International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 15 (2015) pp 36664-35672

A Novel Data Hiding Technique via Mosaic Image Steganography

Shyama Nair V.S
PG Student, Department of Computer Science
Toc H Institute of Science & Technology
Ernakulam, India
syama560@gmail.com

Vishwanath N
Professor, Department of Computer Science
Toc H Institute of Science & Technology
Ernakulam, India
vishwa10370@gmail.com

Abstract- A joint data hiding and a secure image transmission technique is introduced where the functions of data hiding and image transmission can be integrated into one single module seamlessly. On the sender side, the secret data is hidden into an image which is known as the cover image. The image is then transformed automatically into a so-called secret-fragment-visible mosaic image. The mosaic image looks similar to an arbitrary-selected target image and is used to cover the cover image. It is yielded by converting the cover image into image fragments and transforming their color characteristics to be those of the corresponding blocks of the target image. The information required for recovering the cover image is embedded into the created mosaic image by a lossless data hiding scheme using a key. The secure part of this method lies in the generation of the key. The secret key is generated from the features of cover image or target image. This method of key generation ensures the prevention of guesses or breaking the key and provides a more secure way of encryption. In the receiver side, the information for cover image recovery is first extracted and using the information, the cover image is recovered. The image fragments are rejoined to get back the hidden information. The paper is easy to implement and is a much more secure way of transmission of both data and image. This technique provides more security in the areas of transmission and key generation, compared to the other techniques. It has wide range of applications in internet communication, medical images, and biometrics. Good experimental results prove the feasibility of the paper.

Keywords: LSB substitution, mosaic images, key generation, encryption

Introduction

We cannot think of a world without technology. Similarly communication has also become a part of our life. With the faster growth of Internet, people can communicate with each other easily. People communicate through different media like text, image, audio and video. The major concern of most of them is about the security of the data being transmitted through the network.

This introduced a new branch known as cryptology [3] which deals with the art of secret writing. Cryptology was further divided into cryptography and crypt analysis. Cryptography is an encryption technique whereas crypt analysis is done for decryption. Many cryptographic techniques have been implemented as a part of security enhancement. Even though cryptographic techniques were implemented, they supported only secure text transmission. Moreover, the cipher texts generated are meaningless data which would arouse the attacker's attraction. To solve this problem, information hiding techniques have been developed widely both in academia and industry, which can embed a secret piece of text into another media like image, audio or video. This technique was known as Steganography. With the rise in the field of communication, people not only used texts but also images, audios and videos for effective communication.

This introduced many Steganography techniques like image Steganography, which deals with hiding a cover image into another image, audio Steganography dealing with hiding a secret audio file into another audio and video Steganography which hides a video clip into another video. Steganography differs from cryptography in the sense that the cryptography focuses on keeping the contents of a message secret whereas Steganography focuses on keeping the message secret. Once the presence of hidden information is identified, then the purpose of Steganography is partial. Another issue in the transmission of an encrypted text is the key used. In the recent encryption algorithms, the keys used are normal text keys or numbers. Such keys undergo brute force attacks during transmission. Hence key generation and transmission also face a great challenge in the field of security.

Objective

The above mentioned security issues are overcome by the idea of a new technique called Joint Data Hiding [9] and Secure Image Transmission via Mosaic Image Steganography. The main objective of this technique is to solve the issue of randomness in the resultant image. The cover image is converted into a mosaic image [8] by applying certain transformations. The mosaic image looks similar to a pre-selected target image. Hence an attacker does not have any chance of mistaking the image as a secret data carrier. The next issue that is, generating a secure key is solved by generating the key using image features. This technique is reliable and flexible for information security. The user need not remember the keys during encryption and decryption.

Proposed System

The proposed system is divided into 3 modules:

- Data Hiding
- Mosaic Image Creation
- Key Generation

A. Data Hiding

A secret message is hidden into an image using the basic LSB substitution technique [1]. The least significant bits of an image are those bits whose replacement does not distort the image. Thus in order to hide the data into an image the replacement of the least significant is much preferable.

Let I be the RGB color image of size M_I X N_I with 8 bits each for R, G and B. It can be represented as:

I= $\{(x_{ij}, y_{ij}, z_{ij}) | 0 \leq i < M_I, 0 \leq jN_I, x_{ij}\{0,1..255\}, y_{ij} \in \{0,1..255\}, z_{ij} \in \{0,1..255\}\}$

Let M be the secret message to be hidden, which is represented as:

$$M= \{m_i | 0 \leq i < n,\ m_i \in \{0,1\}\}$$

Let the n-bit secret message M be embedded into the k rightmost LSBs of the image I. First, we rearrange the secret message M to form a k-bit virtual image M' which is represented as follows:

$$M'= \{m_i' | 0 \leq i < n',\ m_i' \in \{0,1,2,....2^k - 1\}\}$$

where n'<$M_I X N_I$. The mapping between the n-bit secret message M= $\{m_i\}$ and the rearranged message M'= $\{m_i'\}$ is as shown:

$$m_i' = \sum_{j=0}^{k-1} m_{iXk+j}\ X\ 2^{k-1-j}$$

Then we choose a subset of n' pixels $\{x_{i1}, x_{i2}, x_{i3}, ...x_{in}\}$ from the image I. The embedding process is done by replacing the k bits of $x_{i1}' = x_{i1} - x_{i1} mod\ 2^k + m_i'$.

