Chapter 7

Image Encryption Building Blocks

7.1. Overview

Image encryption deals with the distribution of image into other form or by hiding it into another object. Image encryption is carried out based on different techniques like key based encryption, digital enveloping, steganographic scheme, etc. Some of the existing image encryption techniques are discussed in section 2.6 of chapter 2. The idea of using same numbered pixel from the original image for embedding into same numbered pixel of envelope image, Using of single level encryption and finally applying of XOR operation may reduce the security for image encryption. So the implementation of multiple levels of image encryption, the mapping between different numbered pixels in original image and envelope image and development and use of new Bitwise Masking for Alternate Sequence operation (BWMAS) are carried out for image encryption block. The superiority of the implemented schemes is accessed in respect of other standard algorithms.

Three image encryption schemes are discussed in the current chapter. They are Even Odd block based Digital Enveloping scheme (EODE)[1,2], Cumulative Image encryption scheme using Digital enveloping, Key based encryption with image Partitioning (CIDKP)[3,4] and Cumulative Image encryption approach using Steganographic scheme with Pixel repositioning (CISP)[5]. Satisfactory performances of the implemented schemes are accessed in respect of PSNR, SSIM, BER and other standard parameter values.

In this chapter, Even Odd block based Digital Enveloping scheme (EODE) in section 7.2, Cumulative Image encryption scheme using Digital enveloping, Key based encryption with image Partitioning (CIDKP) in section 7.3, Cumulative Image encryption approach using Steganographic scheme with Pixel repositioning (CISP) in section 7.4 have been discussed. Conclusion is drawn in section 7.5.

7.2. Even Odd block based Digital Enveloping scheme (EODE)

Detection of destination pixel in envelope image corresponding to source pixels in the original image is easier for the traditional digital enveloping scheme as the embedding of original image's pixels and envelope image's pixels are carried out continuously. A new digital enveloping scheme is implemented to solve this issue where the mapping between source pixel and destination pixels is carried out based on their even or odd placement sequence. In EODE[1,2] method, Pixels of the original image are assigned with a sequence number and pixels of the envelope image are divided into blocks. Odd numbered pixels from the original image are mapped into even numbered blocks of envelope image and vice versa for encryption. Two least significant bits of each block of each envelope image's pixels are replaced by the bits of original image's pixels. Decryption is carried out by taking envelope image pixel's bit value corresponding to original image pixels. As the mapping between the bits of original image pixels and envelope image pixels are carried out even and odd block wise rather than in a continuous manner, so the security is increased to a great extent. Figure 7.1 represents the overall procedure for Even Odd block based Digital Enveloping scheme (EODE).

---

[1] **Published in International Journal of Computer sciences and Engineering (ICSE)**, **UGC approved journal,** Volume 6, Issue 5, pp. 170 – 177, DOI: https://doi.org/10.26438/ijcse/v6i5.170177, with title Cumulative Image Encryption Approach

[2] **Published in International Journal of Innovative Technology & Adaptive Management (IJITAM),** Volume 1, Issue 7, pp.15-20, with title An approach of Visual Cryptography Scheme for Color Image by using Even and odd block based digital enveloping

Calculate width (w) and height (h) of original image and width (w1) and height (h1) of envelope image where envelope image is at least 4 times larger than the original image. Convert the envelope image into (w1*h1)/4 numbers of blocks.

↓

Read one pixel from the original image and convert into 32 bits binary representation. If the pixel is odd one then plot it into next even block of envelope image and vice versa.

↓

Two least significant bits of each group (Red, Green, Blue and Alpha) of envelope image pixel are replaced by bits of original image's pixel serially at the time of encryption and construct the enveloped image.

↓

Convert envelope image into (w1*h1)/4 numbers of blocks (where w1 represents width and h1 represents height of envelope image). Read two least significant bits of each group of each pixel in the referred block of the envelope image corresponding to the original image's pixel and construct a decrypted image.

Figure 7.1: Overall procedure for Even Odd block based Digital Enveloping scheme (EODE)

The mapping between the pixels of the original image and blocks of envelope image is represented in Figure 7.2.

Pixels of original image   Blocks of envelope image

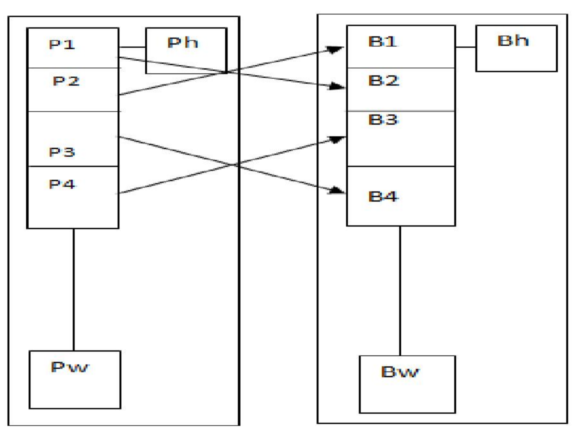


Figure 7.2: Mapping between Pixels of Original Image and Blocks of Envelope Image

P1, P2…PN and B1, B2…BN represent the pixels of original image and blocks of envelope image respectively where Ph, Pw are height and width of original image and Bh, Bw is height and width of envelope image respectively and Bh*Bw=Ph*Pw. The mapping between the bits of the original image's pixels and blocks of envelope image is represented in Figure 7.3.

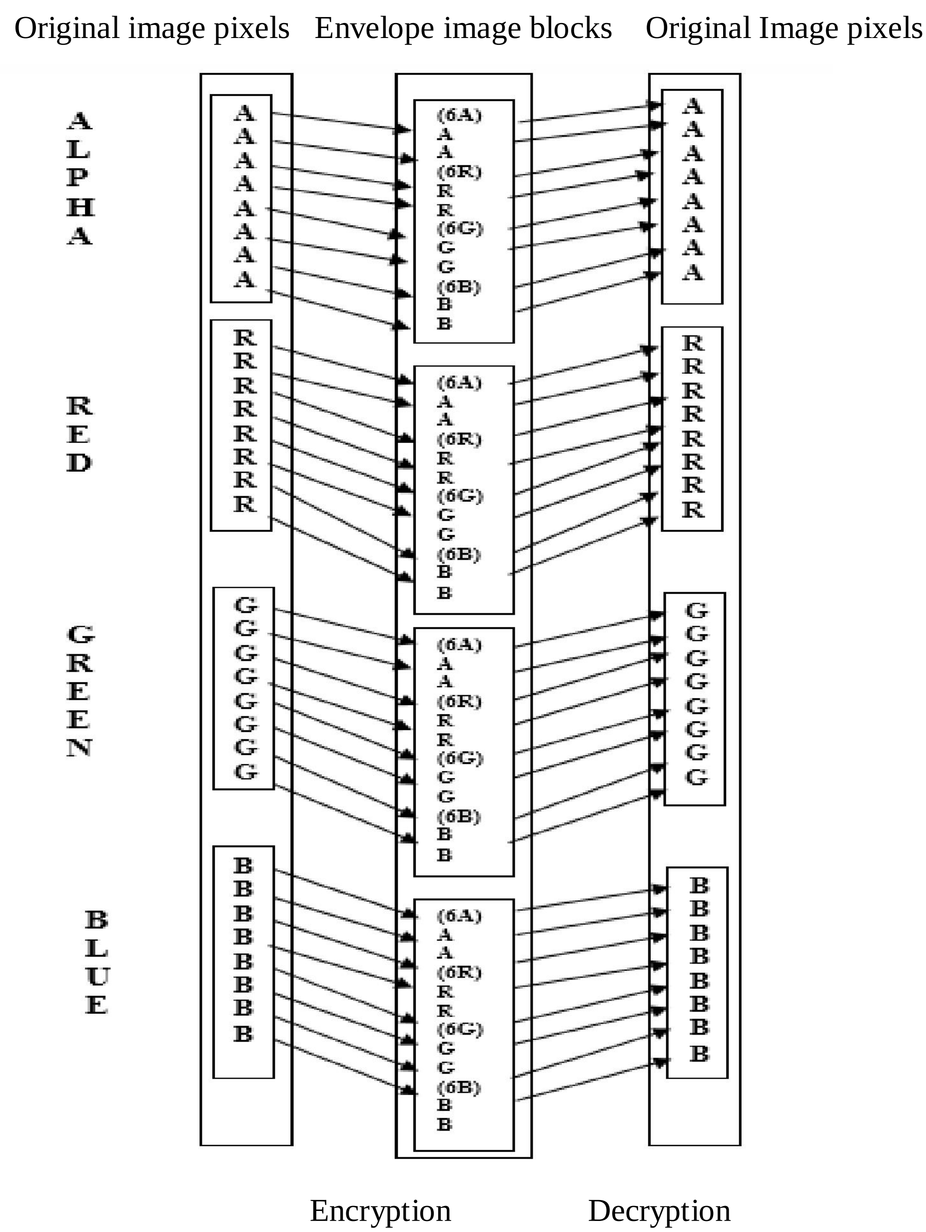


Figure 7.3: Mapping between the bits of Original Image's Pixels and Blocks of Envelope Image

Section 7.2.1 and section 7.2.2 represents the encryption and decryption process respectively. Section 7.2.3 shows the experiment results. Security analysis of even odd block based enveloping scheme are represented in section 7.2.4

7.2.1. Encryption Process

Step 1: Original and envelope image is taken as input. Width (w) and height (h) of original image and width (w1) and height (h1) of envelope image are calculated where [w1*h1>= 4(w*h)].

Step 2: Envelope image is converted into w*h number of blocks where w and h are the width and height of the original image.

Step 3: Pixels of original image and block of envelop image are numbered in a sequential manner. Read and convert one pixel from the original image into 32-bit binary representation. If that pixel is odd numbered pixel, plot the pixel into next even block of envelope image and vice versa. Carry out the same operation for all the pixels of the original image.

Step 4: Continuous bits of the original image's pixels are replacing two least significant bits of each block (alpha, red, green, blue) of the envelope image pixel. Four pixels from envelope image are needed to represent one pixel of the original image as one 8 bit block of the original image pixel is represented by one pixel of envelope image. Size of each block of envelope mage is 4*32=128 bits. At the time of encryption, odd pixels from the original image are mapped with the even block of envelope image and vice versa.

Step 5: Envelope image is generated with the modified pixels and shared to the receiver.

7.2.2. Decryption Process

Step 1: Envelope image is converted into (w1*h1)/4 numbers of blocks where w1 and h1 are the width and height of the envelope image.

Step 2: Two least significant bits from each block of each pixel in the referred envelope image block corresponding to original image pixel are fetched. The decrypted image is constructed.

7.2.3. Experiment Results and Discussions

A. Encryption Process

Enter the name of original image-Twinparrot.jpg

Figure 7.4 shows the original image.

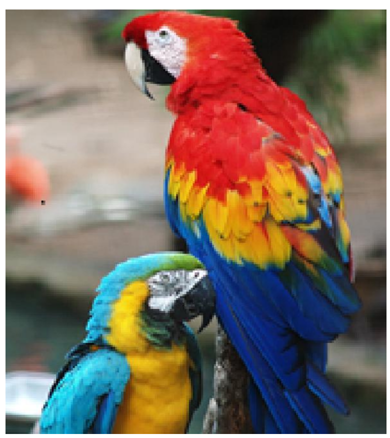


Figure 7.4: Original Image (Twinparrot .jpg)

Size of original image is 200*150 pixels.

Enter the name of envelope image- Greenscenery.jpg

Figure 7.5 shows the envelope Image

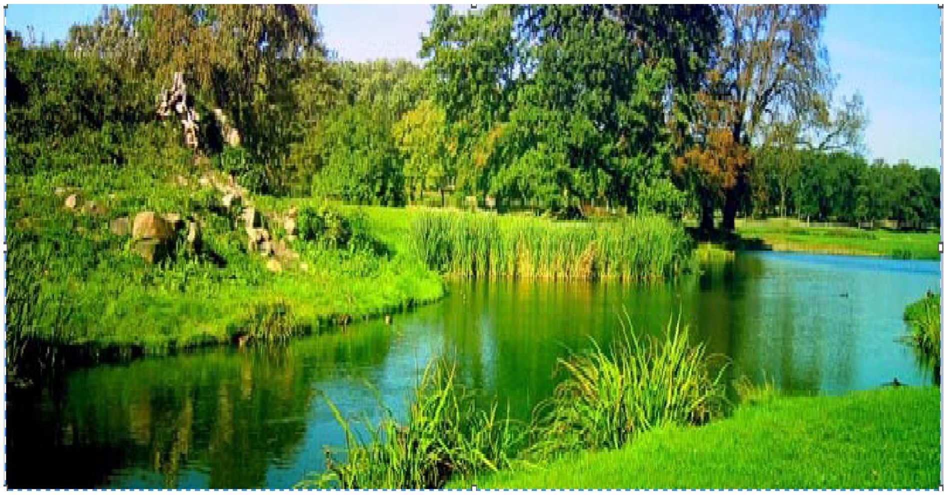


Figure 7.5: Envelope Image (Greenscenery.jpg)

Size of envelope image is 800*600 pixels.

Generated envelope image after embedding the original image is Egreenscenery.jpg which is shown in Figure 7.6.

Figure 7.6: Envelope Image after Embedding (Egreenscenery.jpg)

Size of envelope image after embedding the original image is 800*600 pixels.

B. Decryption Process

Enter the name of envelope image- Egreenscenery.jpg

Figure 7.7 shows the envelope Image.

Figure 7.7: Input Image for Decryption (Egreenscenery.jpg)

Size of envelope image is 800*600 pixels.

Original image generated after de-enveloping is dtwinparrot.jpg which is shown in Figure 7.8.

Figure 7.8: Original Image after De-Enveloping (dtwinparrot.jpg)

Size of the original image is 200*150 pixels.

7.2.4. Security Analysis for Even Odd block based Digital Enveloping scheme (EODE)

Standard image quality measurement parameters are used to determine the performance of implemented image encryption schemes where Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Metric (SSIM), Universal Image Quality Index (Q-Index), Bit Error Rate (BER), Correlation Coefficient (CC) and Normalized Cross-Correlation (NCC) are calculated as per the equations 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9 respectively mentioned in chapter 3.

Table 7.1 represents the values of PSNR, MSE and BER calculated from the output images generated by the EODE scheme where PSNR is quite high and MSE and BER are also satisfactory.

