www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 9 September, 2014 Page No. 8326-8331

Proficient Secret Transmission of Images And Message

[1]S.Uma, [2]V.Valarmathi
1,2 Assistant Professor, Assistant Professor
1,2 Department of CSE,
1,2 SKR ENGINEERING COLLEGE

Abstract- **The concept of data hiding mainly focuses on embedding messages within the other. Many transformations have been made in Least Significant Bit (LSB), but it is a traditional approach. Challenges still remain in storing or embedding text data in images without image distortion. The proposed technique provides a solution for data embedding along with the image security by encrypting the image with the key generated by block permutation. The process of encrypting the given image with the key makes the image to be secure in transmission. In this paper, a novel scheme for separable reversible data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction / image-recovery phases. The input image is obtained from the user and it is converted to gray scale. A key is generated to perform image encryption. Image encryption involves a sequence of steps viz. key permutation, pixels shuffle, block shuffle, height and width shuffle. The final image obtained is used for data hiding. The text to be hidden is encrypted and the position to hide the data is chosen based on the main key. When the receiver has both of the keys, the hidden data and the original image can be obtained.**

Keywords-block permutation, Encrypted image, LSB, Data Extraction, pixels, shuffle

I INTRODUCTION

A. Steganography

The term Steganography refers to the art of covert communications. Its purpose is to hide the presence of communication by embedding messages into innocuous-looking cover objects. Typically, the message is embedded within another object known as a cover Work, by tweaking its properties. The resulting output, known as a stegogramme is engineered such that it is a near identical perceptual model of the cover Work, but it will also contain the hidden message. If anybody intercepts the communication, the stegogramme can be obtained, but as it is so similar to the cover, it is a difficult task to discover the hidden message. Thus steganography ensures that the adversary regards the stegogramme - and thus, the communication - as innocuous.

In modern terms, steganography is usually implemented computationally, where cover Works such as text files, images, audio files, and video files are tweaked in such a way that a secret message can be embedded within them. The techniques are very similar to that of digital watermarking; however one big distinction must be highlighted between the two. In digital watermarking, the focus is on ensuring that nobody can remove or alter the content of the watermarked data, even though it might be plainly obvious that it exists. Steganography on the other hand, focuses on making it extremely difficult to tell that a secret message exists at all. If an unauthorized third party is able to say with high confidence that a file contains a secret message, then steganography has failed.

Steganography also differs from cryptography because the latter does not attempt to hide the fact that a message exists. Instead, cryptography merely obscures the integrity of the information so that it does not make sense to anyone but the creator and the
recipient. The adversary will be able to see that a message exists, and the inverse process of cryptanalysis involves trying to turn the meaningless information into its original form. Hence it is still highly likely that a complete steganographic system might employ cryptographic measures as a safety-net to protect the content of the message in the event that the steganography is broken.

When a steganographic system is developed, it is important to consider what the most appropriate cover Work should be, and also how the stegogramme reaches its recipient. It is possible that an image stegogramme can be sent to a recipient via email. Alternatively it may be posted on a web forum and the recipient can download the image to read the message. Although everyone can see the stegogramme, they will have no reason to expect that it is anything more than just an image. In terms of development,
Usually steganography comprises two algorithms, embedding and extracting. The embedding process is concerned with hiding a secret message within a cover Work, and is the most carefully constructed process of the two. The extracting process is traditionally a much simpler process as it is simply an inverse of the embedding process, where the secret message is revealed at the end.

B. Steganalysis

Steganalysis is the process of finding the hidden data in cover objects. The method is secure when the stego-images do not contain any detectable objects due to message embedding. The secret messages in images can be detected using the message length. Each steganographic method has an upper bound on the maximal safe message length or the bit-rate expressed in bits per pixel or sample that specifies how many bits can be safely embedded in a given image without introducing any statistically detectable artifacts. The selection of cover-images is important because it significantly influences the design of the stego system and its security. Images with a minimum colors, computer drawings, images with a distinct semantic content, such as fonts, should be avoided. The selection of the image format also makes a very big impression on the design of a secure steganographic system. Formats like BMP due to their redundancy make it to be suspicious, even though it provides the biggest space for secure steganography.