In the decryption process, the stego-image is given as the input and the embedded messages can be readily extracted without referring to the original image used for data hiding. We use the same sequence as in the embedding process, and the set of pixels $\{x_{i1}', x_{i2}', x_{i3}', ...x_{in}'\}$ storing the secret message bits are selected from the stego-image. The k bits of

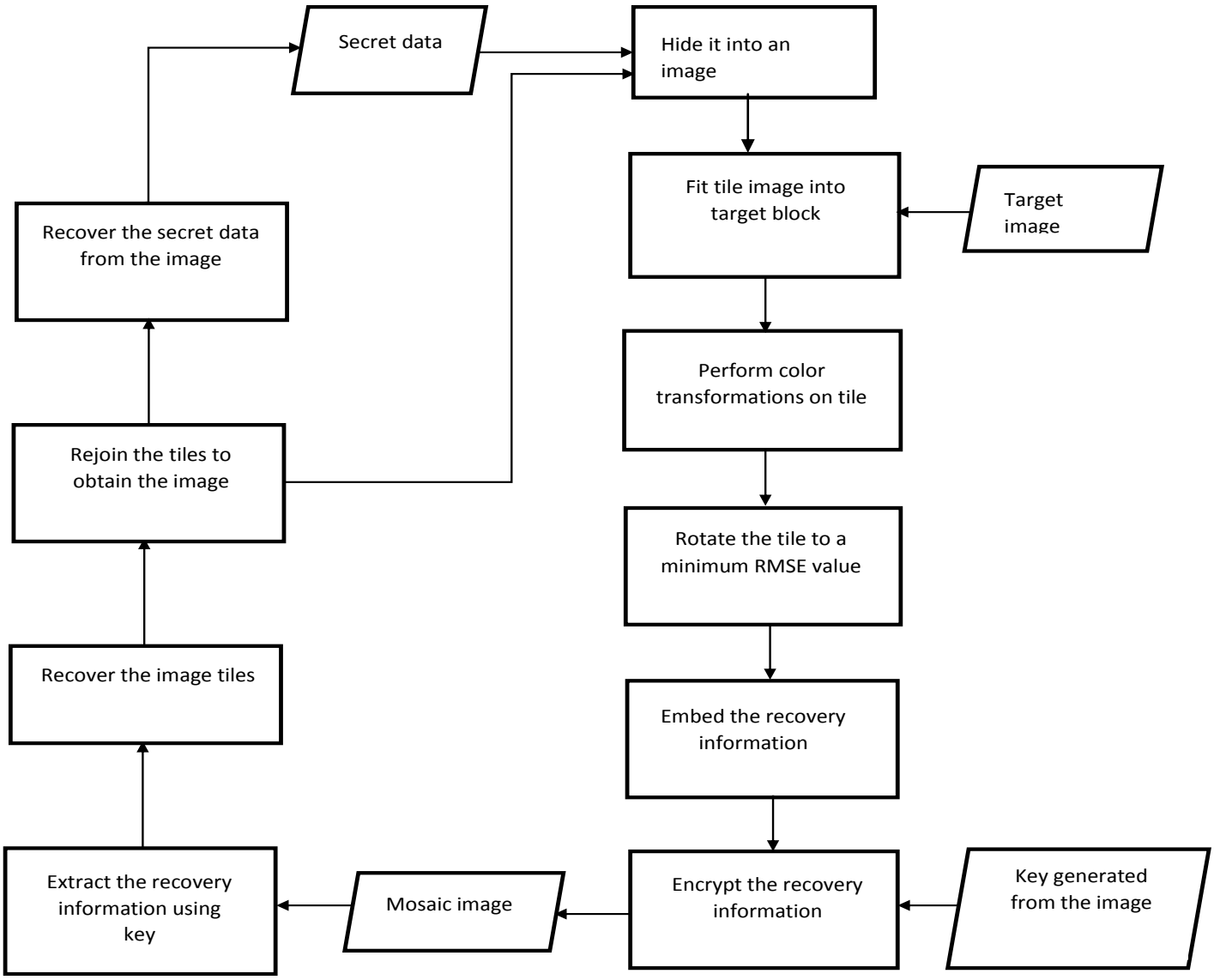


Fig1. Flow diagram of the proposed system

the selected pixels are extracted and used to reconstruct the secret message.

