# A Randomized Secure Data Hiding Algorithm Using File Hybridization for Information Security

K. Venkata Ramana[1],

Department of CSE,
RVR&JC College of Engineering, Guntur, India

Dr.B.Raveendra Babu[2],

Director (Operations),
Delta Technologies (P) Ltd., Hyderabad, India

Sri Ch.Ratna Babu[3],

Department of CSE,
RVR&JC College of Engineering, Guntur, India

**Abstract --The internet and the World Wide Web have revolutionized the way in which digital data is distributed. The growing possibilities of modern communication need special means of security especially on computer network. In this paper a new randomized secure data hiding algorithm using file hybridization is proposed for strengthening the security of information through a combination of cryptography and steganography with random transformation and file hybridization. Cryptography concentrates on rendering the message unreadable to any unauthorized persons who might intercept them and steganography provides security by concealing existence of information being communicated. The proposed method, viz. Randomized Secure Data Hiding Algorithm using File hybridization (RSDHAF), is divided into sender side and receiver side parts which consist of combination of phases. File hybridization technique adds extra mystification to the analyst in finding the secret message. Simulation results show that the method provides high security and the information is safe from various attacks.**

***Keywords- Information Security, Information hiding, Steganography, Cryptography, File hybridization, RSDHAF.***

## I. INTRODUCTION

With the development of the Internet, information processing technologies and the rapid development of communication, it is necessary to share information resources. Nevertheless, the Internet is an open environment so; information security has becoming increasingly important. The different embodiment disciplines of information hiding are shown in following Figure 1. Today, information security technology has two main branches, cryptography and information hiding. Cryptography process data into unintelligible form, reversibly, without data loss. Cryptography aims to prevent unauthorized receivers from decoding the programs by scrambling them [1]. Information hiding is divided into steganography and digital watermarking. Steganography is the art or practice of concealing a message, image, or file within another message, image, or file. Steganography and cryptology are similar in the way that they both are used to protect important information [2]. The difference between the two is that Steganography involves hiding information so it appears that no information is hidden at all. Nowadays the term “Information Hiding” relates to both watermarking and steganography [3]. Watermarking is the technique use to hide information in a digital object (video, audio or image) so that information is robust to adjustments or alterations [2], [3]. By watermarking, the mark itself is invisible or unnoticeable for the human vision system. In addition, it should be impossible to remove a watermark without degrading the quality of the data of the digital object [4]. On the other hand, the main goal of steganography is to hide secret information in the other cover media (video, audio or image) so that other persons will not notice the presence of the information [3], [4]. Although steganography is separate and different from

cryptography, but they are related in the way that they both are used to protect valuable information [4]. In this paper our proposed system uses both steganography and cryptography and provides double layer of security.

In steganography carrier medium is defined as the object that carries the hidden information. Stego-object is the resultant production of steganography that is transmitted to the destination. Stego-key is defined as the key used to extract the hidden data from the stego- object. Data may be embedded in various possible carriers like audio file, document, file headers, digital image and video.

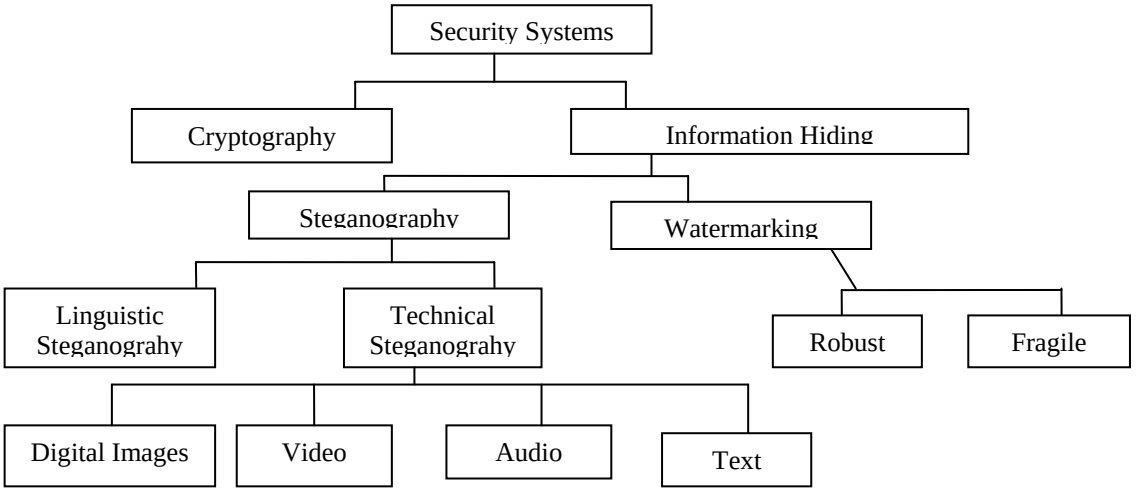


Figure 1. The different embodiment disciplines of Information Hiding.

In our proposed system we are using an efficient carrier media, digital image for steganography. The basic unit of the composition of an image is called pixel. The size of an image can be given in pixels. Pixels are indexed by x and y coordinates with x and y having integer values. Each pixel is generally stored as 24 bit or 8 bit. A 24-bit image are spread over three bytes and each bytes represents red, green, and blue respectively. Colors are obtained by mixing red, green, and blue light in different proportions.

In our proposed system we are combining cryptography and steganography mechanisms with random transformation and file hybridization [5], to have better security. Random transformation and file hybridization makes the tasks of steganalysis difficult. File hybridization results in a hybrid image (merging of one or more images) which makes it difficult to the analyzer to infer which image actually contains the secret message.

RSDHAF has the following phases:

Sender Side:
- Encrypting phase
- Random transformation phase
- Embedding phase
- File hybridization phase

Receiver Side:
- File separation phase
- Extraction phase
- Random re-transformation phase
- Decrypting phase

RSDHAF is divided into two parts as sender side and receiver side. At the sender side four phases are performed to safely hide our secret message in cover image as listed above. In the first phase that is encrypting phase the original message is encrypted to block of colors. Later in the random transformation phase block of colors are transformed to color image based on random key. The resultant color image is embedded into container image using a stego key in the embedding phase. At last in the file hybridization phase a supporting image is selected and then the container file is placed inside the supporting file in a defined region. The resultant hybrid image is the final image which is transmitted to the destination. At the destination, the receiver side four phases are performed to retrieve the original secret message. As listed above in the first phase file separation the container image is extracted. Later in the extraction phase again by using the same stego key which is used at sender side the color image is extracted from the container image. In random re-transformation phase the color

image is transformed into block of colors using the same random key again. Finally by using decryption phase the encrypted block of colors are decrypted to original secret message.

