JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Reversible Data Hiding Using Knight's Tour Algorithm for High Security Applications

[1]K.V.JAYA KRISHNA, [2]Prof. S VARADARAJAN

[1]M. Tech Student, Department of ECE, S.V. University College of Engineering, Tirupati, Andhra Pradesh, India.

[2]Professor, Department of ECE, S.V. University College of Engineering, Tirupati, Andhra Pradesh, India_

Abstract: **The aim of this research is to propose a novel method for Reverse Data Hiding (RDH) based on the Knight's tour algorithm, and to implement it using MATLAB 2013a version for encryption and decryption. RDH is a technique of embedding data into an image in a way that the original image can be restored without any visual distortion. The proposed method exploits the properties of the Knight's tour algorithm, which is a mathematical problem that involves finding a sequence of moves of a knight on a chessboard such that the knight visits every square exactly once. The proposed algorithm uses the Knight's tour to generate a sequence of indices that correspond to the pixels of the cover image, and uses these indices to embed the secret data. The proposed algorithm is evaluated using various metrics such as Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE), Root Mean Square Error (RMSE) to measure the quality of the stego image. The results show that the proposed method achieves high levels of security and invisibility, and can be used for various applications that require secure data transmission and storage.**

IndexTerms **– Data Hiding, Knight's Tour Algorithm, PSNR,MSE, RMSE etc.,**

1. INTRODUCTION

Computers and the Internet are the main communication medium that connects different parts of the world to one global virtual world in modern society. As a result, people can easily exchange information and distance is no longer a communication barrier. It will probably be a security issue for long distance communications. This is really important for confidential data. The solution to this problem leads to the development of a steganography plan. Steganography is a very powerful security tool that provides a high level of security, especially when associated with encryption. Encryption and steganography are well known and widely used technologies that manipulate information and encrypt or hide each entity. Encryption encrypts the message to make it unintelligible. Steganography hides the message from being seen. While both technologies provide security, research is underway to combine encryption and steganography methods in a single system to enhance privacy and security. Send feedback History Saved Community.

A. Cryptography and Steganography

Encryption is a method of storing and transferring certain types of data, so that only intended users can read and process it. The cryptographic system can be broadly categorized as a symmetric key system using a single key for both sender and receiver, a public key system using both keys, a public key known to all, and a private key recipient of the message [1] () Is used. Common terms used in encryption are:

1) Plain text
2) Encrypted text
3) Encryption
4) Decryption
5) Key.

For example, if secret data to be transmitted is encrypted, channel providers who have no knowledge of the encryption key tend to compress the encrypted data due to limited channel resources. Encrypted binary images can be compressed without loss by detecting the low density parity check code syndrome [1], but encrypted gray image lossless compression method compatible with sequential decomposition and turbo puncturing codes The flow is developed in [2]. Using the lossy compression method proposed in [3], the encrypted gray image can be effectively compressed by ignoring excessive coarse information from the coefficients generated from the orthogonal transformation. When compressing data, the receiver can reconstruct the main content of the original image by retrieving the count value. Conversion calculations in encrypted domains have also been studied. Based on the similarity characteristics of the basic cryptosystem, a discrete Fourier transform can be implemented in the encrypted domain [4]. In [5], a composite signal representation method that combines multiple signal samples and treats them as a single sample is used to reduce the computational complexity and size of the encrypted data. There is also a certain amount of work on hidden data in encrypted domains. Buyer Cellular Protocol Watermark [6], a digital media product provider, encrypts and integrates

original data with an encrypted fingerprint public key switch provided by an encrypted domain buyer. After decrypting with a private key, the buyer can get a watermarked product. This protocol allows the seller to find the version watermark of the buyer, although the buyer does not know the original version. Okamoto-Uchiyama has been proposed to improve the speed of greetings using encryption methods [7]. Encryption of public keys by homomorphic complex signal overload calculation The introduction of a mechanism to express large communication bandwidth is greatly reduced [8]. In this case, the data is encrypted and the confidentiality of the data is protected. For example [9].