B. Mosaic Image Creation

This module is further divided into 5 sub modules:

i. Colour Transformation between Blocks

The image in which the secret message is hidden is divided into fragments where each fragment is called a tile image [6]-[7]. A pre-selected target image chosen to hide the cover image is also divided into fragments where each fragment is called a target block. Let each tile image be considered as a set of pixels P_{tile}= $\{p_1, p_2, p_3, \ldots\ldots, p_n\}$ and each target block as a set of pixels P_{target} = $\{p_1', p_2', p_3', \ldots\ldots, p_n'\}$. Each pixel of the tile image is represented as $p_i(r_i, g_i, b_i)$ and each pixel of the target block is represented as $p_i'(r_i', g_i', b_i')$. We now calculate the mean and standard deviation of these pixels using the formulae:

$$\mu_c = \frac{1}{n}\sum_{i=1}^{n} c_i \quad \mu_c' = \frac{1}{n}\sum_{i=1}^{n} c_i' \qquad (1)$$

$$\sigma_c = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i - \mu_c)^2} \quad \sigma_c' \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i' - \mu_c')^2} \qquad (2)$$

Here, c_i and c_i' denote the color channel values of pixels p_i and p_i' respectively where c = r, g or b. The new pixel value $p_i''(r_i'', g_i'', b_i'')$ is calculated using a formula:

$$c_i'' = q_c(c_i - \mu_c) + \mu_c' \qquad (3)$$

Where q_c is known as the standard deviation quotient and is calculated by using the equation $q_c = \sigma_c'/\sigma_c$. It is seen that the new mean and standard deviation of the resulting tile image is same as that of the target block. The original color values of the pixels are calculated by performing the reverse operation:

$$c_i = 1/q_c(c_i'' - \mu_c') + \mu_c \qquad (4)$$

ii. Choosing appropriate target block to fit the tile image

In order to transform the color characteristics of a tile image to a corresponding target block, we should be careful in choosing the respective target block for a particular tile image. For this purpose, we use the standard deviation of the three colors in a block so as to select the most similar target block for the tile image. At first, we calculate the average of the standard deviations of the three colors red, green and blue for each tile image and target block of the cover image and the target image. Next we sort the tile images to form a set Tile = $\{tile_1, tile_2, tile_3, tile_4 \ldots tile_n\}$ and the target block to form a set Target = $\{target_1, target_2, target_3, \ldots.. target_n\}$ in the increasing order of the average values of standard deviations calculated earlier. Finally, we fit the first tile image of the set i.e., $tile_1$ into the first target block in the set i.e., $target_1$, the second tile image into the second target block and so on. This fitting is done using the LSB substitution technique. The LSBs of the target block is replaced with the bits of the tile image. The least significant bits are chosen so as to avoid image distortions.

iii. Rotating the blocks to fit better with minimum RMSE

Even after conducting the color transformations on the tile image, it may not perfectly fit the target block. To avoid the minor distortions, we rotate the resulting tile image into any of the four angles 0°, 90°, 180° or 270° which gives a minimum root mean square error (RMSE) value.

iv. Handling overflows and underflows

After performing the color transformations on the pixels, some of the pixels may have underflows or overflows. An underflow occurs when a particular pixel value is below 0 and an overflow occurs when a particular pixel is above 255. We first convert the pixels above 255 to 255 and the pixels below 0 to 0. Then we calculate the difference between the original values of the pixels and the converted values and note them as residual values. The pixels in the bound of 0 and 255 will thus have a residual value of 0. These values are later used for the recovery of the image. To determine the number of bits needed to represent the residual values we use a different approach. Using two formulae we calculate the smallest possible value in the tile image that is larger than 255, and the largest possible value that is smaller than 0.

$$c_S = \left\lceil \left(\frac{1}{q_c}\right)(255 - \mu_c') + \mu_c \right\rceil \qquad (5)$$

$$c_L = \left\lceil \left(\frac{1}{q_c}\right)(0 - \mu_c') + \mu_c \right\rceil \qquad (6)$$

Next, the residual values are calculated as follows:

$|c_i - c_S|$ for an underflow value of c_i

$|c_L - c_i|$ for an overflow value of c_i

Thus all the possible values of residuals will range from 0 to 255. Hence we require 8 bits to represent them.

v. Embedding the recovery information

In order to recover the image from the created mosaic image, the receiver requires the image recovery information. Thus we embed the recovery information also into the mosaic image so as to recover it during decryption. For this purpose, we use the LSB technique which was earlier used for data hiding. Here, instead of using the normal LSB technique, we apply a simple transformation on the pixels before embedding them. Specifically, let P (x, y) be a pixel pair which is transformed into P' (x', y') using the formulae:

$$x' = 2x - y, \qquad y' = 2y - x \qquad (7)$$

$$x = \left\lceil \frac{2}{3}x' + \frac{1}{3}y' \right\rceil, \qquad y = \left\lceil \frac{1}{3}x' + \frac{2}{3}y' \right\rceil \qquad (8)$$

This technique yields high data hiding capacity compared to the normal LSB technique and the complexity is low.