Table 7.1: Representation of PSNR, MSE and BER values generated from Outputs of EODE Scheme

Original Envelope Image Name	**Image size**	**Embedded Envelope Image Name**	**Image size**	**PSNR (dB)**	**MSE**	**BER**
b_img1.png	335 kb	a_img1.png	455 kb	56.9883	0.1301	0.0026036458333333332
b_img2.jpg	313 kb	a_img2.jpg	418 kb	56.8563	0.1341	0.002580815972222222
b_img3.png	114 kb	a_img3.png	146 kb	56.6661	0.1401	0.0025668402777777777
b_img4.jpg	61 kb	a_img4.jpg	66 kb	47.8873	1.0577	0.02076875
b_img5.png	189 kb	a_img5.png	265 kb	56.4871	0.146	0.0025750868055555557
b_img6.jpg	287 kb	a_img6.jpg	446 kb	57.4712	0.1164	0.0020383076043453403

b_img7.png	78 kb	a_img7.png	130 kb	51.9736	0.4128	0.008265761155462232
b_img8.jpg	102 kb	a_img8.jpg	150 kb	51.974	0.4127	0.008271512943644092
b_img9.png	71 kb	a_img9.png	118 kb	52.0428	0.4063	0.008236846445653966
b_img10.jpg	88 kb	a_img10.jpg	135 kb	51.8368	0.426	0.008230667329027984

Figure 7.9 graphically shows the relationship between PSNR values and image size.

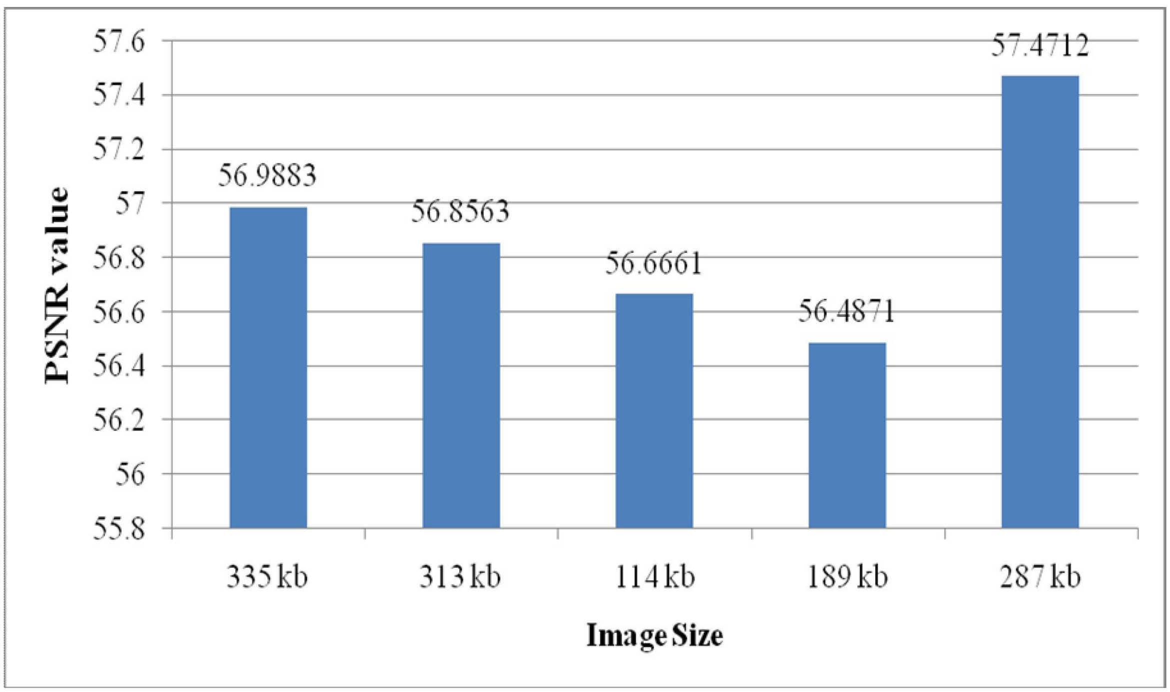


Figure 7.9: Relationship between the PSNR Values and inputted Image Size

Table 7.2 represents the values of SSIM, NCC and Q-INDEX calculated from the output images generated by EODE scheme where all the SSIM, NCC and Q-INDEX are in standard range.

Table 7.2: Representation of SSIM, NCC and Q-INDEX values generated from Outputs of EODE Scheme

Original Envelope Image Name	**Image size**	**Embedded Envelope Image Name**	**Image size**	**SSIM**	**NCC**	**Q-INDEX**
b_img1.png	335 kb	a_img1.png	455 kb	99.8445	0.9999962013165851	0.9999880631622724
b_img2.jpg	313 kb	a_img2.jpg	418 kb	99.8808	0.9999966450051264	0.9999931169160952
b_img3.png	114 kb	a_img3.png	146 kb	99.7278	0.9999966205212796	0.999959350553334
b_img4.jpg	61 kb	a_img4.jpg	66 kb	99.6757	0.9999458966741197	0.99979771178402
b_img5.png	189 kb	a_img5.png	265 kb	99.8242	0.9999926260699378	0.9999650494186986
b_img6.jpg	287 kb	a_img6.jpg	446 kb	99.8888	0.9999921977557237	0.9999869927464404
b_img7.png	78 kb	a_img7.png	130 kb	99.6293	0.9999823308550397	0.9999641394290916
b_img8.jpg	102 kb	a_img8.jpg	150 kb	99.8168	0.9999862486744727	0.9999374330784654
b_img9.png	71 kb	a_img9.png	118 kb	99.7373	0.9999805832705414	0.9999253127373733
b_img10.jpg	88 kb	a_img10.jpg	135 kb	99.7740	0.9999825241799358	0.9999604076908543

Figure 7.10 graphically shows the relationship between the SSIM values and image size.

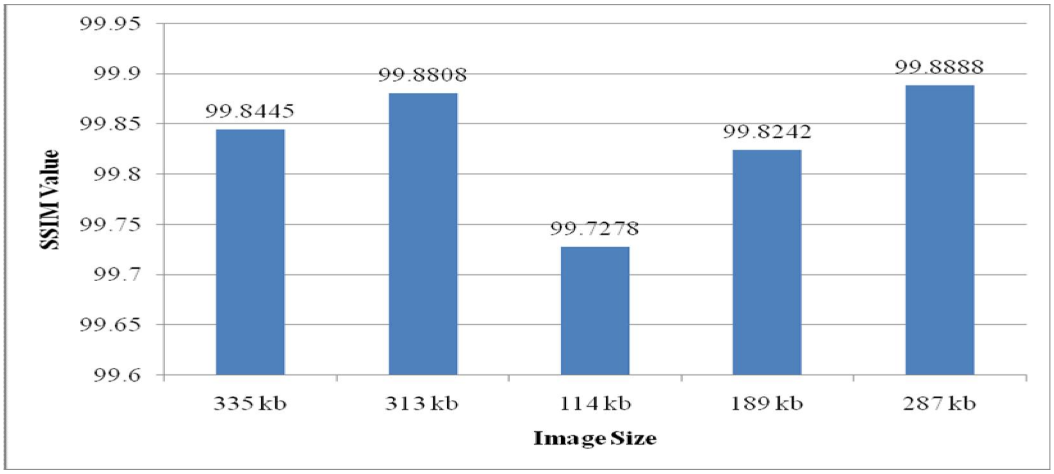


Figure 7.10: Relationship between SSIM Values with inputted Image Size

Table 7.3 represents satisfactory range of values of standard deviation and correlation coefficient calculated from the output images generated by the EODE scheme.

Table 7.3: Representation of Standard Deviation and Correlation Coefficient values generated from Outputs of EODE Scheme

Original Envelope Image Name	Image size (kb)	Embedded Envelope Image Name	Image size (kb)	Standard Deviation of Original Envelope	Standard Deviation of Embedded Envelope	Correlation Coefficient
b_img1.png	335	a_img1.png	455	188.36	188.3683	1.0
b_img2.jpg	313	a_img2.jpg	418	256.2036	256.1875	1.0
b_img3.png	114	a_img3.png	146	108.8819	108.868	1.0
b_img4.jpg	61	a_img4.jpg	66	130.1225	130.0299	0.9998
b_img5.png	189	a_img5.png	265	130.4924	130.5253	1.0
b_img6.jpg	287	a_img6.jpg	446	188.2066	188.2212	1.0
b_img7.png	78	a_img7.png	130	189.7728	189.8233	1.0
b_img8.jpg	102	a_img8.jpg	150	140.3648	140.4211	0.9999
b_img9.png	71	a_img9.png	118	130.4722	130.5188	0.9999
b_img10.jpg	88	a_img10.jpg	135	188.8209	188.8371	1.0

Figure 7.11 graphically represents the satisfactory correlation coefficient values generated from the output images of the EODE scheme.

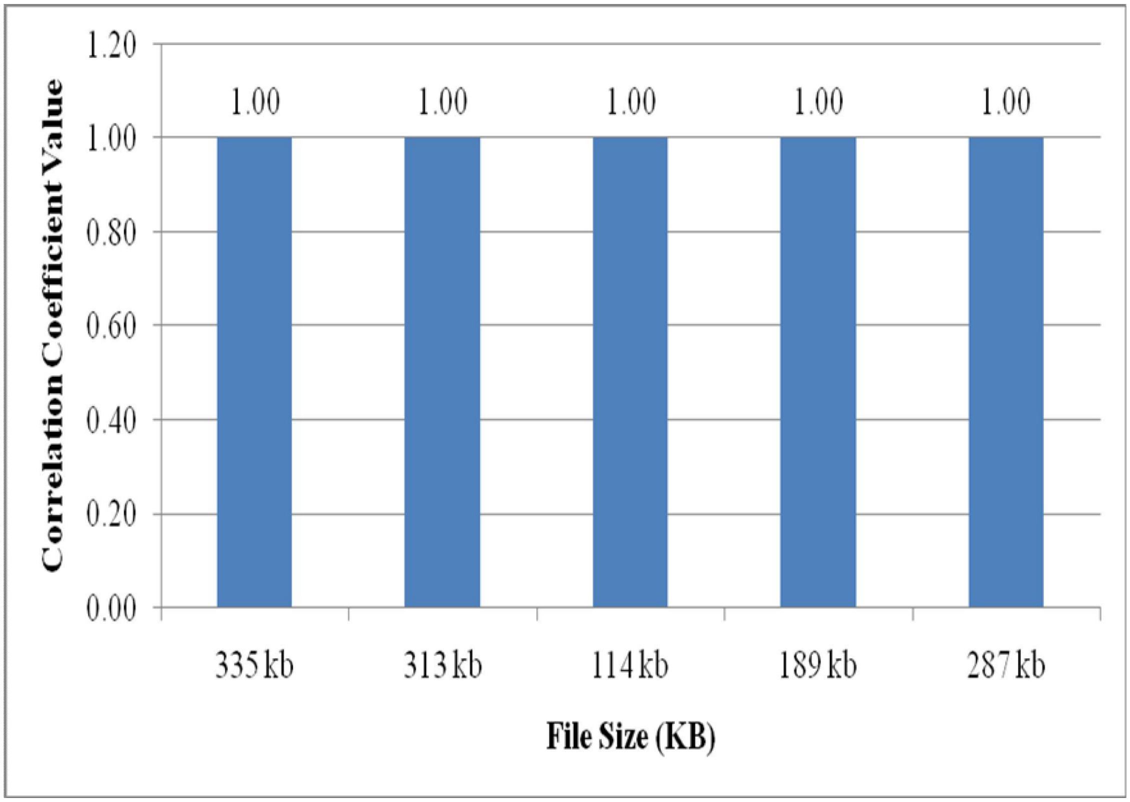


Figure 7.11: Graphical Representation of Correlation Coefficient Values vs Image Size for EODE Scheme

Table 7.4 represents the encryption and decryption time of different files for the EODE scheme where the encryption is carried out using a computer with Core2 Duo 2.20 GHz processor and 1.00 GB RAM.

Table 7.4: Representation of Encryption and Decryption Time of Different Files for EODE Scheme

Original Envelope Image Name	Image size	Embedded Envelope Image Name	Image size	Encryption Time (Milliseconds)	Decryption Time (Milliseconds)
b_img1.png	335 kb	a_img1.png	455 kb	32087	32032
b_img2.jpg	313 kb	a_img2.jpg	418 kb	30780	30729
b_img3.png	114 kb	a_img3.png	146 kb	18721	18693
b_img4.jpg	61 kb	a_img4.jpg	66 kb	12807	12773
b_img5.png	189 kb	a_img5.png	265 kb	16620	16567
b_img6.jpg	287 kb	a_img6.jpg	446 kb	49721	49685
b_img7.png	78 kb	a_img7.png	130 kb	18480	18434
b_img8.jpg	102 kb	a_img8.jpg	150 kb	24113	24079
b_img9.png	71 kb	a_img9.png	118 kb	11815	11765
b_img10.jpg	88 kb	a_img10.jpg	135 kb	18443	18411

Figure 7.12 graphically represents the encryption time of different files for EODE scheme where it is shown that encryption time is independent of size of the inputted file.

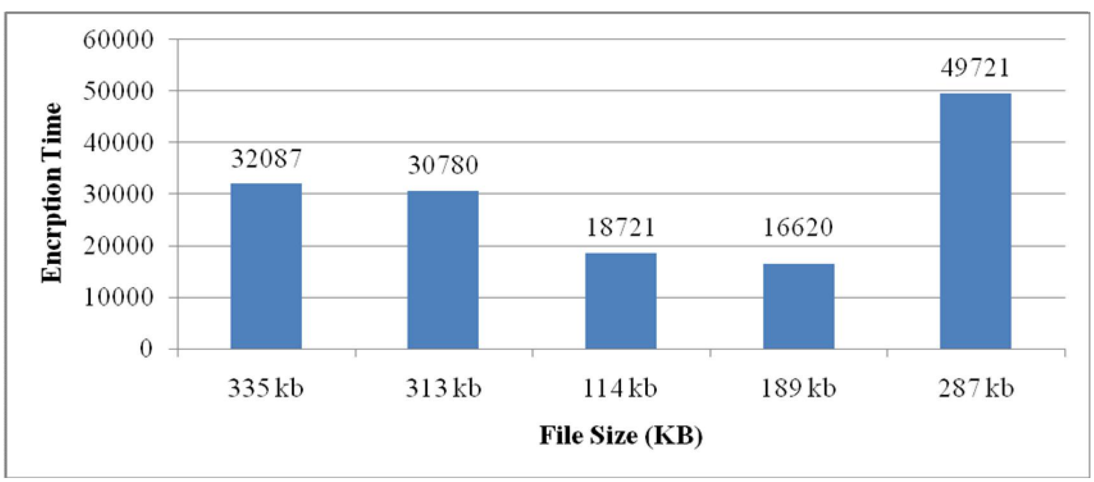


Figure 7.12: Graphical Representation of Encryption Time vs Image Size for EODE Scheme

7.3. Cumulative Image encryption scheme using Digital enveloping, Key based encryption with image Partitioning (CIDKP)

A single level of image encryption is not capable to provide a satisfactory level of security so in CIDKP[3,4] scheme encryption of image is carried out in multiple levels. First, the inputted image is divided into user defined N numbers of pieces where N is a positive integer. Then the resultant image pieces are encrypted by using user defined variable length text keys and unique biometric fingerprint image keys separately and in a cumulative manner. Resultant encrypted image pieces are embedded into user defined envelope image separately using digital enveloping method. Envelope images containing the information of original images are distributed to the receiver through the public network. Implementation of multiple levels of encryptions (image partitioning, Text key based encryption, image key based encryption and digital enveloping), using of user unique biometric fingerprint image as image key and partitioning of the original image into multiple pieces based on user inputs increase the security in great extent. Figure 7.13 represents the overall procedure for Cumulative Image encryption scheme using Digital enveloping, Key based encryption with image Partitioning (CIDKP).