This paper is organized as follows. Section II describes the proposed system sender side, which is used to hide our secret message in the hybrid mage. Section III briefly describes the destination side of proposed system, which is used to retrieve our secret message again. The subsequent Section IV presents (1) Simulation Results of RSDTAF and (2) Analysis and comparison to prior art. The paper is concluded in Section VI.

## II. RANDOMIZED SECURE DATA HIDING ALGORITHM USING FILE HYBRIDIZATION-SENDER SIDE

Algorithm Description

Our algorithm works in four phases at sender side: Encryption phase, Random Transformation phase, Embedding phase and File hybridization phase. Overview of these phases is given in the following Figure 2.

### A. Encryption phase

In this phase, the algorithm encrypts the original message which serves as plaintext. The block of plaintext letters are encrypted into block of colors. The mapping of alphabets (letter/digit) to color is random and unique based on secret key which is shared between sender and receiver. To represent the letters from A to Z and the numbers from 0 to 9 we chose a set of 36 unique colors which are in the same scale/different scale of colors in RGB scale. Then the message is represented as a set of colors. Example color representations are given in the following Table 1.

### B. Random Transformation phase

This phase transforms the block of message colors into color image. The image is considered as a matrix of pixels (x, y) where x takes the values from 0 to 36 (that corresponds to 26 letters, numbers from 0-9 and one for representing the location of the alphabet). Y takes values from 0 to the plain text length (if the size of the message is 1000 bytes, y takes values from 0-1000). In the column (0, 0) is inserted the colors corresponding to the plaintext alphabets. So, the size of the resultant image will be 37× y.

The columns from x=1 to 36 are reserved to store the colors associated to the alphabets. The order of the alphabets in the column is randomly selected by using a random key. The size of the random key is 36. After fixing the random order of alphabets in the image, the transformation process starts by reading the block of colors (representation for message) from the first phase and color the respective column with the correspondent color. If the letter is not present in the message the respective column is colored in black. In this manner our resultant image includes the letters of the plain text and their locations and the correspondent colors. As this transformation is based on random key and random mapping of colors it is very secure and only the authorized destination can read the plain text. An example of color image obtained after this phase is shown in the following Figure 3.

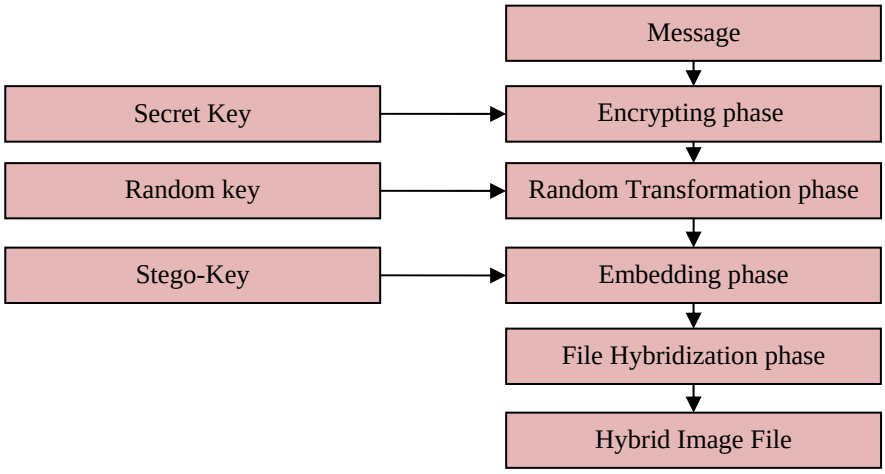


Figure 2. Overview of proposed system sender side.

Table.1: Example color representations of letters and digits

| Letter/Digit | Color Representation in RGB | Letter /Digit | Color Representation in RGB | Letter /Digit | Color Representation in RGB | Letter /Digit | Color Representation in RGB |
|---|---|---|---|---|---|---|---|
| **A** | **0.1.20** | **J** | **5.0.0** | **S** | **78.220.3** | **2** | **45.70.11** |
| **B** | **13.1.2** | **K** | **2.20.0** | **T** | **91.2.4** | **3** | **2.2.9** |
| **C** | **2.1.3** | **L** | **25.0.38** | **U** | **55.66.8** | **4** | **22.33.33** |
| **D** | **225.220.0** | **M** | **55.22.45** | **V** | **34.5.4** | **5** | **5.70.9** |
| **E** | **1.2.39** | **N** | **44.0.2** | **W** | **11.0.0** | **6** | **12.14.14** |
| **F** | **0.1.55** | **O** | **77.0.1** | **X** | **32.220.0** | **7** | **15.17.0** |
| **G** | **22.22.0** | **P** | **88.40.4** | **Y** | **67.10.0** | **8** | **80.70.60** |
| **H** | **5.120.0** | **Q** | **22.11.11** | **Z** | **82.30.5** | **9** | **90.20.55** |
| **I** | **100.20.0** | **R** | **15.0.0** | **1** | **99.120.0** | **0** | **7.30.100** |

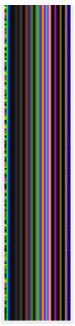

Figure 3. Example color image obtained after random transformation phase for the text "The goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to detect that there is a second secret message present 01"

C. Embedding phase

For higher security the color image produced in random transformation phase is embedded into another image and finally file hybridization is done before transmitting to the receiver. The first step in the embedding phase is selecting the container file. The file to which the secret data is added is called container file. The selection of container file and embedding process is discussed below (see [5] for details).The basic property of this container file is such that even if we change the intensity of any pixel it should look like the original image. appropriate candidate for this purpose are cartoon images, geographical images, background images of any picture or images in any chemical reaction and like others. For example, in the case of twinkling star or any other geographical image we can see that by an appropriate change in color of the object lying in the image can give the same impression of the original image. In order to store the secret data in such a file, the size of the container file should be proportional to the size of the secret data.

In contrast to the traditional LSB schemes [6], [7] and [8] being used for hiding the data, in our embedding process we consider the entire byte for storing the information. Consequently, by the process of replacing the entire byte for embedding information in the container image, only one pixel can be used to store three characters. The selection criteria of choosing the pixels are done randomly by a stego-key. Further, if we replace all pixel values of an image then the entire image generally changes and may look completely like another image of suspicious. So by the use of the concept of hybridization in the next phase it gives the impression of a normal unsuspicious image file.

D. File Hybridization phase

The concept of hybridization may be used in the field of steganography, where more than one file is to be merged and a new hybrid file may consequently be generated. This hybrid file basically consists of two files, namely

- Container image
- Supporting image

Supporting image: To make the image common, we need a supporting image file so that the new hybrid image looks like the original one. The selection of supporting image will depend on the feature of the container image to ensure the above characteristics. There will be two options in this process; either we can put the container file into the supporting file or vice versa.

