MAES BASED IMAGE TRANSMISSION TECHNIQUE FOR IMPROVING THE LEVEL OF SECURITY

Sasikala.Y[1], Riyazuddin.K[2]

[1]*M.Tech Scholar, E.C.E, AITS, Rajampet, Andhra Pradesh,(India)*

[2]*Assistant Professor, E.C.E, AITS, Rajampet, Andhra Pradesh,(India)*

ABSTRACT

The security and performance analysis of encryption has been performed using the histograms, correlation coefficients, information entropy, key sensitivity analysis, differential analysis, key space analysis, encryption/decryption rate analysis etc. A new secure image transmission technique is proposed, which transforms automatically a given large-volume secret image into a secret-fragment-visible mosaic image of same size. A scheme of handling the overflows/underflow in the converted pixel color values by recording the color differences in the untransformed color space is also proposed. The information required for recovering the secret image is embedded into the created mosaic image by a lossless data hiding scheme using a key. In this proposed method the key standards depends on the modification to the Advanced Encryption Standards(MAES) to reflect a high level security and better image encryption

Keywords: Color Transformation, Data Hiding, Image Encryption, Mosaic Image, Secuer Image Transmission

I.INTRODUCTION

Currently it is an important ascept to protect confidential data from unauthorized access. Multimedia content may be text audio, still images, animation and video. The image from various sources are frequently utilized and transmitted through the internet for various applications ,such as medical imaging system, and military image databases. These images usually contain confidential information so that they should be protected from leakages during transmission. There are two types of techniques used for secure image transmission .They are image encryption and data hiding . Image encryption uses natural property of an image, such as high redundancy and strong spatial correlation. Shannon's confusion and diffusion properties are used get an encrypted image, The encrypted image contains noise, the attacker's know the correct key they didn't get secret image. The encrypted image is a noise image so that no one can obtain. The secret image from it unless he/she has the correct key. However, the encrypt image is a meaningless file, which cannot provide additional information before decryption and may arouse an attacker's attention during transmission due to its randomness in form .An alternative to avoid this problem is data hiding that hides a secret message into a cover images so that no one can realize the existence of the secret data, in which the data type of the secret message investigated in this paper is an image .Existing data hiding methods mainly utilize the techniques of LSB substitution, histogram shifting, difference expansion, prediction-error expansion, recursive histogram modification and discrete cosine/wavelet transformations. However, in order to reduce the distortion of the resulting image, an upper bound for the distortion value is usually set on the payload of the cover image. In this paper, a new technique

for secure image transmission is proposed, which transforms a secret image into a meaningful mosaic image with the same size and looking like a preselected target image. The transformation process in controlled by a secret key, and only with the key can a person recover the secret image nearly lossless form the mosaic image. The proposed method is inspired by Lai and Tsai, in which a new type of computer at image, called secret-fragment-visible mosaic image, was proposed. The mosaic image is the result of rearrangement of the fragments of a secret image in disguise of another image called to target image preselected from a database.

fig 1.Result yield by the proposed method. (a) Secret image. (b) Target image. (c) Secret-fragment-visible mosaic image created from (d) By the proposed method.

But an obvious weakness of Lai and Tsai is the requirement of a large image database so that the selected target image. Using their method, the user is not allowed to select freely his/her favourite image for use as the target image. It is therefore desired in this study to remove this weakness of the method, while keeping in merit, that is, it is aimed 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. The proposed method is new in that a meaningful mosaic image is created in contrast with the image encryption method that only creates meaningless noise images. Also, the proposed method can transform a secret image into a distinguishing mosaic image without compression,

II.IDEAS OF THE PROPOSED METHOD

The proposed method includes two main phases as shown by the flow diagram of fig.

1) Mosaic image creation and
2) Secret image recovery.

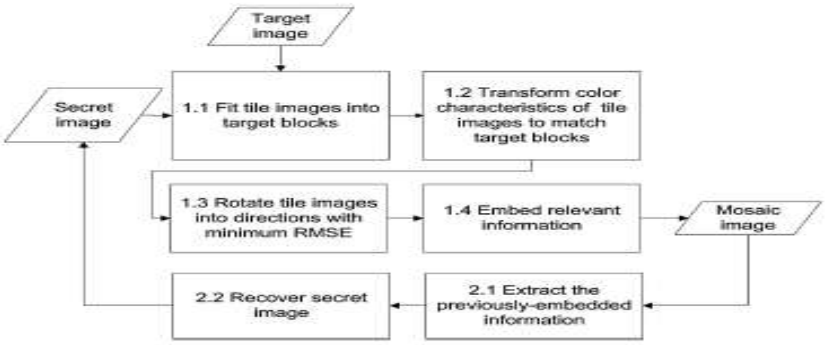

Fig 2: Flow diagram of the proposed method

In the first phase, a mosaic image is yielded, which consists of the fragments of an input secret image with colour corrections according to a similarity criterion based on colour variations. The phases includes four stages

1.Fitting the tile images of the secret image into the target blocks of a preselected target images.