[3] **Presented in International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017)**, **IEEE Xplore,** pp. 130-135, DOI: 10.1109/ ICPCSI. 2017. 8391813, with title Cumulative Image Encryption Approach based on user Defined Operation, Character Repositioning, Text Key and Image Key Encryption Technique and Secret Sharing Scheme

[4] **Published in International Journal of Engineering Research & Technology (IJERT)**, Volume 2, Issue 5, pp. 1341-1349, with title An Approach of Visual Cryptography Scheme for Color Image by Cumulative Encryption using Image Partitioning, Text Key Encryption, Image Key Encryption & Digital Enveloping

The source image is divided into N number of equal pieces by applying image partitioning scheme where N is user defined positive integer.

↓

Combination of characters, special symbols and numbers are considered as text key and unique fingerprint image of the user is considered as image key. Each piece of images is encrypted by using text key and image key encryption method cumulatively.

↓

Finally, all encrypted image pieces are embedded with N numbers of user inputted envelope image separately where N is user defined positive integer. Thus encryption is carried out and all envelope images are shared to the receiver.

↓

For decryption, all envelope images are fetched and follow the reverse process. Thus generate the original image.

Figure 7.13: Overall Procedure for Cumulative Image encryption scheme using Digital enveloping, Key based encryption with image Partitioning (CIDKP)

Figure 7.14 shows the diagrammatic representation of detail encryption and decryption procedure of CIDKP scheme.

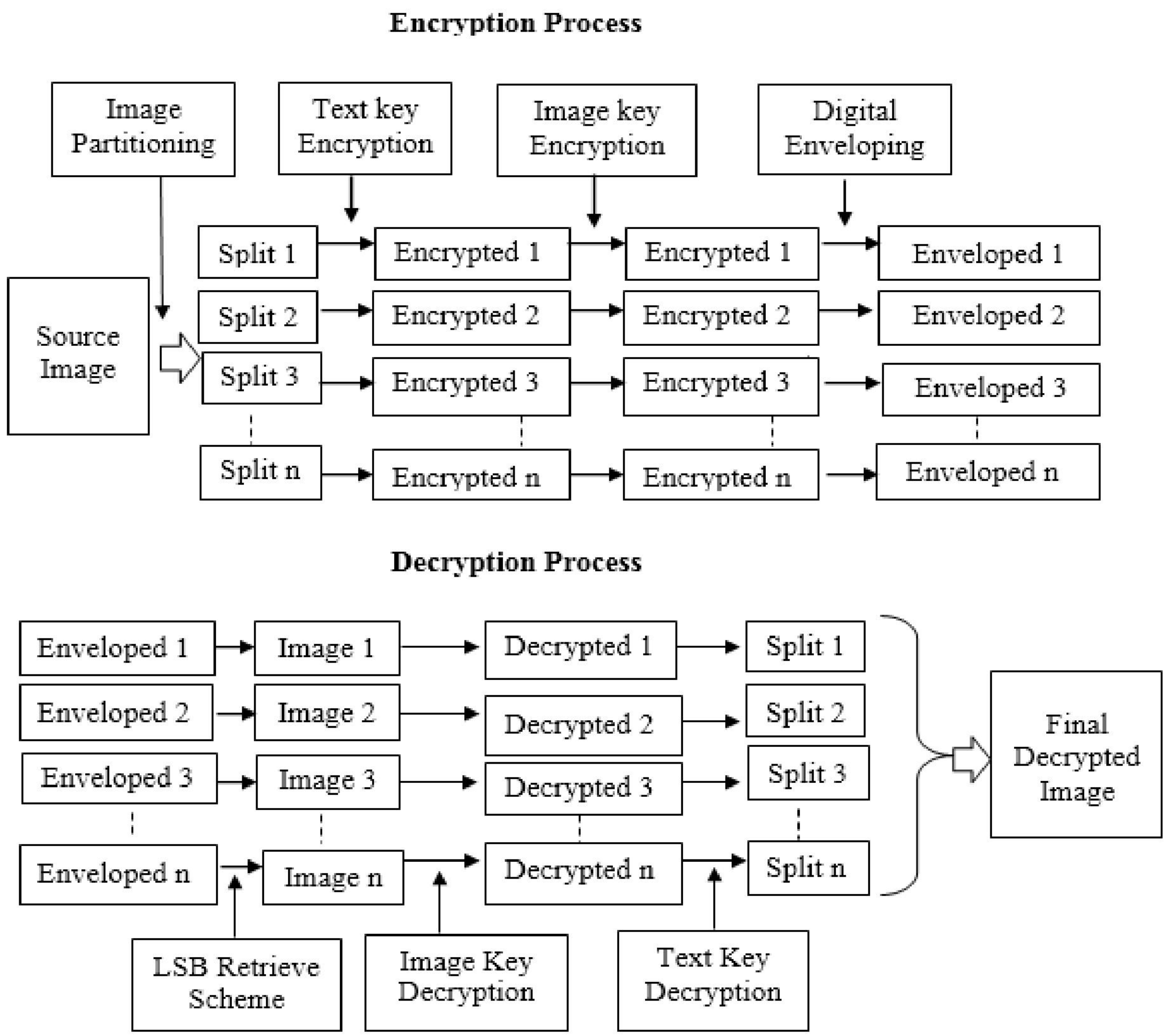


Figure 7.14: Representation of Encryption and Decryption Procedures for CIDKP Scheme.

Section 7.3.1 represents the encryption process. Section 7.3.2 shows the decryption process. Experiment results and security analysis of CIDKP scheme are described in section 7.3.3 and section 7.3.4 respectively.

7.3.1. Encryption process

The encryption process is carried out by executing image partitioning, text key encryption, image key encryption and digital enveloping scheme sequentially.

A. Algorithm for Image Partitioning for CIDKP Scheme

Step 1: Calculate the width and height of the original image. Determine the width and height of split image pieces depending upon the number of image pieces that the original image has to be divided.

Step 2: An array is initialized with the pixels information of all the image pieces. Construct the new images (mini image) with the stored pixel information. Write the information about those mini images into new image files.

B. Algorithm for Text Key Encryption

Step 1: Calculate width (W) and height (H) of the original image.

Step 2: An array named BP[] with size W*H*32 is used to store binary values of each pixel of the original image in the following manner.

For p = 0 to (W*H-1)

{Each pixel value from original image is scanned and converted into 32 bit binary representation

BP [p*32+j] = PIXE.charAt(j)// where PIXE[] array holding the pixel information}

Step 3: Calculate the length (L) of inputted text key which contains characters, special symbols, and numbers. Convert each element of that key into 8-bit binary representation and store those value into an array called K[] in the following manner.

For P = 0 to ((L*8)-1) {K[p] = CK.charAt(j) //where CK[] array holding the key information}.

Step 4: Perform XOR operation between array BP[] and array K[] in the following manner.

For p = 0 to ((W*H*32)/(L*8) -1) {

For j = 0 to (L*8-1){

BP [p*L*8+j] = BP [p*L*8+j] ^ K[j]}}

Construct the encrypted image from array BP[]. Perform same procedure for encrypting all image pieces applying separate text keys.

C. Algorithm for Image key encryption

Step 1: Calculate width (iw) and height (ih) of the original image which is supplied by user.

Step 2: An array IBP[] of size iw*ih*32 is applied for storing each pixel binary values of the original image in the following manner.

For p1 = 0 to (iw*ih-1)

{Scan each pixel value from original image & convert the pixels into 32 bit binary representation

IBP [p1*32+j] = IPIXE.charAt(j) // where IPIXE[] array contains pixel information}

Step 3: Calculate width (kw) and height (kh) of the inputted user fingerprint image used as an image key.

An array KBP[] with size kw*kh*24 is used to store each pixel binary values of image key in the following manner.

For p2 = 0 to (kw*kh-1)

{Scan each pixel value of image key & convert it into 24-bit binary representation as only RGB part is considered.

For k1=0 to 23 {

KBP [p2*24+k1] = KPIXE.charAt(k1) // where KPIXE[] array keeps pixel information}}

Step 4: Perform XOR operation between array IBP[] and array KBP[] in the following manner.

For m = 0 to ((iw*ih*32)/(kw*kh*24) -1) {

For n = 0 to (kw*kh*24-1) {

IBP [m*kw*kh*24+n] = IBP [m*kw*kh*24+n] ^ KBP[n]}}
Construct the encrypted image from array IBP[]. Carry out the same procedure for encrypting all image pieces using separate fingerprint image.

D. Algorithm for Digital Enveloping

Step 1: Take input of original image and envelope image from the user and calculate their width and height where ow and oh are the width and height of the original image and ew and eh are the width and height of envelope image respectively and ew*eh= 4*ow*oh. Convert the original image and envelop image into 32-bit binary representation and store the values into array called O[] with a size of ow*oh*32 and E[] with a size of ew*eh*32 respectively.

Step 2: Bits of the original image are embedded into envelope image in the following manner.
For u = 0 to 4*ow*oh*32 −1
{Replace the bits of envelope image with the bits of the original image for 6th, 7th, 14th, 15th, 22th, 23th, 30th, 31st bit position for each pixel}
Store the value of modified pixels into an array named CI[] with a size of ew*eh*32. Construct the final envelope image.

7.3.2. Decryption Process

Step 1: After receiving of all the envelope images at the receiver end de-enveloping process is carried out to extract the images hidden within envelope images by following the algorithm of section 7.3.1.D. All the resultant images are passed through image key decryption and text key decryption procedures cumulatively by following the algorithms of section 7.3.1.C and 7.3.1.B respectively. Thus generate all the image pieces.

Step 2: All the image pieces are combined together with their proper placement and generate the decrypted image.

7.3.3. Implementations with Experimental Results

A. Encryption Process

Output for Image Partitioning Procedure

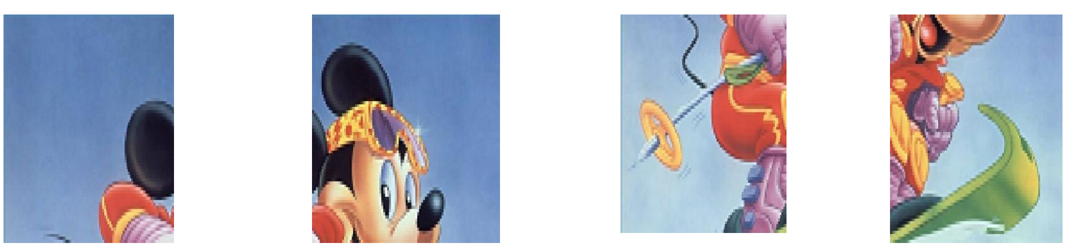
Name of the original image: micki.png

Figure 7.15 and Figure 7.16 represent the original image and the image pieces generated after image portioning method respectively.

Figure 7.15: Original Image

Numbers of pieces have to be generated from the original image are: 4

Image pieces generated after applying the image portioning method are:



imgp0.png imgp1.png **i**mgp2.png imgp3.png

Figure 7.16: Generation of Image Pieces from Image Portioning Procedure

Output of Text Key Encryption Procedure

Figure 7.17 represents the output of the text key encryption process.

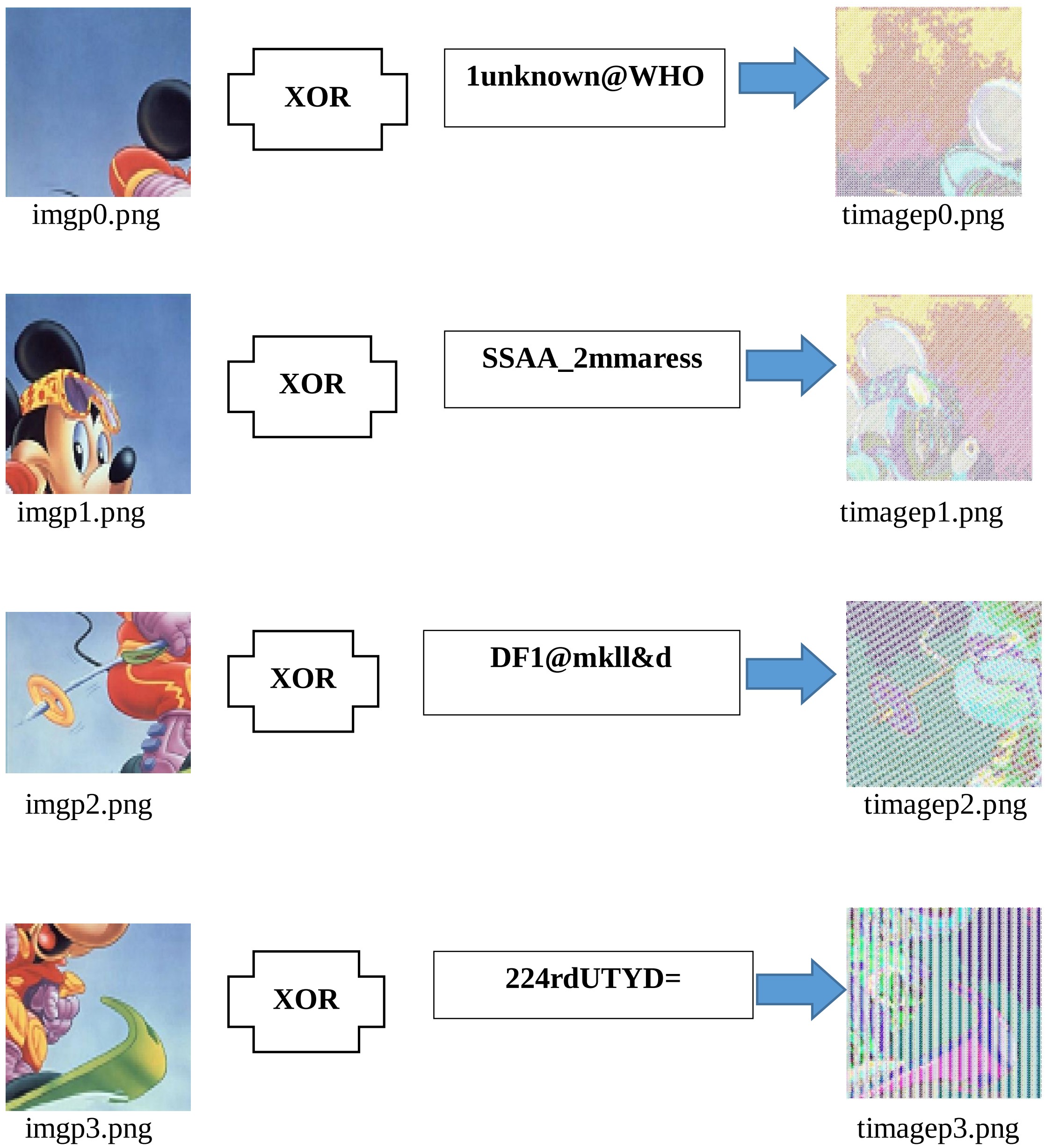


Figure 7.17: Representation of Output of Text Key Encryption Procedure

Output of Image Key Encryption Procedure

Figure 7.18 represents the output of the image key encryption process.

