JPEG Bitstream Encryption using Reversible Data Hiding

Ommi Venkateswara Rao[1], G. Divya[2]
[1]*M.Tech Scholar, Dept. Of ECE, Sai Ganapathi Engineering College, Visakhapatnam-531173,Andhra Pradesh, India.*
[2]*Assistant Professor, Dept. Of ECE, Sai Ganapathi Engineering College, Visakhapatnam-531173,Andhra Pradesh, India.*

Abstract- While most techniques of reversible data hiding in encrypted images (RDH-EI) are developed for uncompressed images, this paper provides a separable reversible data hiding protocol for encrypted JPEG bitstream. We first propose a JPEG encryption algorithm, which enciphers an image to a smaller size and keeps the format compliant to JPEG decoder. After a content owner uploads the encrypted JPEG bitstream to a remote server, a data hider embeds an additional message into the encrypted copy without changing the bitstream size. On the recipient side, the original bitstream can be reconstructed losslessly using an iterative recovery algorithm based on the blocking artifact. Since message extraction and image recovery are separable, anyone who has the embedding key can extract the message from the marked encrypted copy. Experimental results show that the proposed method outperforms a previous work in terms of separation capability, embedding capacity and security.

I. INTRODUCTION

The concept of "What You See Is What You Get, WYSIWYG" which is encountered sometimes while printing images or other material is no longer precise and would not fool a stenographer as it does not always hold true. Images can be more than what can be seen with the Human Visual System, HVS, hence, they can convey more than merely 1000 words.

For decades people strove to develop innovative methods for secret communication. This chapter highlights some historical facts and attacks on methods, also known as steganalysis. A thorough history of steganography can be found in the literature (Johnson & Jajodia, 1998), (Judge, 2001) and (Provos & Honeyman, 2003).

Ancient Steganography

The word steganography is originally derived from Greek words which mean "Covered Writing". It has been used in various forms for thousands of years. In the 5th century BC Histaiacus shaved a slave's head, tattooed a message on his skull and the slave was dispatched with the message after his hair grew back (Johnson & Jajodia, 1998), (Judge, 2001), (Provos & Honeyman, 2003) and (Moulin & Koetter, 2005).

Five hundred years ago, the Italian mathematician Jérôme Cardan reinvented a Chinese ancient method of secret writing. The scenario goes as follows: a paper mask with holes is shared among two parties, this mask is placed over a blank paper and the sender writes his secret message through the holes then takes the mask off and fills the blanksso that the message appears as an innocuous text as shown in Figure 2.4. This is an illustration of the phenomenon. Note that the Grill has no fixed pattern: (left) the mask,(middle) the cover and (right) the secret message revealed. This method is credited to Cardan and is called Cardan Grille (Moulin & Koetter, 2005).

It was also reported that, the Nazis invented several steganographic methods during World War II such as Microdots, and have reused invisible ink and null ciphers. As an example of the latter a message was sent by a Nazi spy that read: "Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils." Using the 2nd letter from each word the secret message is revealed: "Pershing sails from NY June 1"(Judge, 2001), (Lyu & Farid, 2006) and (Kahn, 1996).

In 1945, Morse code was concealed in a drawing, see Figure 1.5. The hidden information is encoded onto the stretch of grass alongside the river (Delahaye, 1996).The long grass denoted a line and the short grass denoted a point. The decoded message read: "Compliments of CPSA MA to our chief Col Harold R. Shaw on his visitto San Antonio May 11th 1945" (Delahaye, 1996).

II. LITERATURE SURVEY

P. K. Naskar and A. Chaudhuri et al. [1] introduced a symmetric encryption technique. In this system, a block size of 4bytes is processed using bit-wise operation such as XOR ingand shifting operation. A secured and efficient data transmissionis attained by combining substitution as well as transposition techniques. The size of the resulting encrypted data remains same as the size of the original secret data. This hasbeen tested and proved using different images. This condition remains same for numerous digital formats like image, audioetc. Many security analyses such as statistical analysis, keysensitivity analysis, and differential analysis was carried out toprove the strength of the proposed algorithm.

Nidhal K. El Abbadi et al. [2] have demonstrated new singular value decomposition (SVD) based encryption approachto encrypt the image for secure transmission. Here,two different matrices are obtained by processing the inputimage through two sequence scrambling process with twodifferent keys. The diagonal values in the resultant matrixare interchanged with SVD matrix. The major disadvantageof this approach is that the increase in the size of the matrixleads to the complexity of the system.

Yuanyuan Sun et al. [3] has proposed an effective image encryption technique that combines Hilbert curves and Julia

curves to protect the original image through encryption process.
In this algorithm, initial keys are generated from Julia sets parameters, which in turn are used to produce a pattern of succession. The obtained cipher image is further secured by diffusion and modulo-arithmetic operation.

Ruisong Ye et al. [4] suggested an image encryption technique based on chaotic system. In this approach, pseudorandom sequence is generated through one tent map. In order to alter the gray value of the image, the 0 and 1 bits are exchanged in a circular fashion. In addition, more pseudorandom gray sequence is generated from Bernoulli and Arnold maps. The obtained gray values are gathered in a bidirectional manner to strive from differential attack efficiently. The main drawback of this system is it is very difficult to analyze the statistical nature of both cipher and plain-images.