In our method, first a container image is selected and then based on our requirements (i.e. the size of container message) we select the appropriate supporting image. If we consider the size of the supporting image as, M1 x N1, then we can place the container image inside the supporting image, in a region defined by A(x, y) and B(x + r, y + s) for a suitable value of r and s, where $0 \leq r \leq M1$ and $0 \leq s \leq N1$.

Since we are hybridizing the file, pixel of the supporting image will be replaced by the corresponding pixel of the container image. As an example we can see in the below Figure 4 merging of a (flower) container file to (plain colors) hybrid supporting file.

After replacing all pixel values of supporting image by the container image we get the resultant hybrid or mixed image which is the final transmitted image to the receiver. Supporting image may be a single image or hybrid image by itself.

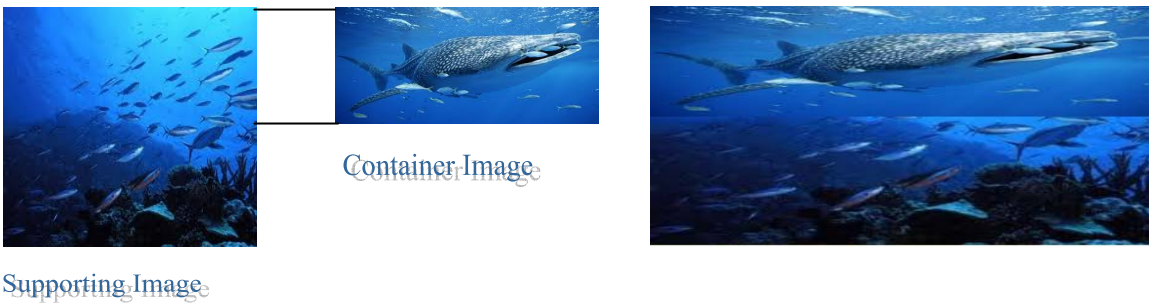


Figure 4. Merging of container image to supporting image before and afterwards.

## III. RANDOMIZED SECURE DATA HIDING ALGORITHM USING FILE HYBRIDIZATION - RECEIVER SIDE

Our algorithm works in four phases at receiver side: File Separation phase, Extraction phase, Random Re-transformation phase and Decryption phase. Overview of these phases is given in the following Figure 5.

### A. File seperation phase

This is the reverse process of file hybridization phase. In this phase container image is obtained from the hybrid image .This process needs values r and s, which define the location of the container file.

### B. Extraction phase

The color image is extracted from the container file in this phase. By using the same stego-key which is used at sender side we can identify the pixels where secret data is embedded. Entire byte is extracted in the identified positions. The color image is transformed to message in the next phase.

### C. Random Re-transformation phase

In this phase the color image is transformed to block of colors which represent the alphabets in the original message. This phase again uses same random key for identifying order of alphabets to have correct mapping of colors to alphabets. This is the reverse transformation to the transformation done at sender's side.

### D. Decryption phase

This the final phase done at receiver side by which we can get back the original message. In this phase the block of colors is mapped to the alphabets based on the secret key which is used for encryption at sender's side.

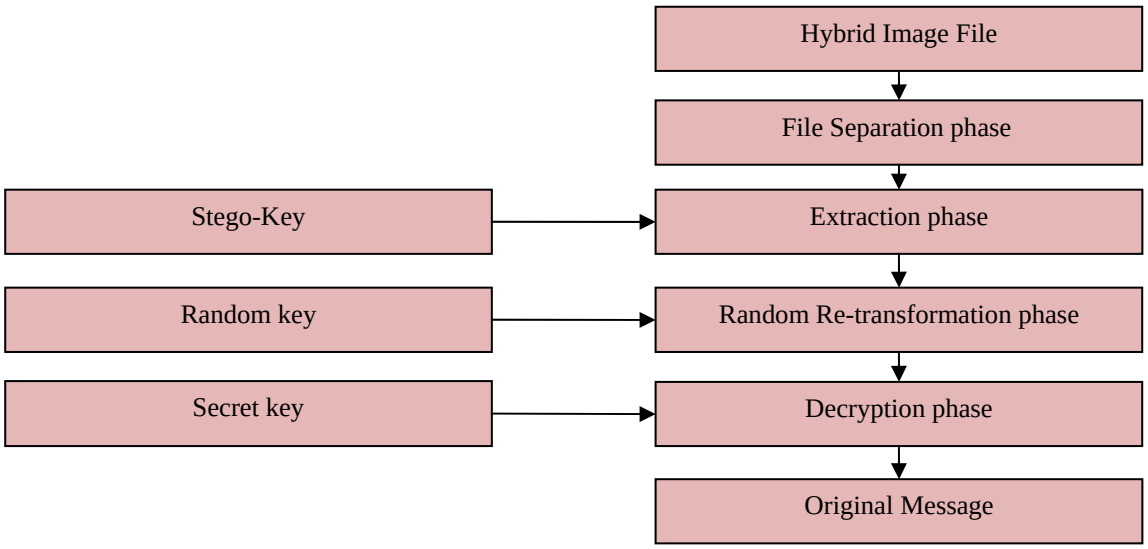


Figure 5. Overview of proposed system receiver side.

## IV. RESULTS

### A. Simulation Results

We have implemented our algorithm in java. To see the performance of our algorithm on different message sizes we have varied the size of the plaintext embedded in the image. Our results show that increasing the text size has increased the size of the resultant image. However the size of the resultant image is still acceptable. Our algorithm performs better for small text size (less than 100 bytes) where we can fix the size of the resultant image and consecutively the encryption time will be fixed. The simulation results for RSDHAF- sender side and receiver side are shown below from figure 6 to figure 14.

### B. Analysis and Comparison

The performance of the proposed technique has been highlighted here with the concept being used in the case of Least Significant Bits (LSB) methods. As the name suggests, the information is stored here by changing the least significant bits of a required number of bytes of the image. Although, this simplicity of replacement gives the advantage of the LSB techniques, in the other hand the message can easily be destructed by the attacker by interchanging and/or placing either zero or one at the respective least significant positions.

Our proposed system provides hierarchical security in various phases. As the final transmitted image is the hybrid image, it helps in creating mystification to the analyst about which part of the image actually contains secret data. If the analyst succeeded in identifying the container image also, it will be very difficult to retrieve the secret data from the container image because of the new embedding process. As the embedding is done by using complete byte at random positions for storing the color image data based on stego-key, it is strong against bit inversion mechanisms used by analysts. The bit inversions (mostly LSB) [9], [10] and [11] done by analyst have less impact on the actual color image retrieved. Major part of the column still contains the color of the alphabet which serves the purpose in identifying the character. The encryption process retains less information in the ciphered data by using random color mapping. Only 36 colors used in the possible color mappings of 256*256*256.

