Modified Texture Synthesis with Dual Encryption Standard

Meethu C M[1], Vijitha G[2]

[1]*M. Tech, Scholar,* [2]*Assistant Professor, Electronics and Communication Engineering*

Jawaharlal College of Engineering and Technology Palakkad, Kerala, India

Abstract*— The earlier stegnography method uses an existing cover image to hide messages, in this algorithm the secret message is concealed into texture image through the process of modified texture synthesis. Texture synthesis is a process which resynthesize a smaller texture image, into a new texture image with a similar local appearance and an arbitrary size. The proposed system aims to enhance the security in data hiding by encrypting the secret message using AES and encrypting the key by DNA digital coding by this way system provide a dual encryption. The proposed system uses Enhanced Least Significant Bit (ELSB) technique, it improves the performance of the LSB method because information is hidden in only one of the three colors that is BLUE color of the carrier image. This minimizes the distortion level which is negligent to human eye. The system offers three advantages firstly, the embedding capacity is proportional to the size of stego texture image. Secondly, the system is robust against steganalysis attack. Third, the reversible capability used in this results in the recovery of the source texture image and message.

Index Terms*—Texture Synthesis, DNA digital coding, Enhanced LSB.

I. INTRODUCTION

The growth of high-speed computer networks and that of the Internet, in particular, has increased the ease of information communication. Ironically, the cause for the development is also the apprehension use of digital formatted data. In comparison with analog media, digital media offers several distinct advantages but this type of advancement in the field of data communication in other sense has hiked the fear of getting the data intercepted at the time of sending it from the sender to the receiver. In the field of data communication, security-issues have got the top priority. The degree of security provided by a security tool has become the main evolutionary criteria of it. Classical cryptography is one of the ways to secure plain text messages. At the time of data transmission, security is also implemented by introducing the concept of steganography, watermarking, etc. Major image steganographic algorithms use an existing image as a cover medium to hide data. The main drawback of this approaches was the size of the cover image is fixed, hence if we need to embed more message the image get distorted. This problem is overcomed in the proposed approaches here the small texture image resamples into a new texture image with a similar local appearance and an arbitrary size.

This paper proposes a new modified texture synthesis approach to enhance the embedding capacity of image and also improves the secrecy in hiding the message by dual encryption.

In the first phase, an initial source texture is synthesized to form an arbitrary sized synthetic texture using a secret key. This phase includes creation of an index table and a composition image. During the second phase, the message to be hidden is embedded into this synthetic texture. In the third phase, the receiver takes this stego-synthetic texture and by using the secret key that was used by the sender, the receiver recovers the initial source texture. This source texture can be again used for second round of texture synthesis, if needed. The concepts of Digital Imaging are covered in the following Digital Image, Image Pre-processing, Image Analysis and Classification.

II. LITERATURE SURVEY

Secret message can be embedded in RGB 24 bit color image. This is achieved by applying the concept of the linked list data structures to link the secret messages in the images [5], [7]. First, the secret message that is to be transmitted is embedded in the LSB's of 24 bit RGB color space. In this approach instead of just hiding the data pixel by pixel and plane by plane, the procedure involves hiding the data based on the intensity of the pixels. The bits are hidden randomly in the plane instead of hiding them adjacent to each other and the planes are transmitted sporadically thus making it difficult to guess and intercept the transmitted data.

Least significant bit (LSB) insertion [14] is a common, simple approach to embedding information in a cover file. Unfortunately, it is vulnerable to even a slight image manipulation. Converting an image from a format like GIF or BMP, which reconstructs the original message exactly (lossless compression) to a JPEG, which does not (lossy compression), and then back could destroy the information. hidden in the LSBs. 24-bit images. To hide an image in the LSBs of each byte of a 24-bit image, 3 bits can be stored in each pixel. If the message to be hidden is compressed before you embed it, a large amount of information can be hided. To the human eye, the resulting stego-image will look identical to the cover image.

A simple image-based method of generating novel visual

appearance in which a new image is synthesized by stitching together small patches of existing images was developed. This process is called image quilting [1]. First, quilting is used as a fast and very simple texture synthesis algorithm which produces surprisingly good results for a wide range of textures. Second, the algorithm is extended to perform texture transfer – rendering an object with a texture taken from a different object. The minimum cost path along this surface is found out and make that the boundary of the new block. It is found out using the following equation.

$$E_{i,j} = e_{i,j} + \min(E_{i-1j-1}, E_{i-1j}, E_{i-1j+1}) \quad (1)$$

The complete quilting algorithm is as follows. First the algorithm goes through the image to be synthesized in raster scan order in steps of one block (minus the overlap). For every location, the input texture is searched for a set of blocks that satisfy the overlap constraints (above and left) within some error tolerance. One such block is randomly picked. The error surface between the newly chosen block and the old blocks is computed at the overlap region.

