Journal of Engineering Sciences
Vol 10, Issue 9, Sept /2019
ISSN NO: 0377-9254

NEW SYSTEM OF REVERSIBLE INFORMATION COVERING UP IN ENCODED JPEG BITSTREAMS

[1]Zeba Afreen, [2]Dr. G .Kalaimani
[1]PG Scholar, MTech, Dept of CSE, Shadan Women's College of Engineering and Technology HYD, T.S.
[2]Professor, Dept of CSE, Shadan Women's College of Engineering and Technology HYD, T.S.

Abstract— This script initiate a distinctive anatomy of reversible information covering up in encoded JP EXPERT GROUP bitstream. Initially did a JP EXPERT GROUP encrypting calculation to encipher a JP EXPERT GROUP photo to littler dimensions and remains the arrangement consistent to JP EXPERT GROUP decoders. Once a photo proprietor relocates encode of JP EXPERT GROUP bitstreams to distributive storage, the server installs added letters beneath those cipher text to enhance a considerable scrambled JP EXPERT GROUP bitstream. During information concealing, a joined installing calculation includes two phases, the Huffman cipher mapping & the arranged histogram moving. The inserting alignment is reversible. When an accustomed applicant wants a downloading task, the server removes extra messages from the stamped scrambled JP EXPERT GROUP bitstream and recoups the actual encoded bit-stream lossless. Subsequent to downloading, the sender acquires the original JP EXPERT GROUP bitstream by an immediate decoding. The suggested structure out-performs past tasks of RDH-EI. Begins with the assignments of implanting/extraction and bitstream recuperation are altogether cultivated by server, the snap proprietor, the permitted client has no added tasks aside from JP EXPERT GROUP encoding or decoding. Second, the implanting payload is greater than cutting edge works
.

Keywords—Reversible data hiding, information hiding, image recovery, JPEG encryption

I. INTRODUCTION

Reversible information covering up in encrypted pictures (RDH-EI) is a rising system started from reversible information stowing away in plain content pictures, which has been explored by numerous analysts. In the distributed storage situation, this procedure is acknowledged by a convention that contains three parties,an picture proprietor, a cloud server and an authorized client. The picture proprietor encodes the substance before transferring pictures onto a distributed storage server. The server conceals extra messages into the encrypted pictures. The RDH-EI convention ensures that the shrouded message can be actually separated by the server, and the first substance of the pictures can be losslessly recouped by the authorized client.

RDH-EI is particularly helpful for naming the ciphertext in distributed storage, as appeared in Fig. 1. At the point when picture proprietors want to ensure their protection, RDH-EI first gives a safe encryption calculation to the proprietors to scramble their pictures before up-stacking. On the cloud side, RDH-EI enables the server to mark an encrypted picture by information covering up, e.g., concealing the personalities, timestamps, and comments into the ciphertext to produce a stamped encrypted duplicate. In this manner, the names are appended inside the ciphertext, giving a superior administration to chairmen. In the mean time, stockpiling overheads can likewise be spared. Then again, when an authorized client downloads the encrypted picture from the cloud, the first substance can be losslessly recouped after picture decoding. In customary frameworks of document the executives, the server develops a metadata record to record the data of the transferred pictures. The RDH-EI system gives an elective way, which suits extra data of the picture inside the encrypted bitstream. In this manner, no metadata documents are required any longer for marking the transferred picture

