VOL. 10, NO. 7, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2015 Asian Research Publishing Network (ARPN). All rights reserved.

www.arpnjournals.com

SECURE DATA EMBEDDING USING REVERSIBLE DATA HIDING FOR ENCRYPTED IMAGES

R. Selveeswari and P. R. Vijayalakshmi
Computer Science and Engineering, K.L.N. College of Engineering, Madurai, India
E-Mail: selveeswari@gmail.com

ABSTRACT

The common methodology in data embedding is that in the sender side, the data is to be hidden within the pixel values of the cover image that is producing the stego image. In the receiver side the embedded data in the stego image is retrieved by reversing the same process that is employed for embedding. The data that is to be hidden which is embedded in the AES encrypted image using Reversible Data Hiding (RDH) method. The RDH improves the data hiding process. In existing the data retrieval process the error is accrue it affect the data, in our proposed improve the data hiding and retrieval accuracy. A content owner encrypts the original image using an encryption key and reserves the room for data hiding. Then, a data-hider hides the data in reserve space of the image. If a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. AES algorithm uses two different keys for encryption and decryption. Each byte of the state is combined with a block of the round key using bitwise XOR. Then each byte is replaced with another according to a lookup table. The last three rows of the state are shifted cyclically a certain number of steps. Four bytes in each column were combined. In the receiver side the same procedures are reversed in order to obtain the secret message and the cover image. The performance of the process is measured with the help of the calculation of the Compression Ratio, PSNR and MSE. The measured performance indicates that the proposed system is capable of hiding the messages with better accuracy and the process is reversible also and hence has numerous applications.

Keywords: reversible data hiding, adaptive pixel pair matching, advanced encryption standard, modification direction, diamond encoding, least significant bit, mean square error, peak signal to noise ratio

INTRODUCTION

Steganography comes from Greek words stegos and grafia and its meaning is cover and writing which defines it as covered writing. In image steganography the information is hidden exclusively in images. Steganography is the art and science of secret communication. It is the practice of encoding/embedding secret information in a manner such that the existence of the information is invisible. The original file can be referred to as cover text or cover image. After inserting the secret message it is referred to as stego-medium. A stego-key is used for hiding/encoding process to restrict detection or extraction of the embedded data. Steganography differs from cryptography. Steganography hide the messages inside the cover medium, many Carrier formats. Breaking of steganography is known as Steganalysis.

Image data hiding methods have many applications. Image data hiding represents a class of processes used to embed data into cover images. Robustness is one of the basic requirements for image data hiding. In most cases, the cover image cannot be inverted back to the original image after the hidden data are retrieved. In Existing Wien Hong and Tung-Shou Chen [5] proposed a adaptive pixel pair matching in which the cover image segmented into a number of non-overlapping two pixel blocks [5]. Then, selected the each block from top-down and left-right in turn for data embedding process. The block construction vector (x,y) is defined as x = I(2t), y=I(2t+1)and cover image I sized m×n, and t is block index The embedded secret data bit stream is transformed into l-ary digit sequence. Constructs a cover image embed a message digit sd, coordinates (a, b) in the cover image is selected according to the embedding sequence Q, and calculate the modulus distance (sd-f (a, b)) mod B. It is between sd and f (a, b), then replace (a, b) with (a+aD, b+bD). The secret image bits were concealed into coordinates of host image.

In this paper a framework is proposed to the data along with the cover image. In the proposed work the cover image is encrypted by AES algorithm and also the secret data is encrypted then embedded into the encrypted cover image by Reversible data hiding process. This work increases the security of data and reduces the distortions after the image recovery. Data is extracted by reverse process of embedding.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 presents Algorithm explanation. Section 4 gives Implementation details. Conclusion is in Section 5.