The block is pasted onto the texture. The size of the block is the only parameter controlled by the user and it depends on the properties of a given texture, the block must be big enough to capture the relevant structures in the texture, but small enough so that the interaction between these structures is left up to the algorithm. While the algorithm is particularly effective for semi-structured textures (which were always the hardest for statistical texture synthesis), the performance is quite good on stochastic textures as well. The two most typical problems are excessive repetition, and mismatched or distorted boundaries. Both are mostly due to the input texture not containing enough variability.

An algorithm for synthesizing textures from an input sample was developed. This patch-based sampling algorithm is fast [9] and it makes high-quality texture synthesis a real-time process. For generating textures of the same size and comparable quality, patch-based sampling is orders of magnitude faster than existing algorithms. The patch-based sampling algorithm works well for a wide variety of textures ranging from regular to stochastic. By sampling patches according to a nonparametric estimation of the local conditional MRF density function, mismatching features can be avoided across patch boundaries.

A new patch-based texture synthesis method have been introduced [3],[11]. The core of the proposed method consists of two main components: (1) a feature-weighted similarity measurement to search for the best match and (2) a dynamically prioritized-based pixel re-synthesis to reduce discontinuity at the boundary of adjacent patches. The proposed method is then enhanced with a view warping technique to better synthesize non-frontal-parallel textures (NFPTs) that cannot be synthesized well by traditional texture synthesis methods. The proposed patch-based texture synthesis method includes two major steps: (1) search in a sample texture for the best match for the current output neighborhood and (2) re-synthesize the mismatched pixels at the boundary of two adjacent patches. First, an initial patch is selected from input texture randomly and paste it to the left bottom corner of the output texture. Second, an adjacent patch is found that has the best matched neighborhood in the input texture constrained by an L-shape neighborhood. Then the best matched patch is pasted to the output image in a scan-line order patch by patch and a minimum-error-cut path is found in overlapped region. Above steps are repeated until a complete output image is formed. Finally, the repairing regions are re-synthesized. If necessary, a Gaussian filtering can be applied on re-synthesized pixels for smoother synthesis results.

III. PROPOSED METHODOLOGY

In the proposed methodology steganography uses reversible texture synthesis is used for hiding the secret messages. A texture synthesis process synthesizes a new texture image from a small texture image with a similar local appearance and arbitrary size. The patches are combined together to form the composition image in which we are embedding our secret message. The project includes mainly two major steps:

- Message Embedding Procedure
- Message Extraction and Authentication

In the first phase, an initial source texture is synthesized to form an arbitrary sized synthetic texture using a secret key. This phase includes creation of an index table and a composition image. During the second phase, the message to be hidden is embedded into this synthetic texture. In the third phase, the receiver takes this stego-synthetic texture and by using the secret key that was used by the sender, the receiver recovers the initial source texture. In the fourth phase, again texture synthesis is performed at the receiver side. Figure 1 shows the block diagram of sender side. The sender side consists of six processes, namely:

1. DNA Cryptography
2. AES Encryption
3. Enhanced LSB
4. Index Table Generation
5. Composition image generation
6. Texture Synthesis