2.Transforming the colour characteristic of each tile image in the secret image to become that of the corresponding target block in the target images.

3.Rotating each tile image into a direction with the minimum RMSE value with respect to its corresponding target block;

4.Embedding relevant information into to created mosaic image for future recovery of the secret image. In the second phase, the embedded information is extracted to recovery nearly losslessly the secret image from the generated mosaic image. The phase includes two stages.1) Extracting the embedded information for secret image recovery from the mosaic image and 2) Recovering the secret image using the extracted information.

III.IDEAS OF MOSAIC IMAGE GENERATION

Problems encounterd in generating mosaic images are discussed in this section with solution to them proposed.

3.1 Color Transformation between blocks

In the first phase of the proposed method, each tile image *T* in the given secret image is fit into a target block *B* in a preselected target image. Since the color characteristics of *T* and *B* are different from each-other , how to change their color distributions to make them look alike is the main issue here. Reinhard *et al*. proposed a color transfer scheme in this aspect, which converts the color characteristic of an image to be that of another in the l$\alpha\beta$ color space. This idea is an answer to the issue and is adopted in this paper, except that the RGB color space instead of the l$\alpha\beta$ one is used to reduce the volume of the required information for recovery of the original secret image. More specifically, let *T* and *B* be described as two pixel sets $\{p_1,p_2,p_3,\}$,and $\{p'_1,p'_2,p'_3,\}$, respectively. Let the color of each *pi* be denoted by (*ri*, *gi*, *bi*) and that of each*p'i* by (*r'i*, *g'i*, *b'i*). At first, we compute the means and standard. deviations of *T* and *B*, respectively, in each of the three color channels R, G, and B by the following formulas:

$$\mu_C = 1/n\sum_{i=1}^{n} ci \qquad \mu_C = 1/n\sum_{i=1}^{n} ci' \qquad (1)$$

$$\sigma_C = \sqrt{1/n\sum_{i=1}^{n}(ci - \mu i)^2} \qquad \sigma'_C = \sqrt{1/n\sum_{i=1}^{n}(ci' - \mu i')^2} \qquad (2)$$

in which *ci* and c_i' denote the C-channel values of pixels *pi* and *pi'*, respectively, with *c* = *r*, *g*, or *b* and C=R, G,or B.Next, we compute new color values ($\acute{r}i$, $\acute{g}i$, $\acute{b}i$). for each *pi* in *T* by 2

$$c_{i''} = q_c(c_i - \mu_c) + \mu_c' \qquad (3)$$

we use the following formula which is the inverse of (3):

$$ci = (1/qc)(ci'' - \mu_c'') + \mu_c' \qquad (4)$$

Further more, we have to embed into the created mosaic image sufficient information about the new tile image *T'* for use in the later stage of recovering the original secret image. For this, theoretically we can use (4) to compute the original pixel value of p_i. However, the involved mean and standard deviation values in the formula are all real numbers, and it is impractical to embed real numbers, each with many digits, in the generated mosaic image. Therefore, we limit the numbers of bits used to represent relevant parameter values in (3) and (4). Specifically, for each color channel we allow each of the means of *T* and *B* to have 8 bits with its value in the range of 0 to 255, and the standard deviation quotient q_c in (3) to have 7 bits with its value in the range of 0.1 to 12.8. That is, each mean is changed to be the closest value in the range of 0 to 255, and each q_c is changed to

be the closest value in the range of 0.1 to 12.8. We do not allow *qc* to be 0 because otherwise the original pixel value cannot be recovered back by (4) for the reason that 1/*qc* in (4) is not defined when *qc* =0.