The information hidden into the mosaic image for later recovery of the cover image includes:

- The index of the target block

- The optimal rotation angle of the tile image
- The means and standard deviation quotient of tile and target block
- The residual values

These information are integrated to form a stream of bits:

$$M=t_1t_2 \ldots\ldots t_m r_1r_2 m_1m_2 \ldots\ldots m_{48} q_1q_2 \ldots\ldots q_{21} d_1d_2 \ldots\ldots d_k$$

Here, $t_1t_2 \ldots\ldots t_m$ represents the index of the target block, r_1r_2 represents the optimal rotation angle of tile image, $m_1m_2 \ldots\ldots m_{48}$ represents the mean of tile and target block in which the mean of n pixels in one color value uses 8 bits. $q_1q_2 \ldots\ldots q_{21}$ represents standard deviation quotient in which the value in a single color channel uses 7 bits. $d_1d_2 \ldots\ldots d_k$ represents the residual values of the pixels.

C. Key Generation

The recovery information required to recover the cover image and hence the data should be encrypted before it can be embedded into the mosaic image. In normal cases, we use text keys to encrypt the information which can be subjected to brute force attacks. To avoid this problem, we have a different approach in encrypting the information. Instead of using the normal keys we generate a key from any of the two images used in this technique. The key is generated from the image features [4]. This technique includes 5 sub modules:

i. GLCM

Gray level co-occurrence matrix [2] is used to find the gray level co-occurrence properties of an image. It calculates the number of occurrences in an image. It is also called gray level spatial dependence matrix.

ii. Feature extraction

This is a dimensionality reduction technique. It transforms a large set of redundant input values into a reduced set of features. This set of features is called feature vector. The process of transformation is called feature extraction.

iii. Principal component analysis

It is a multivariate which is based on Eigen values and Eigen vectors. It focuses on multi-dimensional data. It transforms the data from its co-ordinate system to a new co-ordinate system. The features extracted from the image are inputted for Principal component analysis for selecting higher rank features.

iv. Feature selection

This is a dimensional reduction of a data. It selects the features which has the highest rank from an image. The extracted features [5] are fed as inputs for the principal component analysis and thus the features are selected.

v. *v. Key*

vi. The output obtained after applying all these techniques is the random key. This key is used for the encryption and decryption of the image recovery information. The technique follows a symmetric key encryption.

vii. Algorithms of the Proposed Method

Algorithm 1 Data Hiding

Input: Secret data and image
Output: Image with hidden data
Steps:
Step 1- Select a colour image C with 8 bits for each colour, to hide the data.
Step 2- Select an n-bit secret data M to hide into the image.
Step 3-Convert the secret message into a k-bit virtual image M'.
Step 4- The message is then embedded into the k LSBs of the image.
Step 5- The data can be extracted by performing the reverse operation.

Algorithm 2 Mosaic Image Creation

Input: Image in which data is hidden (cover image) and target image.
Output: Mosaic image
Steps:
Step 1: Divide the cover image into blocks where each block is known as tile.
Step 2: Similarly, divide the target image into blocks where each block is called a target block.
Step 3: The mean and standard deviation of each tile and target block is calculated using formulae.
Step 4: Arrange the tile images and the target blocks in the increasing order of average values of their standard deviations.
Step 5: Embed the first tile image in the set to first target block with minimum standard deviation variations.
Step 6: Rotate the tile image to an angle with minimum RMSE value.
Step 7: Calculate the residual values of the pixels with underflow and overflow.
Step 8: Generate a stream of bits that contain the recovery information which include the index of the target block, mean and standard deviation of tile and target block, rotation angle and the residual values.
Step 9: Encrypt the recovery information using a secure key.
Step 10: Embed the recovery information into the generated mosaic image using LSB substitution technique.

Algorithm 3 Key Generation

Input: The cover image or the target image.
Output: Secure key
Steps:
Step 1: Select either the cover image or the target image from the two.
Step 2: Calculate the grey-level co-occurrence matrix of the selected image.
Step 3: Calculate the grey-level co-occurrence properties of the grey level co-occurrence matrix.
Step 4: Among the several properties calculated in the above step, retrieve the best four properties.
Step 5: Round all the property values and find the sum of these properties.
Step 6: The property values that are in decimal are converted to binary.
Step 7: The maximum and minimum length converted binary values are calculated.

