Data Transaction", International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.6, pp.1-4, 2017.

Authors Profile

Mr. V. Sridhar pursued Bachelor of Technology from Jawaharlal Nehru Technilogical University , Hyderabad in 2004 and Master of Tecnology from Osmania University in the year 2010. He is currently pursuing Ph.D from Osmania University and currently working as Assistant Professor in Department of Computer Science & Engineering, MVSR Engineering College, Nadergul, Hyderabad, India since 2004. He is a Life member of ISTE Chapter since 2014. He qualified in UGC-NET conducted in June, 2012. He taught various subjects as part of teaching curriculum which includes Programming in C & C++, Object Oriented Programming through JAVA, Database Management Systems, Distributed Systems, Web Programming & Servies etc. He has more than 13 year of teaching Experience. His research work focuses on Cloud security, Information processing, Information retrieval etc.

Mrs. M. Dyna pursed Bachelor of Technology in 2004 and Master of Tecnology in 2012 from Jawaharlal Nehru Technological University , Hyderabad. She is currently working as Assistant Professor in Department of Computer Science & Engineering, MVSR Engineering College, Nadergul, Hyderabad, India since 2007. She is a Life member of ISTE Chapter since 2014. She taught various subjects as part of teaching which includes Programming in C & C++,t Discrete Mathematics, Software Engineering, Embedded Systems, Web Programming & Servies etc. She has more than 10 year of teaching Experience. Her research work focuses on Software Engineering Paradigms, Big Data Analytics, Maching Learning etc.

