ENHANCING SECURITY ON MOSAIC IMAGES USING CRYPTOGRAPHIC ALGORITHM AND COMPRESSION TECHNIQUE

Nilam Gandhe[1], Vikash Kumar[2]

[1] *Computer Science And Engineering, A.C.E.Wardha, Maharashtra, India*
[2] *Computer Science And Engineering, A.C.E.Wardha, Maharashtra, India*

ABSTRACT

A new technique is proposed for secure image transmission, which automatically transforms large-volume secret image into secret-fragment-visible mosaic image of the same size. The mosaic image is obtained by dividing the secret image into fragments and transforming their color characteristics to be those of the corresponding blocks of the target image. It looks similar to an arbitrarily selected target image and may be used as a camouflage of the secret image. We proposed a approach to enhance security of the mosaic image by providing better Encryption logic which provide faster encryption. And for the sake of transmission we proposed JPEG lossless compression technique which provide good compression ratio.

Keyword : - *Image Encryption, Decryption, Mosaic Image, Image Compression, jpeg.*

1. INTRODUCTION

Today, for various applications images are frequently utilized and transmitted from various sources through the internet, these images usually contain secret personal information so they should be protected from leakages during transmissions. Many methods have been proposed for securing image transmission, two common methods are image encryption and data hiding.

In the process of data encryption like images the encrypted image is a noise image so that no buddy can obtain the secret image from it without the correct key. However, the encrypted image is called a meaningless image, which do not give additional information before its decryption and may arouse an attacker's attention at the time of transmission because its in shuffle form. The other method is Data hiding which used to avoid this problem [6] that hides a secret message into a cover image because of that no one can realize the existence of the secret data.

A main problem of the method for hiding data in images is the difficulty to embed a large amount of message data into a single image. In case, if one wants to hide a secret image into a cover image with the same size, the secret image must be highly compressed in advance. But, for many applications, such as keeping or transmitting medical pictures, military images, legal documents, etc., such type of data compression technique are usually impractical. The user cannot select freely his/her favorite image for use as the target image. Therefore in this study to remove this drawback of the method while keeping its merit, it is needed to design a new method that can transform a secret image into a secret-fragment- visible mosaic image of the same size that has the visual appearance of any freely selected target image without the need of a database [2].

A new technique is use for secure image transmission, that transforms a secret image into a meaningful mosaic image with the same size and looks like a preselected target image. The given secret image is first divided into rectangular fragments called tile images, which are then fit into similar blocks in the target image, called target blocks, according to a similarity criterion based on color variations. Next, the color characteristic of each tile image

is transformed to that of the corresponding target block in the target image, resulting in a mosaic image which looks like the target image. For loading and transferring of image efficiently the lossless compression is applied on mosaic image. And encryption algorithm [8] to improve the security of mosaic image so any one cryptographic algorithm is applied on compressed mosaic image to securely transferring.

2. LITRATURE REVIEW

The author in this paper[1] is shows a technique for the transmission of the secret image securely with no loss. This method convert the secret image into a mosaic tile image having the same size looking like that of the target image which is preselected from a database. This color transformation is controlled and the secret image is get back without loss from the mosaic tile image with the help of the extracted relevant information which are generated for the recovery of the image.

The author in this paper[2] is presented A new type of computer art image called secret-fragment-visible mosaic image , which is created automatically from a given secret image which are divided into a small fragments and by composing all those small fragments to become a target image in a mosaic form, an effect of embedding the given image visibly but secretly in the resulting mosaic image. To create this type of mosaic image from a given secret color image, need to transformed the 3-D color space into a new 1-D color scale, based on finding the similarity of a new image for selecting a target image from a database that is the most similar to the given secret image.

The author in this paper[3] is describes a method for a more general form of color correction that receive one image's color characteristics from another. They described core strategy in every way is to choose or select an appropriate color space and then to apply simple operations there. When a typical three channel image is described in any of the most conventional color spaces, there will be correlations between the different channels' values.

The author in this paper[4] is describes a new image encryption scheme using a secret key of 144-bits is proposed. In the substitution process of the scheme, image is divided into blocks and subsequently into color components. Each color component is modified by performing bitwise operation which depends on secret key as well as a few most significant bits of its previous and next color component.