3.2 Choosing Appropriate Target Blocks And Rotating Bloc To Fit Better With Smaller Rmse Value

In transforming the color characteristic of a tile image *T* to be that of a corresponding target block *B* as described above, how to choose an appropriate *B* for each *T* is an issue. For this, we use the standard deviation of the colors in the block as a measure to select the most similar *B* for each *T*. Specially, we sort all the tile images to form a sequence, S_{tile}, and all the target blocks to form another, S_{target} , according to the average values of the standard deviations of the three color channels. Then, we fit the first in S_{tile} into the first in S_{target}, fit the second in S_{tile} into the second in S_{target} , and so on. Additionally, after a target block *B* is chosen to fit a tile image *T* and after the color characteristic of *T* is transformed, we conduct a further improvement on the color similarity between the resulting tile image *T'* and the target block *B* by rotating *T* ' into one of the four directions, 0^o, 90^o, 180^o, and 270^o, which yields a rotated version of *T*" with the minimum root mean square error (RMSE) value with respect to *B* among the four directions for final use to fit *T* into *B*.

3.3 Hndling Overflows/Underflows in Color Transformation

After the color transformation process is conducted as described previously, some pixel values in the new tile image *T* ' might have overflows or underflows. To deal with this problem, we convert such values to be non-overflow or non underflow ones and record the value differences as residuals for use in later recovery. Specifically, we convert all the transformed pixel values in *T'* not smaller than 255 to be 255, and all those not larger than 0 to be 0. Next, we compute the differences between the original pixel values and the converted ones as the residuals and record them as part of the information associated with *T*" Accordingly, the pixel values, which are just on the bound of 255 or 0, however, cannot be distinguished from those with overflow/underflow values during later recovery since all the pixel values with overflows/underflows are converted to be 255 or 0 n. To solve this problem, we record the residual values in the untransformed color space rather than in the transformed one. That is, by using the following two formulas, we compute first the smallest possible color value *cS* (with *c* = *r*, *g*, or *b*) in *T* that becomes larger than 255, as well as the largest possible value *cL* in *T* that becomes smaller than 0, respectively, after the color transformation process has been conducted

$$C_S=[(1/q_c)(225-\mu_c')+\mu_c] \quad (5)$$

$$C_S=[(1/q_c)(225-\mu_c')+\mu_c]$$

Next, for an untransformed value c_i which yields an overflow after the color transformation, we compute its residual as$|c_i - c_S|$; and for c_i which yields an underflow, we compute its residual as $|c_L - c_i|$. Then, the possible values of the residuals of c_i will all lie in the range of 0 to 255 as can be verified. Consequently, we can simply record each of them with 8 bits. And finally, because the residual values are centralized around zero, we use further in this study the Huffman encoding scheme to encode the residuals in order to reduce the number of required bits to represent them.

3.4 Embedding Information For Secret Image Recovery

In order to recover the secret image from the mosaic image, we have to embed relevant recovery information into the mosaic image. For this, we adopt a technique proposed by Coltuc and Chassery and apply it to the least significant bits of the pixels in the created mosaic image to conduct data embedding. Unlike the classical LSB replacement methods , which substitute LSBs with message bits directly, the reversible contrast mapping method applies simple integer transformations to pairs of pixel values. Specifically, the method conducts forward and backward integer transformations as follows, respectively, in which (x,y) are a pair of pixel values and (x', y') are the transformed ones.The method yields high data embedding capacities close to the highest bit rates and has the lowest complexity reported so far. The information required to recover a tile image T which is mapped to a target block B includes: 1) the index of B; 2) the optimal rotation angle of T; 3) the truncated means of T and B and the standard deviation quotients, of all color channels; and 4) the overflow/underflow residuals. These data items for recovering a tile image T are integrated as a five-component bit stream of the form

$$\boldsymbol{M = t_1t_2. . .t_mr_1r_2m_1m_2. . .m_{48}q_1q_2. . .q_{21}d_1d_2. . .d_k}$$

in which the bit segments $t_1t_2 \ldots t_m$, r_1r_2, $m_1m_2 \ldots m_{48}$, $q_1q_2 \ldots q_{21}$, and $d_1d_2 \ldots d_k$ represent the values of the index of B, the rotation angle of T, the means of T and B, the standard deviation quotients, and the residuals, respectively. In more detail, the numbers of required bits for the five data items in M are discussed below: 1) the index of B needs m bits to represent, with m computed by

$$\boldsymbol{m = [\log[(WS \times HS)/NT]\ ]}$$

in which WS and HS are respectively the width and height of the secret image S, and NT is the size of the target image T; 2) it needs two bits to represent the rotation angle of T because there are four possible rotation directions; 3) 48 bits are required to represent the means of T and B because we use eight bits to represent a mean value in each color channel; 4) it needs 21 bits to represent the quotients of T over B in the three color channels with each channel requiring 7 bits; and 5) the total number k of required bits for representing all the residuals depends on the number of overflows or underflows in T".