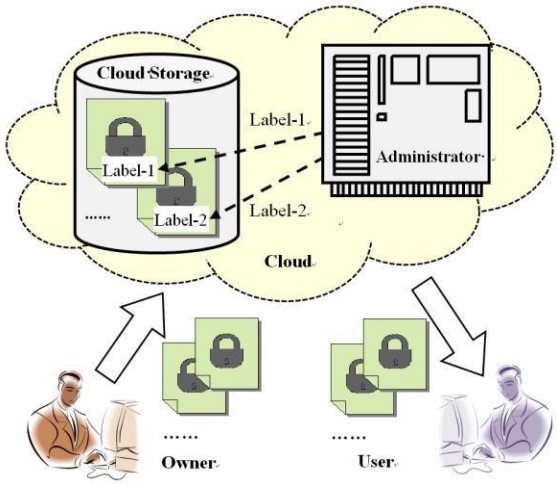

Fig 1: RDH-EI for cloud storage

Numerous RDH-EI strategies were proposed in the past fiveyears. Most techniques were intended for uncompressed pictures, while a few works were for JPEG bitstreams. Since JPEG pictures are broadly utilized on Internet, this paper centers around RDH-EI in JPEG bitstreams. We propose a novel RDH-EI

system to make this procedure increasingly functional. Contrasted and past works, two accomplishments are accomplished in the proposed strategy. To begin with, the assignments of information em-bedding, information extraction and bitstream recuperation are altogether done by the server. There are no additional calculation loads for proprietor/client aside from encryption/decoding. Second, we accomplish a bigger implanting payload than best in class takes a shot at JPEG RDH-EI.

Accomplished in the proposed strategy initially, the assignments of information em-bedding, information extraction and bitstream recuperation are altogether done by the server. There are no additional calculation troubles for possess er/client aside from encryption/decoding. Second, we accomplish a bigger inserting payload than best in class takes a shot at JPEG RDH-EI.

II. RELATED WORK

RDH-EI for Uncompressed Images

RDH-EI was first acknowledged in uncompressed nature pictures. The substance proprietor encodes the first picture utilizing a stream figure calculation and after ward transfers the encrypted picture onto the cloud. The cloud server inserts extra bits into ciphertext by flipping three least significant bits (LSB) of half pixels in each square. On the beneficiary end, the authorized client unscrambles the checked encrypted picture and creates two possibility for each square by flipping LSBs once more. Since the first square of a nature picture is smoother than the meddled square, one shrouded bit can be removed and the first square can be recuperated. This technique was improved in utilizing aside coordinate calculation to explore the spatial connection between's neigh exhausting squares. The flipping based methodologies were improved to decrease mistakes by contrasting more neighbor pixels. In any case, when the pixels in a square have indistinguishable qualities, information extraction and picture recuperation may fall flat. A swapping and moving based RDH-EI strategy was proposed to over come this disadvantage. AI was likewise utilized in RDH-EI to improve the implanting capacity.

Despite the fact that the strategies in have great implanting and recuperation capacities, information extraction must be done after picture unscrambling. This restriction makes the system less valuable in distributed storage. Distinct calculations were proposed to determine the problem. With a pseudo haphazardly created framework, the server packs some LSB-planes of the encrypted pixels to less bits to produce save spaces for concealing extra mes-sages. Hence, the shrouded bits can be separated legitimately from the checked encrypted picture. On the beneficiary side, the first substance are recouped by evaluating LSBs utilizing MSBs of neigh exhausting pixels. This technique was improved in by selecting proper bitplanes from the encrypted picture. Distributed source coding (DSC) is utilized to plan RDH-EI. The server packs some chosen bits in the encrypted picture to conceal extra messages. An a lot higher limit can be accomplished utilizing a Low-Density-Parity-Check (LDPC) based Slepian-Wolf encoder. This technique was additionally improved utilizing a dynamic recuperation calculation, which accomplishes a superior inserting rate.RDH-EI with higher embedding limit was proposed by keeping up certain redundancies amid picture encryption.