The author in this paper[5] is describes JPEG: Still Image Data Compression Standard Here, W. B. Pennebaker tries to explain that the main impediment in many applications is the perceived length of data required to represent a digital image. For this we would need an image compression standard to maintain the quality and clarity of the images after compression. To meet all the needs of the JPEG standard for image compression includes two basic methods having different operation modes: A predictive method for "lossless" compression and a DCT method for "lossy" compression.

3. PROPOSED WORK

In the first phase, a mosaic image is yielded, which consists of the fragments of an input secret image with color corrections according to a similarity criterion based on color variations.

Fig- 1. In the proposed method. (a) Secret image. (b) Target image. (c) mosaic image created from (a) and (b) by the proposed method.

These phase includes Three stages: 1) fitting the tile images of the secret image into the target blocks of a preselected target image; 2) converting the color characteristic of each tile image in the secret image to the corresponding target block in the target image; 3) rotating each tile image into a different direction with the minimum RMSE value with respect to its corresponding target block. In second phase we will perform lossless compression on mosaic image and then perform the encryption on compressed image then we transmit the image securely. In the Third phase, the secret image is recover nearly losslessly from the generated mosaic image. These phase includes two stages: : 1) we will first decrypt the encrypted image and 2)perform decompression on image for recovering the secret image from mosaic image.

The proposed method includes phases as shown by the flow diagram.