Step 8: To get the maximum length, append the required number of bits to the left of the binary values.
Step 9: Compare all the binary bit positions to check whether all the properties have same value (either 0 or 1).
Step 10: Use an array to store the matching positions.
Step 11:Calculate the sum of all the matching positions and their men.
Step 12: Calculate the sum of all the property values(decimal).
Step 13:Round the sum to a four digit value.
Step 14: Divide the sum by performing binary division.
Step 15:Assign characters for each of the quotient values obtained after the binary division and store them in an array.
Step 16:Repeat the last 2 steps until we get a quotient 1.
Step 17: Store the character array into another array after reversing it.
Step 18:Concatenate the two arrays and form a new array.
Step 19: Choose the first 56 bits or the last 56 bits of the new character array and remove the remaining characters.
Step 20:The sum of all the values of the ASCII characters is calculated.
Step 21: Divide the 4 digits sum into 2 digits.
Step 22: Sum the 2 digits and take it as one value if the digits are greater than 26.
Step 23: Perform the above step for the other two digit number
Step 24: Finally, the key for encryption has been generated.

Results and Observations

The proposed technique comprises of two phases. In the first phase, we hide the secret data into an image which is chosen randomly. Succeeding, we choose a target image to hide the image with the secret data hidden in it. In this way, we generate a mosaic image, and encrypt the recovery information using a secure key before embedding it into the created mosaic image. In phase two, we perform the decryption using the same key to extract the recovery information. Using the recovery information, we decrypt the image and thus the secret data.

The output is tested with several inputs. Here we check the RMSE value of the tile images by taking the tiles of different sizes. Successions of experiments have been conducted to test the proposed method using many secret and target images with different sizes. To prove that the created mosaic image looks like the pre-selected target image, we make use of the quality metric of root mean square error (RMSE), where the root mean square 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 Fig2. The created mosaic image which is created using Fig 2(A) as the cover image and Fig 2(B) as the target image is shown in Fig 2(C).

(A) (B)

(C) (D)

Fig2. Experimental results of the proposed method.
(A) Cover image (B) Target image (C) Mosaic image created with tile image size 8x8 (D) Recovered cover image using a correct key with RMSE=0.948

The tile image size is 8x8. The recovered cover image using a correct key is shown in Fig2(D) which looks nearly identical to the original cover image shown in Fig 2(A) with RMSE = 0.948 with respect to the cover image. Moreover, in Fig 3, we see that a wrong key is used to recover the image. Thus we obtain a noise image.

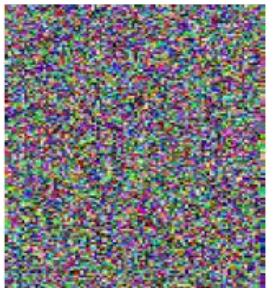


Fig3. Recovered cover image using a wrong key

Fig 4(A) through Fig 4(D) show more results using different tile image sizes. From the Figs, we can see that the created mosaic image retains more details of the target image when the tile image is smaller. We can also see that when the mosaic image is enlarged, it can be visualized as blocks; but if it is observed as a whole, it looks exactly like the target image.

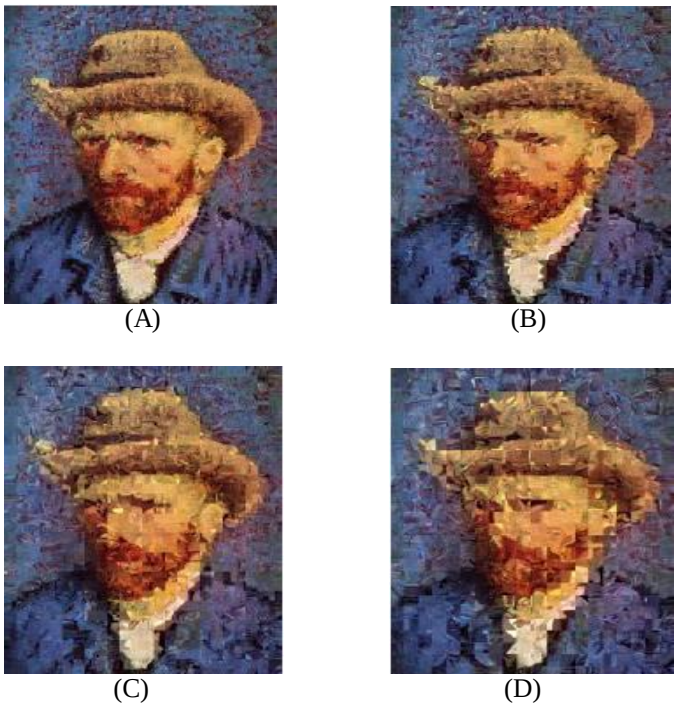


Fig4. (A)Mosaic image created with tile image size 16x16. (B)Mosaic image created with tile image size 24x24. (C)Mosaic image created with tile image size 32x32. (D)Mosaic image created with tile image size 40x40.