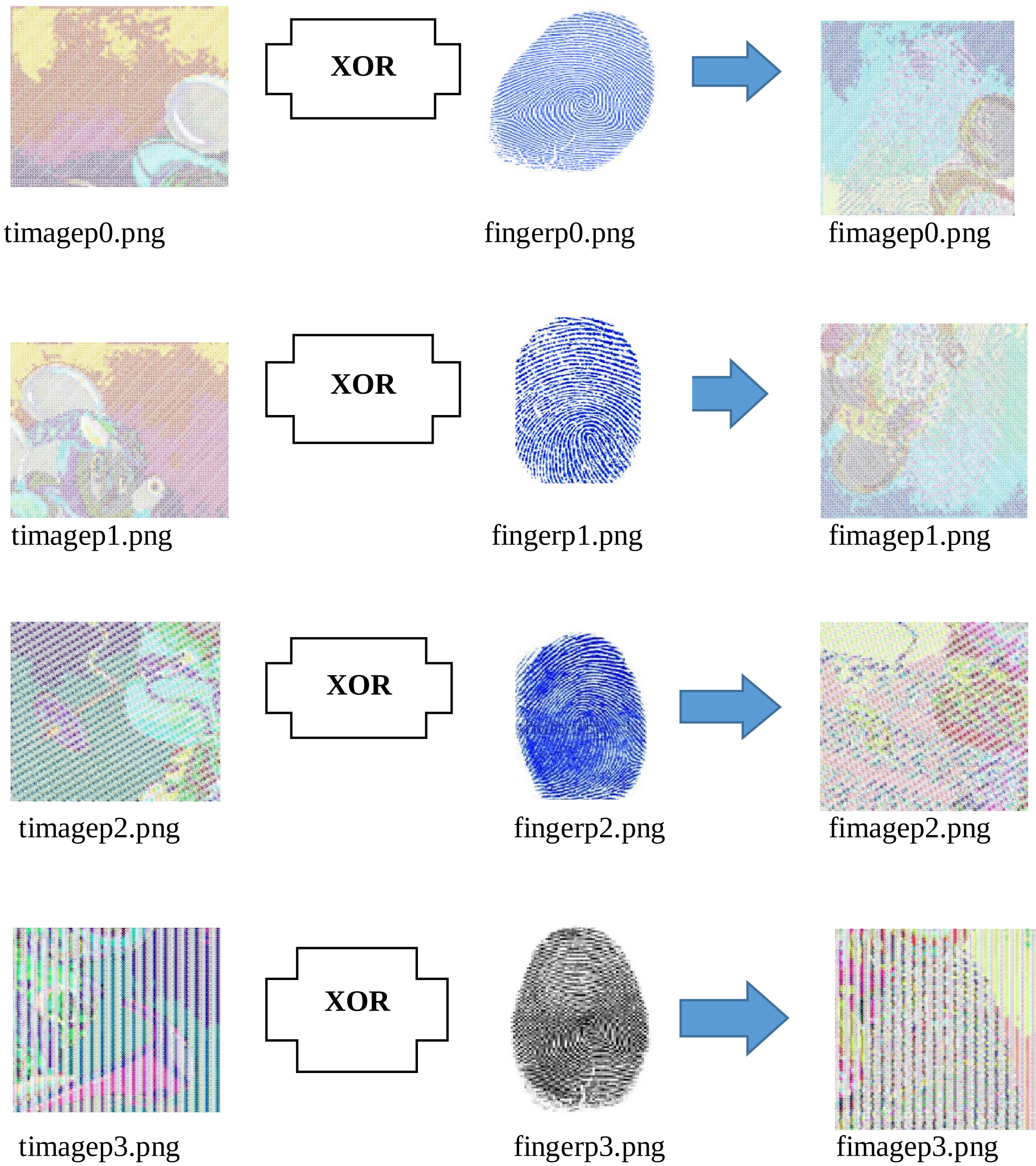


Figure 7.18: Representation of Output of Image Key Encryption Procedure

Output of Digital Enveloping Procedure

Figure 7.19 represents the output of the digital enveloping process.

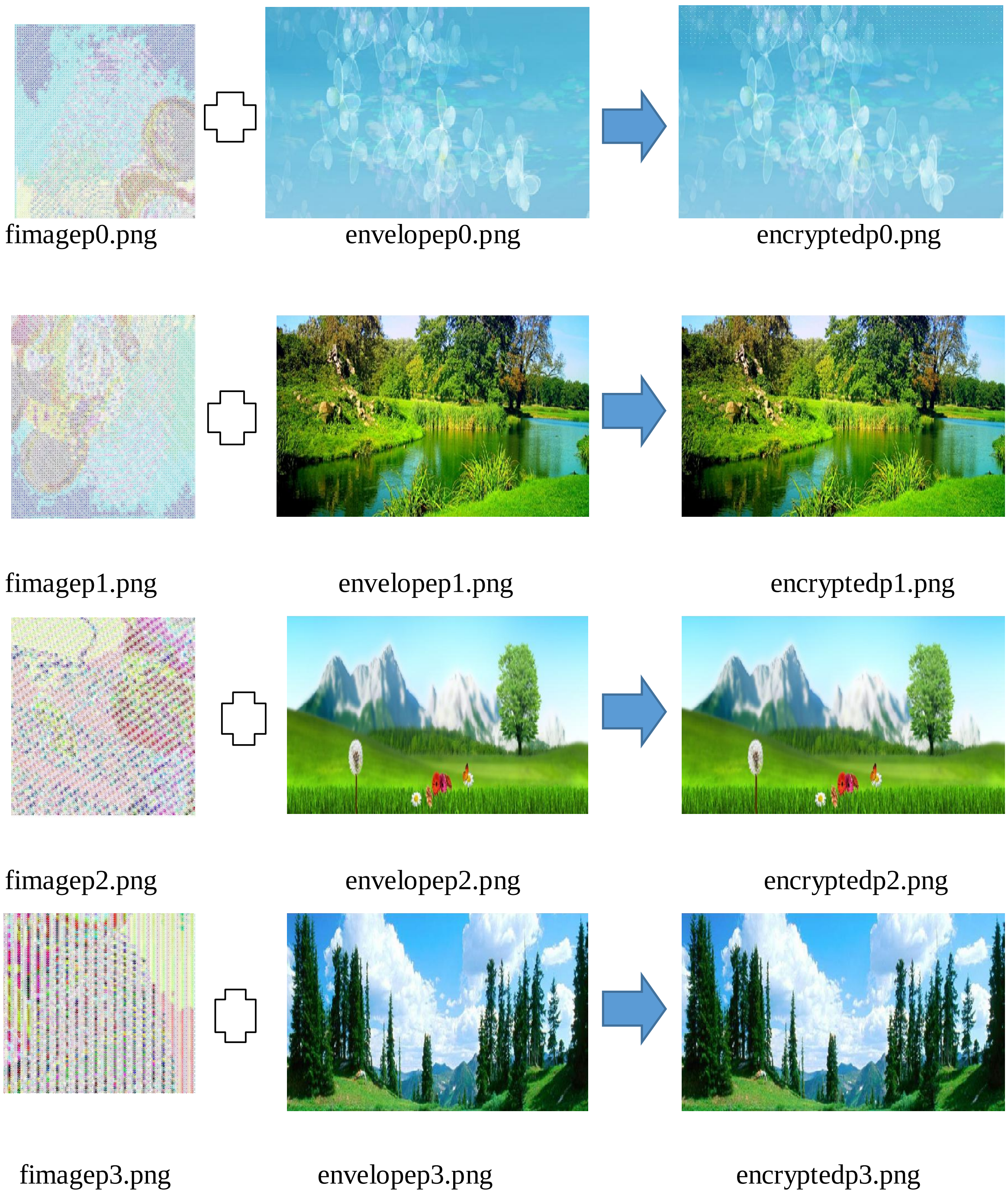


Figure 7.19: Representation of Output of Digital Enveloping Procedure

B. Decryption Process

Output of De-Enveloping Procedure

Figure 7.20 represents the output of the de-enveloping process.