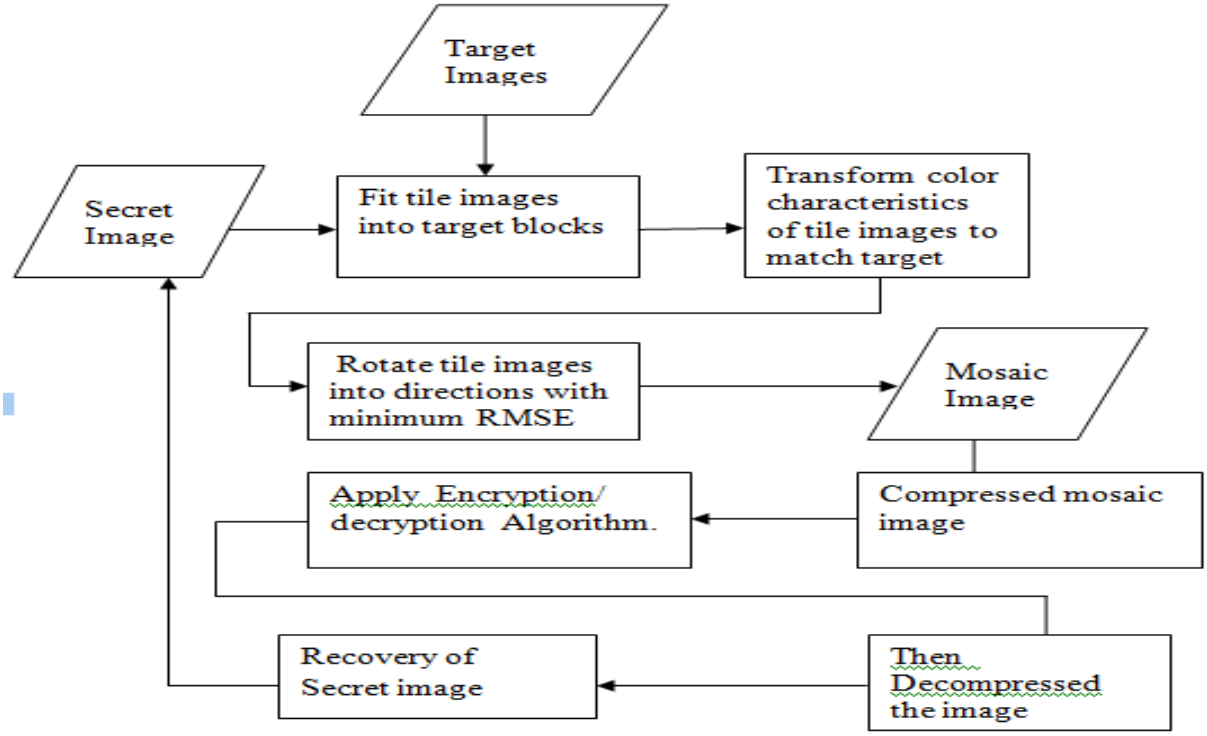

Fig-2. Flow Diagram of Proposed Method

4. ALGORITHMS OF THE PROPOSED METHOD

4.1 ALGORITHM 1 MOSAIC IMAGE CREATION

Input: a secret image *S*, a target image *T*, and a secret key *K*.
Output: a secret-fragment-visible mosaic image *F*.
Steps:
Stage 1. fitting the tile images into the target blocks.
Step 1. If the size of the target image *T* is different from that of the secret image *S*, change the size of *T* to be identical to that of *S*; and divide the secret image *S* into *n* tile images *{T*1, *T*2, . . . , *Tn}* as well as the target image *T* into *n* target blocks *{B*1, *B*2, . . . , *Bn}* with each *Ti* or *Bi* being of size *NT* .
Step 2. Compute the means and the standard deviations of each tile image *Ti* and each target block *Bj* for the three color channels according to (1) and (2); and compute accordingly the average standard deviations for *Ti* and *Bj* , respectively, for *i* = 1 through *n* and *j* = 1 through *n*.
Step 3. Sort the tile images in the set *Stile = {T*1, *T*2, . . . , *Tn}* and the target blocks in the set *Starget = {B*1,*B*2, . . . , *Bn}* according to the computed average standard deviation values of the blocks; map in order the blocks in the sorted

Stile to those in the sorted *Starget* in a 1-to-1 manner; and reorder the mappings according to the indices of the tile images, resulting in a *mapping sequence L* of the form: $T1 \rightarrow Bj1$, $T2 \rightarrow Bj2$, . . . , $Tn \rightarrow Bjn$.
Step 4. Create a mosaic image *F* by fitting the tile images into the corresponding target blocks according to *L*.
Stage 2. performing color conversions between the tile images and the target blocks.
Step 5. Create a *counting table TB* with 256 entries, each with an index corresponding to a residual value, and assign an initial value of zero to each entry (note that each residual value will be in the range of 0 to 255).
Step 6. For each mapping $Ti \rightarrow Bji$ in sequence *L*, represent the means μc and $\mu\_ c$ of *Ti* and *Bji*, respectively, by eight bits; and represent the standard deviation quotient *qc* appearing in (3) by seven bits, according to the scheme described in Section III(A) where *c* = *r*, *g*, or *b*.
Step 7. For each pixel *pi* in each tile image *Ti* of mosaic image *F* with color value *ci* where *c* = *r*, *g*, or *b*, transform *ci* into a new value *c__i* by (3); if *c__ i* is not smaller than 255 or if it is not larger than 0, then change *c__ i* to be 255 or 0, respectively; compute a residual value *Ri* for pixel *pi* and increment by 1 the count in the entry in the counting table *TB* whose index is identical to *Ri*.
Stage 3. rotating the tile images.
Step 8. Compute the RMSE values of each color transformed tile image *Ti* in *F* with respect to its corresponding target block *Bji* after rotating *Ti* into each of the directions θ =0o, 90o, 180o and 270o; and rotate *Ti* into the *optimal* direction θo with the smallest RMSE value.
Stage 4. embedding the secret image recovery information.
Step 9. Construct a Huffman table *HT* using the content of the counting table *TB* to encode all the residual values computed previously.
Step 10. For each tile image *Ti* in mosaic image *F*, construct a bit stream *Mi* for recovering *Ti* including the bit-segments which encode the data items of: 1) the index of the corresponding target block *Bji*; 2) the optimal rotation angle $\theta°$ of *Ti*; 3) the means of *Ti* and *Bji* and the related standard deviation quotients of all three color channels; and 4) the bit sequence for overflows/underflows with residuals in *Ti* encoded by the Huffman table *HT* constructed in Step 9.
Step 11. Concatenate the bit streams *Mi* of all *Ti* in *F* in a raster-scan order to form a total bit stream *Mt* ; use the secret key *K* to encrypt *Mt* into another bit stream *M_ t* ; and embed *M_ t* into *F* by the reversible contrast mapping scheme proposed in [9].
Step 12. Construct a bit stream *I* including: 1) the number of conducted iterations *Ni* for embedding *M_t*; 2) the number of pixel pairs *Npair* used in the last iteration; and 3) the Huffman table *HT* constructed for the residuals; and embed the bit stream *I* into mosaic image *F* by the same scheme used in Step 11.