In modern years, research on signal processing in the encryption domain has been increased. In this approach, the traditional signal processing usually takes place before encryption or after decryption. When the secret data to be transmitted are encrypted, a channel provider without the knowledge of the cryptographic key may try to compress the encrypted data due to the restricted channel resource. The encrypted binary image can be compressed with a lossless manner by finding the patterns of low-density parity-check codes. With the lossy compression method, an encrypted image can be professionally compressed by neglecting the excessively rough and fine information of coefficients produced from orthogonal transform. While having the compressed data, a receiver may reconstruct the primary content of original image by retrieving the values of coefficients. Based on the homomorphic properties of the original cryptosystem, the DCT [discrete Fourier transform] in the encrypted field can be implemented. A complex signal representation method packing together a number of signal samples and processing them as a unique sample is used to reduce the complication of reckoning and the size of encrypted data.

A content owner can encrypts the original image using an encryption key which may be public/private, and a data-hider can embed additional data into the encrypted image using a data-hiding key or using any of the embedding algorithm or techniques though he does not know the original content. With an encrypted image comprising additional data, a receiver may first decrypt the message according to the encryption key, and then mine the embedded data and recover the original image according to the data-hiding key/data hiding techniques. In this method, the extraction of data is not separable from the content decryption. In other words, the added data must be extracted from the decrypted image, so that the primary content of original image is revealed before data extraction, and, if data-hiding key is known and not the encryption key, he cannot extract any information from the encrypted image containing added data.

The above methodologies have been reviewed in order to introduce a new scheme which encrypts and decrypts both image and data in a very secure manner.

II. Related Work

Chung-Ming Wang et al. [1], have proposed an approach that avoids falling-off-boundary problem by using pixel-value differencing and the modulus function. First, a difference value from two consecutive pixels by utilizing the pixel-value differencing technique (PVD) has been derived. The hiding capacity of the two consecutive pixels depends on the difference value. The authors have identified the smoother area is, the less secret data can be hidden and the more edges an area has, the more secret data can be embedded. Second, the remainder of the two consecutive pixels has been computed by using the modulus operation, and then secret data can be embedded into the two pixels by modifying their remainder. The values of the two consecutive pixels are scarcely changed after the embedding of the secret message by the optimal alteration algorithm. This scheme is secure against the RS detection attack.

Dr. EktaWalia et al.[2], have proposed a LSB based Steganography that embeds the text message in least significant bits of digital picture but it is vulnerable to even a small image manipulation. Converting an image from a format like GIF or BMP, which reconstructs the original message exactly, the lossless compression like JPEG, this does not. The reverse could destroy the information hidden in the LSBs. DCT based Steganography embed the text message in least significant bits of the Discrete Cosine (DC) coefficient of digital picture. Comparison of LSB based and DCT based stego-images using PSNR ratio shows that PSNR ratio of DCT based steganography scheme is high as compared to LSB based steganography scheme for all types of images such as gray scale, color, black and white. DCT based steganography scheme works perfectly with minimal distortion of the image quality as compared to LSB based steganography scheme. It is known that even though the amount of secret data that can be hidden using this technique is very small as compared to LSB based steganography scheme still, DCT based steganography scheme is of the minimum distortion of image quality.

Kumar et al [3], the author proposed a J2 steganography algorithm was based on hiding data in the spatial domain by making changes in the frequency domain. J2 had problems such as lower capacity along with lack of first order histogram restoration. In J3, global histogram is preserved along with higher capacity and it also includes matrix encoding techniques where the number of coefficient changes is reduced if the payload is smaller than the maximum capacity. The advantage with J3 was high capacity with full histogram restoration as compared to other existing algorithms. J3 has outperformed other algorithms in terms of detection rate. The next algorithm was J4, an improvement over J3 which not only restores the global histogram but also individual histograms. The estimation of matrix encoding enables to choose the best encoding rate and optimize in terms of minimizing the number of coefficient changes. These features are then used for training a number of cover and stego images using a SVM classifier. Based on the trained data, another set of cover and stego images are used for estimation of prediction accuracy of those images.