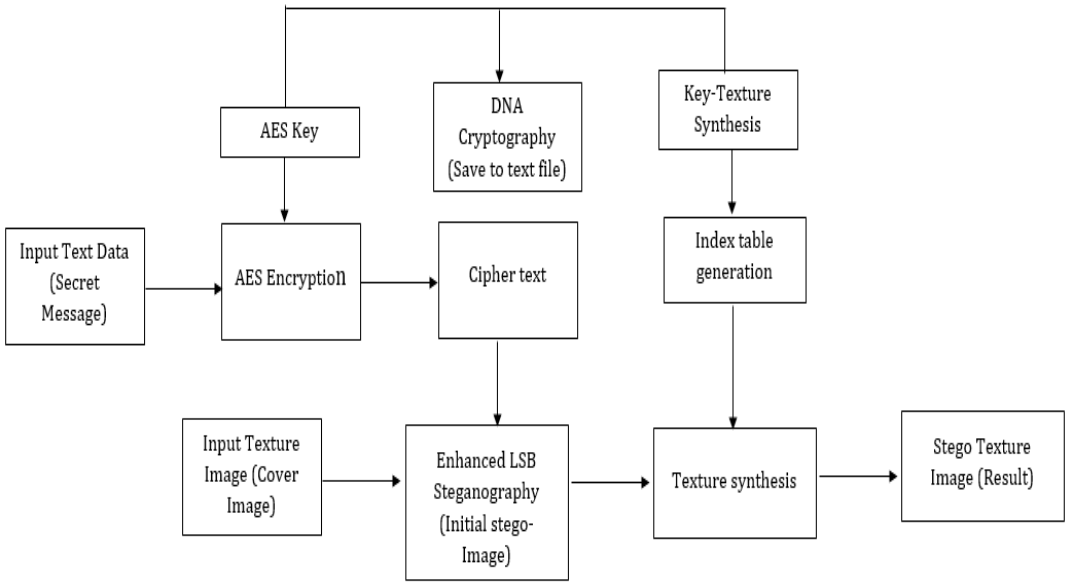

Fig.3.1 Block Diagram of Proposed System at Sender Side

A, Message Embedding Procedures

1. DNA Digital Coding

In this method, the key for AES and Index table is encrypted using DNA coding. The binary digital coding encoded by two state 0 or 1 and a combination of 0 and 1. But DNA digital coding can be encoded by four kind of base that is ADENINE (A) and THYMINE (T) or CYTOSINE (C) and GUANINE (G) [16]. There are possibly 4! =24 pattern by encoding format like (0123/ATGC).

Every bit have 2 bits like A=00, T=01, G=10, and C=11 and by using ATGC, key combinations is generated and give numbering respectively that is given in the table. Here we use DNA digital coding for encrypting the key for AES and index table generation.

Table.1.DNA digital coding

Binary value	DNA digital coding
00	A
01	T
10	G
11	C

2. AES Encryption Algorithm

This is then subjected to AES encryption algorithm. AES has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. AES operates on a 4 × 4 column major order matrix of bytes.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key. The steps are as follows:

- Key Expansions—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.
- Add RoundKey—each byte of the state is combined with a block of the round key using bitwise XOR.
- SubBytes—a non-linear substitution step where each byte is replaced with another according to a look up table.
- ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
- MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- AddRoundKey
- Final Round (no MixColumns)
  - SubBytes
  - ShiftRows
  - AddRoundKey

AES is performed as a first level of security. The security is again improved by performing the method.

3. Enhanced LSB Technique

The existing Least Significant Bit Algorithm has been analyzed and found to have a more amount of distortion, so a new method has been proposed "Enhanced Least Significant Bit (ELSB)". It improves the performance of the LSB method because information is hidden in only one of the three colors that is BLUE color of the carrier image. This minimizes the distortion level which is negligent to human eye. Enhanced LSB method that would introduce more efficiency and less distortion would store the 3 bits of information to hide in the same color [20].

Using the same example, the 3 bits of information will be introduced in the 3 LSB bits of blue color.
(10101000-10101111-10101111):
Results obtained hiding the message 111 in the pixel 10101000-10101000-10101000 with the ELSB method.

Table 2 Result with ELSB Method

	Hexadecimal	**Decimal**	Red	Green	Blue
Original Pixel	A8A8A8	11053224	168	168	168
Modified Pixel	A8A8AF	11053231	168	168	175

4. Index table Generation

The first process of this project is the index table generation where here will create an index table to preserve the location of the source patch set inside the synthetic texture. The index table will allow us to access the synthetic texture and extract the source texture wholly. The texture of any size according to our wish can be generated using this index table. The index table is used to record the location of the source patch set *SP* in the synthetic texture. The index table allows to access the synthetic texture and retrieve the source texture completely.

For the patch distribution, positioning a source texture patch on the borders of the synthetic texture is avoided. This will encourage the borders to be produced by message-oriented texture synthesis, enhancing the image quality of the synthetic texture [8]. The index table has the initial values of −1 for each entry, which shows that the table is blank. Now, the values need to be re-assigned, when the source patch ID in the synthetic texture is distributed. In this implementation, a random seed for patch ID distribution is employed, which increases the security of the steganographic algorithm making it more difficult for malicious attackers to extract the source texture. As a result, the index table will be scattered with different values.