RELATED WORKS

Chi-Kwong Chan and L.M. Cheng [1] proposed LSB substitution method to embed secret messages into r-rightmost Least Significant Bit of cover image pixels. The pixel with even values will be decreased by one or kept unmodified. The Pixel with odd values will be increased by one or kept unmodified. The optimal pixel adjustment process of such method is used only for checking the embedding error between the original cover-image and the stego-image. The final stego image is obtained by the direct replacement of the k least significant bits of cover image pixels with k message bits. Therefore, the imbalanced embedding distortion emerges and it is vulnerable to steganalysis.

Jarno Mielikainen [2] proposed an LSB matching revisited algorithm in which used gray scale images for message embedding in which two cover image pixels are used at a time. The gray- level values of those two pixels are x_i and xi_{+1}. After the message embedding, the value of the ith message bit m_i is equal to the LSB of stego image's ith pixel y_i. The value of the i+1th message bit m_{i+1} is a function of y_i and y_{i+1}.The embedding is performed by using a pair of pixel as a unit, where the LSB of the first pixel carries one bit of information and a function of the two pixel values carries another bit of information. This method allows a selection of addition/subtraction of y_i to carry information randomly because the selection can set a binary function f ($y_{i,}$ y_{i+1}) to the desired value. Therefore, the modified method allows embedding the same payload as LSB matching but fewer changes to the cover image.

Xinpeng Zhang and Shuozhong Wang [6] proposed a stenographic method. In which each secret digit in a (2n +1)-ary notational system is carried by n cover pixels, where n is a system parameter, and at most, only one pixel is increased or decreased by 1. In each block of n cover pixels, there are 2n possible states of only one pixel value plus 1 or minus 1. The 2n states of alteration plus the case in which no pixel is modified form (2n +1) different cases. Therefore, (2n+1)-ary notational secret digit is embedded into the cover pixels by changing the state. Before data embedding, the secret data can be converted into sequences of digits through preprocessing with (2n+1)-ary notational representation. Suppose n=5, the secret data stream $S_{(5)}$ can conceal into blocks of two cover pixels by modifying at most one pixel value. The gray values of two cover pixels as p_1 and $p_{2,}$ where the transformed 5-ary secret digit is to be embedded. Based on the block of two pixels gray value and function of weighted sum modulo 5 is defined the extraction process. The directions of modification are fully exploited by the proposed method of exploiting Modification Direction method.

Ruey-Ming Chao *et al* [4] proposed a Diamond Encoding method. In this method, cover image is partitioned into non overlapping blocks of two consecutive pixels and secret message is transformed into series of k-ary digits. In each block diamond encoding technique is applied to calculate the diamond characteristic value and one secret k-ary digit is concealed into the diamond characteristic value. This value is modified the secret digit and that can be obtained by adjusting pixel values in a block. This scheme is designed in such a way and the difference between the cover-block and the stego-block is never more than the embedding parameter k,and the block embedding capacity is equal to $\log_2(2k^2 + 2k + 1)$. The diamond encoding method minimizes the distortion after the DCV alteration to perform better visual quality.

J. Wang *et al*. [3] used section-wise exploiting modification direction method for embedding. In this method, cover images are divided into pixel sections. The number of pixels in each section can either be equal or not. And then the section is divided into two groups such as selective and descriptive group. A pointer is assigned each group which is called its section name. The total no of pixels in these group as n, can be converted into (2n+1)-ary digit which is used to define the variations of SP and DP whose value varies from 0 to 2n. Once values of SP and DP found. Based on SP and DP cut off values create SP and DP table. In which cover image pixels are permuted randomly and embedding direction is modified through converting of secret message into sequence of (2n+1) -ary secret digits. So, that secret digit is in the region [0, 2n].

Shilpa Sreekumar and Vincy Salam [7] proposed a embedding method based on reversible data hiding. In which both data and image were encrypted and extracted reversibly without any errors. Data encrypted by advanced Encryption standard algorithm and image is encrypted by Blowfish algorithm and also implemented in digital video watermarking. Encrypted image divided into several blocks by flipping 3 LSB of the half of pixels in each block, which is vacated for embedded bit. Reversible data hiding find out space for LSB embedding of pixels into other pixels and then encrypt image. The position of LSB in encrypted image is used to embed data. The secret data is encoded by Huffman algorithm which is used to compress the data before encryption. AES is known as a substitution permutation which is a combination of substitution and combination.AES is fast in software and hardware. The data is extracted and image is recovered by finding flipped part of one block.