Mohammed A.F. Al-Husainy,[4], proposed a main a method which uses enough number of bits from each pixel in an image, take as 7-bits, to map them to 26 alphabetic English characters i.e. 'a'…'z' with some special characters that are mostly using in writing a secret message. An English message text is written by using the alphabetic characters, which hare 26 letters ('a'…'z'). Some special characters are useful in writing messages which are giving the reader a good understanding of the message. Some of these characters that are adopted in the study are, ('space character', '.', ',', '(' , ')' , '"'). Therefore, the total numbers of characters that are used to write a message become 32-characters. This means that at least 5-bits needed to represent these 23-characters in any digital system. Now, a gray scale image is using 256 gray scales for each pixel in it. This means that need (1-byte = 8-bits) per pixel is needed to produce ($2^{(8\text{-bits})}$ approx. 256) gray scales. The main operation of the algorithm is to map each 4-case from (2^7 = 128) of the 7 Most Significant Bits (MSBs) in a pixel to one of the 32-cases of the above mentioned characters in the message. The algorithm's goal for using 4-cases instead of one case is to increase the probability of finding the matched pixels in the image that are mapped to a character in the message.

W. Liu et al[5] Lossless compression of encrypted sources can be achieved through Slepian-Wolf coding. For encrypted real-world sources, such as images, the key to improve the compression efficiency is how the source dependency is exploited. Approaches in the literature that make use of Markov properties in the Slepian-Wolf decoder do not work well for grayscale images. In this correspondence, we propose a resolution progressive compression scheme which compresses an encrypted image progressively in resolution, such that the decoder can observe a low-resolution version of the image, study local statistics based on it, and use the statistics to decode the next resolution level.

Siddharth Singh and Tanveer J. Siddiqui[6] A new robust steganography algorithm based on discrete cosine transform (DCT), Arnold transform and chaotic system is proposed. The chaotic system is used to generate a random sequence to be used for spreading data in the middle frequency band DCT coefficient of the cover image. The security is further enhanced by scrambling the secret data using Arnold Cat map before embedding. The recovery process is blind. A series of experiments is conducted to prove the security and robustness of the proposed algorithm. The experimental results demonstrate that the proposed algorithm achieves higher security and robustness against JPEG compression, addition of noise, low pass filtering and cropping attacks as compared to other existing algorithms for data hiding in the DCT domain.

Xinpeng Zhang[7] a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data.With an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

C.Anuradha et al[8] A Secure and authenticated discrete reversible data hiding in cipher images deals with security and authentication. In the first phase, a content owner encrypts the original uncompressed image using an encryption key. Then, a data hider may compress the least significant bits of the encrypted image using a data hiding key to create a sparse space to accommodate some additional data. With an encrypted image containing additional data, if a receiver has the data hiding key, receiver can extract the additional data though receiver does not know the image content. If the receiver has the encryption key, can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data hiding key and the encryption key, can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large. It is also a drawback because if the receiver has any one key as known, and then he can take any one information from the encrypted data. In order to achieve authentication SHA-1 algorithm is being used.

Masoud Nosrati[9] Hiding information inside images is a popular technique nowadays. An image with a secret message inside can easily be spread over the World Wide Web or in newsgroups. The use of steganography in newsgroups has been researched by

German steganographic expert Niels Provos, who created a scanning cluster which detects the presence of hidden messages inside images that were posted on the net. However, after checking one million images, no hidden messages were found, so the practical use of steganography still seems to be limited.

To hide a message inside an image without changing its visible properties, the cover source can be altered in ”noisy” areas with many color variations, so less attention will be drawn to the modifications. The most common methods to make these alterations involve the usage of the least-significant bit or LSB, masking, filtering and transformations on the cover image. These techniques can be used with varying degrees of success on different types of image files.

X. Zhang[11] Image Encryption: psuedo-random permutation Compression of Encrypted Image discarding the excessively rough and fine information of coefficients in the transform domain Image Reconstruction: reconstruct the principle content of the original image by iteratively updating the values of the coefficients, with the help of spatial correlation in natural image.

III. PROPOSED SCHEME

The proposed scheme is comprises of image encryption, data embedding and data-decryption/image-recovery phases. The content owner can upload the gray scale image/JPEG image, an encryption key is used for the encrypting the selected image. The encrypted image is taken by the user and the additional data which needs secure transmission can be embedded inside the image using the LSB(Least Significant Bit) technique which selects the LSB position

randomly for embedding the text, which is hard for the intruders to get the secret message. After encryption the user can use any of the
secret key and then encrypt the image and text. Fig[1] shows the process of encryption and decryption.