5. Patch Based Composition

The second step that has to be used in this project is to attach the source patches into a workbench to create a composition image. First here will set up an empty image as the workbench where the size of the workbench is proportional to the synthetic texture. By referring to the source patch IDs stored in the index table, we then attach the source patches into the workbench. During the attaching process, if no imbrications of the source patches are found, we can attach the source patches directly into the workbench.

6. Composition Image Creation

The second process of the algorithm is to paste the source patches into a workbench to produce a composition image. First, a blank image is established as the workbench where the size of the workbench is equal to the synthetic texture. By referring to the source patch IDs stored in the index table, the source patches are then pasted into the workbench. During the pasting process, if no overlapping of the source patches is encountered, the source patches are pasted directly into the workbench.

7. Texture Synthesis

Now, the composition image is generated where the source patches have been pasted. The remaining areas are blank. This blank area has to be filled with the same texture. For this the texture synthesis is used. [8] A texture synthesis process re-samples a small texture image drawn by an artist or captured in a photograph in order to synthesize a new texture image with a similar local appearance and arbitrary size.

Texture synthesis is an alternative way to create textures. Because synthetic textures can be made of any size, visual repetition is avoided. Texture synthesis can also produce tileable images by properly handling the boundary conditions. Potential applications of texture synthesis are also broad; some examples are image de-noising, occlusion fill-in, and compression.

B. MESSAGE EXTRACTION

The message extraction process were done at the receiver side. This module consists of the following steps:

1. Reverse DNA Cryptography
2. Reverse Index table generation
3. Source texture recovery
4. Reverse composition image creation
5. Reverse Enhanced LSB and AES
6. AES Decryption

The same index table as that of the sender side is created by sharing the keys. The same Random function is used here. The key length and the number of rows and columns used at sender side is passed to the receiver side for creating the index table. The dimension of both the index tables will be the same.

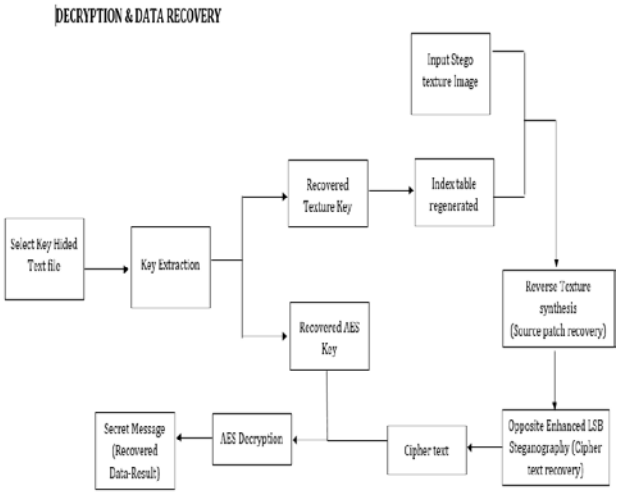

Fig.3.2 The Block Diagram of Proposed System at Reeiver Side.

The stego synthetic texture and the index table is now available at the receiver side. By referring to the locations of the source patches in the index table and by comparing with the corresponding positions in the stego synthetic texture, the source patches can be obtained. The composition image is subjected to texture synthesis. The resulting image resembles the stego synthetic texture. The stego synthetic texture is now taken and using extract string function, then decrypted secret message.

IV RESULT AND ANALYSIS

The proposed system has a login phase as part of security hence only authorized person can hide the data. The system can be divided in to three different phases they are Registration, Login and Verification. The system itself check whether the entered user id and password match then only it will grant the permission to hide data.

In the case of hiding message first we have to give key for AES encryption and Index table generation this key is encrypted using DNA encryption as a sort of enhancing security. Next, we have to select the source image to hide the data. The secret message to be embedded into the image is encrypted using AES encryption algorithm. Then Enhanced LSB, Source Patches, Key Indexing and Patch compositions were held out to get a Stego Synthetic Texture.