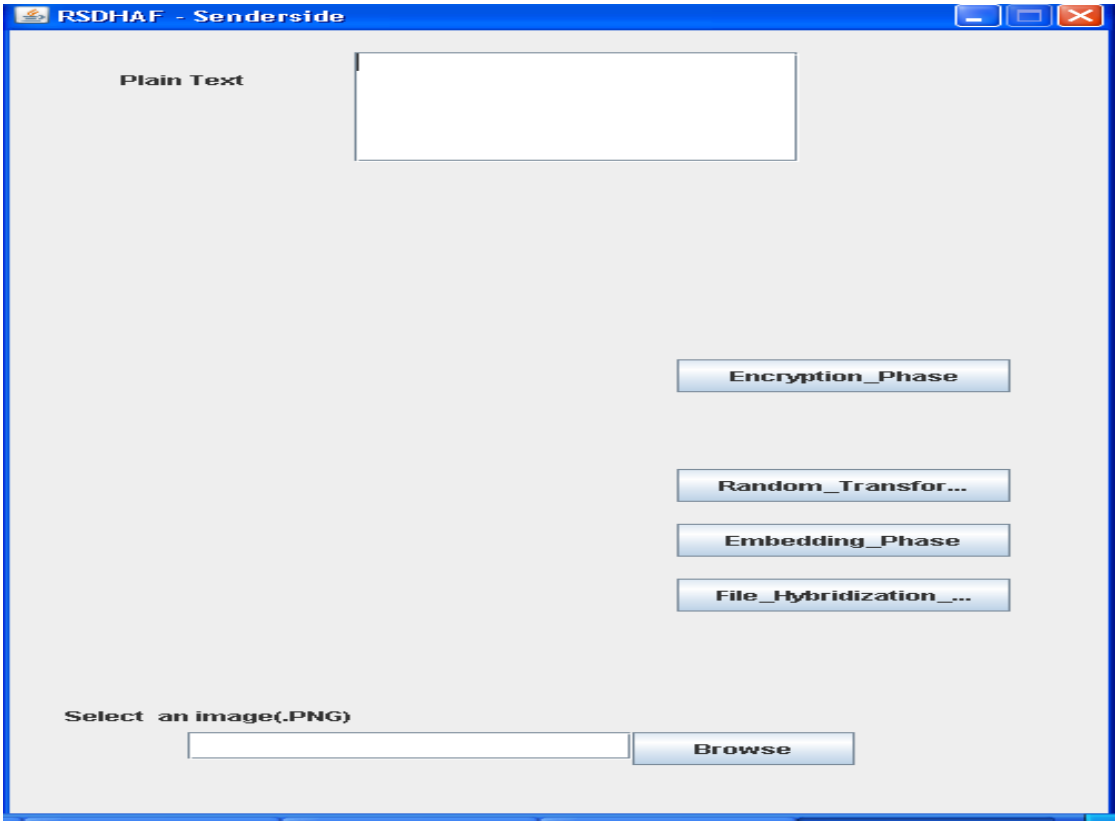


Figure 6. RSDHAF Sender side - Initial page.

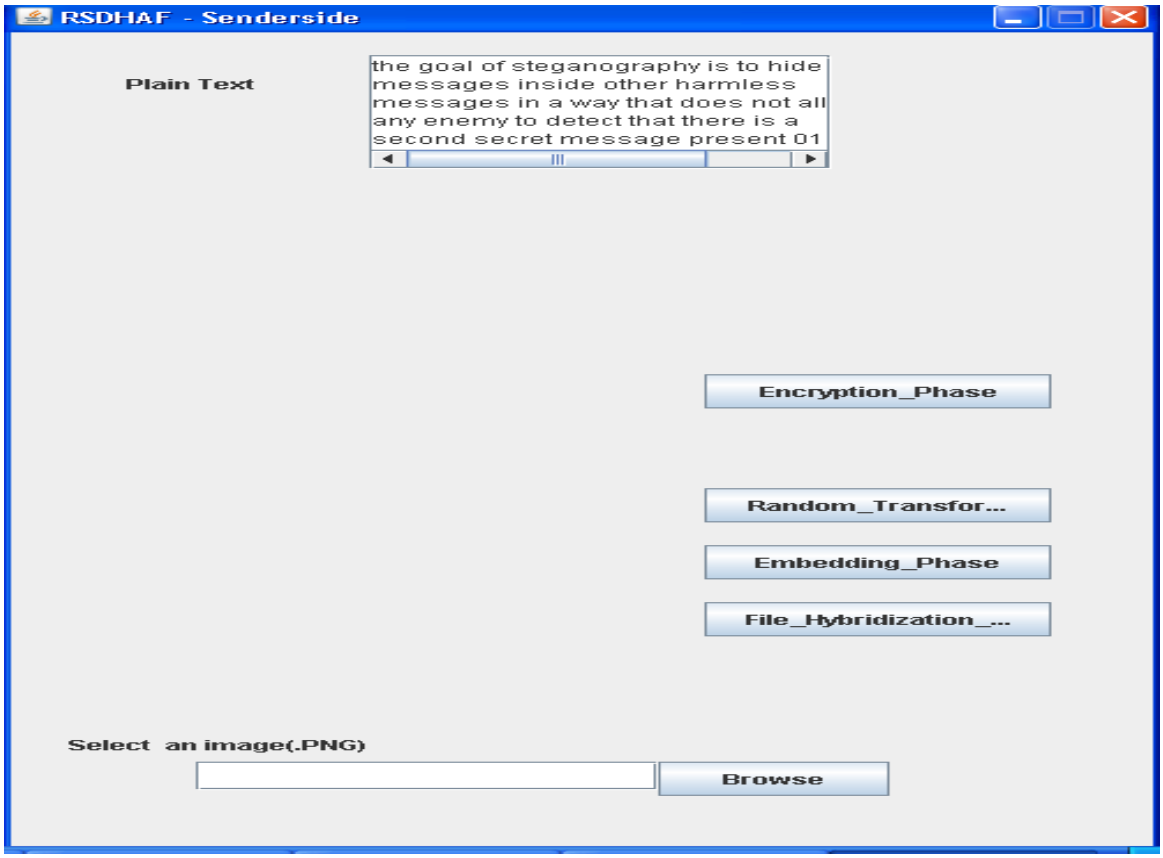


Figure 7. RSDHAF Sender side - After entering plain text.

.