ALGORITHM

Advanced encryption standard

AES is a block cipher with a block length of 128 bits.AES permits three different key lengths for processing such as 128, 192, or 256 bits. In this processing the key length is 12 bits. Encryption has taken 10 rounds to process 128-bit keys, 12 rounds to process 192-bit keys, and 14 rounds to process 256-bit keys. Except the last round in each case, all other rounds are same. Each round of processing consists one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption. Figure-1 represents 128-bit block as represented as a 4×4 matrix of bytes:

$$\begin{bmatrix} byte0 & byte4 & byte8 & byte12 \\ byte1 & byte5 & byte9 & byte13 \\ byte2 & byte6 & byte10 & byte14 \\ byte3 & byte7 & byte11 & byte15 \end{bmatrix}$$

Figure-1. 4x4 matrix of bytes.

In 128-bit input block, the first four bytes occupy the first column of 4x4 matrix of bytes and second four bytes occupy second column and so on. The 4 × 4 matrix of bytes is called as state array. AES also use word. A word consists of four bytes which contains32 bits. In state array, each column use a word, that is same as in each row. In each round, the input state array is processed and produces the output state array and in the last round the output state array is rearranged into output block which contains 128 bit. The decryption algorithm differs substantially from the encryption algorithm. Same steps are used in encryption and decryption, the order in which the steps are conceded different. AES requires the block size to be 128 bits, the original Rijndael's cipher works with any block size and any key size which is a multiple of 32. The state array for the different block sizes has four rows in the Rijndael cipher. The number of column is depending on block size. For example, when the block size is 192, the Rijndael cipher has need of a state array to consist of 4 rows and 6 columns. Encryption of a 128-bit block has following steps: Byte Substitution, Shift Rows, Mix Columns, Add Round Key, and AES Round.

Byte substitution

The Substitute bytes stage has an S-box which is used to perform byte-by-byte substitution of the block. S-box is 8-bit wide box which is used on every byte and also S-box is a permutation of all 256 8-bit values, constructed using a transformation which treats the values as polynomials in GF (2^8) - this value is fixed. Decryption requires the inverse of the table. A simple substitution of each byte uses one S-box of 16x16 bytes containing a permutation of all 256 8-bit values. Each byte of state is replaced by byte indexed by row (left 4-bits) and column (right 4-bits)example byte {95} is replaced by byte in row 9 column 5which has value {2A}S-box constructed using defined transformation of values in GF (256)S-box constructed using a simple math formula using a non-linear function: 1/x.

As this diagram from Stallings Figure-2 shows, the Byte Substitution operates on each byte of state independently, with the input byte used to index a row/column in the table to retrieve the substituted value.

Shift rows

The Shift Rows stage creates a permutation of the data, and the other steps are involving substitutions. The state is treated as a block of columns which provides diffusion of values between columns. This performs a circular rotation on each row of 0, 1, 2 and 3 places. During decryption, circular shift is performed in opposite direction for each row. This row shift moves an individual byte from one column to another and is a linear distance of a multiple of 4 bytes, and ensures that the 4 bytes of one column are spread out to four different columns. A circular byte shift in each, 1st row is unchanged, 2nd row does 1 byte circular shift to left, 3rd row does 2 byte circular shift to left, 4th row does 3 byte circular shift to left. Decrypt inverts using shifts to right since state is processed by columns, this step permutes bytes between the columns. Stallings Figure-3 illustrates the Shift Rows permutation.

Mix columns

The Mix Columns stage is a substitution that makes use of arithmetic over GF (2^8). Each byte of a column is mapped into a new value that is a function of all four bytes in that column. It is designed as a matrix multiplication where each byte is treated as a polynomial in GF (2^8).