IV. ALGORITHMS OF THE PROPOSED METHOD

Based on the above discussions, the detailed algorithms for mosaic image creation and secret image recovery may now be described respectively as Algorithms 1 and 2.

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 $\{T1, T2, \ldots, Tn\}$ as well as the target image T into n target blocks $\{B1, B2, \ldots, Bn\}$ 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 thethree 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 $S_{tile} = \{T1, T2, \ldots, Tn\}$ and the target blocks in the set $S_{target} = \{B1, B2, \ldots, Bn\}$ according to the computed average standard deviation values of the blocks; map in order the blocks in the sorted S_{tile} to those in the sorted S_{target} 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 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* by the way described in Section III(C); 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 till image *Ti* in *F* with respect to its corresponding target block *Bji* after rotating *Ti* into each of the directions $\theta = 0^o$, 90^o, 180^o and 270^o;and rotate *Ti* into the *optimal* direction θ^o 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* in the way as described in Section III(D), 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 bitstream M'_t; and embed M'_t into *F* by the reversible contrast mapping scheme proposed

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.

Algorithm 2 Secret image recovery

Input: a mosaic image *F* with *n* tile images $\{T_1, T_2, \ldots, T_n\}$ 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 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 *θ*° 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 *θ*° 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.

V. EXPERIMENTAL RESULTS

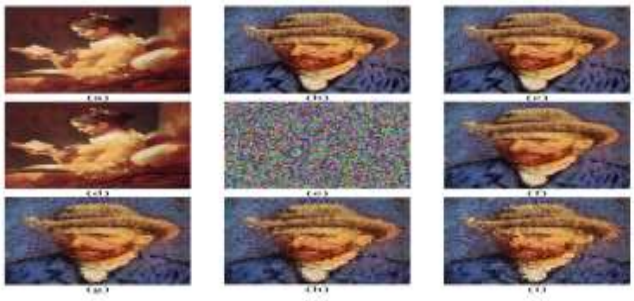

Fig. 3. Experimental Result of Mosaic Image Creation. (a) Secret image.(b) Target image. (c) Mosaic image created with tile image size 8×8. (d) Recovered secret image using a correct key with RMSE= 0.948 with respect to secret image (a). (e) Recovered secret image using a wrong key. (f)-(i) Mosaic images created with different tile image sizes: 16×16, 24×24, 32×32, and 40×40. with respect to the secret image.

A series of experiments have been conducted to test the proposed method using many secret and target images with sizes 1024×768 or 768×1024. To show that the created mosaic image looks like the preselected target image, the quality metric of root mean square error (RMSE) is utilized, which is defined as the square root of the mean square difference between the pixel values of the two images. An example of the experimental results is shown in Fig. 3; Fig. 3(c) shows the created mosaic image using Fig. 3(a) as the secret image and Fig. 3(b) as the target image. The tile image size is 8×8. The recovered secret image using a correct key is shown in Fig. 3(d) which looks nearly identical to the original secret image shown in Fig. 3(a) with RMSE= 0.948

VI. SECURITY CONSIDERATIONS

In order to increase the security of the proposed method, the embedded information for later recovery is encrypted with a secret key as seen in Algorithm 1. Only the receiver who has the key can decode the secret image. However, an eavesdropper who does not have the key may still try all possible permutations of the tile images in the mosaic image to get the secret image back. Fortunately, the number of all possible permutations here is $n!$, and so the probability for him/her to correctly guess the permutation is $p=1/n!$ which is very small in value. For example, for the typical case in which we divide a secret image of size 1024×768 into tile images with block size 8×8, the value n is (1024×768)/(8×8) = 12,288. So the probability to guess the permutation correctly without the key.