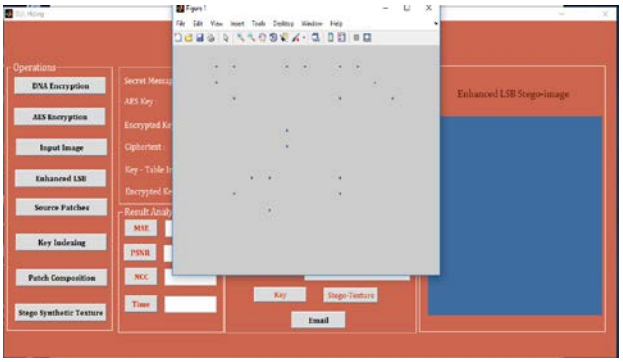

Fig.4.1. Screenshot of source patch at sender side

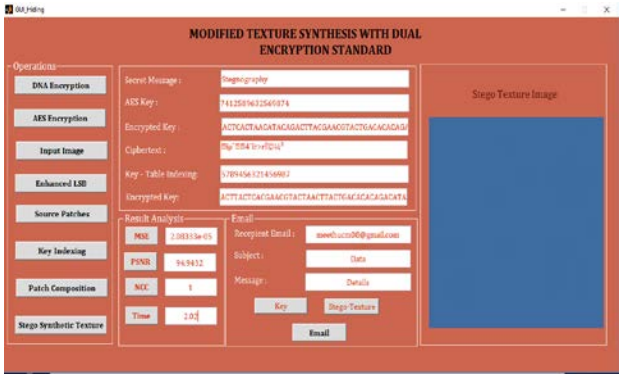

Fig.4. 2 Screenshot of the result at receiver side.

Fig.4.3 Analysis of Non-Texture image

At the receiver side where the stego image and key text file are retrieved using these two the AES and Index table key were extracted and then perform reverse key indexing and source patch recovery and patch composition is held out and then perform reverse enhanced LSB now we get cipher text this cipher text has to be converted to plain text to retrieve the original secret message this is done by AES decryption at the receiver end. Finally, the secret message hided is decoded without making distortion to

stego image.

The analysis of the proposed system is done using different analysis parameters to identify the accuracy a of the system. Here the parameters like Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Normal Cross Correlation and Computation Time are measured.

The above figure 4.3 shows the analysis of non-texture images. In this analysis five different non-texture images were selected and their parameters such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Normalized Cross Correlation (NCC) and Time for computation are calculated and are plotted below accordingly.

The figure 4.4 shows the analysis of Texture images. In this analysis five different Texture images were selected and their parameters such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Normalized Cross Correlation (NCC) and Time for computation are calculated and are plotted below accordingly.

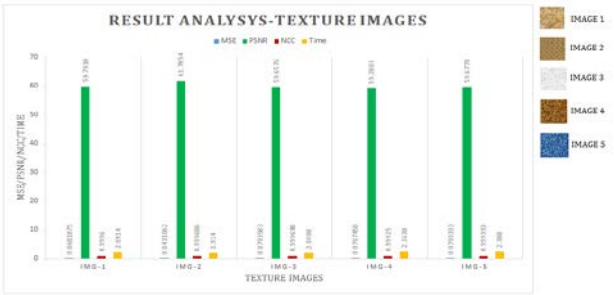

Fig.4.4. Analysis of Texture Images.

The table 3 shows the comparative analysis of existing texture synthesis approach with the proposed modified texture synthesis approach.

Table 3 Comparision of Convolutional and Modified Texture Synthesis

PRAMETERS	CONVOLUTIONAL TEXTURE SYNTHESYS	MESSAGE ORRIENTED MODIFIED TEXTURE SYNTHESYS
1. Embedding capacity	Single layer hiding is possible	3 times increased compared to old methods
2. Computational Time	Computing time is comparatively less	computing time is slightly increased
3. Synthesized results	A pure large texture	A large texture containing source texture and message
4. Image distortion	Visible	Slightly visible
5. Security	Less secure	Security increased because of dual encryption method

IV CONCLUSION

This paper proposes a reversible steganographic algorithm using texture synthesis. Given an original source texture, our scheme can produce a large stego synthetic texture concealing secret messages. The proposed system aims to enhance the security in data hiding by encrypting the secret message using AES and encrypting the key by DNA digital coding by this way system provide a dual encryption. Our method provides reversibility to retrieve the original source texture from the stego synthetic textures, making possible a second round of texture synthesis if needed. This paper also introduces Enhanced Least Significant Bit (ELSB) technique, it improves the performance of the LSB method because information is hidden in only one of the three colors that is BLUE color of the carrier image. This minimizes the distortion level which is negligent to human eye. The presented algorithm is secure and robust against RS steganalysis attack. I believe the proposed scheme offers substantial benefits and provide an opportunity to extend steganographic application. We can also embed this process in other cover medias like audio, video etc.