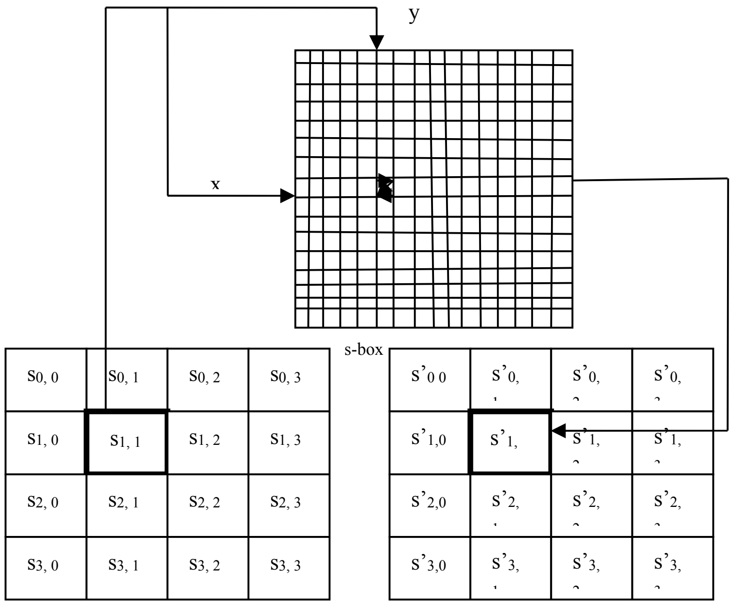

Figure-2. Byte substitution.

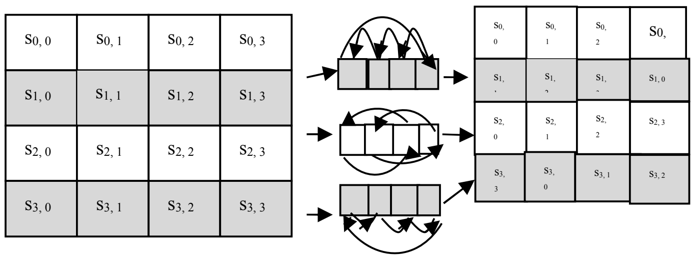

Figure-3. Shift rows.

The inverse used for decryption involves a different set of constants. The constants used are based on a linear code with maximal distance between code words - this gives good mixing of the bytes within each column. Combined with the "shift rows" step provides good avalanche, so that within a few rounds, all output bits depend on all input bits. Each column is processed separately. Each byte is replaced by a value depend on all 4 bytes in the column. Effectively a matrix multiplication in GF (2^8) using prime poly m(x) =$x^8+x^4+x^3+x+1$. Stalling Figure-4 illustrates the Mix Columns transformation.

Add round key

Add Round Key stage which is a bitwise XOR of the current block with a portion of the expanded key. In this step which makes use of the key and obscures the result. But the other steps provide confusion/diffusion/non-linearity. Hence cipher as a series of XOR with key then scramble/permute block repeated.XOR state with 128-bits of the round key. Again processed by column (though effectively a series of byte operations).Inverse for decryption identical. Since XOR own inverse, with reversed keys. For encryption these process are performed in cover image and data.