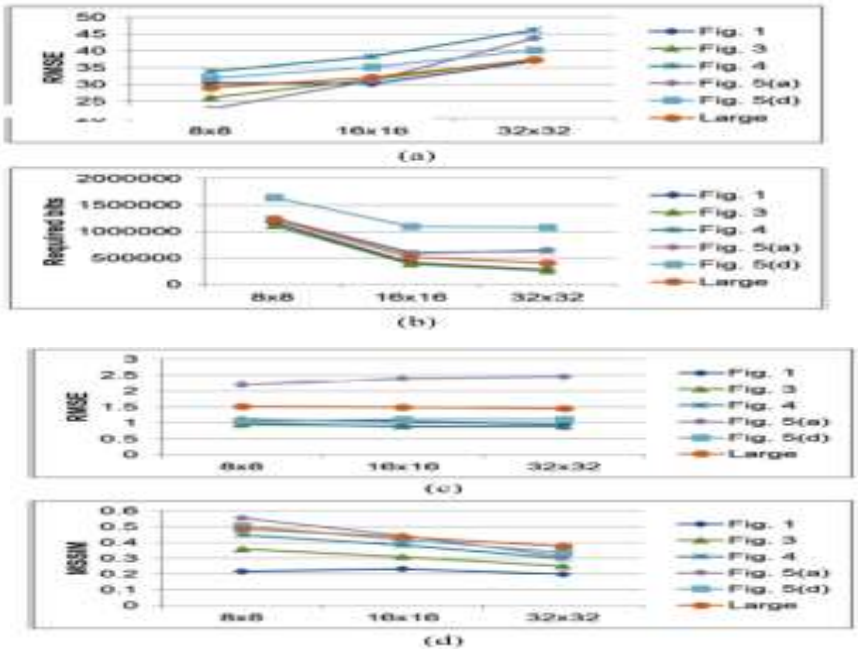

Fig4:. Plots of Trends of Various Parameters Versus Different Tile Image Sizes (8×8, 16×16, 32×32) With Input Secret Images Shown Previously And Coming From A Large Dataset. (a) RMSE values of created mosaic images with respect to target images. (b) Numbers of required bits embedded for recovering secret images. (c) RMSE values of recovered secret images with respect to original ones. (d) MSSIM values of created mosaic images with respect to target images.

is $1/n! = 1/(12{,}288!)$. So breaking the system by this way of guessing is computationally infeasible.

will observe the content of the mosaic image with a correct permutation, and try to figure useful information out of it. For example, an attacker might analyze the spatial continuity of the mosaic image in order to estimate a rough version of the secret image. To increase the security of the proposed method against this type of attack, one possible way to is to use the key to randomize important information of a secret image, such as the positions of the pixels in the secret image, before transforming the secret image into a mosaic image by the proposed method. Consequently, only authorized users with the key can know the correct secret image while an attacker cannot.

VII. CONCLUSION

A new secure image transmission method has been proposed, which not only can create meaningful mosaic images but also can transform a secret image into a mosaic one with the same data size for use as a camouflage of the secret image. By the use of proper pixel color transformations as well as a skillful scheme for handling overflows and underflows in the converted values of the pixels' colors, secret-fragmentvisible mosaic images with very high visual similarities to arbitrarily-selected target images can be created with no need of a target image database. Also, the original secret images can be recovered nearly losslessly from the created mosaic images. Good experimental results have shown the feasibility of the proposed method by using MAES. Future studies may be directed to applying the proposed method to images of color models other than the RGB.

REFERENCES

[1] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcat. Chaos*, vol. 8, no.6, pp. 1259–1284, 1998.

[2] G. Chen,Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps,"*Chaos Solit. Fract.*, vol. 21, no. 3, pp. 749–761, 2004.

.[3] H. S. Kwok and W. K. S. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos Solit. Fract.*, vol. 32, no. 4, pp. 1518–1529, 2007.

[4] S. Behnia, A. Akhshani, H. Mahmodi, and A. Akhavan, "A novel algorithm for image encryption based on mixture of chaotic maps," *Chaos Solit. Fract.*, vol. 35, no. 2, pp. 408–419, 2008.

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

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

[7] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*., vol. 37, pp. 469–474, Mar. 2004.

[8] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol*., vol. 16, no. 3, pp. 354–362, Mar. 2006.

[9] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. VideoTechnol*., vol. 13, no. 8, pp. 890–896, Aug. 2003.

BIOGRAPHICAL NOTES

Mis. Y. Sasikala is presently pursuing M. Tech. final year in Electronics and Communication Engineering Department (Specialization in Digital Electronics Communication Systems) from A.I.T.S Rajampet, Andhra Pradesh, India.

Mr. K. Riyazuddin is working as a Assistant Professor in Electronics and Communication Engineering Department, A.I.T.S Rajampet, Andhra Pradesh, India.