Another sort of RDH-EI is acknowledged by preprocessing the first picture. A few works require the picture proprietor to abandon save rooms in the plaintext picture before picture encryption. We name this extra task of abandoning save rooms in plaintext as preprocessing. From that point forward, the picture proprietor entombs the handled picture and transfers it onto the server. Picture encryption ought not be accounted as preprocessing,since encryption is required in all RDH-E Imethods.LSBs of certain pixels are inserted into different pixels utilizing customary RDH for plaintext pictures. Subsequently, these LSBs are abandoned as extra rooms. The handled picture is then encrypted and transferred to the cloud. On cloud, the server inserts extra bits into the encrypted picture utilizing the predefined save rooms, which gives a high installing rate.A few pixels are utilized to appraise the rest before encryption. Subsequent to scrambling the pixels and estimation mistakes, a last form of the encrypted picture is planned. The server acknowledges information stowing away by altering the estimation mistakes.Fix level scanty portrayal is utilized to investigate the relationships of neighbor pixels. Another RDH-EI approach was acknowledged by histogram moving in the spatial permuted pictures. Picture change was proposed to encode one picture to another. Extra bits are inserted by RDH in plaintext pictures. The preprocessing based techniques accomplish much better implanting rates, yet require additional RDH tasks before picture encryption.

RDH-EI for JPEG Bitstreams

The previously mentioned RDH-EI is for uncompressed pictures. Be that as it may, those techniques are not helpful in numerous applications in light of the fact that most pictures transmitted over Internet are compacted, e.g., the prominent JPEG. As a result, some RDH-EI works were proposed for JPEG bitstreams.

The first RDH-EI for JPEG bitstreams was proposed. This plan starts with a JPEG encryption calculation, in which the annexed bits of Huffman codes are encrypted by a stream figure, and all Huffman codes are kept unaltered. After encryption, the JPEG document estimate is protected and the arrangement is consistent to JPEG decoders. On cloud, the server chooses the encrypted bitstreams of certain squares as hopefuls. Extra bits are encoded by LDPC-based error correction codes (ECC) and inserted into the helpful applicant bitstream by flipping the LSB delicate he encrypted added bit that the AC coefficients in every competitor square. After the authorized client downloads and unscrambles the stamped encrypted bitstream, LSBs of the attached bits of helpful competitor s are flippedaga into gauge the extra bits utilizing a predefined blocking antique capacity and an ECC decoder. In the mean time, the first bitstream are losslessly recuperated by the separated bits. This technique is improved, in which the implanting roomwas saved before bitstream encryption. Despite the fact that the implanting limit is bigger, the preprocessing requires the picture proprietor to complete an extra calculation. Another arrangement of reversibly concealing information in encrypted JPEG bitstream was proposed, in which picture encryption and information installing are consolidated together. By scrambling the JPEG structure, the picture is encrypted and the extra bits are implanted. The technique can't be utilized in distributed storage since the server is unfit to insert bits into the encrypted bitstream.

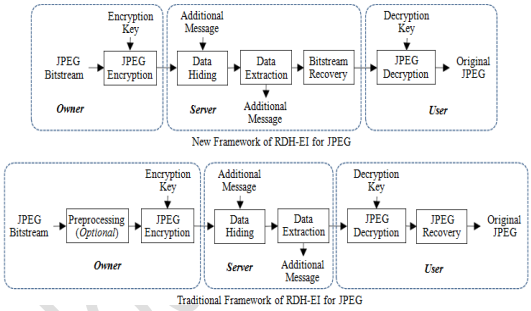

Fig 2: Framework of RDH-EI

To upgrade the security of JPEG encryption and improve the installing ability, another RDH-EI for JPEG bit stream was proposed. Amid JPEG bit stream encryption, another JPEG bitstream is built by choosing a segment of squares from the entire picture. Bit surges of the rest squares are encoded and covered up in the JEPG header. With a pressure calculation, some annexed bits of the encoded JPEG bit stream are compacted to oblige date extra bits. On the beneficiary side, the approved client utilizes an iterative calculation to recuperation the first JPEG bits tream as indicated by the variety of blocking ancient rarities. Contrasted this technique has bigger limit and better security, and the inserted bits can be straightforwardly removed by the server.