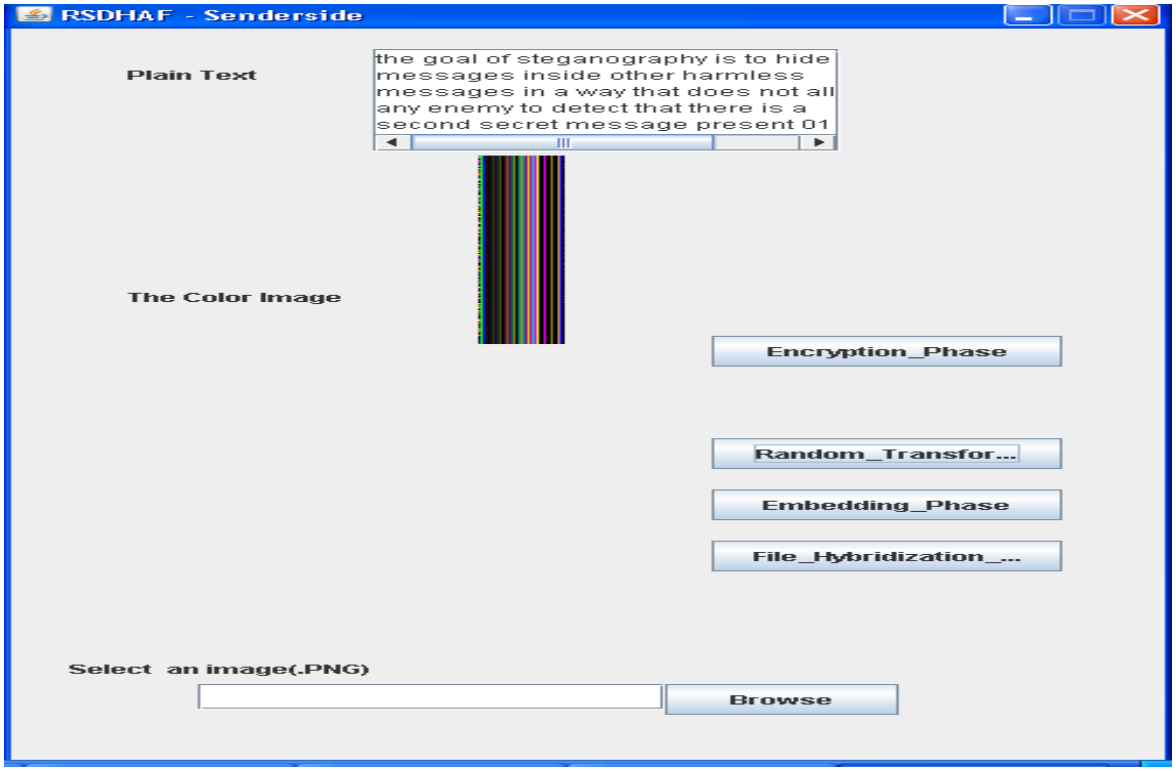


Figure 8. RSDHAF Sender side - After Random Transformation Phase.

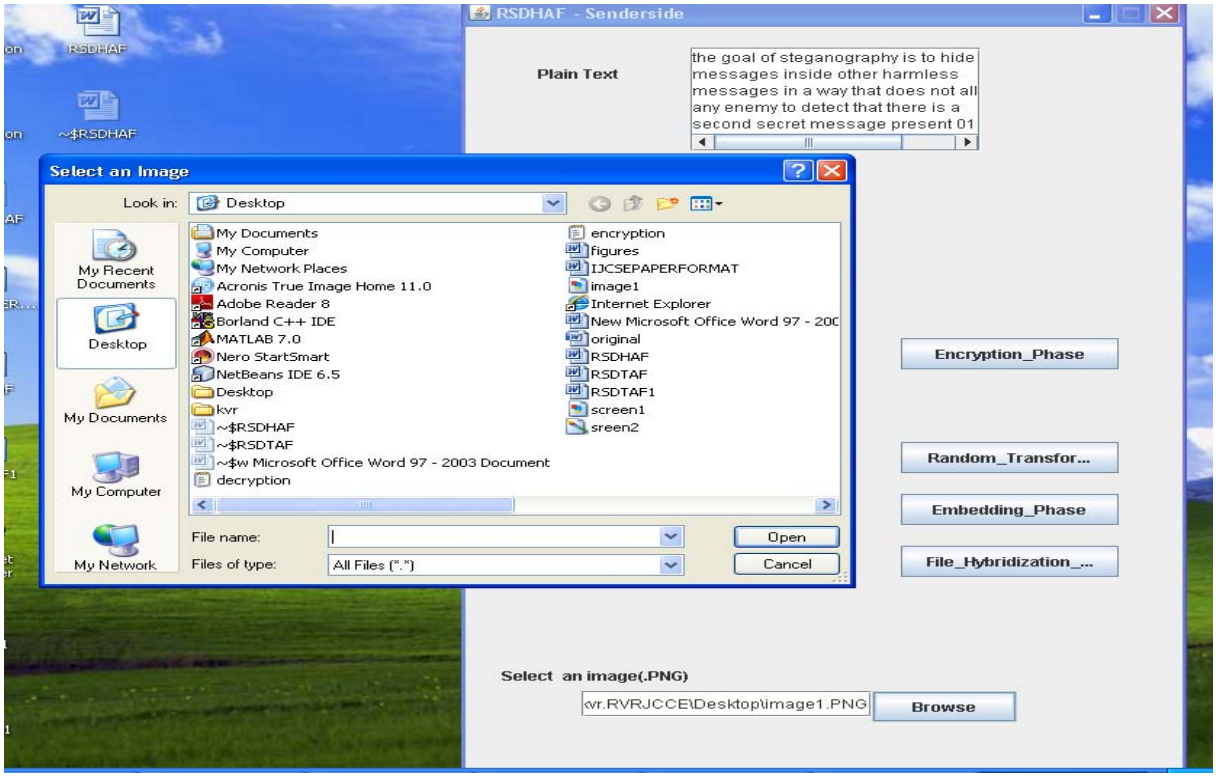


Figure 9. RSDHAF Sender side - Giiving path to select container image.