In this paper The proposed method for Reverse Data Hiding (RDH) using Knight's tour algorithm can be donewhich is efficient and provides high levels of security and invisibility. It can be used for various applications such as secure data transmission, steganography, and digital watermarking.

The organizational framework of this study divides the research work in the different sections. The Literature review is presented in section 2. Further, in section 3 shown Concept of Existing System,, in section 4 shown the Methodology and section 5 shown the Performance metrics used in this work. Simulation Results work is shown in 6. Conclusion and future work are presented by last sections 7.

2. LITERATURE REVIEW

In paper [1] The Generative Topographic Mapping (GTM) algorithm is proposed as a stochastic reconstruction of the SOM (self-organizing map). The GTM algorithm captures the data structure by modeling nonlinear transformations into a multidimensional data spacethat can be used as a visualization tool in a small dimension of potential variable space. The purpose of this white paper is to extend the GTM algorithm to handle multivariate time series. The standard GTM algorithm assumes that the data is independent samples and equally distributed. However, i.i.d. This assumption is clearly inappropriate for time series. In this paper, we propose an extension of GTM for multivariable time series called GTM-ARHMM, which assumes that the time series are generated by the ARCHMM which is hidden by autoregressive.

In this paper [2], the main motivation in normal steganography is to maintain a high-quality stegano-graphic image without a doubt. Sometimes it is important that the hidden image quality is maintained. The way to get high quality hidden images is the motivation of this work. To solve this problem, prior to the masking procedure, the pixels of the secret image are analyzed to generate an optimal codebook. The most common pixel values are encoded with the shortest code to minimize stego image distortion. In addition to testing the logo for a simple hidden image distortion for evaluation, we also tested high resolution images. Using the proposed method, PSNR values of more than 45.6 dB were obtained in the still image even for high-resolution hidden images. PSNR of confidential images after recovery was more than 50dB. We can conclude that the proposed method can provide good results regardless of the type of hidden image.

In this paper [3], Copyrights Variations (CNVs) are an important genetic component in human disease studies. While re-sequencing the entire high-efficiency genome provides multiple data sources for NVC detection, the computer algorithms must be adapted to different types or sizes of NVCs depending on different experimental models. A hidden Markov model has been implemented to obtain the optimal output and resolution of CNV detection in shallow areas. A new aspect of the algorithm is the inference of the probability of performing deletions jointly in different regions. By integrating all relevant information into the complete model, this method can detect medium depth (100-2000 bp) with low depth (this method applies to simulated data and medium size deletion.

In this paper [4] Steganography, a data concealment technology, is becoming increasingly important for the development of Internet communications. As a result, a variety of steganographic algorithms have recently been proposed (eg, Ni et al.) We developed a lossless data masking algorithm based on histogram modification. To increase the Ni algorithm by random permutation and histogram re-quantization to increase it, it is not easy to break security with a random attack by applying random permutation, and by adopting histogram re-quantization, the integrated capacity can be multiplied by about 3. When we approach, it is visually impossible to distinguish between images and stegoimage. In this paper [5] The Internet is always vulnerable to unauthorized interception from around the world. The importance of reducing the amount of information detected during transmission is becoming a problem today. Decryption is the solution to the problem, but when the password is decrypted, the secret of the information no longer exists. Data masking for confidentiality, Copyright protection of digital media. Importance is also given to the use of technology used to process information. A traditional LSB modification technique that distributes message bits randomly in an image and makes it more difficult for unauthorized people to retrieve the original message is a secret. Other experiment results are shown here. All experiments are performed using Matlab 2013a simulation software.

3. EXISTING METHOD

1. ENCODER

Fig. 1 shows the flowchart of the proposed scheme. It is noted that the shadow and blank layers are embedded equally, i.e., each layer is embedded with half of secret data. With specific block size and optimal thresholds (T1, T2), the embedding procedure for shadow layer is given below.