In spite of the fact that RDH-EI techniques for JPEG have great capacities in information covering up and picture recuperation, there exists one issue that the beneficiary must complete a recuperation task after unscrambling. This recuperation trouble on the beneficiary side restrains the genuine utilization of RDH-EI methods, since clients dependably would like to do nothing with the exception of picture encryption and unscrambling. To this end, this paper proposes another JPEG RDH-EI structure, in which no recuperation task is required for the proprietor or the client.

- **New Framework**

The proposed RDH-EI framework focuses on labeling the encrypted JPEG images on cloud storage. There are three parties, including the image owner, the cloud server and the authorized user. The owner encrypts a JPEG bitstream and uploads it to the cloud. The cloud server embeds additional messages into the encrypted bitstream to generate a marked encrypted bitstream. The hidden messages can be extracted from the marke dencrypted bit stream.When an authorized requires a downloading operation, the server recovers the original encrypted bitstream. After decryption, the user obtains the original JPEG image. The procedure is shown in Fig.2.

The proposed RDH-EI system centers around naming the encoded JPEG pictures on distributed storage. There are three gatherings, including the picture proprietor, the cloud server and the approved client. The proprietor encodes a JPEG bit stream and transfers it to the cloud. The cloud server implants extra messages into the encoded bit stream to produce a stamped scrambled piece stream. The concealed messages can be removed from the stamped encoded bit stream. At the point when an approved requires a downloading task, the server recoups the first encoded bit stream. After decoding, the client acquires the first JPEG picture. The methodology is appeared in Fig.2.

Fig. 3 demonstrates the customary system utilized in past works. The client needs to execute the JPEG recuperation after unscrambling. Once in a while the proprietor is likewise required to do preprocessing, i.e., producing save rooms before encryption. In Fig. 2, we propose to achieve all calculation undertakings by the cloud server. Along these lines, information extraction and bit stream recuperation are straightforward to the proprietor or

the client.

The customary structure keeps the length of scrambled piece stream unaltered after information concealing, which prompts a constrained inserting limit. To install more bits into the scrambled piece stream, the proposed structure permits the addition of bit stream length. With a condition that the measure of installed bits must be bigger than the bit stream increase, the star presented system utilizes a two-stage consolidated implanting calculation to build the inserting payload and dispense with the bit stream increase.

➢ **JPEG Encryption and Decryption**

In this area, we propose an encryption and decoding calculation for JPEG bit streams. The proposed calculation point sat cover the substance of a JPEG picture, and keep the scrambled piece stream agreeable to the JPEG decoders that are generally utilized. For curtness, we talk about the gauge JPEG for grayscale pictures. Fig. 4 demonstrates a disentangled sentence structure of the standard JPEG. We list a piece of the terms utilized in JPEG in Table I. In the accompanying passages, we utilize these abbreviations for curtness dialogs.

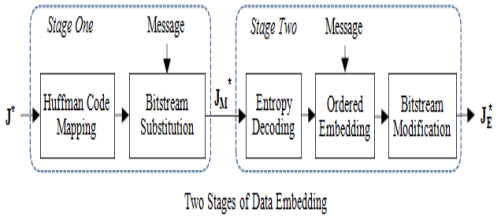

Fig 3: Stages of data embedding

III. IMPLEMENTATION

➢ **User Interface Design**

The User Interface Design plays an important role for the user to move login the Application. This module has created for the security purpose. In this login page we have to enter user name and password, it will check username and password, if valid means directly go to home page, invalid username or password means show the error message and redirect to registration page. So we are preventing from unauthorized user entering into the login page to user page. It will provide a good security for our project.

➢ **Upload videos**

In this module user is going to access the application after successful login, he will give the input for video and text message. In this module we will try to split the video in to number of different parts, and as for our convenience we will consider the 1st part of the splitted video and extract its STREAMS. If it is transmitted like this technique with side information then it is easy to retrieve the video with same quality.

➢ **Split the video into Frames**

In this module we receive the splitting video from the database. Receive the video for extracting frames. Extracting frames by using Java Media Framework (JMF). And then every video must be converted in to frames by using the JMF. Every frame from the splitting videos must be stored in database for frame indexing. Thus, the module used for extracting videos in to frames.