Fig 5(A) shows a comparison of the results yielded by the proposed method, where the Fig is the input cover image. Fig 5(B) is the selected target image. Fig 5(C) is the mosaic image. Fig 5(D) is that created by the proposed method. It can be seen from these results that the mosaic image yielded by the proposed method has a smaller RMSE value with respect to the target image, implying that it is more similar to the target image in appearance. The other results of the experiments also show the same conclusion. And more importantly, the proposed method allows users to select their favorite images for uses as target images. Fig 5(E) shows two other experimental results of mosaic image creation, where the utilized cover images both contain many structures. Fig 5(F) and Fig 5(I) are the target images; Fig 5(G) and Fig 5(J) are the created mosaic images with image sizes 8x8. It can be seen from Fig5(G) and Fig 5(J) that each created mosaic image still has the visual appearance of the pre-selected target image even when the cover image contains many structural elements. Especially, the cover image of Fig 5(H) is a nearly black-and-white document image, which means that the proposed method can be utilized for secure transmissions of confidential document images.

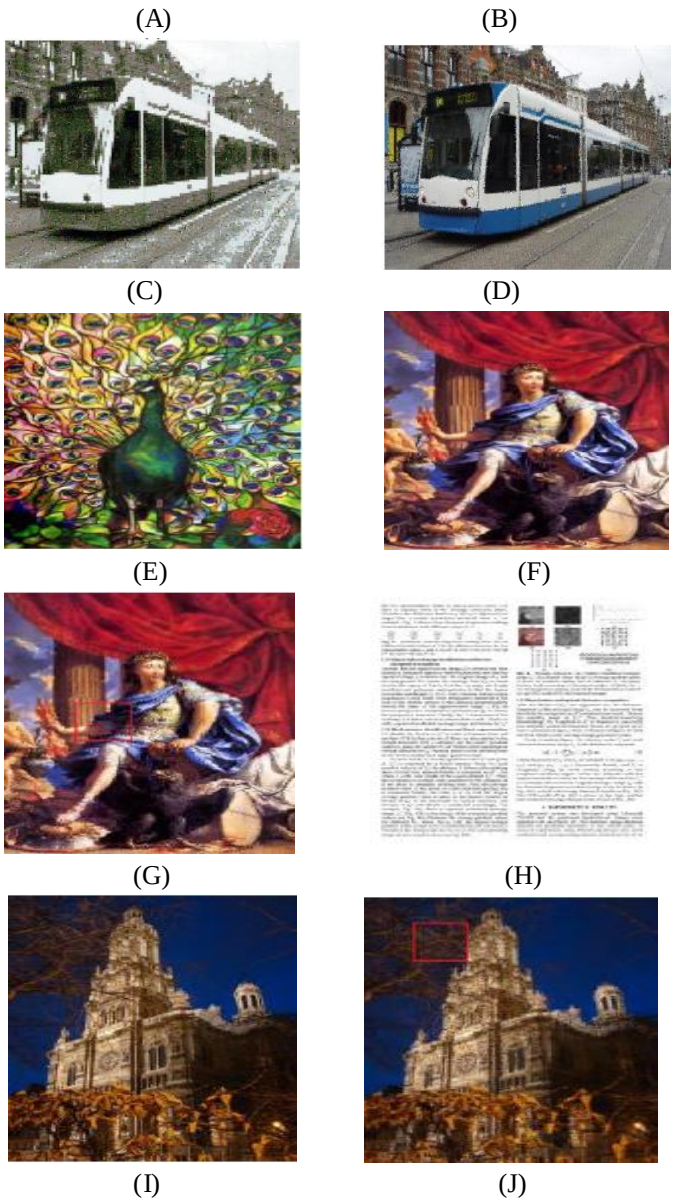


Fig5. (A)Cover image (B)Target image (C)Mosaic image created from Fig5(A) and Fig5(B) with RMSE=47.651 (D)Mosaic image created from Fig5(A) and Fig5(B) with RMSE=33.935 (E)Cover image (F)Target image (G)Mosaic image created from Fig5(E) and Fig5(F). (H)Cover image(I)Target image(J)Mosaic image created from Figs 5(H) and 5(I).

However, since the mosaic image is yielded by dividing the cover image into tile images and transforming their color characteristics to be those of the corresponding target blocks, the global color characteristics of a transformed tile image and its corresponding target block are the same but the color distributions of them may be quite different. Hence, although the mosaic image has the visual appearance of the target image, the details of each fragment in the mosaic image may have low similarity to those of its corresponding target block. An input-output model in tabular form is provided in the table below, where the input data is related to the output data through the processes for easier understanding.