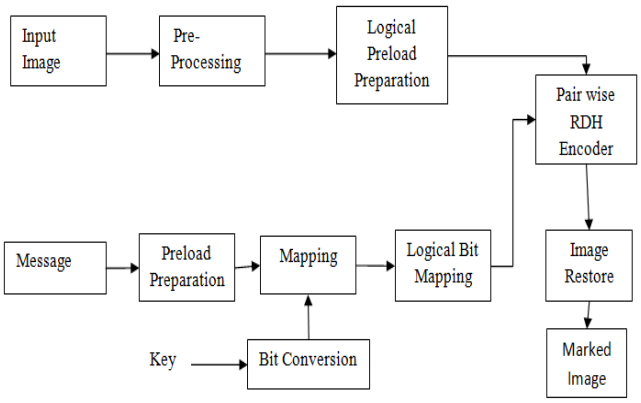

Fig.1: Existing System Block Diagram at Encoder

Reverse Data Hiding (RDH) is a technique used to embed a secret message within an image in such a way that the modification is difficult to detect. The process of RDH involves several steps, which are explained below:

Input Image: The first step is to choose an input image that will be used to embed the secret message.

Pre-Processing: The input image is pre-processed to remove any noise or distortions that may be present in the image.

Logical Preload Preparation: The preload is a set of bits that are used to hide the message. The logical preload is prepared by performing some logical operations on the preload bits.

Message Preload Preparation: The message to be hidden is pre-processed to convert it into a binary form that can be embedded in the image.

Mapping: A mapping function is used to map the binary message bits to the pixels of the input image.

Logical Bit Mapping: The logical bit mapping is a technique used to ensure that the embedded message does not significantly affect the visual quality of the image.

Key: A secret key is used to encrypt the message before embedding it in the image. This key is known only to the sender and receiver of the message.

Bit Conversion: The binary message bits are converted into pairs of bits to reduce the probability of errors during the embedding process.

Pair wise RDH Encoder: The pair wise RDH encoder is used to embed the message pairs in the image pixels. The encoder ensures that the message is embedded in such a way that it is not easily detectable.

Image Restore: The restored image is generated by extracting the embedded message from the marked image.

Marked Image: The marked image is the final image that is produced by embedding the message in the input image.

2. DECODER

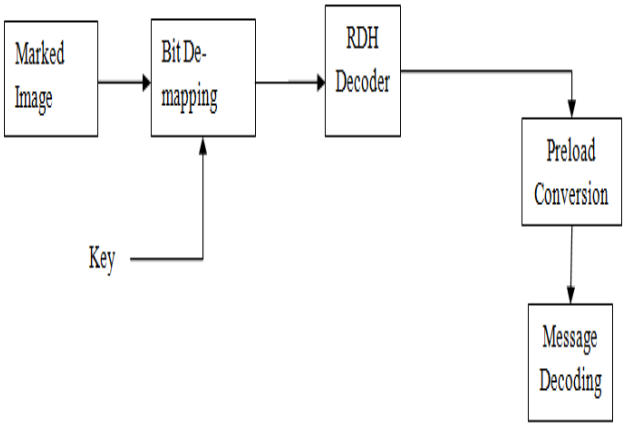

Fig.2: Existing System Block Diagram at Decoder

Reverse Data Hiding (RDH) is the process of extracting a hidden message from a marked image. The following are the steps involved in RDH by marked image:

Marked Image: The marked image is the image that contains the hidden message. This image is used as input to the RDH decoder.

Key: The secret key that was used to encrypt the message before embedding is required to extract the hidden message from the marked image. The key is known only to the sender and receiver of the message.

Bit De-mapping: The bits in the marked image are extracted by reversing the bit mapping function that was used during the embedding process. This step is required to retrieve the embedded message bits.

RDH Decoder: The RDH decoder is a program that is used to extract the hidden message from the marked image. The decoder uses the extracted bits and the logical preload to retrieve the original message.

Preload Conversion: The logical preload that was used during the embedding process is converted back to its original form using the reverse logical operations.