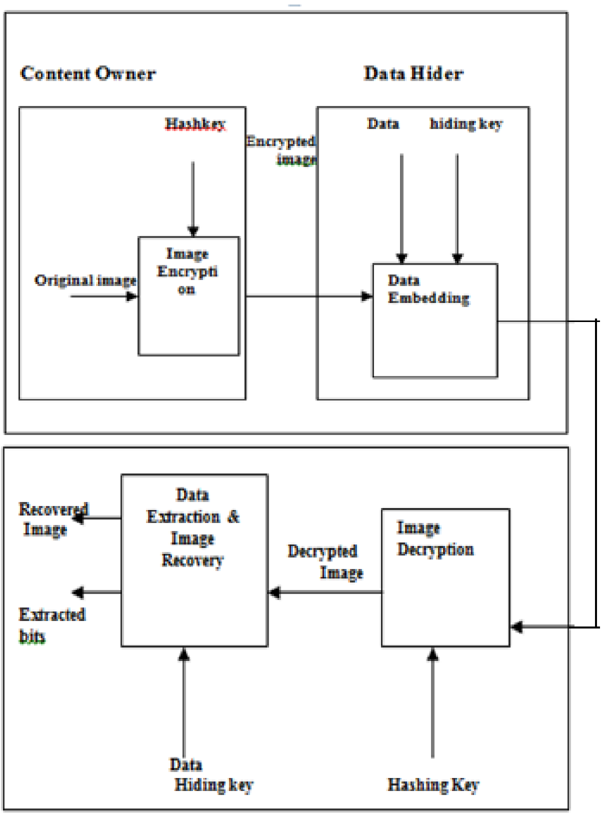


Fig 1 Process of Encryption and Decryption

At the receiver side, the data inside in the formed space can be easily retrieved from the encrypted image containing the added data according to the key used for data-hiding. As the data embedding only affects the LSB, a decryption with the encryption key obtains the image similar to that of the original image. While knowing and using both of the encryption and data-hiding keys, the embedded additional data and the encrypted image can be effectively extracted and the original image can be faultlessly recovered by exploiting the spatial correlation in natural image.

A. IMAGE ENCRYPTION

Assume the original image with a size of N1 *N2 is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits. Symbolize the bits of a pixel as bi,j,0, bi,j,1 ,……,bi,j,7 where 1<= i<=N1 and 1<=j<=N2 , the gray value as pi,j , and the number of pixels as N(N=N1*N2). That implies

$$b_{i,j,u} = \lfloor p_{i,j}/2^{u} \rfloor mod\ 2, u - 0,1, \dots\dots ,7$$

and

$$p_{i,j} = \sum_{u=0}^{7} b_{i,j,u} . 2^{u}$$

In encryption phase, the exclusive-or results of the original bits and pseudo-random bits are calculated

$$B_{i,j,u} = b_{i,j,u} \oplus r_{i,j,u}$$

Where ri,j,u are determined by an encryption key using a standard stream cipher. Then, Bi,j,u are concatenated orderly as the encrypted data.

If the blocks are very small then the objects and its edges don't appear clearly. In this block permutation the block share permuted horizontally in the image. The permutation of blocks along vertical side is also similar to horizontal side block permutation. At the receiver the original image can be obtained by the inverse permutation of the blocks. The entire array of these permuted pixels forms the encrypted image. The encrypted image obtained from the block permutation technique is transmitted to the receiver end through the insecure channel. At the receiver the encrypted image is decrypted using the same set of keys and same pseudo random index generator.
Steps:

- Get the gray scale of the image, the vales between [0, 255].
- Convert the values in binary form and store in a matrix.
- Generate a key, and this key is the main key used for both encrypting image and text.
- The generated alphabetical key is then converted to the 8 bit binary key.
- The key and the image values are performed XOR and the matrix values are reversed.
- The matrix values are swapped in case of shuffling. Swapping is performed based on the size of height and width of an image.
- The next step is the shuffled values are converted to decimal values, using this encrypted image is obtained.
- The encrypted image is divided into 4 blocks, even and odd blocks are swapped respectively.
- The last step in image encryption is array transpose is done.

These steps give a perfectly encrypted image which is tough to guess by any third party.

B. DATA HIDING ALGORITHM

The Advanced Encryption Standard (AES) is a symmetric-key block cipher and it is based on a design principle known as a substitution-permutation network. AES is a variant of Rijndael which consists of a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits.