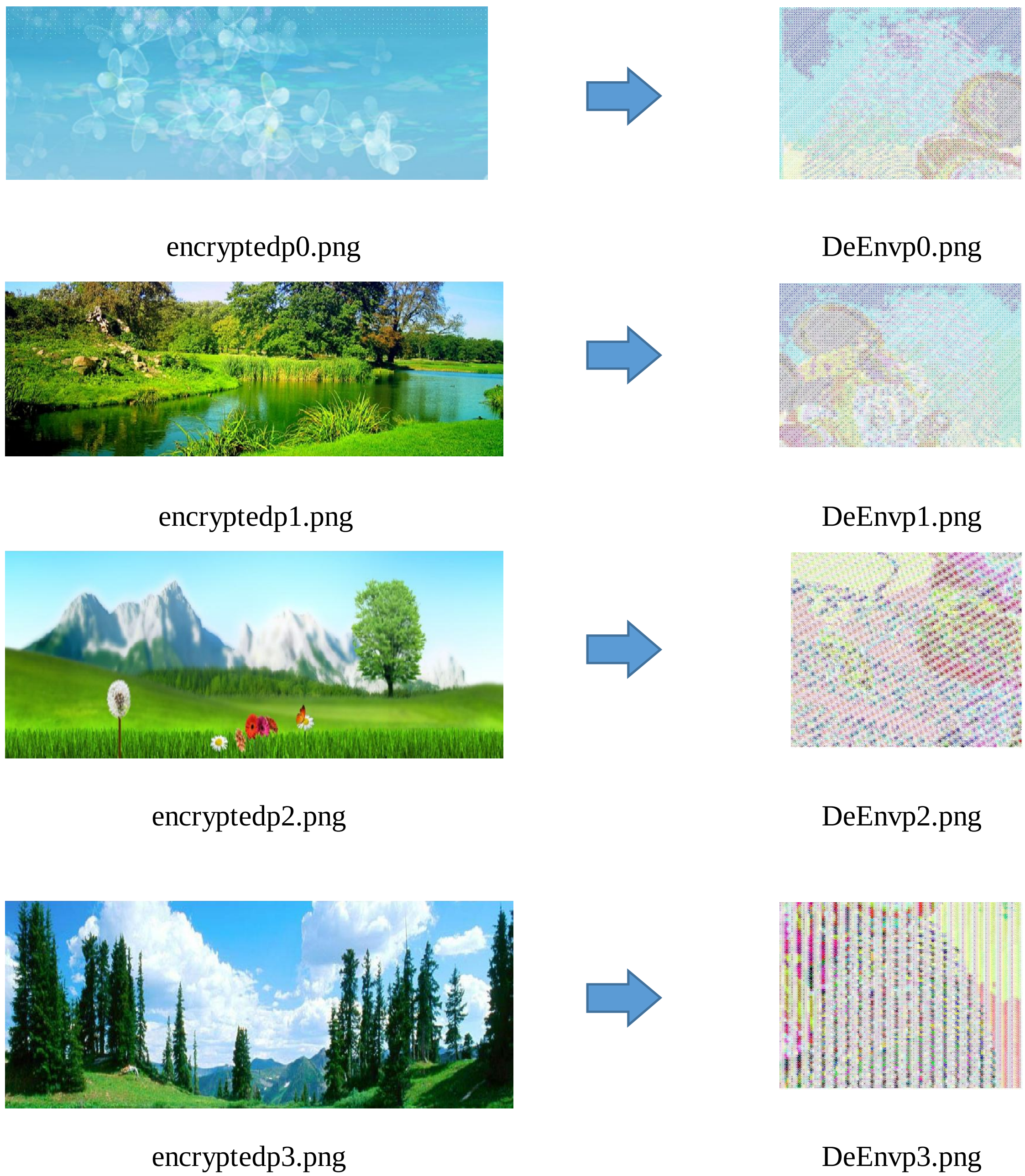


Figure 7.20: Representation of Output of De-Enveloping Procedure

Output of Image Key Decryption Procedure

Figure 7.21 represents the output of the image key decryption process

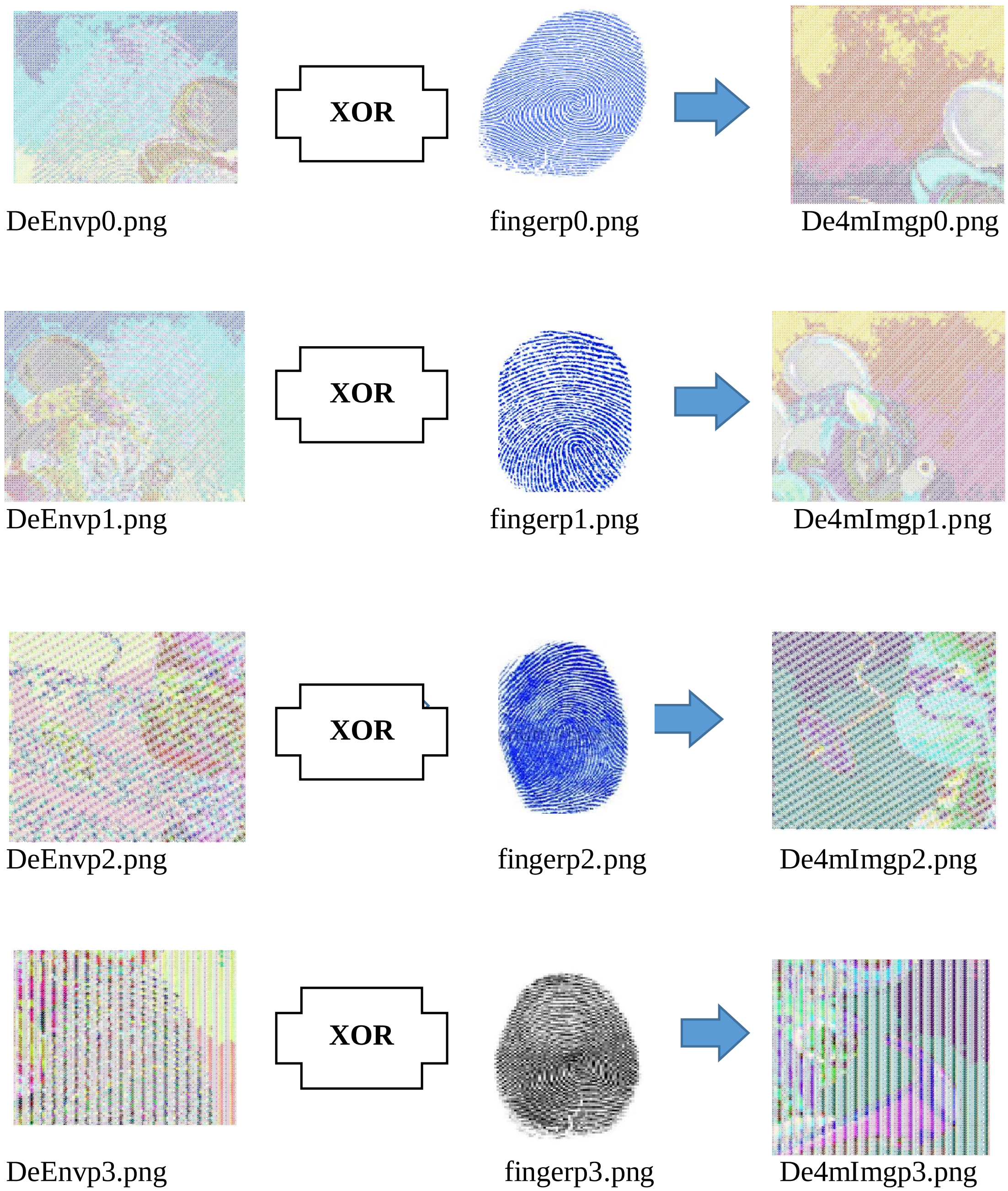


Figure 7.21: Representation of Output of Image Key Decryption Procedure

Output of Text Key Decryption Procedure

Figure 7.22 represents the output of the text key decryption process

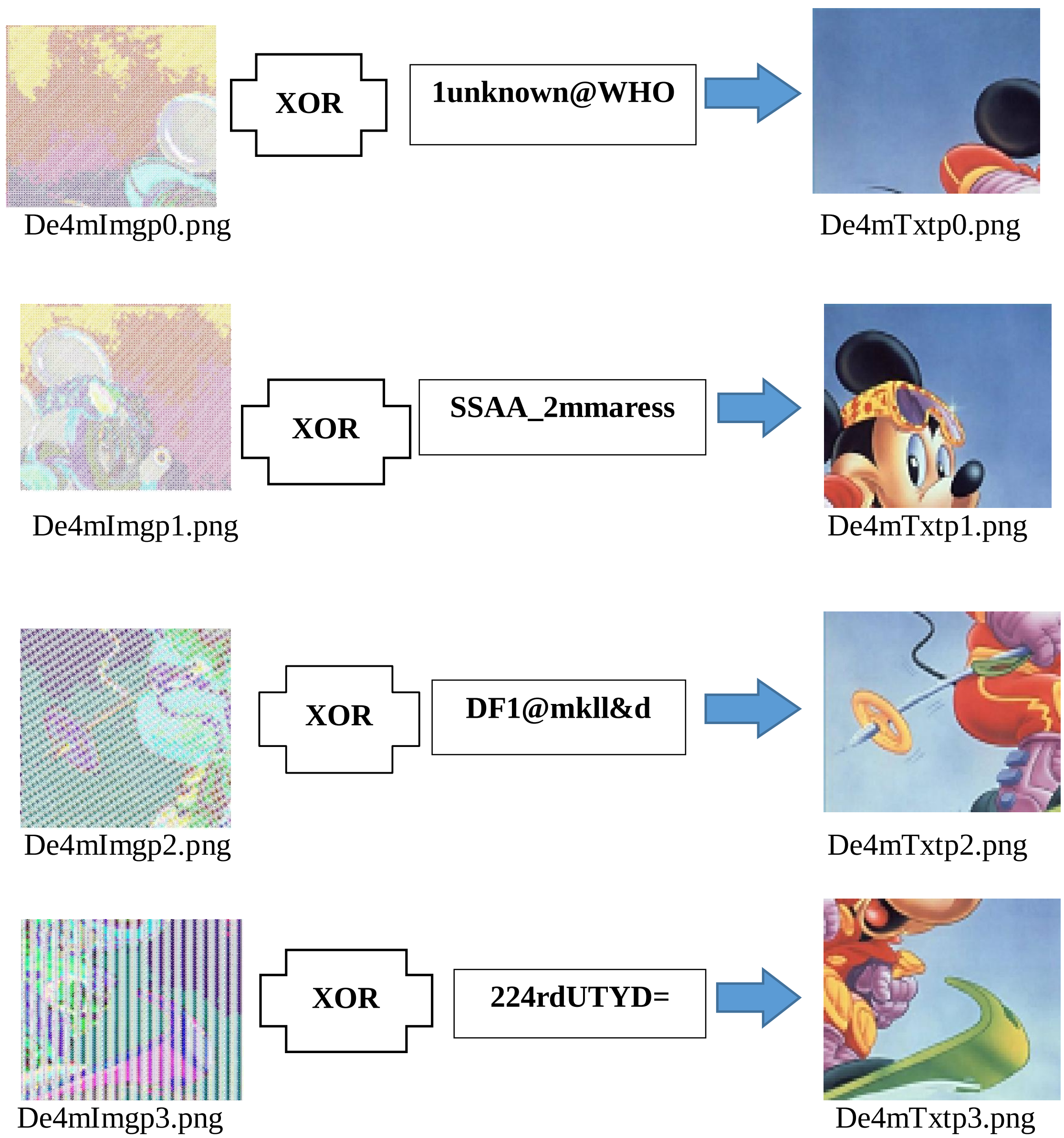


Figure 7.22: Representation of Output of Text Key Decryption Procedure

Output of Image Reconstruction Procedure

Figure 7.23 represents the output of the image reconstruction process

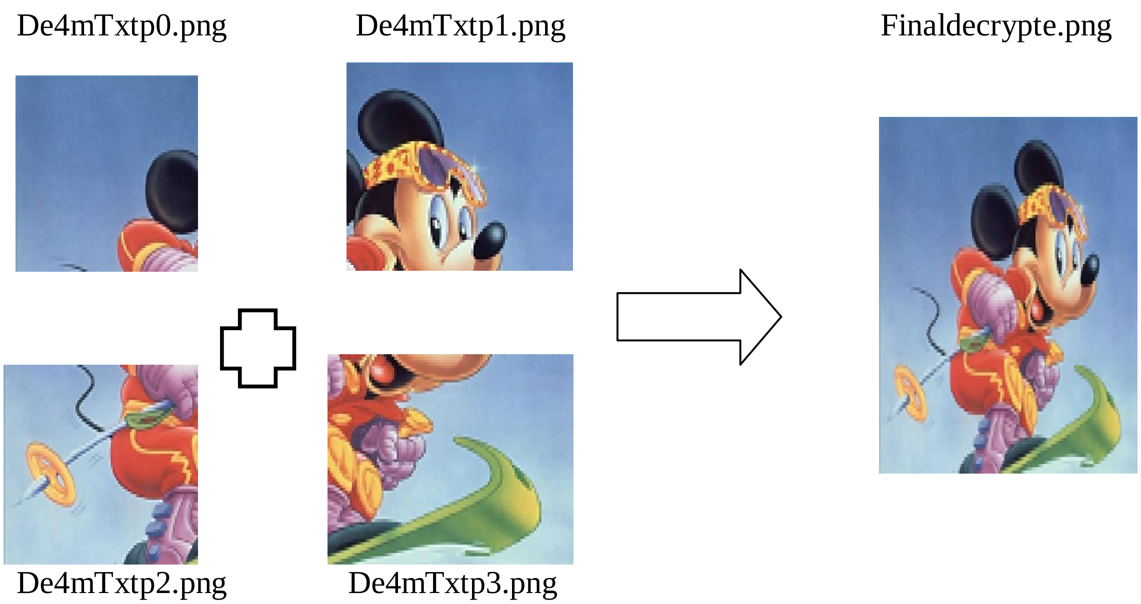


Figure 7.23: Representation of Output of Image Reconstruction Procedure

7.3.4. Security Analysis of Cumulative Image encryption scheme using Digital enveloping, Key based encryption with image Partitioning (CIDKP)

Standard image quality measurement parameters are used to determine the performance of implemented image encryption schemes where Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Metric (SSIM), Universal Image Quality Index (Q-Index), Bit Error Rate (BER), Correlation Coefficient (CC) and Normalized Cross-Correlation (NCC) are calculated as per the equations 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9 respectively mentioned in chapter 3.

Table 7.5 represents the encryption time, PSNR, MSE and BER values originated from the output generated by CIDKP scheme where outputs are taken from three levels of encryptions such as digital enveloping, image key encryption and text key encryption. When digital enveloping is considered then PSNR is high and MSE and BER are low as the content of the original envelope and embedded envelope don't differ in great extent but at the time of image or text key encryption, PSNR is extremely low and MSE and BER are high as the original image mostly differs with the encrypted image.

Table 7.5: Representation of PSNR, MSE and BER Values originated from Output of CIDKP Scheme

Original Image Name	Image Size (KB)	Compared Image Name	Image Size (KB)	PSNR (dB)	MSE	BER	Remarks
b-env1.png	335	a-env1.png	454	52.2793	0.3847	0.00757934 0277777778	Results from digital enveloping
b-env2.jpg	313	a-env2.jpg	427	52.314	0.3817	0.00753758 6805555555	
b-env3.png	114	a-env3.png	170	52.3591	0.3777	0.00748732 6388888889	
b-env4.jpg	189	a-env4.jpg	285	52.4794	0.3674	0.00748151 0416666667	
b-env5.png	287	a-env5.png	475	53.5654	0.2861	0.00581978 8178278744	
b-ori_part0.png	26	a-i_e_p0.png	29	8.7301	8710.9877	0.17110413 985413986	Results from image key encryption
b-ori_part1.jpg	23	a-i_e_p1.jpg	27	12.3472	5137.557	0.16942069 620641048	
b-ori_part2.png	27	a-i_e_p2.png	28	11.5719	4527.807	0.15181252 68125268	
b-ori_part3.jpg	37	a-i_e_p3.jpg	39	10.5041	5789.8302	0.15590909 787338358	
b-i_part0.png	9	a-t_e_p0.png	12	7.9502	10424.636	0.14818325 979040264	Results from text

b-i_part1.jpg	13	a-t_e_p1.jpg	13	7.2391	12279.224	0.15403988171845315	key encryption
b-i_part2.png	14	a-t_e_p2.png	19	11.1884	4945.7947	0.1448987252558681	
b-i_part3.jpg	14	a-t_e_p3.jpg	15	9.9867	6522.4741	0.1552790034932892	

Figure 7.24 shows image size vs. PSNR graph for CIDKP Scheme.

Figure 7.24: Representation of Image Size vs. PSNR Graph for CIDKP Scheme

Table 7.6 represents SSIM, NCC and Q-INDEX values for CIDKP scheme where outputs are produced from digital enveloping, image key encryption and text key encryption. At the time of digital enveloping, both SSIM and Q-INDEX are closer to '1' as content of the original envelope and embedded envelope are quite similar but at the time of image or text key encryption both SSIM and Q-INDEX values stay farthest from '1' as extremely less similarity is present between original image and encrypted image

Table 7.6: Representation of SSIM, NCC and Q-INDE Values originated from Output of CIDKP Scheme

Original Image Name	Image Size (KB)	Compared Image Name	Image Size (KB)	SSIM	NCC	Q-INDEX	Remarks
b-env1.png	335	a-env1.png	454	99.7589	0.9999889225324158	0.9999650713759347	Results from digital enveloping
b-env2.jpg	313	a-env2.jpg	427	99.6737	0.9999904953928216	0.9999816768805317	
b-env3.png	114	a-env3.png	170	99.7341	0.9999908459267128	0.9999090380077028	
b-env4.jpg	189	a-env4.jpg	285	99.5728	0.9999815410333526	0.9999344989697043	
b-env5.png	287	a-env5.png	475	99.7285	0.9999808165786295	0.9999751197837937	
b-ori_part0.png	26	a-i_e_p0.png	29	23.7945	0.9587984678855476	0.2657903822031001	Results from image key encryption
b-ori_part1.jpg	23	a-i_e_p1.jpg	27	32.7865	0.9136175825703966	0.039283554426930066	
b-ori_part2.png	27	a-i_e_p2.png	28	30.6045	0.9565210011472334	0.20912465780883754	
b-ori_part3.jpg	37	a-i_e_p3.jpg	39	28.7645	0.9386296007784183	0.1338086373638244	
b-i_part0.png	9	a-t_e_p0.png	12	21.7695	0.9079175831908377	-0.23812678665981526	Results from text key encryption
b-i_part1.jpg	13	a-t_e_p1.jpg	13	21.6754	0.8847447841100304	-0.242931267835754	
b-i_part2.png	14	a-t_e_p2.png	19	32.7657	0.9363874325216011	-0.29440387887556757	
b-i_part3.jpg	14	a-t_e_p3.jpg	15	24.8765	0.9131979469880628	-0.2049305450177468	

Figure 7.25 represents image size vs. Q-INDEX value graph for CIDKP Scheme.