Message Decoding: The retrieved message bits are converted back to the original message by reversing the pre-processing steps that were used during the embedding process.

4. PROPOSED METHOD

1. ENCRYPTION

The basic idea of Knight's algorithm is to divide the cover image into blocks of equal size and then embed data into each block by modifying the least significant bit (LSB) of each pixel in the block. Specifically, the algorithm replaces the LSB of each pixel with a bit from the message to be embedded. If the bit to be embedded is the same as the original LSB of the pixel, no change is made; otherwise, the LSB is flipped to match the bit to be embedded.

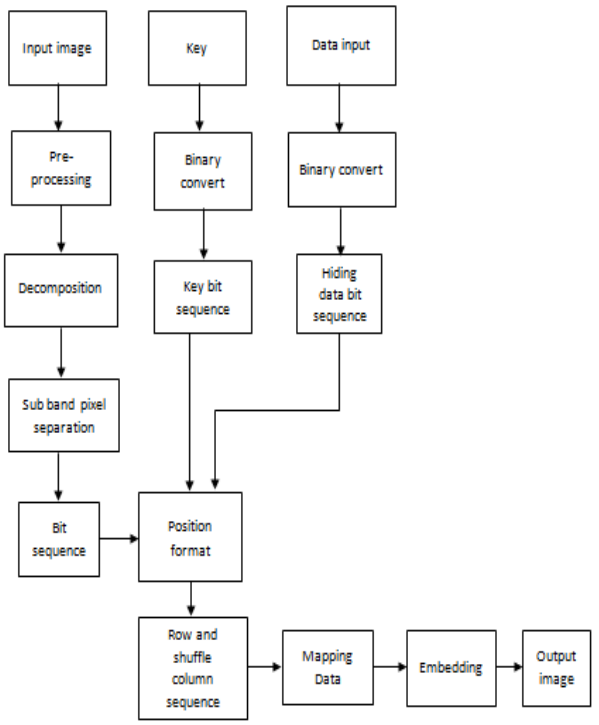

Fig.3: Proposed System Block Diagram at Encryption

Reversible data hiding is a technique that allows for the embedding of additional information into an image, while maintaining the ability to fully recover the original image and the embedded data.

The following is a high-level overview of the reversible data hiding process:

Input Image: The process starts with an input image that will serve as the carrier for the data to be embedded.

Pre-processing: The input image may be pre-processed to enhance its quality, reduce noise, or make it more suitable for the data hiding process.

Decomposition: The input image is then decomposed into sub-bands using a suitable transform such as wavelet or DCT.

Sub-band pixel separation: Each sub-band is separated into its individual pixels, which will be used for embedding the data.

Bit sequence: The data to be embedded is converted into a sequence of bits.

Key: A secret key is generated and used to determine the location and value of the embedded data. The key is kept secret and is required for successful recovery of the embedded data.

Binary Convert: The key is converted into binary format.

Key bit sequence: The binary key is used to generate a sequence of key bits that will be used in subsequent steps.

Data input: The data to be embedded is converted into binary format.

Binary Convert: The binary data is converted into binary format.

Hiding data bit sequence: The binary data is processed to create a sequence of hiding data bits that will be used to embed the data.

Position format: A suitable format is chosen to represent the position of the pixels where the data will be embedded.

Row and shuffle column sequence: The pixel positions are then organized into a sequence that will be used for embedding the data. This sequence may involve sorting the pixels based on their row and column positions.

Mapping data: The hiding data bit sequence is mapped onto the pixel positions in the sequence created in step 13, using the key bit sequence to determine the value of each hiding data bit.

Embedding: The data is embedded into the pixels at the positions determined in step 14. This may involve modifying the pixel values slightly in order to encode the additional data.

Output Image: The final step is to generate an output image that contains both the original image and the embedded data. This output image should be identical to the input image, except for the additional data that has been embedded.