- Key Expansion - round keys are derived from the cipher key using Rijndael's key schedule.
- Initial Round -Add Round Key - each byte of the state is combined with the round key using bitwise XOR.
- Sub Bytes - a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows - a transposition step where each row of the state is shifted cyclically a certain number of steps.
- Mix Columns - a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- Add Round Key
- Final Round (no Mix Columns)
- Sub Bytes
- Shift Rows
- Add Round Key

C. LSB ALGORITHM

Key generated to hide data in the particular pixel positions.

- The average of the ASCII values of the character is found out and that is used as initial position.
- For further attachment increment the positions based on the

average value.

D.DATA EXTRACTION

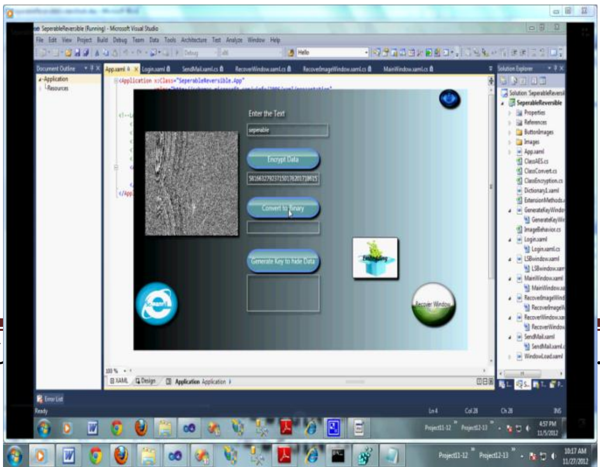
- Key to extract the data from various pixel positions is given.
- Using the key binary bits hidden in the LSB of the pixel is extracted.
- Key used to encrypt the text is given.
- Finally the text is decrypted and the original text is found out.

E.IMAGE DECRYPTION

- The first step is the image pixel values are transposed.
- The encrypted image is divided into 4 blocks, even and odd blocks are swapped respectively.
- The matrix values are swapped in case of shuffling. Swapping is performed again based on the size of height and width of an image.
- The next step is the shuffled values are regained back and converted to decimal values.
- The alphabetical key received from the sender is then converted to the 8 bit binary key.
- The key and the encrypted image values are performed XOR and the matrix values are reversed.
- Convert the values in binary form and store in a matrix.
- Get the gray scale of the image, which is of the values between [0, 255]. These steps lead to the decrypted gray scale image.

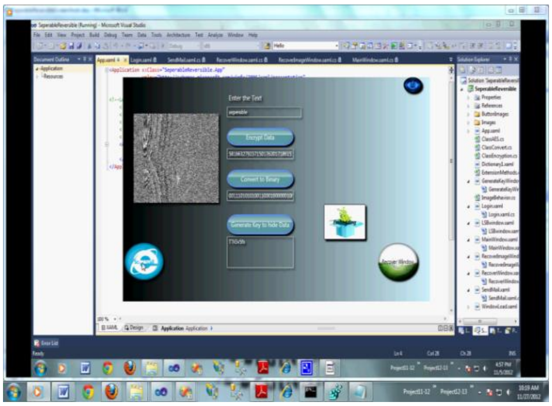
IV RESULTS AND DISCUSSION

The implementation results shows the original image uploaded by user and it is converted to gray scale. The secret key is generated and using it image is encrypted in four steps Height and width of the image is shuffled to get a perfectly encrypted image. User entered text is embedded into the encrypted image, then the process of encryption takes place. The data which is entered by the user is encrypted with the generated key and using the key data hiding positions are chosen. The data which is embedded in the encrypted image is extracted and using the respective key. After the text is extracted the original image is decrypted back using the encryption key

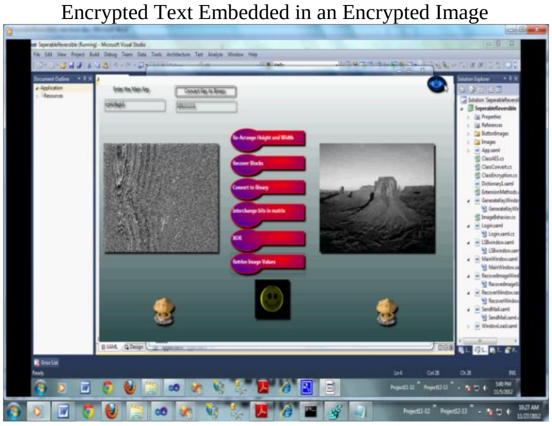

SeperableReversible (Running) - Microsoft Visual Studio
Enter the Text
seperable
Encrypt Data
Convert to Binary
Generate Key to hide Data
Solution Explorer