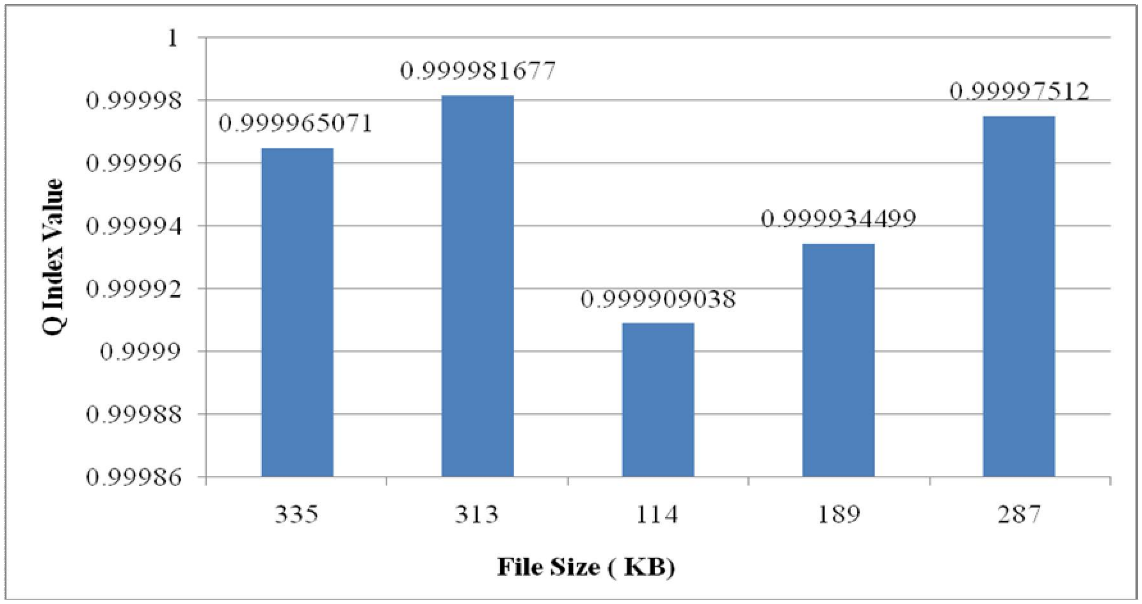


Figure 7.25: Representation of Image Size vs. Q-INDEX Value Graph for CIDKP Scheme

Table 7.7 represents standard deviation and correlation coefficient values for CIDKP scheme

Table 7.7: Representation of Standard Deviation and Correlation Coefficient Values originated from Output of CIDKP Scheme

Original Image Name	**Image size (KB)**	**Compared Image Name**	**Image size (KB)**	**Standard Deviation of Original Envelope**	**Standard Deviation of Compared Envelope**	**Correlation Coefficient**	**Remarks**
b-env1.png	335	a-env1.png	454	188.36	188.4209	1.0	Results from digital enveloping
b-env2.jpg	313	a-env2.jpg	427	256.2036	256.14	1.0	
b-env3.png	114	a-env3.png	170	108.8819	108.8213	0.9999	
b-env4.jpg	189	a-env4.jpg	285	130.4924	130.5336	0.9999	
b-env5.png	287	a-env5.png	475	188.2066	188.2179	1.0	
b-ori_part0.png	26	a-i_e_p0.png	29	104.3969	41.6862	0.4353	Results from image
b-ori_part1.jpg	23	a-i_e_p1.jpg	27	145.9205	41.9845	0.0834	

b-ori_part2.png	27	a-i_e_p2.png	28	112.9967	100.7594	0.2148	key encryption
b-ori_part3.jpg	37	a-i_e_p3.jpg	39	124.5767	124.3321	0.1372	
b-i_part0.png	9	a-t_e_p0.png	12	110.2919	38.8915	0.4192	Results from text key encryption
b-i_part1.jpg	13	a-t_e_p1.jpg	13	145.9205	36.6184	0.5754	
b-i_part2.png	14	a-t_e_p2.png	19	110.5981	88.8414	0.3043	
b-i_part3.jpg	14	a-t_e_p3.jpg	15	122.4516	124.5962	0.2076	

Figure 7.26 graphically shows the correlation coefficient values originated from the output images of CIDKP scheme.

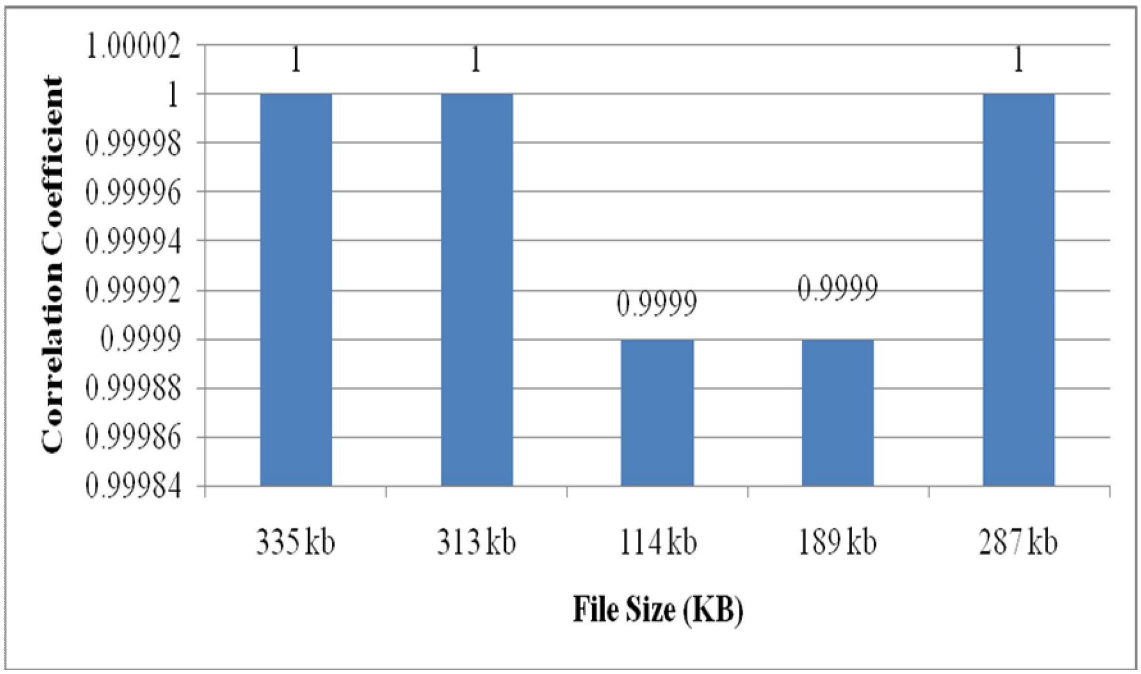


Figure 7.26: Graphical Representation of Inputted Image Size vs Correlation Coefficient Values for CIDKP Scheme

Table 7.8 shows the encryption and decryption time for different types of files which are encrypted by the CIDKP scheme where all the operation is carried out using a computer having Core 2 Duo 2.20 GHz processor and 1.00 GB RAM.

Table 7.8: Representation of Decryption and Encryption Time of Different Files encrypted by CIDKP Scheme

Original image Name	Image size (KB)	Compared Image Name	Image size (KB)	Encryption Time (Millisec)	Decryption Time (Millisec)	Remarks
b-env1.png	335	a-env1.png	454	27153	27112	Results from digital enveloping
b-env2.jpg	313	a-env2.jpg	427	31720	31687	
b-env3.png	114	a-env3.png	170	25747	25723	
b-env4.jpg	189	a-env4.jpg	285	24547	24518	
b-env5.png	287	a-env5.png	475	25447	25396	
b-ori_part0.png	26	a-i_e_p0.png	29	21910	21874	Results from image key encryption
b-ori_part1.jpg	23	a-i_e_p1.jpg	27	20900	20853	
b-ori_part2.png	27	a-i_e_p2.png	28	21500	21457	
b-ori_part3.jpg	37	a-i_e_p3.jpg	39	35930	35912	
b-i_part0.png	9	a-t_e_p0.png	12	71482	71437	Results from text key encryption
b-i_part1.jpg	13	a-t_e_p1.jpg	13	20510	20486	
b-i_part2.png	14	a-t_e_p2.png	19	21060	21015	
b-i_part3.jpg	14	a-t_e_p3.jpg	15	16670	16623	

Figure 7.27 graphically shows the encryption time for different files which are encrypted by the CIDKP scheme where it is represented that encryption time does not directly depend with the size of inputted file.

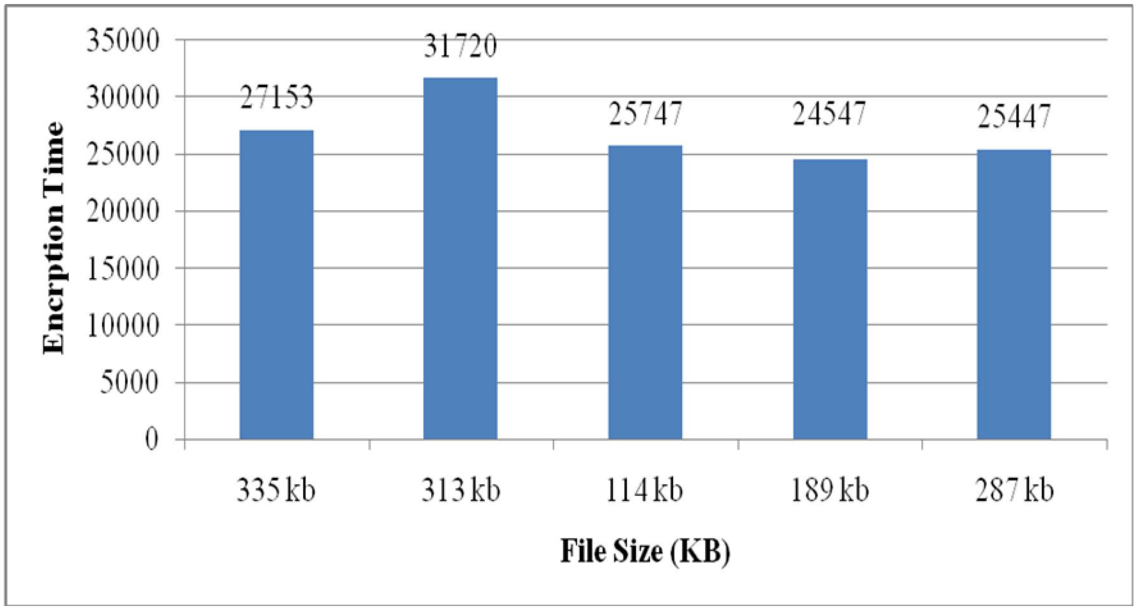


Figure 7.27: Graphical Representation of Image Size vs Encryption Time for CIDKP Scheme

7.4. Cumulative Image encryption approach using Steganographic scheme with Pixel repositioning (CISP)

Steganography is the process of hiding information of one object within another object. Traditional enveloping scheme suffers from security as the embedding of source pixel to the destination pixel is carried out by applying XOR operation in a sequential manner. So CISP[5] technique is developed based on new user defined BWMAS operation (Bitwise Masking for Alternate Sequence) where the pixels sequences for both original and envelope images are repositioned based on the user defined pixel sequencing algorithms.

---

[5] **Presented in IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN 2017)**, **IEEE Xplore,** pp. 309–314, DOI: 10.1109 / ICRCICN.2017.8234526, with title An Approach for Secured Image Encryption Scheme using User defined Operator based Steganographic Technique with Pixels Repositioning Scheme

In the current scheme first, all pixels of the envelope and the original image are repositioned by applying prime and non-prime pixel repositioning scheme. Again the resultant pixels of both the images are sequenced by applying one of the user defined pixels repositioning methods (PRONE (Positional Reverse Odd Normal Even), CRENO (Continuous Reverse Even Normal Odd), PRENO (Positional Reverse Even Normal Odd), CRONE (Continuous Reverse Odd Normal Even)) separately. Finally, user defined BWMAS (Bitwise Masking for Alternate Sequence) operation is carried out between the resultant bits of original image and least significant bits of each block of each pixel of envelope image. Implementation of prime or non-prime pixels rearrangement method and followed by pixel repositioning schemes (PRONE, CRENO, PRENO, CRONE) and introducing of user defined BWMAS operation for the first time provide great security in respect of hiding of information. Figure 7.28 represents the overall procedure for Cumulative Image encryption approach using Steganographic scheme with Pixel repositioning (CISP).

Inputted envelope image and the original image are converted into pixels and stored into two separate arrays. All the pixels of the envelope and original images are rearranged by using user specified non-prime or prime positioned pixel rearrangement scheme.

Pixels of resultant original and envelope image are rearranged separately based on their position by using one of the user defined pixel repositioning schemes (PRONE, PRENO, CRENO, CRONE) chosen by the user.

User defined BWMAS operation is carried out between the bits of original image and least significant bits of envelope image for encryption. The encrypted enveloped image is generated and it is send to the user. Reverse procedures are followed for decryption.

Figure 7.28: Overall Procedure for Cumulative Image encryption approach using Steganographic scheme with Pixel repositioning (CISP)

Section 7.4.1 represents the basic definition of different terminology. Section 7.4.2 and section 7.4.3 represents the encryption and decryption procedure respectively. Experiment results and security analysis of CISP scheme are represented in section 7.4.4 and 7.4.5 respectively.

7.4.1. Different Terminology

A. Pixel Repositioning Schemes

PRONE (Positional Reverse Odd Normal Even)

Detailed description is given in section 8.3.2 of chapter 8 (Cipher & Pixel Block Sequencing Methodologies).

PRENO (Positional Reverse Even Normal Odd)

Detailed description is given in section 8.3.2 of chapter 8 (Cipher & Pixel Block Sequencing Methodologies).

CRONE (Continuously Reverse Odd Normal Even)

Detailed description is given in section 8.3.2 of chapter 8 (Cipher & Pixel Block Sequencing Methodologies).

CRENO (Continuously Reverse Even Normal Odd)

Detailed description is given in section 8.3.2 of chapter 8 (Cipher & Pixel Block Sequencing Methodologies).

B. BWMAS Operation (Bit Wise Masking for Alternate Sequence)

Detailed description is given in section 8.3.1of chapter 8 (BWMAS Operation (Bit Wise Masking for Alternate Sequence)).

7.4.2. Encryption Process

A. Algorithm for Bit Formation from Original and Envelope Image for CISP Scheme

Step 1: Calculate width (w2) and height (h2) of the original image and convert all the pixel value of the original image into bits which are kept in an array called O[] with a size of w2*h2*32.

Step 2: Calculate width (w4) and height (h4) of envelope image and convert all the pixel of envelope image into bits which are stored in an array named E[] with a size of w4*h4*32.

B. Prime and Non-Prime positioned Pixels Rearrangement Algorithm

Step 1: Continuous prime positioned pixels from the original image are stored into an array named F2[] continuously. If two scattered prime positioned pixels are found from the original image then they are stored together in F2[] side by side and all the non-prime positioned pixels placed between these two scattered prime positioned pixels are stored into array F2[] just after the location where two prime positioned pixels are stored. Carry out the same operation until all the pixels of the original image are visited. An array named F4[] is used to store the pixels from envelope image by following the same procedure.

C. Algorithm of Steganographic Technique using BWMAS Operation

Step 1: BWMAS operation is carried out between the bits of each pixel of original and envelope image in the following manner.

E[Kth pixel's starting bit position (where K= 0 to w4*h4-1) + P(where P = any(6, 7, 14, 15,22,23,30,31))] = E[same bit position specified in array E in left side of "="] (BWMAS) O[M (where M = 0 to w2*h2*32 -1)].Where w2, h2 are width and height of original image and w4, h4 are the width and height of envelope image.