TABLE.1. Input-Output Process Table

INPUT	PROCESS	OUTPUT
SECRET DATA	USER INPUT	ACCEPT INPUT DATA
COVER IMAGE	HIDE THE DATA INTO IT	COVER IMAGE WITH THE HIDDEN DATA
TARGET IMAGE	ARBITRARILY SELECTED	ACCEPT AS COVER IMAGE
BLOCK OF TILES OF EQUAL SIZE	MATLAB FUNCTION MAT2CELL	TILE IMAGES AND TARGET BLOCKS
HIDING TILE IMAGE INTO TARGET BLOCKS	MOSAIC IMAGE STEGANOGRAPHY	MOSAIC IMAGE
SECURE KEY	RECOVERY INFORMATION ENCRYPTION	MOSAIC IMAGE WITH RECOVERY INFORMATION
SECURE KEY	IMAGE RECOVERY INFORMATION DECRYPTION	MOSAIC IMAGE
RECOVERY INFORMATION	MOSAIC IMAGE DECRYPTION	COVER IMAGE

The data embedding process is also found to give good results. Experimental results show that the image before data embedding and after embedding looks exactly the same. Thus the technique avoids image distortions. An example of the results is shown in Figs 6(A) and 6(B).

(A) (B)

Fig6. Experimental results of data hiding. (A)Input image to hide secret data. (B)Image with secret data

Only using the correct key, we can retrieve the data from the image. The data retrieved using the correct key is shown in Fig 7.

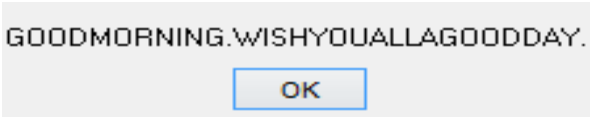
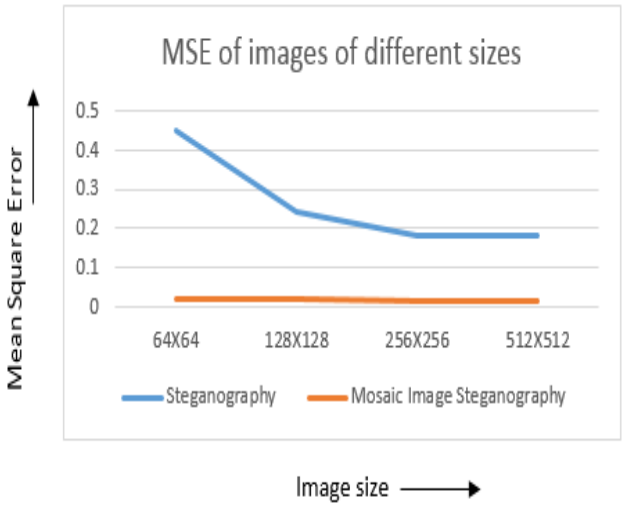
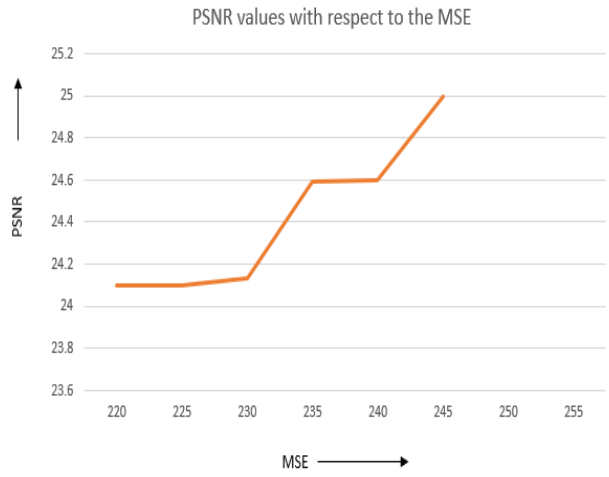


Fig7. Data recovered using the correct key

We now compare the existing Steganography technique with the proposed system. For that, we plot a graph by comparing the mean square error value of the mosaic image with respect to target image to test the distortion. As we can see, the mean square error decrease as the image dimension becomes greater which is represented in Fig 8(A). On the other hand, we see from Fig 8(B) that as the MSE decreases, the PSNR (peak signal to noise ratio) increases. The proposed method is thus efficient and accurate than the existing method.



(A)



(B)

Fig8. (A) MSE values of mosaic images with respect to target image. (B) PSNR values for various values of MSE.

Fig 9 shows the result of this comparison. As we can see, the Mosaic Image Steganography technique has less image distortions compared to the normal Steganography technique. The table given below shows the comparative results of the proposed method. Thus the accuracy of the proposed system is more than the existing system.