2. DECRYPTION

Reversible data hiding is a technique that allows for the embedding of additional information into an image, while maintaining the ability to fully recover the original image and the embedded data. The below figure shows the Decryption

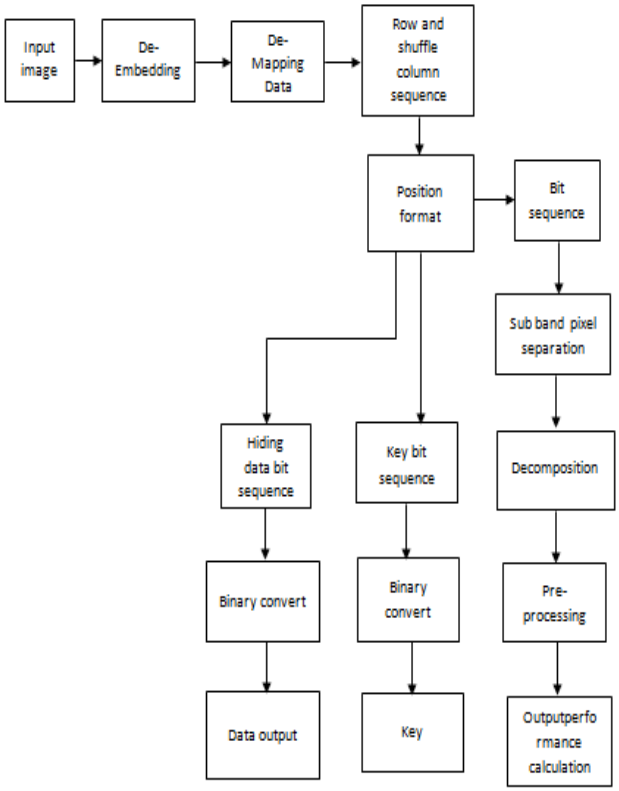

Fig.4: Proposed System Block Diagram at Decryption

Input Image: The input image is the carrier image where the data will be hidden.

De-embedding: The process of extracting the hidden data from the carrier image is known as de-embedding.

De-mapping Data: The extracted data is mapped back to its original format.

Row and Shuffle Column Sequence: The rows and columns of the image are shuffled based on a particular sequence.

Position Format: The position of the data bits is identified and stored in a specific format.

Hiding Data Bit Sequence: The data bits to be hidden are arranged in a specific sequence.

Binary Convert: The data bits are converted into binary format.
Data Output: The data bits are embedded in the carrier image, and the resulting image is the output of the process.
Key Bit Sequence: A sequence of key bits is used to encrypt the data.
Binary Convert: The key bits are converted into binary format.
Key: The key is used to decrypt the data.
Bit Sequence: The data bits are separated into sub-bands, and the pixels are analyzed based on their frequencies.
Sub-Band Pixel Separation: The sub-bands are separated, and the pixels are analyzed based on their frequencies.
Decomposition: The image is decomposed into different frequency bands.
Pre-Processing: The pre-processing step involves filtering and other image enhancement techniques to improve the quality of the image.
Output Performance Calculation: The performance of the output image is calculated based on various parameters, such as image quality, compression ratio, and data hiding capacity.

.

5. PERFORMANCE METRICS

Mean Square Error (MSE):

Mean Square Error is the averaged value of the square of the pixel-by-pixel difference between the original image and stego-image. It gives us a measure of the error produced in the cover image due to the data embedding process.

$$MSE=(mxn)^{-1}\sum^{m}_{i=1}\sum^{n}_{i=1}I[I(i,j)-k(I,j)]^2$$

A lower value of MSE indicates a good quality embedding.