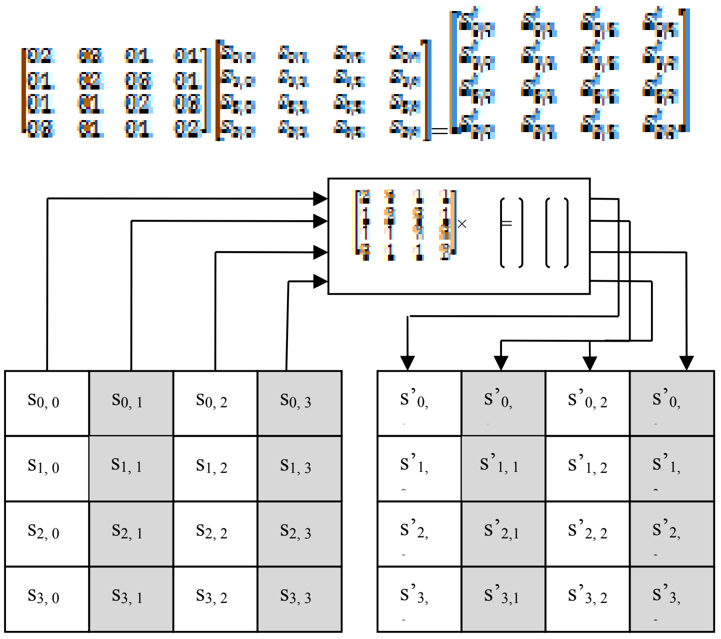

Figure-4. Mix columns transformation.

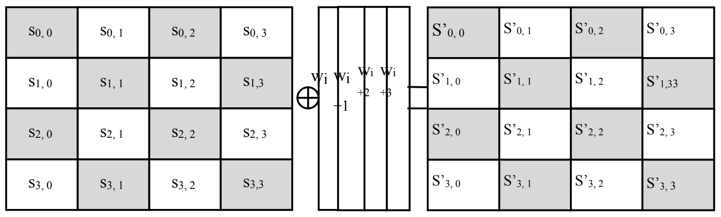

Figure-5. Add round key.

Stallings Figure-5 illustrates the Add Round Key stage, which likesByte Substitution, operates on each byte of state independently.

Embedding phase

The Embedding process can be employed by the following RDH process steps. The encrypted secret message is embedded in the image using Reversible data hiding method. The encrypted image is combined with the cover image by means of the combining the pixels. The pixels combine by means of the shifting and placing of the rows in the image. The secret image is now placed in the cover image. The Figure-6 shows the original image in 3X3 matrix format.

2	5	6
3	0	1
8	9	10

Figure-6. Cover Image 3x 3 matrix.

0	0	0	0	0
0	2	5	6	0
0	3	0	1	0
0	8	9	10	0
0	0	0	0	0

Figure-7. Space allocation of original image.

Figure illustrates the space allocation of original image and is 5X5 image. For example data is HAI, and key is DK. Convert key and HAI as bit. If change HAI as bit. The values are 100100100101001110000001. The word HAI contain each letter get 8 bits, hence the overall word get 24bit (3 X 8) and the key all covert to change the bit. The word got 16 bit (2 X 8). LSB select least values. The least values are zero. The first letter of the data is H and last letter of key was D. Each word get 8 bit. The first 4bit of the data letter and the first 4bit of key letter are combined. Hence get 8bits, and the image each pixel get 8 bit. The first least value is 0 it contains 8 bits 00000000 the first 4bit of data is 1001 an the first 4bit of key is 0001 then combine the both values and the new bit is 10010001.New bit is replaced by 00000000 that process is done in all pixels. Ten data is stored in least bit pixels
Data = Hai, Key = DE, Data bits = 100100100101001110000001, Key bits = 0001000101011011, First least pixel bit= 00000000, New combine bit=10010101, that is replace by pixel bit= 10010101, Decode (10010101) = 149.

149	0	0	0	0
0	2	5	6	0
0	3	0	1	0
0	8	9	10	0
0	0	0	0	0

Figure-8. Data embedded in the image.

Figure-8 shows the final image in which the data embedded in one pixel and that process done in each data and key. The key is reversibly stored in the data.

Extraction phase

The encrypted secret data obtained by employing the data hiding process in reverse manner. The exact shifted rows identified. From the identified rows the encrypted secret pixels obtained. The obtained pixels placed in the corresponding rows of the images and the obtained row values reconstructed into data. Thus the encrypted secret data are obtained.

Decryption phase

The decryption process is exactly reversed to obtain the secret data from the encrypted data. The same steps that are used in the encryption process is applied in the image with the new key to retrieve the secret image pixels.