➢ **Water Marking**

This is fourth module of our project in this module first we encrypt plain text into Encrypted (cipher) format using the RDH-EI technique, after encryption is done the cipher text is embedded in to the user selected frame. After embedding encrypted text into frame reconstruct the all frame into single video and send to the destination.

➢ **Decrypt The Frame**

In this module we going to decrypt the text and showing the original message to the receiver. First we receive the video, to split the video into frame and chose the encrypted frame and extract the encrypted text into frame. After extract the encrypted text we decrypt the text into original format using the Reversible data hiding in encrypted images (RDH-EI).

➢ **SYSTEM ARCHITECTURE**

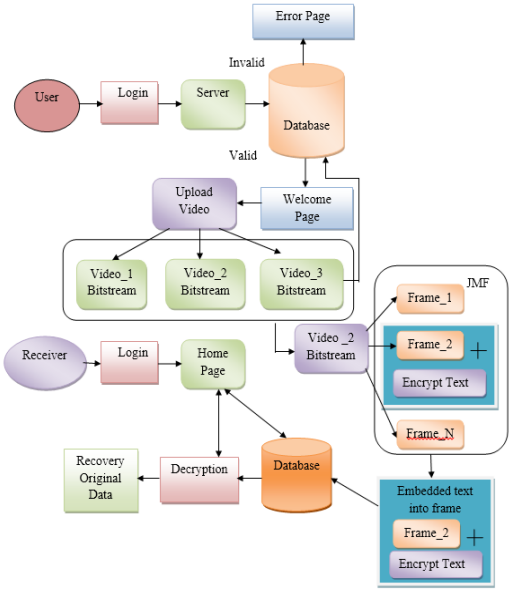

Fig 4: System Architecture

Journal of Engineering Sciences

Vol 10, Issue 9, Sept /2019
ISSN NO: 0377-9254

V. ALGORITHM

Input: Encrypted JPEG Bitstream $\mathbf{J}^*$, Additional Message $\mathbf{M}_1$
Output: Marked Encrypted Bitstream $\mathbf{J}_M^*$

1. Extract JPEG header $\mathbf{JH}^*$ and all entropy-coded segments $\mathbf{ECS}_{s(i)}^*$ from $\mathbf{J}^*$
2. Construct 162 AC Huffman codes, and divide them into two categories: the *frozen* codes and the *active* codes.
3. Extract all AC Huffman codes from $\mathbf{ECS}_{s(i)}^*$, and find the *active* codes that are used in $\mathbf{J}^*$.
4. Mapping the *used active* codes with the *unused* using (3) and (4) to represent additional bits.
5. Record the mapping by modifying $\mathbf{JH}^*$ to generate a new JPEG header $\mathbf{JH}^{*\prime}$.
6. Date embedding: according to $\mathbf{M}_1$, substitute the codes in all $\mathbf{ECS}_{s(i)}^*$ with the mapped codes to generate $\mathbf{ECS}_{s(i)}^{*\prime}$.
7. Integrate $\mathbf{JH}^{*\prime}$ with $\mathbf{ECS}_{s(i)}^{*\prime}$ to generate the marked $\mathbf{J}_M^*$.

Input: Marked Encrypted Bitstream $\mathbf{J}_M^*$, Additional Message $\mathbf{M}_2$
Output: Marked Encrypted Bitstream $\mathbf{J}_E^*$