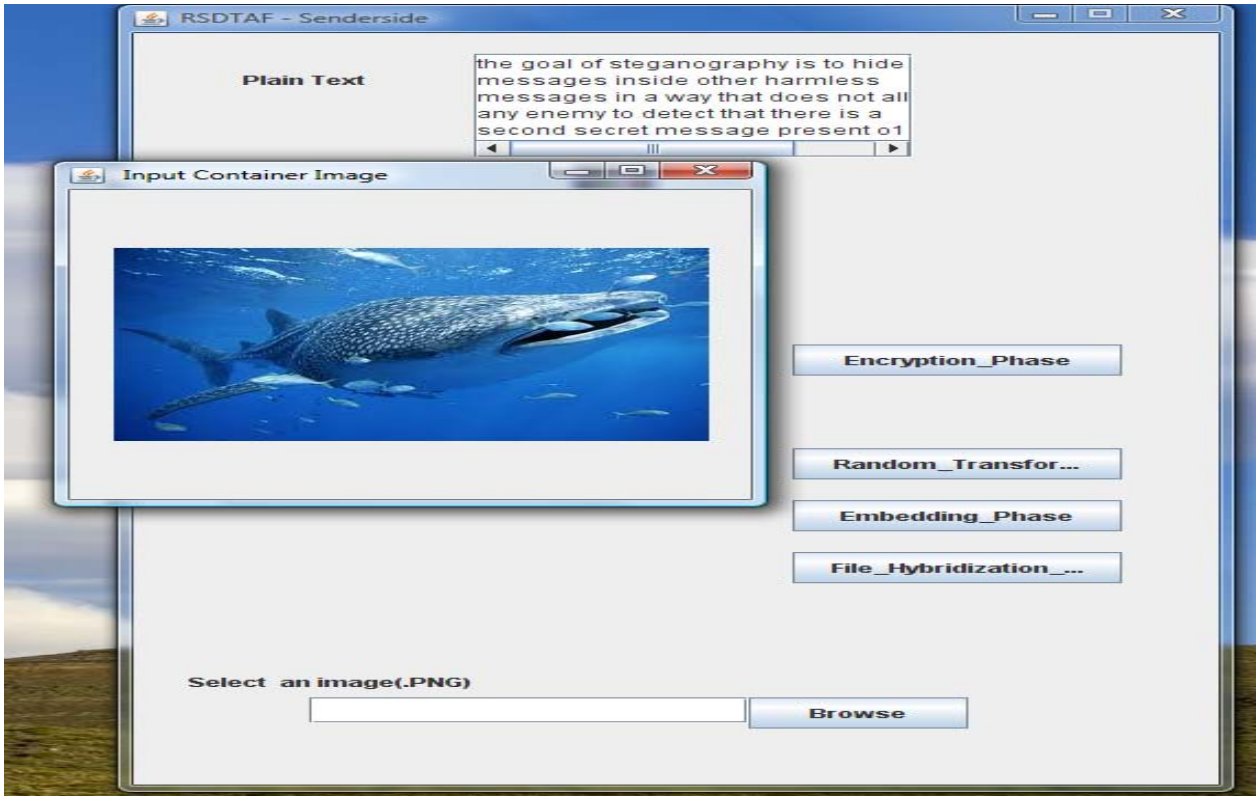


Figure 10. RSDHAF Sender side - After embedding color image into container image.

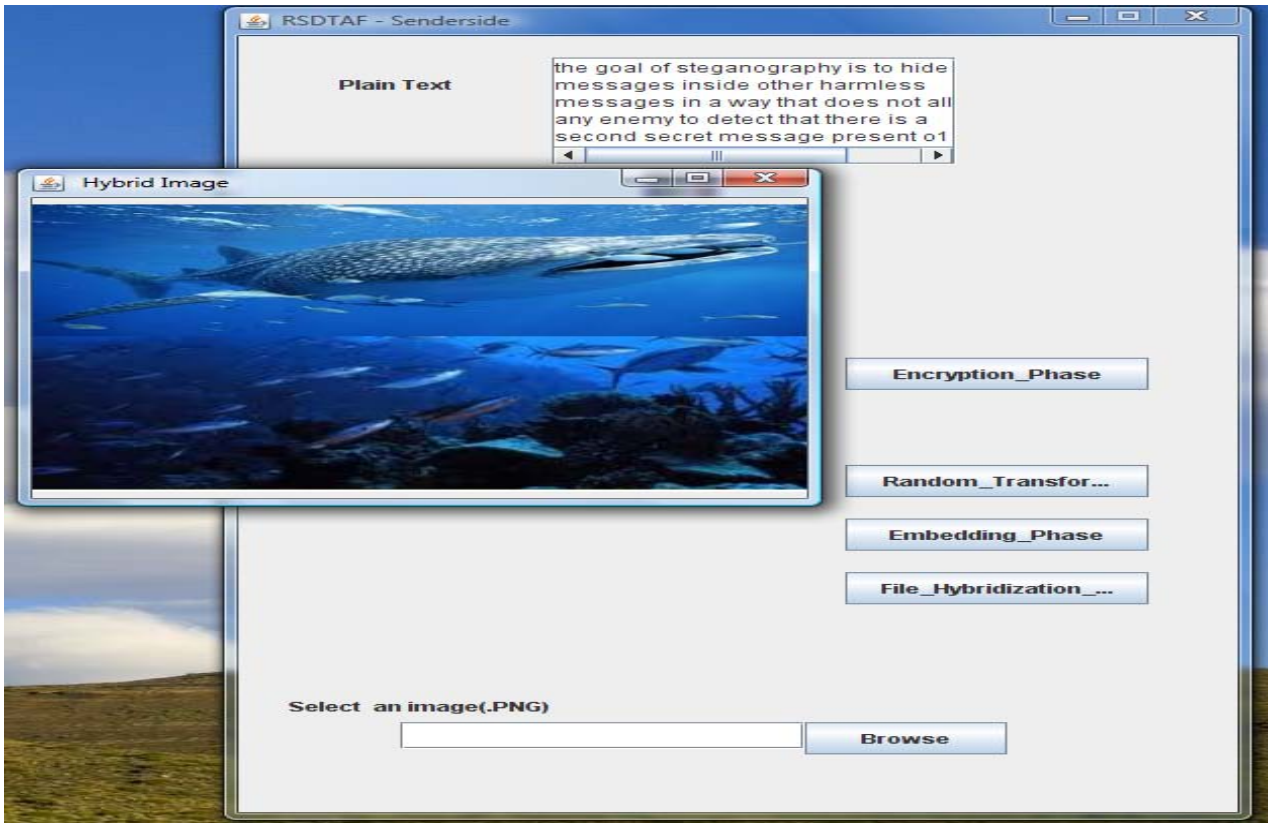


Figure 11. RSDHAF Sender side – After File Hybridization phase obtaining final hybrid image to be transmitted.