IMPLEMENTATION AND RESULTS

This work used following six modules for embedding process such as 1. Preprocessing 2. Encryption 3. Embedding 4. Extraction 5.Decryption 6. Performance of process. Preprocessing includes basic steps like reading the image, resizing the image and filtering the image by applying any filtering technique. The image is retrieved from the image database and preprocessed by median filter. The median filter is a common filter which is used to remove all type of noises occurring in images. It divides the image into small matrices and it identifies the noisy pixels and replaces it with the median of the neighboring pixels and then a set of steps are applied to embedding and extracting the data. Figure-9 shows flow of data during embedding and extraction process.

VOL. 10, NO. 7, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

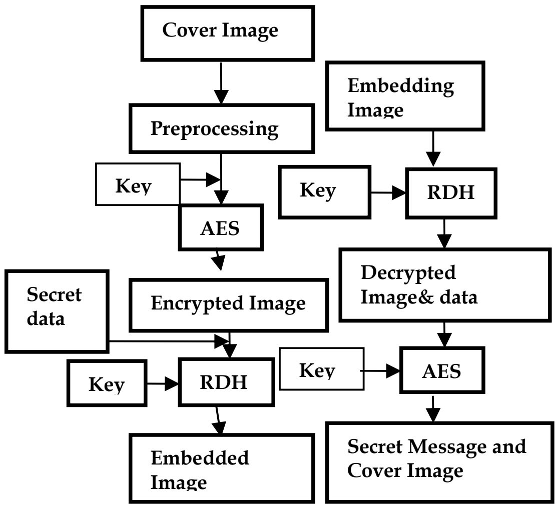

Figure-9. Data flow diagram.

After the data embedding is done, the PSNR value is calculated through MATLAB simulator. The performance of the process is measured by measuring the PSNR, MSE value. The PSNR and the MSE value show the similarity between the input secret image and the resulting image. The PSNR value should be high and the MSE value should be low and it is calculated from the Equation (1) and Equation (2).

$$PSNR = 10.\log_{10}(MAX_I^2 | MSE) \qquad (1)$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \qquad (2)$$

Where m: width of the image
n: height of the image
m*n: number of pixels

The simulation result is as followed:

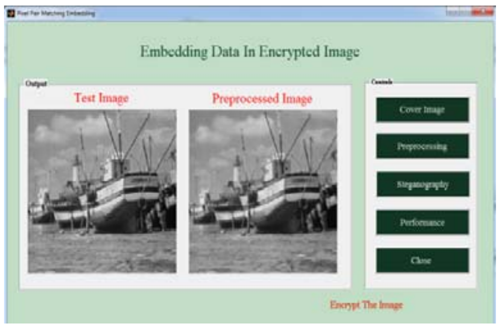

Figure-10. Cover image is selected and preprocessed.

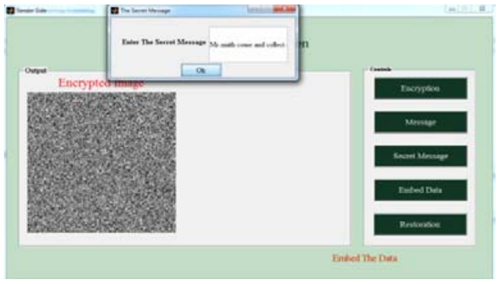

Figure-11. Cover image is encrypted and secret data is entered in textbox above the image.

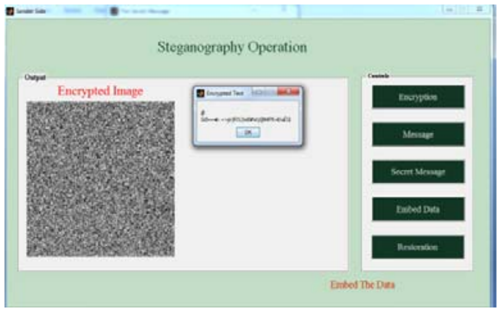

Figure-12. Secret data is encrypted by AES.