The work of P. K. Naskar et al. [5] focused on linear geometry approach based on symmetric encryption technique. Itis based on block ciphering method which uses linear geometry for implementing ciphering operation. The resultant bytes from cipher process are shuffled further for N position, whereN is the size of secret image. High security of encrypted imageis achieved by combining transposition and substitution techniques. The size of the encrypted image remains fixed after and before the encryption process.

Ibrahim M. Harram et al. [6] simulated AES algorithm based on data Encryption in Vb.net frame work. The main focus of AES algorithm is to ensure high data security and confidentiality. The AES codes inVb.NET frame work determines the properties and methods for using the AES algorithm. Data encryption and decryption are carried out with the help of AES codes to improve image integrity.

Jun-Xin Chen et al. [7] introduced a double image encryption scheme. This approach centralizes cross-image pixel scrambling especially in gyrator domain. In this method an iterative architecture was designed to enhance the security level of cryptosystem. Two images are given as input and each input is shuffled by the proposed cross-image pixel scrambling approach, which resulted in a uniform distribution of the pixel across the input images. The acquired scrambled images are sub-divided into real and imaginary parts like a complex function and converted into gyrator domain. The simulation result shows better performance when compared with other encryption schemes

Sunil VK Gaddam et al. [8] have presented an approach using cancellable template for generating cryptographic key. This approach consists of three phases and the cryptographic key is developed with good performance.

Abdullah Sharaf Alghamdi et al. [9] have presented a Bio chaotic stream cipher approach. In this technique image encryption is carried out by employing a biometric key in abio-chaotic function. The first step in Bio-Chaotic Algorithm(BCA) is to generate the initial condition and convert into secret key using LFSR. The second step is to generate the biometric key. The key is generated by performing XOR operation of secret key and iris template in parallel fashion. The resultant biometric key again performs XOR operation with different blocks of the iris template to ensure high security.

U. Uludag et al. [10] utilizes different techniques to preserve cryptographic key with the help of biometric pattern of a person whose information is stored in the database. It is ensured that the key is not disclosed without effective biometric authentication. The main drawback is that the biometric key varies drastically due to improper representation of a biometric identifier and impaired biometric feature extraction and matching algorithms.

Andrew Teoh Beng Jin et al. [11] have proposed a two factor authenticator scheme. This system mainly focuses on two features namely user specific finger print feature and tokenized pseudo-random number. It is generated from integrated wavelet and Fourier- Mellin Transform (WFMT) and produces a group of compact code called bio-hashing. The benefit of bio-hashing is that it gives zero error rates at any operating point.

Alghamdi S. Abdullah et al. [12] introduced a Bio-chaotic stream cipher technique. Here, biometric keys are used to carry out encryption process for the images present in the electronic media. Bio-chaotic function is preferred to increase security in the images. The encrypted and decrypted data are estimated using bio-chaotic function and keys used for this process are generated from the biometric string.

III. EXISTING SYSTEM

JPEG Theory

JPEG is an image compression standard used for storing images in a compressed format. It stands for Joint Photographic Experts Group. The remarkable quality of JPEG is that it achieves high compression ratios with little loss in quality.JPEG format is quite popular and is used in a number of devices such as digital cameras and is also the format of choice when exchanging large sized images in a bandwidth constrained environment such as the Internet. The JPEG algorithm is best suited for photographs and paintings of realistic scenes with smooth variations of tone and color. JPEG is not suited for images with many edges and sharp variations as this can lead to many artifact in the resultant image. In these situations it is best to use lossless formats such as PNG, TIFF or GIF.

It is for this reason that JPEG is not used in medical and scientific applications where the image needs to reproduce the exact data as captured and the slightest of errors may snowball into bigger ones. A JPEG image may undergo further losses if it is frequently edited and then saved. The operation of

decompression and recompression may further degrade the quality of the image. To remedy this, the image should be edited and saved in a lossless format and only converted to JPEG format just before final transmittal to the desired medium. This ensures minimum losses due to frequent saving. Image files saved in the JPEG format commonly have the extensions such as .jpg, .jpeg or .jpe

A JPEG image consists of a sequence of *segments*, each beginning with a *marker*, each of which begins with a 0xFF byte followed by a byte indicating what kind of marker it is. Some markers consist of just those two bytes; others are followed by two bytes indicating the length of marker-specific payload data that follows. (The length includes the two bytes for the length, but not the two bytes for the marker.) Some markers are followed by entropy-coded data; the length of such a marker does not include the entropy-coded data. Note that consecutive 0xFF bytes are used as fill bytes for padding purposes. Within the entropy-coded data, after any 0xFF byte, a 0x00 byte is inserted by the encoder before the next byte, so that there does not appear to be a marker where none is intended, preventing framing errors. Decoders must skip this 0x00 byte. This technique, called *byte stuffing*, is only applied to the entropy-coded data, not to marker payload data.