Peak Signal to Noise Ratio (PSNR)

The Peak Signal to Noise Ratio (PSNR) is the ratio between maximum possible power and corrupting noise that affect representation of image. In other words, PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually expressed as decibel scale. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale and the performance of the proposed algorithms was evaluated in terms of the visual quality and the peak-signal-to-noise-ratio (PSNR). The PSNR is commonly used as measure of quality reconstruction of image. The signal in this case is original data and the noise is the error introduced. High value of PSNR indicates the high quality of image. It is defined via the Mean Square Error (MSE) and corresponding distortion metric. The following methodology to find out the PSNR value for a given gray image is used:

$$PSNR = 10x\log\ (MAX^2/MSE)$$

Where, MAXi is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, MAXi is 255. More generally, when samples are represented B bits per sample, MAXi is 2B−1 and MSE is Mean Square Error between the filtered image and the original image

Root Mean Squared Error (RMSE)

The Root Mean Square Error (RMSE) is given by as the squared root of MSE. The root mean square error (RMSE) measures the amount of change per pixel due to the processing. The RMSE between a reference or original image, image1-K (i, j) and the enhanced image, image2- I (i, j) is given by

$$RMSE = \sqrt{\frac{1}{M*N}\sum_{i=0,\ j=0}^{M-1,\ N-1}[I(i,j)-K(i,j)]^2}$$

Or simply,

$$RMSE = \sqrt{MSE}$$

This shows that a higher PSNR and higher value of MSE & RMSE provides a higher image quality. PSNR, MSE and RMSE can be used to evaluate the quality of an image, the bigger the values for these metrics, the less distortion. MSE is smaller, the performance is worse, which means the filteredimage is close to the original and similarity index is very high, closer to 1.

6. SIMULATION RESULTS

1. Encryption

Step 1: Take input image

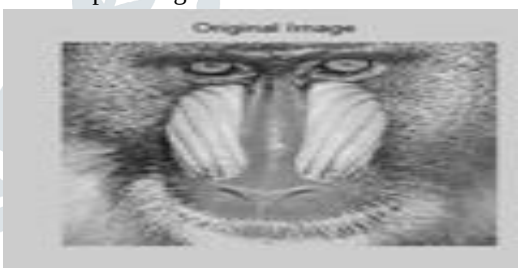

Fig.5: Input image at encryption

Step 2: Enter the encryption message length:**22**
Step 3: Enter the encryption message to hide**: baboon**
Step 4: Enter the data key between [0-255]: **253**

2. Decryption

Step 1: Enter the data key between [0-255]: **253**
Step 2: Decryption Process (After giving Correct Data Key it should be processed and decrypted the message)

- Data key matched and extracting will be done
- Extraction process extracts data exactly the same as embedded data
- Success
- Embedding rate equal to: 1.3209

Step 3: Decrypted message**: baboon**
Step 4: The reconstructed image is exactly the same as the original shown in below figure

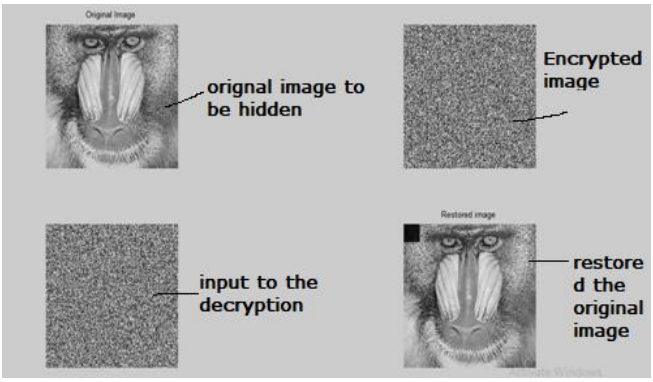

Fig.6: Output image at decryption (Decrypted image)