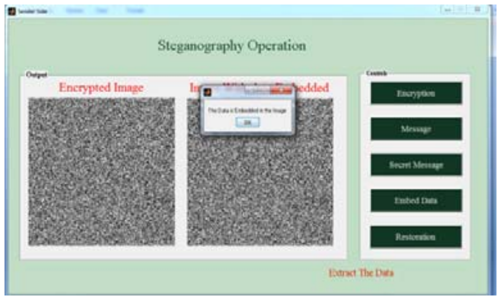

Figure-13. RDH is performed in encrypted image.

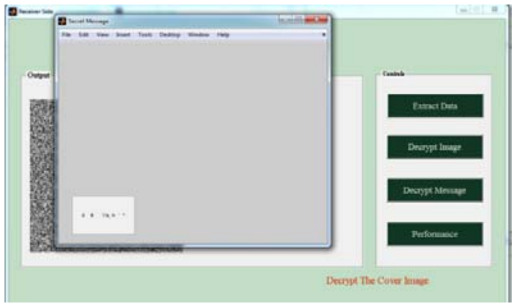

Figure-14. Encrypted data extracted.

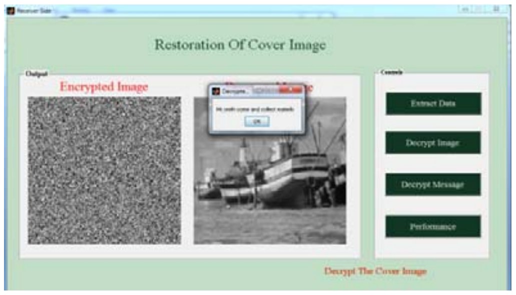

Figure-15. Original secret data and cover image is obtained by decryption and PSNR is 97.6343.

5. CONCLUSIONS

An embedding process which is based on the encryption process is employed. The encryption of the data ensures the security of the secret data and encryption of cover image is provided additional security to the data. RDH method is employed to embed the encrypted data into encrypted image. The combination of encryption and embedding process is increased the performance of the embedding process, and thus makes sure that the secret message can be obtained only by the authorized person. The AES technique is employed for the encryption of the secret image and the reversible data hiding is employed for the embedding of the encrypted bits in the image. The distortions in the images are reduced which is proved by the obtained PSNR and the MSE values. These methods also improve the embedding capacity.

REFERENCES

[1] Chi-Kwong Chan, L.M.Cheng “Hiding data in images by simple LSB substitution”, ELSEVIER, The Journal of the Pattern Recognition. Vol. 37, pp. 469-474, 2004.

[2] J. Mielikainen, “LSB matching revisited”, IEEE Signal Processing Letters. Vol. 13, No. 5, pp. 285-287, May 2006.

[3] J. Wang, Y. Sun, H. Xu, K. Chen, H. J. Kim, and S. H. Joo, “An improved section-wise exploiting modification direction method,” Signal processing. Vol. 90, No. 11, pp. 2954-2964, 2010.

[4] Ruey-Ming Chao, Hsien-ChuWu, Chih-Chiang Lee, and Yen-Ping Chu, “A Novel Image Data Hiding Scheme with Diamond Encoding”, EURASIP Journal on Information security 2009, DOI: 10.1155/2009/658047, Article ID 658047.

[5] Wien Hong and Tung-Shou Chen, “A Novel Data Embedding Method Using Adaptive Pixel Pair Matching”, IEEE Transactions on Information Forensics and Security. Vol. 7, No. 1, February 2012.

[6] X.Zhang and S.Wang, “Efficient Stenographic embedding by exploiting modification direction”, IEEE Communication Letters. Vol. 10, No. 11, pp. 781-783, November 2006.

[7] Silpa Sreekumar, Vincy Salam, “Advanced Reversible data hiding with encrypted data“, International Journal of Engineering Trends and Technology. Vol. 13, No. 7, July 2014.