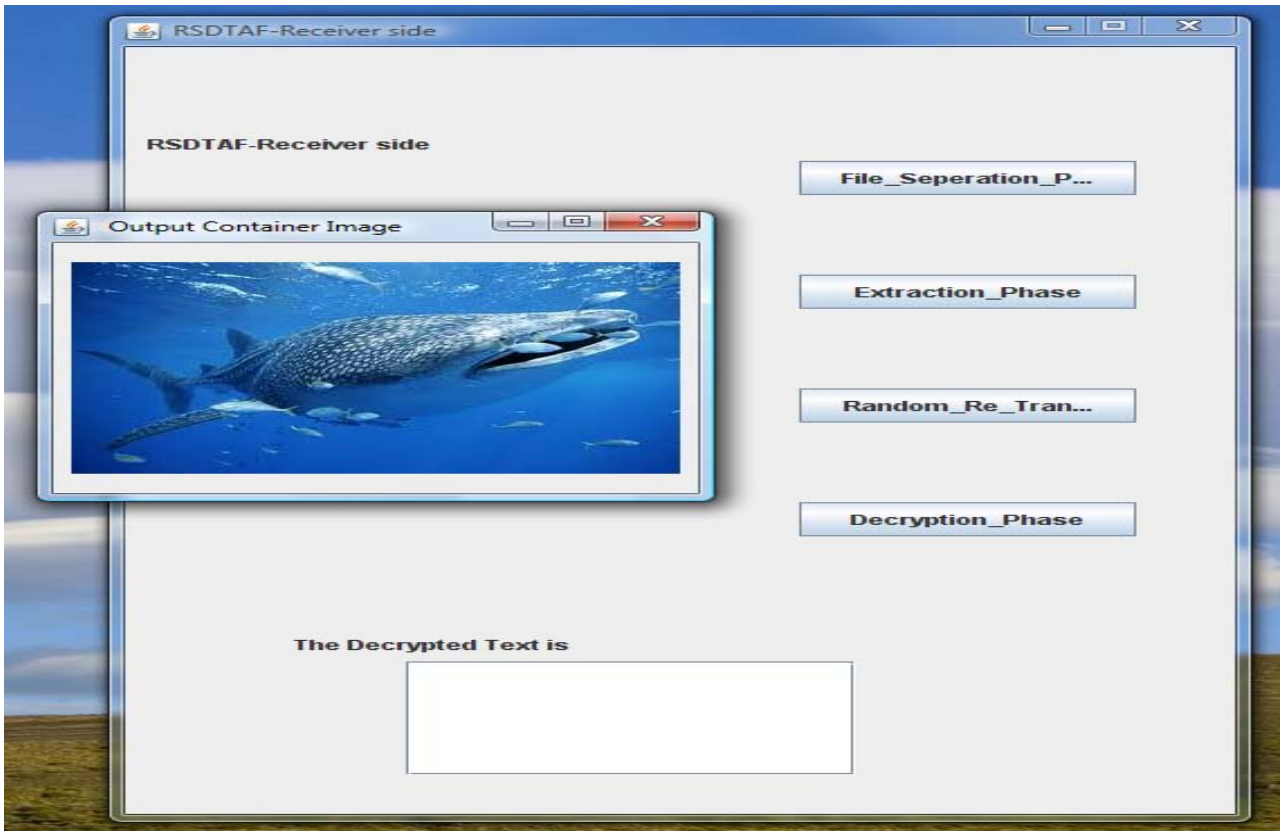


Figure 12. RSDHAF Receiver side - After File Seperation phase obtaining container image.

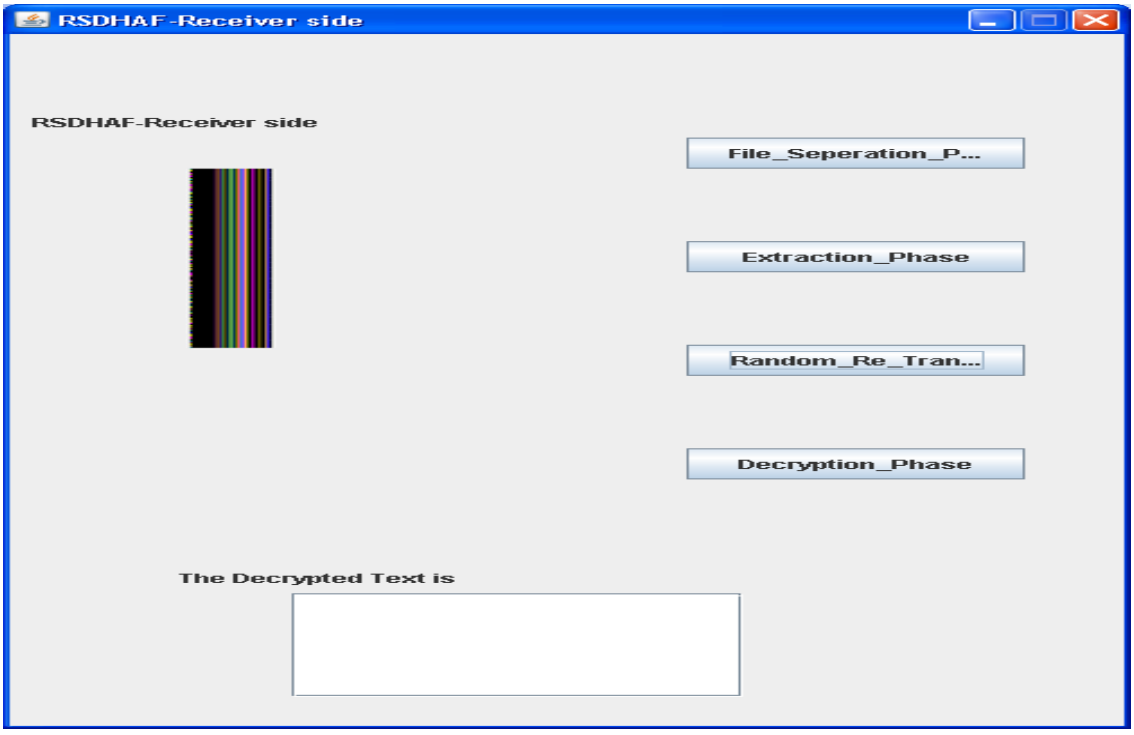


Figure 13. RSDHAF Receiver side – After Random Re-Transformation phase color image is obtained.