Table: I Comparison of Existing and Proposed system

S.NO	PSNR	MSE	RMSE
Existing System	48	0.45	0.67
Proposed System	56	0.19	0.43

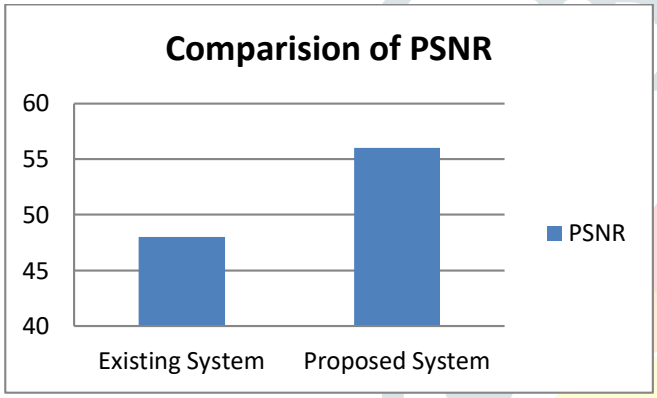

Fig.7: Comparison of PSNR for Existing system and proposed system

7. CONCLUSION AND FUTURESCOPE

This paper enhance the security of reverse data hiding is to use encryption and decryption techniques in conjunction with the Knight's Algorithm. This ensures that even if an attacker gains access to the cover image and the embedded secret data, they would not be able to decipher the data without the correct key. On the other hand, an encoder and decoder can be used for reverse data hiding. The encoder is used to embed the secret data into the cover image, and the decoder is used to extract the data.When comparing the two approaches, it is important to note that encryption and decryption offer a higher level of security, as they require the use of a key to access the embedded data.

Future Work

The Knight's Algorithm can be combined with other data hiding methods to create hybrid techniques that offer even greater security and capacity. For example, it could be combined with other encryption techniques to create a more secure system.

ACKNOWLEDGEMENT

The satisfaction that accompanies with the successful completion of the model would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

REFERENCES

1. Nobuhiko Yamaguchi “Visualizing states inautoregressive hidden Markov models using generative topographic mapping” 2012 8th International Conference on Natural Computation29-31 May 2012.
2. Yu-Ching Lu Goutam Chakraborty ; Tzu-Chuen Lu “Hidden content quality aware stego-image hiding method using re-encoding strategy” 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)8-10 Nov. 2017.
3. YufengShen ;YiweiGu ; ItsikPe'er “Poster: A Hidden Markov Model for Copy Number Variant prediction from Whole genome resequencing data”.
4. C.Y. Teng ; Y.H. Shiau ; C.C. ChenA data hiding algorithm based on histogram re-quantization 5th International Conference on Computer Sciences and Convergence Information Technology 30 Nov.-2 Dec. 2010.
5. 2011 IEEE 1st International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)3-5 Feb. 2011.
6. VandanaThakur ;MonjulSaikiaHiding secret image in video 2013 International Conference on Intelligent Systems and Signal Processing (ISSP)1-2 March 2013
7. T. Filler, J. Judas, J. Fridrich, Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes, Information Forensics and Security, IEEE Transactions on, 6, 920-935 (2011).
8. J. Tian, “Reversible data embedding using a difference expansion,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890– 896, Aug. 2003.
9. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Lossless generalized-LSB data embedding,” IEEE Trans. Image Process., vol. 14, no. 2, pp. 253–266, Feb. 2005.
10. 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, 2006.
11. L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, “Reversible image watermarking using interpolation technique,” IEEE Trans. Inf. Forensics Secur., vol. 5, no. 1, pp. 187–193, 2010.
12. X. Zhang, “Reversible data hiding in encrypted image,” IEEE.Signal Process. Lett.,vol. 18, no. 4, pp. 255–258, Apr. 2011.
13. X. Zhang, C. Qin, and G. Sun, “Reversible data hiding in encrypted images using pseudorandom sequence modulation,” in Proc. IWDW 2012, LNCS, vol. 7809, pp. 358–367, 2013.
14. W. Hong, T.-S. Chen, and H.-Y. Wu, “An improved reversible data hiding in encrypted images using side

match," IEEE Signal Process. Lett. vol. 19, no. 4, pp. 199–202, Apr. 2012.

15. W. Hong, T.-S. Chen, J. Chen, Y.-H. Kao, H.-Y. Wu, and M.C. Wu, "Reversible data embedment for encrypted cartoon images using unbalanced bit flipping," in Proc. ICSI 2013, LNCS, vol. 7929, pp. 208–214, 2013.