REFERENCES

[1] A. A. Efros and W. T. Freeman (2001), “Image quilting for texture synthesis and transfer,” in Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn, pp. 341–346

[2] Adib Akl, Charles Yaacoub,(2015) Senior Member, IEEE, Marc Donias, Jean-Pierre Da Costa, and Christian Germain, “Texture Synthesis Using the Structure Tensor”,IEEE Transactions on Image Processing, vol. 24, no. 11.

[3] Chung-Ren Yan, Tong-Yee Lee, (2009)“Texture Synthesis With Prioritized Pixel Re-Synthesis”, Journal of Information Science and Engineering, vol.25, 389-402,

[4] Dhananjay Yadav, Vipul Singhal, Devesh Kumar Bandil,(2010) “Reversible Data Hiding Techniques”, International Journal of Computer Science Engineering.

[5] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn (1999) “Information hiding-a survey,”Proc. IEEE, vol. 87, no. 7, pp. 1062–1078.

[6] H. Otori and S. Kuriyama, (2009)“Texture synthesis for mobile data communications,” IEEE Comput. Graph. Appl., vol. 29, no. 6, pp. 74–81.

[7] Komal Patel, Sumit Utareja, Hitesh Gupta,(2013) “ A Survey of Information Hiding Techniques”, International Journal of Emerging Technology and Advanced Engineering, vol 3, Issue 1.

[8] Kuo-Chen Wu and Chung-Ming Wang (2015) Steganography Using Reversible Texture Synthesis, IEEE Transactions on Image Processing, vol. 24.

[9] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum,(2001) "Real-time texture synthesis by patch-based sampling," ACM Trans. Graph., vol. 20, no. 3, pp. 127–150, 2001.

[10] L.-Y. Wei and M. Levoy (2000)"Fast texture synthesis using tree-structured vector quantization," in Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn., pp. 479–488

[11] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen,(2003) "Wang tiles for image and texture generation," ACM Trans. Graph., vol. 22, no. 3, pp. 287–294.

[12] M. Naseem, Ibrahim M. Hussain, M. Kamran Khan, Aisha Ajmal (2011) "An Optimum Modified Bit Plane Splicing LSB Algorithm for Secret Data Hiding", Int. Journal of Computer Applications (0975 – 8887), vol. 29, no.12.

[13] Masoud Nosrati, Ronak Karimi Mehdi Hariri, (2012)"Reversible Data Hiding: Principles, Techniques, and Recent Studies", World Applied Programming, Vol 2, Issue 5, 349-353

[14] N. F. Johnson and S. Jajodia (1998) "Exploring steganography: Seeing the unseen," Computer, vol. 31, no. 2, pp. 26–34.

[15] N. Provos and P. Honeyman,(2003)"Hide and seek: An introduction to steganography,"IEEE Security Privacy, vol. 1, no. 3, pp. 32–44.

[16] Prajapati Ashishkumar B and Prajapati Barkha (2016) "Implementation Of DNA Cryptography In Cloud Computing And Using Socket Programming". International Conference on Computer Communication and Informatics.

[17] Q Ke, Xie Dong-qing,(2011) "An High-capacity Steganographic Scheme for 3D Point Cloud Models Using Self-similarity Partition", International Journal of Modeling and Optimization, vol. 1, no. 1.

[18] Samir Kumar Bandyopadhyay, Indra Kanta Maitra (2010), "An Alternative Approach of Steganography using Reference Image" , Int. Journal of Advancements in Technology, vol.1, no.1.

[19] Sanjay Bajpai, Kanak Saxena,(2014) "A High End Capacity in Digital Image Steganography: Empowering Security by Mottling through Morphing",International Conference on Communications, Signal Processing and Computers.

[20] Shilpa Gupta, Geeta Gujral ,and Neha Aggarwal (2012),"Enhanced Least Significant Bit algorithm For Image Steganography".IJCEM International Journal of Computational Engineering & Management, Vol. 15 Issue 4, July 2012 ISSN (Online): 2230-7893.

[21] Usha B A, Dr. N K Srinath, Dr. N K Cauvery, (2013)"Data Embedding Technique in Image Steganography Using Neural Network", Int. Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 5.

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