4.2 ALGORITHM 2 SECRET IMAGE RECOVERY

Input: a mosaic image *F* with *n* tile images *{T*1, *T*2, . . . ,*Tn}* and the secret key *K*.
Output: the secret image *S*.
Steps:
Stage 1. extracting the secret image recovery information.
Step 1. Extract from *F* the bit stream *I* by a reverse version of the scheme proposed in [9] and decode them to obtain the following data items: 1) the number of iterations *Ni* for embedding *M_ t* ; 2) the total number of used pixel pairs *Npair* in the last iteration; and 3) the Huffman table *HT* for encoding the values of the residuals of the overflows or underflows.
Step 2. Extract the bit stream *M_t* using the values of *Ni* and *Npair* by the same scheme used in the last step.
Step 3. Decrypt the bit stream *M_ t* into *Mt* by *K*.
Step 4. Decompose *Mt* into *n* bit streams *M*1 through *Mn* for the *n to-be constructed* tile images *T*1 through *Tn* in *S*, respectively.
Step 5. Decode *Mi* for each tile image *Ti* to obtain the following data items: 1) the index *ji* of the block *Bji* in *F* corresponding to *Ti*; 2) the optimal rotation angle $\theta°$ of *Ti*; 3) the means of *Ti* and *Bji* and the related standard deviation quotients of all color channels; and 4) the overflow/underflow residual values in *Ti* decoded by the Huffman table *HT*.
Stage 2. recovering the secret image.
Step 6. Recover one by one in a raster-scan order the tile images *Ti*, *i* = 1 through *n*, of the desired secret image *S* by the following steps: 1) rotate in the reverse direction the block indexed by *ji*, namely *Bji*, in *F* through the optimal angle $\theta°$ and fit the resulting block content into *Ti* to form an *initial* tile image *Ti*; 2) use the extracted means and related standard deviation quotients to recover the original pixel values in *Ti* according to (4); 3) use the extracted means, standard deviation quotients, and (5) to compute the two parameters *cS* and *cL*; 4) scan *Ti* to find out pixels with values 255 or 0 which indicate that overflows or underflows, respectively, have occurred there; 5)add

respectively the values *cS* or *cL* to the corresponding residual values of the found pixels; and 6) take the results as the final pixel values, resulting in a *final* tile image *Ti*.
Step 7. Compose all the final tile images to form the desired secret image *S* as output.

4.3. CRYPTOGRAPHYC ALGORITHM BY AES ENCRYPTION

AES is based on a design principle known as a substitution-permutation network[7], combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.
AES operates on a 4×4 column-major order matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field[3].
The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext. The number of cycles of repetition are as follows:

- 10 cycles of repetition for 128-bit keys.
- 12 cycles of repetition for 192-bit keys.
- 14 cycles of repetition for 256-bit keys.

4.4. PERFORMING COMPRESSION ON MOSAIC IMAGE

JPEG uses a lossy form of compression based on the discrete cosine transform (DCT)[12]. A perceptual model based loosely on the human psycho visual system discards high-frequency information and color hue. In the transform domain, the process of reducing information is called quantization. In simpler terms, quantization is a method for optimally reducing a large number scale (with different occurrences of each number) into a smaller one, and the transform-domain is a convenient representation of the image because the high-frequency coefficients[11], which contribute less to the overall picture than other coefficients, are characteristically small-values with high compressibility. The quantized coefficients are then sequenced and losslessly packed into the output bit stream. Nearly all software implementations of JPEG permit user control over the compression-ratio allowing the user to trade off picture-quality for smaller file size.

5. EXPERIMENTAL RESULTS

Fig-3. Select Target Image

Fig-4. Select Secret Image

Fig-5. Created Mosaic Image

Fig-6. Recreated Secret Image

To maintain the clarity of recreated secret image we need to maintain the PNSR value for that purpose we calculated the MSE and Average Distance of original secret image and recreated secret image.