D. Algorithm for Image Construction

Construction of Alpha, Red, Green, Blue blocks of each pixel of the image is carried out and store the bit values of each pixel into an array named IC[] from where a new image is generated.

E. Algorithm of main () Function

Step 1: Call Algorithm for Bit Formation from Original and Envelope Image.

Step 2: Call Prime and Non-Prime positioned Pixels Rearrangement Algorithm.

Step 3: Call algorithm for user specified pixel repositioning technique from available pixel repositioning schemes (PRONE, CRENO, PRENO, CRONE) for original and enveloped image separately.

Step 4: Call Algorithm of Steganographic Technique using BWMAS Operation.

Step 5: Call Algorithm for Image Construction.

7.4.3. Decryption Process

A. Algorithm for main () Function

Step 1: Call algorithm of the steganographic technique using BWMAS operation. That retrieved bit value of the original image and the values are stored into an array called O1[] of size w2*h2*32.

Step 2: Call reverse algorithm for user specified pixel repositioning technique applied at the time of encryption from available reverse pixel repositioning schemes (R_PRONE, R_CRENO, R_PRENO, R_CRONE) for array O1[].

Step 3: Call reverse prime and non-prime positioned pixels rearrangement algorithm for array O1[].

Step 4: Call Algorithm for Image Construction where the input is taken from array O1[]. Thus generates the final decrypted image.

7.4.4. Experiment Result and Discussions

A. Encryption Process

Figure 7.29 and Figure 7.30 represent the original image and envelope image respectively.
Enter the name of the original image- Ori.jpg.

Figure 7.29: Original Image (Ori.jpg)

Size of the original image is 200*200 Pixels.

Enter the name of envelope image- Env.jpg.

Figure 7.30: Envelope Image (Env.jpg)

Size of the envelope image is 550*371 pixels.

Output of Prime and Non-Prime positioned Pixels Rearrangement Algorithm for Original and Envelope Image

All the pixels of the original and envelope image are provided with a sequence number and are stored in arrays in a serial manner. After applying the prime and non-prime pixels rearrangement algorithm the sequence of pixels of original and envelope image is modified. Figure 7.31 represents the sequence of pixels in the array for original or envelope image where only 25 numbers of pixels are shown among the entire size of the original image or envelope image.

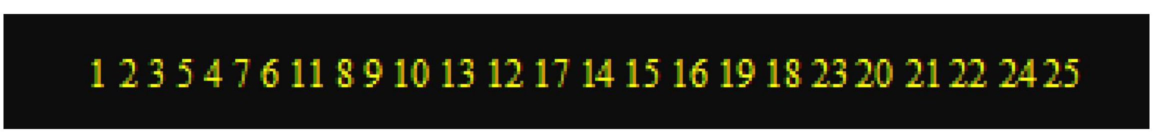


Figure 7.31: Pixels Sequence of Original or Envelope Image after applying Prime and Non-Prime positioned Pixels Rearrangement Algorithm

Output of Pixel Repositioning Algorithm for Original and Envelope Image

Figure 7.32 represents the user inputs for pixel repositioning algorithm for original and envelope image at encryption time.

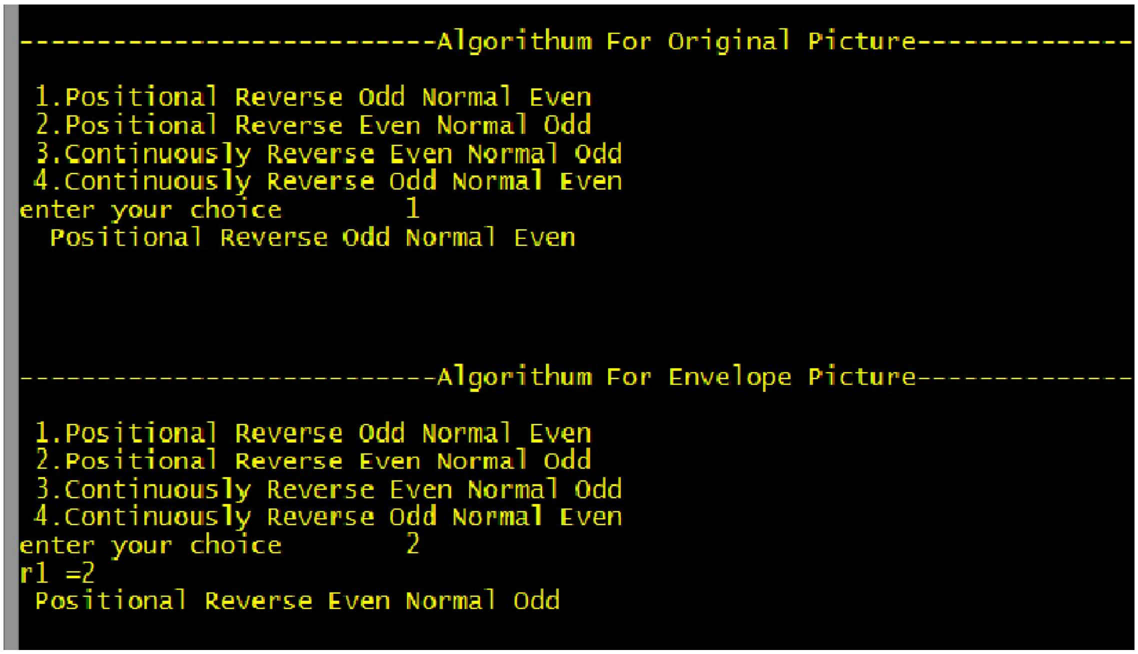


Figure 7.32: User Inputs for Pixel Repositioning Algorithm for Original and Envelope Image at Encryption Time

Output of Positional Reverse Odd Normal Even (PRONE) Algorithm for Original Image

Figure 7.33 represents the sequence of pixels from the original image in the array after applying positional reverse odd normal even (PRONE) algorithm where only first 25 numbers of pixels are shown among all pixels of the original image.

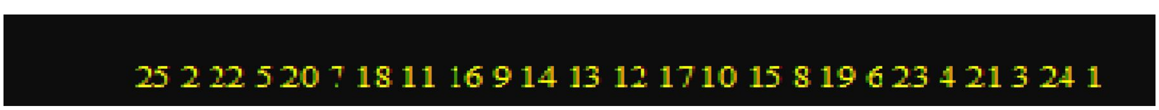


Figure 7.33: Sequence of Pixels of Original Image after applying PRONE Algorithm

Output of Positional Reverse Even Normal Odd (PRENO) Algorithm for Envelope Image

Figure 7.34 shows the sequence of pixels of envelope image in the contained array after applying positional reverse even normal odd (PRENO) algorithm where only first 25 numbers of pixels from envelope image are shown in the result.

Figure 7.34: Sequence of Pixels of Envelope Image after applying PRENO Algorithm

Output of Algorithm for Steganographic Technique using BWMAS Operation

Figure 7.35 represents the envelope image where all the pixels of the original image are merged by applying the BWMAS operation.

Figure 7.35: Envelope Image after Embedding Original Image (Eenv.png)

D. Decryption Process

Figure 7.36 represents the user inputted envelope image in which the information of the original image is embedded.
Enter the name of the envelope image- Eenv.png.

Figure 7.36: Input Image for Decryption (Eenv.png)

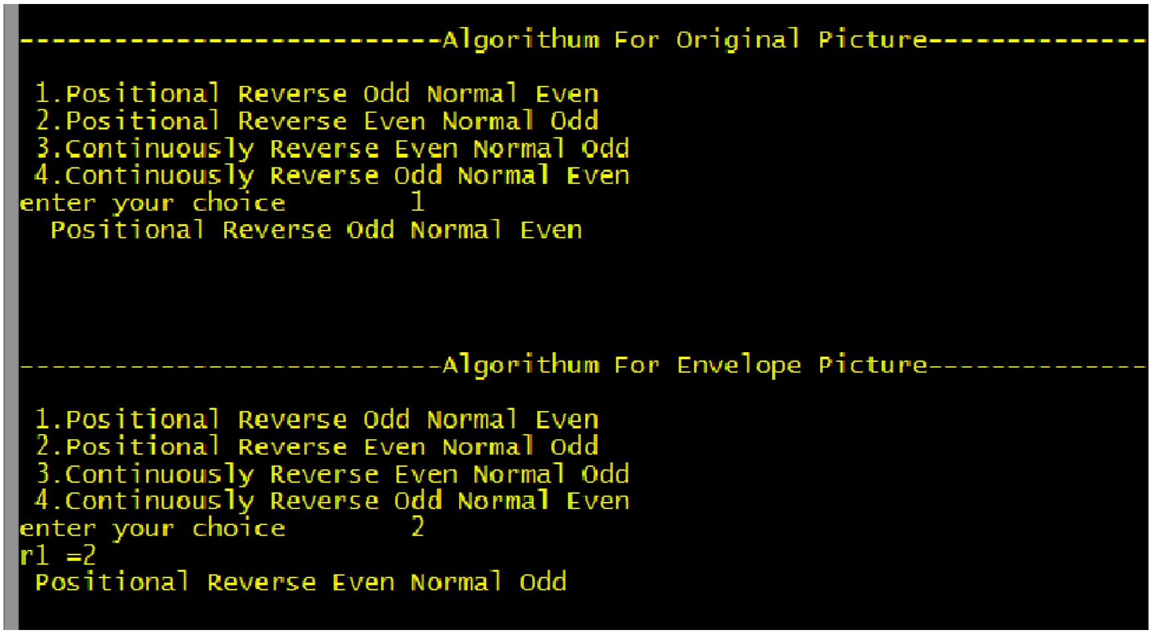
Size of the envelope image is 550*371 pixels.

Output of Algorithm for Steganographic Technique using BWMAS Operation for Decryption

BWMAS operation is carried out in between actual envelope image and the envelope image in which the information of the original image is merged. Resultant values are stored in an array named O1[]. Only the first 25 number of pixels of the original image is shown in Figure 7.37.

Figure 7.37: Pixel Sequence for Original Image generated from Algorithm for Steganographic Technique using BWMAS Operation for Decryption

Figure 7.38 shows user inputs for pixel repositioning algorithm for the original image at decryption time.


```
---------------------------Algorithum For Original Picture--------------
 1.Positional Reverse Odd Normal Even
 2.Positional Reverse Even Normal Odd
 3.Continuously Reverse Even Normal Odd
 4.Continuously Reverse Odd Normal Even
enter your choice        1
  Positional Reverse Odd Normal Even



---------------------------Algorithum For Envelope Picture--------------
 1.Positional Reverse Odd Normal Even
 2.Positional Reverse Even Normal Odd
 3.Continuously Reverse Even Normal Odd
 4.Continuously Reverse Odd Normal Even
enter your choice        2
r1 =2
 Positional Reverse Even Normal Odd
```

Figure 7.38: User Inputs for Pixel Repositioning Algorithm for Original and Envelope Image at Decryption Time

Output of Reverse Positional Reverse Odd Normal Even (R_PRONE) Algorithm for Original Image

As the user has selected the PRONE algorithm, the scheme executes Reverse PRONE (R_PRONE) algorithm for carrying out the decryption process. Figure 7.39 shows the sequence of pixels of array O1[] after applying reverse positional reverse odd normal even (R_PRONE) algorithm where only the first 25 numbers of pixels are shown.


```
1 2 3 5 4 7 6 11 8 9 10 13 12 17 14 15 16 19 18 23 20 21 22 24 25
```

Figure 7.39: Pixel Sequence for Original Image after applying Reverse Positional Reverse Odd Normal Even (R_PRONE) Algorithm

Output of Reverse Prime and Non-Prime positioned Pixels Rearrangement Algorithm for Original Image

The scheme executes reverse prime and non-prime positioned pixels rearrangement algorithm for carrying out the decryption process. Figure 7.40 represents the sequence of pixels for array O1[] where only the first 25 numbers of pixels are shown.

Figure 7.40: Pixels Sequence for Original Image after applying Reverse Prime and Non-Prime positioned Pixels Rearrangement Algorithm

Output for Algorithm of Image Construction

Figure 7.41 represents the original image which is constructed from the array O1[] by applying image construction algorithm.

Figure 7.41: Original Image after De Enveloping (De_Ori.jpeg)

Size of the decrypted image is 200*200 pixels.

7.4.5. Security Analysis for Cumulative Image encryption approach using Steganographic scheme with Pixel repositioning (CISP)

Standard image quality measurement parameters are used to determine the performance of implemented image encryption schemes where Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Metric (SSIM), Universal Image Quality Index (Q-Index), Bit Error Rate (BER), Correlation Coefficient (CC) and Normalized Cross-Correlation (NCC) are calculated as per the equations 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9 respectively mentioned in chapter 3. Table 7.9 represents encryption time, PSNR, BER and MSE values generated from the outputs of CISP scheme where the value of MSE and BER are quite low and PSNR value is quite high.

Table 7.9: Representation of MSE, BER, PSNR Values generated from Outputs of CISP Scheme

Original Envelope Image Name	Image size	Embedded Envelope Image Name	Image size	PSNR (dB)	MSE	BER
b-ori_env1.png	335 kb	a- enc_env1.png	379 kb	44.3506	2.3879	0.04173315 9722222225
b-ori_env2.jpg	313 kb	a- enc_env2.jpg	469 kb	44.9504	2.0799	0.04143654 513888889
b-ori_env3.png	114 kb	a- enc_env3.png	298 kb	44.9405	2.0847	0.04127048 611111111
b-ori_env4.jpg	61 kb	a- enc_env4.jpg	63 kb	44.639	2.2345	0.04165416 6666666666
b-ori_env5.png	189 kb	a- enc_env5.png	268 kb	45.1287	1.9962	0.03445616 3194444445
b-ori_env6.jpg	287 kb	a- enc_env6.jpg	428 kb	44.5293	2.2917	0.04175345 095156416
b-ori_env7.png	78 kb	a- enc_env7.png	74 kb	43.7026	2.7722	0.04361872 123505213
b-ori_env8.jpg	102 kb	a- enc_env8.jpg	134 kb	44.2056	2.469	0.04238201 689021361
b-ori_env9.png	71 kb	a- enc_env9.png	134 kb	44.2931	2.4197	0.04428170 87250536
b-ori_env10.jpg	88 kb	a- enc_env10.jpg	126 kb	44.4239	2.348	0.04214936 247723133

Figure 7.42 shows the PSNR values vs. image size graphs.