1. Extract all entropy-coded segments $\mathbf{ECS}_{s(i)}^{*\prime}$ from $\mathbf{J}_M^*$
2. Decode all $\mathbf{ECS}_{s(i)}^{*\prime}$ into DCT blocks $\{\mathbf{B}_1, \mathbf{B}_2, \ldots \mathbf{B}_n\}$, and obtain a series of (R, V) pairs by zigzag scanning.
3. In each $\mathbf{B}_i$, collect the position of the last non-zero coefficient P_i, and count the number ρ_i of (R, V) pairs with $(0,\pm1)$.
4. Sort all blocks into $\{\mathbf{B}_{o(1)}, \ldots, \mathbf{B}_{o(n)}\}$ s.t. $P_{o(1)} \le P_{o(2)} \le \ldots \le P_{o(n)}$, and choose the first w_L blocks $\mathbf{B}_{o(1)}$~$\mathbf{B}_{o(w_L)}$ as carriers.
5. Construct a histogram $h(v)$ of all (R, V) pairs with R=0 in carrier blocks $\mathbf{B}_{o(1)}$~$\mathbf{B}_{o(w_L)}$.
6. Data embedding: implement (6) to hide $\mathbf{M}_2$ into $\mathbf{B}_{o(1)}$~$\mathbf{B}_{o(w_L)}$ by modifying $\mathbf{ECS}_{s(i)}^{*\prime}$ to $\mathbf{ECS}_{s(1)}^{*\prime\prime}$,
7. Integrate $\mathbf{JH}^{*\prime}$ with $\mathbf{ECS}_{s(i)}^{*\prime\prime}$ to generate the marked $\mathbf{J}_E^*$.

VI. RESULTS

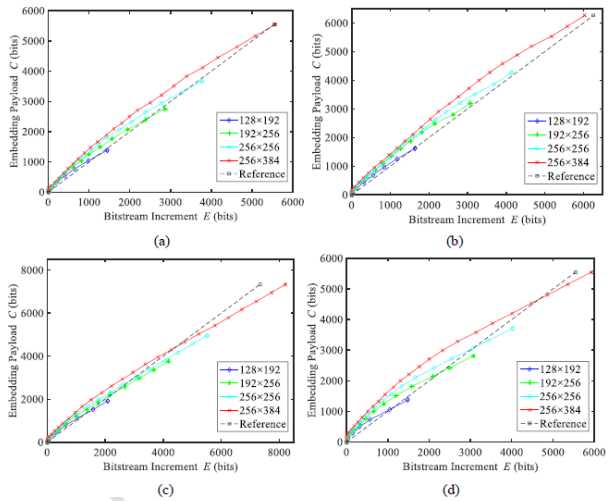

Fig 5: Embedding payload vs Bitstream increment (a) the original JPEG image Lena, (b) Peppers, (c) Lake, (d) Airplane

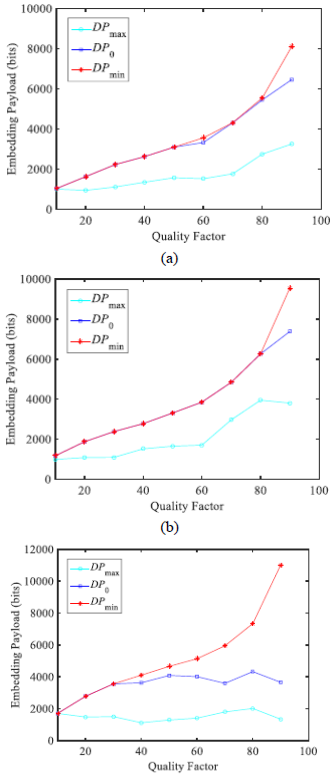

Fig 6: Embedding Payload vs. Quality Factor, (a) the original JPEG image Lena, (b) Peppers, (c) Lake

VII. FUTURE ENHANCEMENT

The proposed framework outperforms the previous JPEG RDH-EI frameworks on three aspects. First, the proposed embedding method provides a much larger payload than the other methods. Second, the proposed framework frees the computation burdens on both the owner and the user sides. While pre-processing or post-processing is required in the traditional works, the propose framework requires the owner or the user to no extra tasks except encryption or decryption. Finally, the proposed JPEG encryption algorithm is also secure against the ciphertext-only attack.