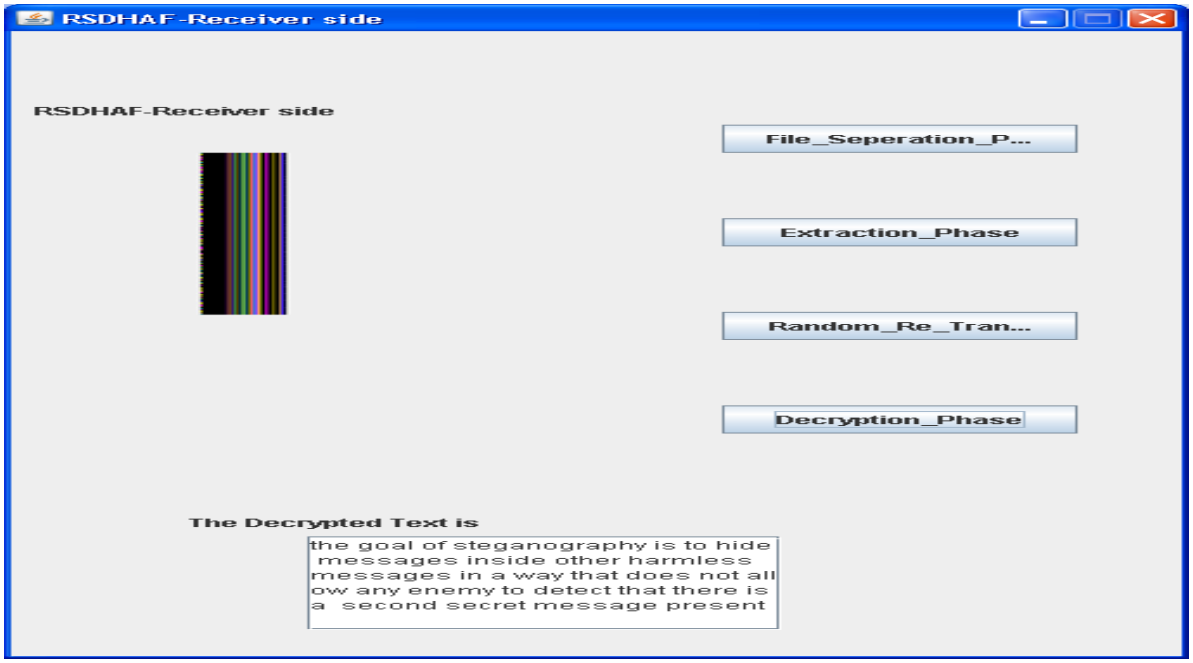


Figure 14. RSDHAF Receiver Side – Original message is obtained after decryption phase.

## V. CONCLUSION

Our goal in this paper is to propose a new steganography mechanism that allows the hiding of a data in a colored image with better security. The suitability of steganography as a tool to conceal highly sensitive information has been discussed by using a new methodology sharing the concept of hybridization and a multilevel of security of data is achieved. This suggests that an image containing encrypted data can be transmitted to anybody any where across the world in a complete secured form. Industries like music, film, publishing and organization like ministry and military will definitely be highly benefited by the use of such techniques. We can conclude here by saying that combination of both steganography and cryptography can provide us a double layer of protection.

## References

[1] B. Macq and J.J. Quisquater, “Cryptology for digital TV broadcasting" ,Proceedings of the IEEE, vol. 83, no. 6, pp. 944-956, Jun. 1995.
[2] A.A.Zaidan, B.B.Zaidan, Fazidah Othman, “New Technique of Hidden Data in PE-File with in Unused Area One”, International
[3] Journal of Computer and Electrical Engineering (IJCEE), Vol.1, No.5,ISSN: 1793-8198, 2009, pp 669-678.
[4] A.A.Zaidan, B.B.Zaidan, M.M.Abdulrazzaq, R.Z.Raji, and S.M.Mohammed, “ Implementation Stage for High Securing Cover-File of Hidden Data Using Computation Between Cryptography and Steganography”, International Conference on Computer Engineering and Applications (ICCEA09), Telecom Technology and Applications (TTA), Vol.19, Session 6, p.p 482-489, ISBN: 978-1-84626-017-9,June 6 (2009), Manila, Philippines
[5] A.W. Naji, Shihab A. Hameed, B.B.Zaidan, Wajdi F. Al-Khateeb,Othman O. Khalifa, A.A.Zaidan and Teddy S. Gunawan, “ Novel
[6] Framework for Hidden Data in the Image Page within Executable File Using Computation between Advance Encryption Standared and Distortion Techniques”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 3,No 1 ISSN: 1947-5500, 2009, P.P 73-78.
[7] G. Sahoo and R. K. Tiwari, “ Designing an embedded Algorithm for Data Hiding using Steganographic Techniques by File Hybridization ” , International Journal of Computer Science and Network Security (IJCSNS),Vol.8,No.1,January 2008.
[8] Mohammed A.F, Al-Husainy, “Image Steganography by Mapping Pixels to Letters”, Journal of Computer Science 5 (1): 33-38, ISSN 1549-3636 ,2009 .
[9] S. Katzenbeisser, F. Petitcolas, “ Information Hiding Techniques for Steganography and digital watermarking ”,2000, pp 17-76.
[10] Yogendra Kumar Jain, R. R. Ahirwal, “A Novel Image Steganography Method with Adaptive Number of Least Significant Bits Modification Based on Private Stego-Keys ”, International Journal of Computer Science and Security (IJCSS), Vol.4,No.1, ISSN (Online): 1985-1553 , March 2010, pp 40-49
[11] Anderson. R. J , F. A. P. Petitcolas, “ On the limits of steganography”, IEEE J. Selected Areas in Commun., Vol .16, Issue 4, ISSN: 0733-8716 , 1998, pp 474 – 481.
[12] A. D. Ker, “A general framework for structural analysis of LSB replacement”, in M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez- González, editors, Information Hiding, 7th International Workshop, volume 3727 of Lecture Notes in Computer Science, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.
[13] A. D. Ker and R. Böhme, “ Revisiting weighted stego-image steganalysis”, in E. J. Delp and P. W. Wong, editors, Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, volume 6819, pages 5 1–5 17, San Jose, CA, January 27–31, 2008.

## AUTHORS PROFILE

[1]**Ms K. Venkata Ramana** obtained her B. Tech in Computer Science and Engineering from R.V.R. & J.C. College of Engineering, Guntur. She received her M.Tech. in Computer Science and Engineering at R.V.R. s& J.C. College of Engineering, Guntur. She is pursuing Ph.D. in Computer Science and Engineering at Acharaya Nagarjuna University, Guntur. She is currently working as Asst. Professor at R.V.R. & J.C. College of Engineering, Guntur. She has 10 years of teaching experience. Her research areas of interest include Information Security, Cryptography & Network Security and Computer Networks.

[2]**Dr B. Raveendra Babu** obtained his Masters in Computer Science and Engineering from Anna University, Chennai. He received his Ph.D. in Applied Mathematics at S.V University, Tirupati. He is currently leading a Team as Director (Operations), M/s. Delta Technologies (P) Ltd., Madhapur, Hyderabad. He has 26 years of teaching experience. He has more than 25 international & national publications to his credit. His research areas of interest include VLDB, Image Processing, Pattern analysis and Information Security.

[3]**Mr Ch. Ratna Babu** obtained his B.Tech in Computer Science and Engineering from R.V.R.&J.C. College of Engineering, Guntur. He received his M.Tech. in Computer Science and Engineering at R.V.R. & J.C. College of Engineering, Guntur. He is pursuing Ph.D. in Computer Science and Engineering at Acharaya Nagarjuna University, Guntur. He is currently working as Asst. Professor at R.V.R. & J.C. College of Engineering, Guntur. He has 10 years of teaching experience. His research areas of interest include Cryptography & Network Security, Information Security, Computer Networks, and Image Processing.