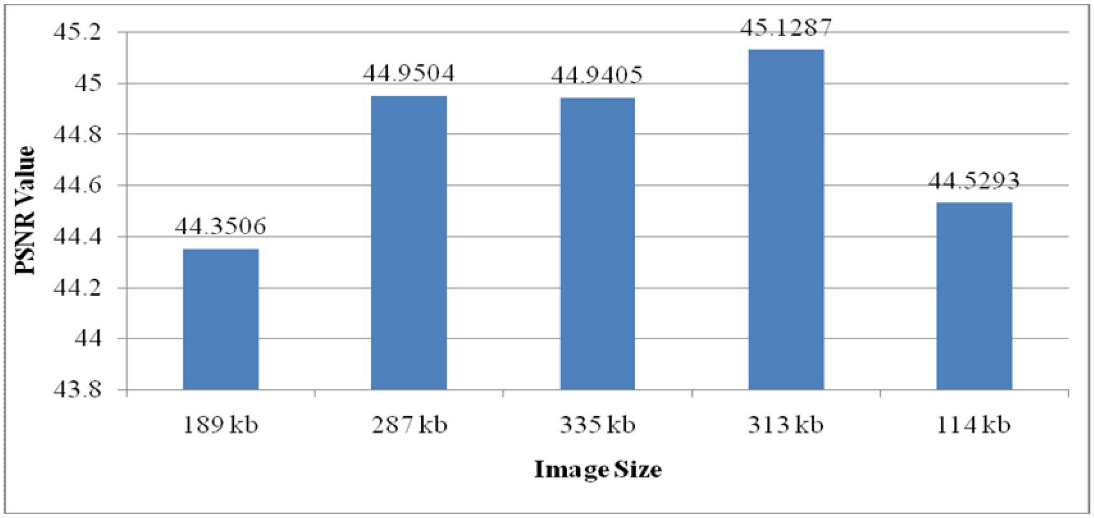


Figure 7.42: Representation of CISP Scheme's PSNR Values vs. Image Size Graph

Table 7.10 represents the SSIM, NCC and Q-INDEX values generated from the outputs of CISP scheme where all the SSIM, NCC and Q-INDEX values are satisfactory and they are in standard range.

Table 7.10: Representation of SSIM, NCC and Q-INDEX Values originated from the Output Images of CISP Scheme

Original Envelope Image Name	Image size	Embedded Envelope Image Name	Image size	SSIM	NCC	Q-INDEX
b-ori_env1.png	335 kb	a- enc_env1.png	379 kb	98.5057	0.99993255 94731652	0.99973675 88816646
b-ori_env2.jpg	313 kb	a- enc_env2.jpg	469 kb	98.5274	0.99994906 47561634	0.99990640 039524
b-ori_env3.png	114 kb	a- enc_env3.png	298 kb	96.3594	0.99994942 50090051	0.99945702 76889952
b-ori_env4.jpg	61 kb	a- enc_env4.jpg	63 kb	99.3649	0.99989703 6179409	0.99963413 21284478

b-ori_env5.png	189 kb	a- enc_env5.png	268 kb	97.7845	0.99990952 31371945	0.99962858 77166567
b-ori_env6.jpg	287 kb	a- enc_env6.jpg	428 kb	97.9717	0.99986522 72234807	0.99979658 13942105
b-ori_env7.png	78 kb	a- enc_env7.png	74 kb	99.0685	0.99993347 89959666	0.99981386 47298151
b-ori_env8.jpg	102 kb	a- enc_env8.jpg	134 kb	99.3189	0.99992383 29295699	0.99954002 21053972
b-ori_env9.png	71 kb	a- enc_env9.png	134 kb	98.7612	0.99989971 40523898	0.99960890 98579719
b-ori_env10.jpg	88 kb	a- enc_env10.jpg	126 kb	98.9790	0.99991900 28757109	0.99980487 36953291

Figure 7.43 represents the image size vs SSIM value generated from the outputs of the CISP scheme graph where SSIM values are in satisfactory range.

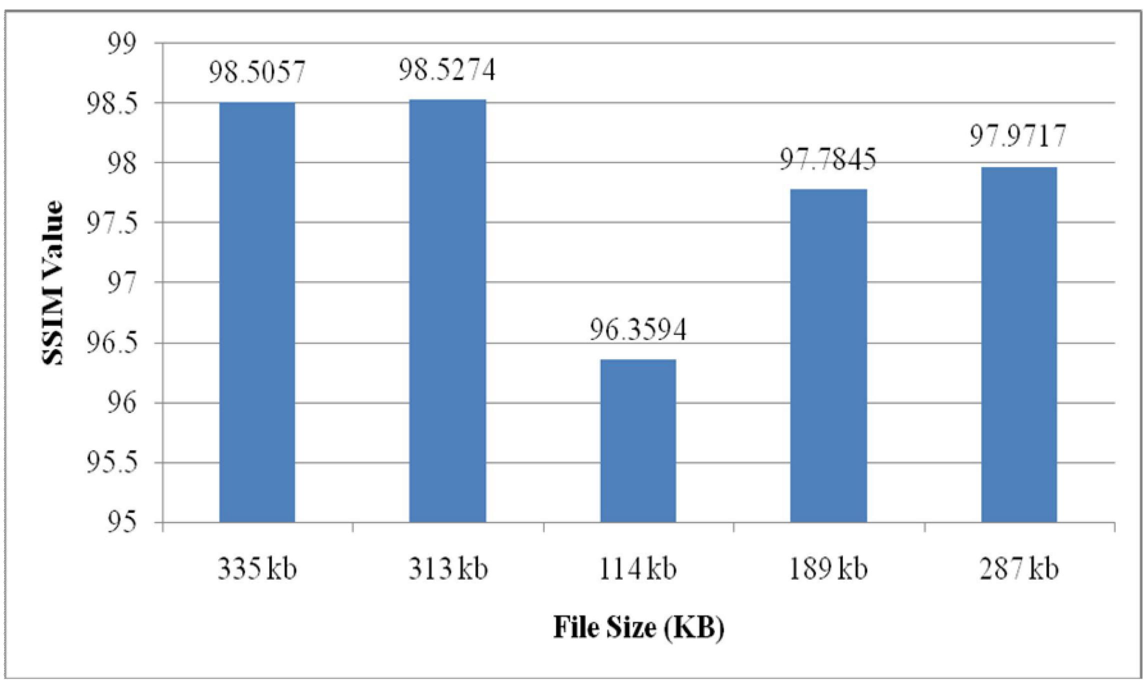


Figure 7.43: Representation of Image Size vs SSIM Value Graph for CISP Scheme

Table 7.11 shows standard deviation values for original envelope image and embedded envelope image and correlation coefficient values generated from the output images from CISP scheme where all the standard deviation and correlation coefficient values are in satisfactory range.

Table 7.11: Representation of Standard Deviation and Correlation Coefficient Values generated from CISP Scheme

Original Envelope Image Name	**Image size**	**Embedded Envelope Image Name**	**Image size**	**Standard Deviation of Original Envelope**	**Standard Deviation of Embedded Envelope**	**Correlation Coefficient**
b-ori_env1.png	335 kb	a- enc_env1.png	379 kb	188.36	188.4263	0.9998
b-ori_env2.jpg	313 kb	a- enc_env2.jpg	469 kb	256.2036	255.864	0.9999
b-ori_env3.png	114 kb	a- enc_env3.png	298 kb	108.8819	108.3077	0.9995
b-ori_env4.jpg	61 kb	a- enc_env4.jpg	63 kb	130.1225	130.0969	0.9997
b-ori_env5.png	189 kb	a- enc_env5.png	268 kb	130.4924	130.4822	0.9997
b-ori_env6.jpg	287 kb	a- enc_env6.jpg	428 kb	188.2066	188.2136	0.9999
b-ori_env7.png	78 kb	a- enc_env7.png	74 kb	189.7728	189.7273	0.9999
b-ori_env8.jpg	102 kb	a- enc_env8.jpg	134 kb	140.3648	140.5042	0.9996
b-ori_env9.png	71 kb	a- enc_env9.png	134 kb	130.4722	130.5133	0.9997
b-ori_env10.jpg	88 kb	a- enc_env10.jpg	126 kb	188.8209	188.7456	0.9999

Figure 7.44 represents correlation coefficient values for CISP scheme graphically where all the values lie between satisfactory ranges.

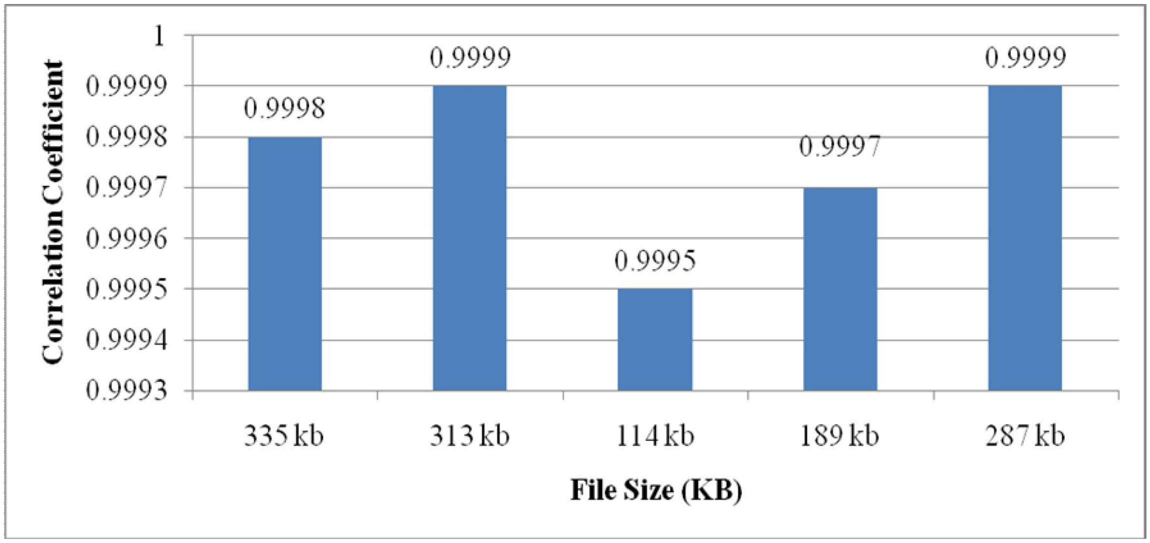


Figure 7.44: Representation of CISP Scheme's Image Size vs Correlation Coefficient Graph

Encryption of different files has been carried out by CISP scheme with the help of a computer having Core 2 Duo 2.20 GHz processor and 1.00 GB RAM. Table 7.12 shows the encryption and decryption time of different files which are encrypted by the CISP scheme.

Table 7.12: Encryption and Decryption Time of Different Files encrypted by CISP Scheme

Original Envelope Image Name	Image size	Embedded Envelope Image Name	Image size	Encryption Time (Milliseconds)	Decryption Time (Milliseconds)
b-ori_env1.png	335 kb	a- enc_env1.png	379 kb	39680	39642
b-ori_env2.jpg	313 kb	a- enc_env2.jpg	469 kb	36046	36013
b-ori_env3.png	114 kb	a- enc_env3.png	298 kb	33414	33375
b-ori_env4.jpg	61 kb	a- enc_env4.jpg	63 kb	29342	29314
b-ori_env5.png	189 kb	a- enc_env5.png	268 kb	52468	52426

b-ori_env6.jpg	287 kb	a- enc_env6.jpg	428 kb	41790	41757
b-ori_env7.png	78 kb	a- enc_env7.png	74 kb	36340	36311
b-ori_env8.jpg	102 kb	a- enc_env8.jpg	134 kb	28260	28223
b-ori_env9.png	71 kb	a- enc_env9.png	134 kb	26580	26545
b-ori_env10.jpg	88 kb	a- enc_env10.jpg	126 kb	38110	38069

Figure 7.45 graphically shows encryption time of different files which are encrypted by CISP scheme where it is noticed that encryption time is not relate with inputted file size.

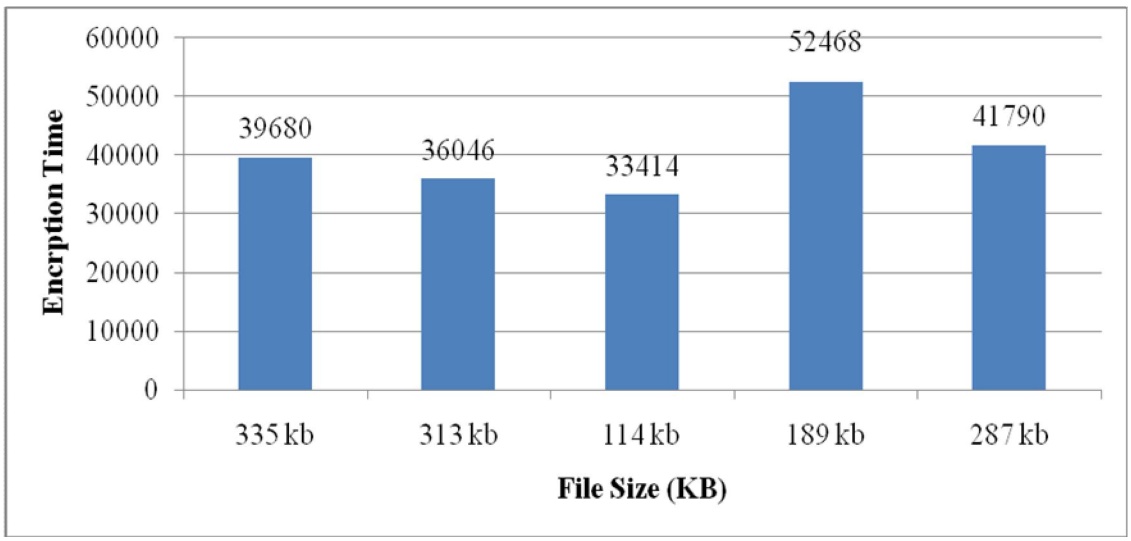


Figure 7.45: Representation of CISP Scheme's Encryption Time vs Image Size Graph

7.5. Conclusion

In EODE scheme the bit values of the pixel of the original image are embedded with the bit values of pixels of envelop image in even and odd block wise rather than continuously, thus increase the security of the implemented scheme in great extent.

In CIDKP scheme, image partitioning into user defined pieces, using of sender unique biometric images as image keys, using of separate keys for encrypting each image pieces for multiple times and reconstruction of an original image based on the proper placement of image pieces have increased the security in great extent.

CISP scheme introduces user defined BWMAS (Bitwise Masking for Alternate Sequence) operation rather than using XOR operation. Multiple levels of encryptions are carried out using prime/non-prime pixels rearrangement, user defined pixels repositioning scheme and BWMAS operation. Thus the security is increased to a great extent.

Appreciable performances are measured for the implemented schemes in respect of standard parameters like PSNR, SSIM, NCC, MSE, Q-INDEX, BER, Correlation Coefficient values. Thus it may be concluded that the implemented schemes may provide great security in image encryption.