VIII. CONCLUSION

This paper proposes another system of reversible information covering up in scrambled JPEG bit streams. A JPEG encryption and decoding calculation is created to conceal the substance of the first picture. The encryption is position consistent to the prominent JPEG decoders. On the cloud side, the server can embed a lot of extra bits into the scrambled piece stream. We propose to do the implanting utilizing a mix of code mapping and requested installing. When an approved client requires a downloading activity,

the server removes the extra message and recoups the first encoded bit stream. Since the recuperation has been done before downloading, the client gets a picture precisely equivalent to the first.

The proposed system outflanks the past JPEG RDH-EI structures on three viewpoints. In the first place, the proposed embedding strategy gives an a lot bigger payload than different strategies. Second, the proposed system liberates the calculation troubles on both the proprietor and the client sides. While pre-preparing or post handling is required in the conventional works,the propose structure requires the proprietor or the client to no additional undertakings aside from encryption or unscrambling. At long last, the proposed JPEG encryption calculation is additionally secure against the figure message just assault.

REFERENCES

[1].Z. Ni, Y. -Q. Shi, N. Ansari, and W. Su, "Reversible Data Hid-ing," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 354-362, 2006.
[2] W. -L. Tai, C. -M. Yeh, and C. -C. Chang, "Reversible data hiding based on histogram modification of pixel differences," IEEE Transactions on Circuits and Systems for Video Tech-nology, vol. 19, no. 6, pp. 906–910, 2009.
[3] X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Processing Letters, vol. 18, no. 4, pp. 255–258, 2011.
[4] W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Processing Letters, vol. 19, no. 4, pp. 199–202, 2012.
[5] X. Liao, and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," Journal of Visual Communication and Image Representation, vol. 28, pp. 21–27, 2015.
[6] Z. Qian, S. Dai, F. Jiang, and X. Zhang, "Improved joint reversible data hiding in encrypted images", Journal of Visual Communication and Image Representation, vol. 40, pp. 732-738, 2016.
[7] J. Zhou, W. Sun, L. Dong, et al. "Secure reversible image data hiding over encrypted domain via key modulation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 441-452, 2016.
[8] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Transactions on Information Forensics Security, vol. 7, no. 2, pp. 826–832, 2012.
[9] X. Wu, and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," Signal processing, vol. 104, pp. 387-400, 2014.
[10] Z. Qian, and X. Zhang, "Reversible data hiding in encrypted image with distributed source encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 4, pp. 636-646, 2016.
[11] Z. Qian, X. Zhang, and G. Feng, "Reversible data hiding in encrypted images based on progressive recovery", IEEE Signal Processing Letters, vol. 23, no. 11, pp. 1672-1676, 2016.
[12] F. Huang, J. Huang, and Y. Q. Shi, "New framework for reversible data hiding in encrypted domain". IEEE Transactions on Information Forensics and Security, vol. 11, no. 12, pp. 2777-2789, 2016.
[13] K. Ma, W. Zhang, X. Zhao, et al. "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Transactions on Information Forensics Security, vol. 8, no. 3, pp. 553-562, 2013.
[14] W. Zhang, K. Ma and N. Yu, "Reversibility improved data hiding in encrypted images," Signal Processing, vol. 94, pp. 118–127, 2014.

AUTHOR'S PROFILE

Ms. ZEBA AFREEN has completed her B.Tech from Shadan women's college of engineering and technology, Khairtabad, TS District, JNTU University Hyderabad. Presently, she is pursuing her Masters in Computer Science and Engineering from Shadan Women's college of Engineering and technology, Hyderabad, TS, India.

Mrs. Dr. G. KALAIMANI has completed B.Tech (CSE) from Madras University, Tamil Nadu, M.Tech (CSE) from Mahendra engineering college, Anna University, Tamil Nadu , P.hd (CSE) from Mahendra engineering college, Anna University, Tamil Nadu Currently she is working as an Professor of CSE Department in Shadan Women's college of Engineering and technology, Hyderabad, TS. India.