Chart -1. Comparison of MSE value

Chart -2. Comparison of Average Distance

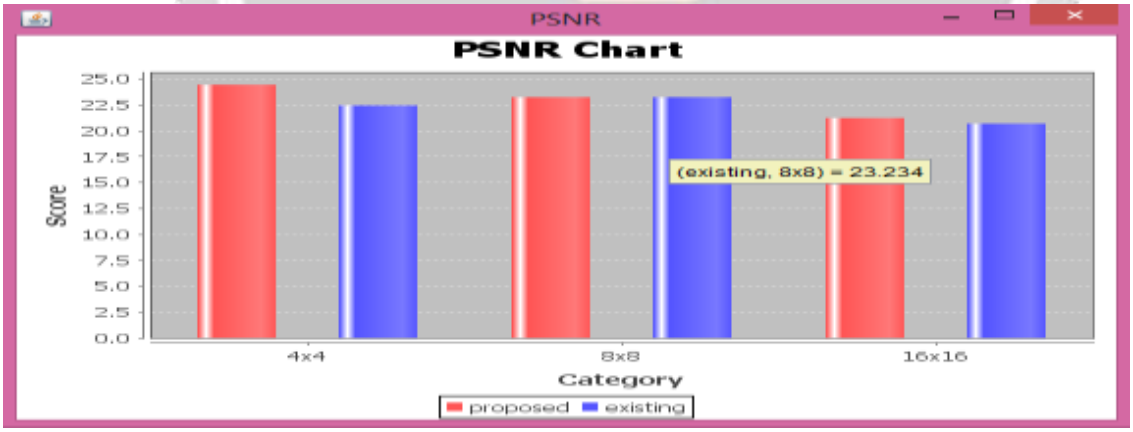

Chart -3. Comparison of PNSR Value

6. CONCLUSION

The proposed method to securely transmit a secret image, which can create mosaic images which also can transform a secret image into a mosaic image with the same size of data for concealing the secret image. The technique encryption algorithm is use to improve the security of mosaic image and also there is need to transfer an image by compressing it allows to loading and transferring it in an efficient form and to recover it with minimum loss. We maintain PSNR ratio of the recreate Image. In future studies ,We Can implement this approach for creation of mosaic video , which give more security in video stenography**.**

7. REFERENCES

[1] Ya-Lin Lee, Wen-Hsiang Tsai, "A New Secure Image Transmission Technique via Secret-fragment-Visible Mosaic Images by Nearly Reversible Color Transformations," IEEE Transactions on Circuits and systems for video Technology, vol. 24, no. 4, April 2014.

[2] I. J. Lai and W. H. Tsai, "Secret-fragment-visible mosaic image—A new computer art and its application to information hiding," *IEEE Trans. Inf.Forens. Secur.*, vol. 6, no. 3, pp. 936–945, Sep. 2011.

[3] E. Reinhard, M. A shikhmin, B. Gooch, and P. Shirley, "Color transfer between images," IEEE Comput. Graph. Appl., vol. 21, no. 5, pp. 34–41,Sep.–Oct. 2001.

[4] Narendra K Pareek "design and analysis of a novel digital Image encryption scheme," International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012

[5] W. B. Pennebaker and J. L. Mitchell, "JPEG: Still Image Data Compression Standard", New York, NY, USA: Van Nostrand Reinhold,pp. 34–38, 1993.

[6] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2775–2785, Jul. 2013.

[7] V. Patidar, N. K. Pareek, G. Purohit, and K. K. Sud, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," *Opt. Commun.*, vol. 284, no. 19, pp. 4331–4339, 2011.

[8] Ismail Amr Ismail, Mohammed Amin and Hossam Diab, (2010) "A digital image encryption algorithm based a composition of two chaotic logistic map", International Journal of Network Security, Vol. 11, No. 1, pp. 1-10.

[9] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr. 2007.

[10] C. H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution," *Pattern Recog.*, vol. 41, no. 8,pp. 2674–2683, 2008.

[11] D. Xiao, X. Liao, and P. Wei, "Analysis and improvement of a chaos based image encryption algorithm," *Chaos Solit. Fract.*, vol. 40, no. 5,pp. 2191–2199, 2009.

[12] C. C. Chang, C. C. Lin, C. S. Tseng, and W. L. Tai, "Reversible hiding in DCT-based compressed images," *Inf. Sci.*, vol. 177, no. 13, pp. 2768–2786, 2007.