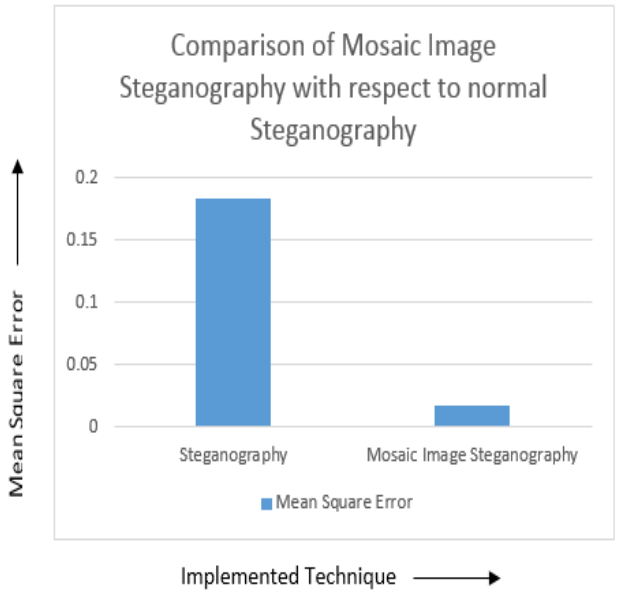


Fig9. Comparison of the Mosaic Image Steganography technique with respect to the existing Steganography technique.

TABLE.2. Comparison of the proposed system with respect to existing technique

IMAGE SIZE	STEGANOGRAPHY	MOSAIC IMAGE STEGANOGRAPHY
64X64	0.4490	0.021
128X128	0.2419	0.0185
256X256	0.1839	0.0168
512X512	0.1810	0.0151

Security Consideration

As a part of security, we encrypt the image and data recovery information with a secure key. In order to enhance security, the key is generated from either the cover image or the target image using the image features. This prevents brute force attacks on using the normal text keys. Only the receiver who knows the key can get the recovery information. Once he obtains the recovery information, he can extract the cover image and data using these information.

However, an attacker may try all the possible permutations to get back the cover image and hence the data. But, the number of possible permutations is n! , hence the probability for an eavesdropper to find the correct permutation is p=1/n! which is a very small value. Hence breaking the system in this manner is practically infeasible.

Conclusion

The various drawbacks found in the surveys done so far have brought in an idea of introducing a new technique where the data hiding and image transmission is combined to a single module. The data is first hidden into a cover image using the LSB data hiding technique. The image is then divided into small fragments and embedded into a pre-selected target image. No image databases are used in this technique for the selection of target images. The resultant image is known as mosaic image. The image recovery information is also embedded in the mosaic image after encrypting it using a key. The security is further enhanced by generating the key from the image. For generating the key, various features of the image can be used. The key is thus safe from brute force attacks. The mosaic image is recovered using the recovery information. From the mosaic image the cover image and thus the secret data is obtained by performing the reverse operations.

Acknowledgement

We would like to thank all the reviewers for many useful comments and feedbacks that improve the paper presentation.

References

[1] Chia-Chen Lin, Yi-Hui Chen and Chin-Chen Chang,"LSB-Based High-Capacity Data Embedding Scheme For Digital Images", International Journal of Innovative Computing, Vol.5, November 2009.

[2] Alaa Eleyan and Hasan Dem˙irel, "Co-occurrence matrix and its statistical features as a new approach for face recognition", Turk J Elec Eng& Comp Sci, Vol.19, No.1, 2011.

[3] William Stallings, "Overview in Cryptography and Network Security", 5th edition, Pearson Education, Inc, 2006, ISBN 10: 0-13-609704-9.

[4] B. Santhi, K.S. Ravichandran, A.P. Arun and L. Chakkarapani, "A Novel Cryptographic Key Generation Method Using Image Features", Research Journal of Information Technology Vol. 4, No.2,June 2012.

[5] Fengxi Song, Zhongwei Guo and Dayong Mei, "Feature selection using principal component analysis", International Conference on System Science, Engineering Design and Manufacturing Informatization, 2010.

[6] Ya-Lin Lee and 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.

[7] Soumi C.G, Joona George and Janahanlal Stephen, "Genetic Algorithm based Mosaic Image Steganography for Enhanced Security", ACEEE Int. J. on Signal and Image Processing, Vol. 5, No.1, January 2014.

[8] I-Jen Lai and Wen-Hsiang Tsai,"Secret-Fragment-Visible Mosaic Image–A New Computer Art and Its Application to Information Hiding", IEEE Transactions on Information Forensics and Security, Vol. 6, No. 3, September 2011.

[9] N.S. Soleimani Zakeri and M.A. Balafar, " A Review on Data Hiding upon Digital Images", International Journal on Technical and Physical Problems of Engineering.

[10] Chi-Kwong Chsn and L.M Cheng, "Hiding Data in Images by Simple LSB Substitution ", Pattern Recognition, August 2003.