Uploaded Image converted to Gray Scale

Key Generated and used for Image Encryption


SeperableReversible (Running) - Microsoft Visual Studio
Enter the Text
seperable
Encrypt Data
Convert to Binary
Generate Key to hide Data
Solution Explorer

Encrypted Text Embedded in an Encrypted Image


SeperableReversible (Running) - Microsoft Visual Studio
Re-Arrange Height and Width
Recover Blocks
Convert to Binary
Interchange bits in matrix
XOR
Switches Image Values
Solution Explorer

Original Image after Decryption

V CONCLUSION

A separable reversible data hiding scheme for encrypted image is proposed, which comprises of image encryption, data embedding and data-extraction / image-recovery phases. In the first part, the user needed to login for authentication. The input image is obtained from the user and it is converted to gray scale. The key which is generated is used for image encryption. Image encryption is carried out in sequence of steps viz. key permutation, pixels shuffle, block shuffle, height and width shuffle. Then that image was used for data hiding. The text was first encrypted then position to hide the data was chosen based on the main key. This was done with the help of compressing data of certain particular colors and by collecting the redundant bits in an image. So with an encrypted image containing additional data, the receiver will extract the secret data using only the key which is used for data-hiding, also obtain an image similar to the original version one using only the encryption key. When the receiver has both the keys, both the data and the original image can be acquired. In the future, a comprehensive combination of image encryption and data hiding compatible with lossy compression deserves further investigation.

This work can be extended for large amount of data embedding along with the image security by encrypting the image with a key image. The noise during the retrieval of the image is a challenging issue which is to be noted. Future work should center on development of stronger embedding algorithms whose output can survive image manipulations and those that can make use of more permanent embedding procedures. This will facilitate the use of steganography in more sensitive application areas like in computer digital forensics and in enhancement of security in electronic commerce and trading applications.

REFERENCES

[1]Chung-Ming Wang, Nan-I Wu, Chwei-Shyong Tsai, Min-Shiang Hwang, “A high quality steganographic method with pixel-value differencing and modulus function”, Elsevier, Journal of Systems and Software, Volume 81, Issue 1, January 2008, Pages 150–158.

[2]Dr. EktaWalia, Payal Jain, Navdeep, “An Analysis of LSB & DCT based steganography”, Global Journal of Computer Science and Technology, P a g e | 4 Vol. 10 Issue 1 (Ver 1.0), April 2010.

[3]Kumar, Mahendra, ProQuest® Dissertations & Theses, “Steganography and steganalysis of Joint Picture Expert Group (JPEG) images”, DAI/B 73-06, p., Feb 2012.

[4]Mohammed A.F. Al-Husainy, “Image Steganography by Mapping Pixels to Letters”, Journal of Computer Science, Science Publications 5 (1): 33-38, 2009.

[5]W. Liu, W. Zeng, L. Dong, and Q. Yao, “Efficient compression of en- crypted grayscale images,” IEEE Trans. Image Process., vol. 19, no. 4,pp. 1097–1102, Apr. 2010.

[6]Siddharth Singh and Tanveer J. Siddiqui, “A Security Enhanced Robust Steganography algorithm for Data Hiding”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012.

[7]Xinpeng Zhang, “Separable Reversible Data Hiding in Encrypted Image”, IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, april 2012.

[8] C Anuradha, S Lavanya - Secure and Authenticated Reversible Data Hiding in Encrypted Image International Journal, 2013 - ijarcsse.com

[9] Masoud Nosrati * Ronak Karimi Mehdi Hariri “An introduction to steganography methods” World Applied Programming, Vol (1), No (3), August 2011. 191-195 ISSN: 2222-2510 ©2011 WAP journal. www.waprogramming.com

[10]http://www.tifr.res.in/~sanyal/papers/A_Survey_on_Various_ Data_ Hiding_ Techn- iques_and_Their_Comparative_Analysis

[11]http://eprints.utm.my/4339/1/71847

[12] X. Zhang, “Lossy compression and iterative reconstruction for encrypted image,” IEEE Trans. Inform. Forensics Security, vol. 6, no. 1, pp. 53–58, Feb. 2011