Encoding

Many of the options in the JPEG standard are not commonly used, and as mentioned above, most image software uses the simpler JFIF format when creating a JPEG file, which among other things specifies the encoding method. Here is a brief description of one of the more common methods of encoding when applied to an input that has 24 bits per pixel (eight each of red, green, and blue). This particular option is a lossy data compression method.

Color space transformation

First, the image should be converted from RGB into a different color space called YCbCr. It has three components Y, Cb and Cr: the Y component represents the brightness of a pixel, the Cb and Cr components represent the chrominance (split into blue and red components). This is the same color space as used by digital color television as well as digital video including video DVDs, and is similar to the way color is represented in analog PAL video and MAC but not by analog NTSC, which uses the YIQ color space. The YCbCr color space conversion allows greater compression without a significant effect on perceptual image quality (or greater perceptual image quality for the same compression). The compression is more efficient as the brightness information, which is more important to the eventual perceptual quality of the image, is confined to a single channel, more closely representing the human visual system.

This conversion to YCbCr is specified in the JFIF standard, and should be performed for the resulting JPEG file to have maximum compatibility. However, some JPEG implementations in "highest quality" mode do not apply this step and instead keep the colour information inthe RGB color model, where the image is stored in separate channels for red, green and blue luminance. This results in less efficient compression, and would not likely be used if file size was an issue.

Down sampling

Due to the densities of color- and brightness-sensitive receptors in the human eye, humans can see considerably more fine detail in the brightness of an image (the Y component) than in the color of an image (the Cb and Cr components). Using this knowledge, encoders can be designed to compress images more efficiently.

The transformation into the YCbCr color model enables the next step, which is to reduce the spatial resolution of the Cb and Cr components (called "down sampling" or "Chroma sub sampling"). The ratios at which the down sampling can be done on JPEG are 4:4:4 (no down sampling), 4:2:2 (reduce by factor of 2 in horizontal direction), and most commonly 4:2:0 (reduce by factor of 2 in horizontal and vertical directions). For the rest of the compression process, Y, Cb and Cr are processed separately and in a very similar manner.

Block splitting

After sub sampling, each channel must be split into 8×8 blocks of pixels. Depending on chroma sub sampling, this yields (Minimum Coded Unit) MCU blocks of size 8×8 (4:4:4 – no sub sampling), 16×8 (4:2:2), or most commonly 16×16 (4:2:0).

If the data for a channel does not represent an integer number of blocks then the encoder must fill the remaining area of the incomplete blocks with some form of dummy data. Filling the edge pixels with a fixed color (typically black) creates ringing artifact along the visible part of the border; repeating the edge pixels is a common technique that reduces the visible border, but it can still create artifact.

IV. PROPOSED SYSTEM

Signal processing in encrypted domain (**SPED**) for privacy preserving has attracted considerable research interests in recent years. In cloud computing and delegated calculation, users who are unwilling to reveal contents of the original signal may send an encrypted copy to a remote server. The server has to accomplish signal processing in the encrypted domain. Many approaches have been proposed for different applications, for example, compressing encrypted images, signal transformation in ciphertexts, pattern recognition in encrypted domain, watermarking in encrypted multimedia, data searching in encrypted dataset, etc. Reversible data hiding in encrypted images (RDH-EI) is another topic of SPED.

RHD-EI allows a server to embed additional message into an encrypted image uploaded by the content owner, and guarantees that the original content can be losslessly recovered after decryption on the recipient side. Generally, reversibility

is closely related to the embedding payload. If the original image can be losslessly recovered when the payload does not exceed the achievable capacity, we say it is *reversible*. Meanwhile, RDH-EI protocols are always designed for natural images. Since a natural image always contains large smooth areas, *i.e.*, redundancies, one can embed data into the original image and losslessly recover it. Unlike robust watermarking, reversible data hiding are widely used when perfect image reconstruction and data extraction are emphasized while robustness against malicious attacks is not considered.

RDH-EI is useful in many applications. For example, in cloud storage as shown in Fig. 1, an image owner may store images in the cloud. Before uploading the images, the owner encrypts the contents to preserve privacy. For management purposes, the cloud administrator can embed labels, such as user information, timestamps and remarks, into the ciphertexts. Therefore, labels are attached inside these ciphertexts, and storage overheads can be saved. The embedded information can also be extracted exactly by the administrator or authorized users. Meanwhile, when an authorized user downloads the encrypted image containing additional message from the cloud, RDH-EI protocol also guarantees that the original content can be losslessly recovered after decryption.

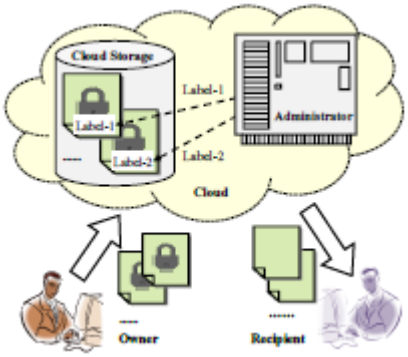

Fig.1: An example of RDH-EI application

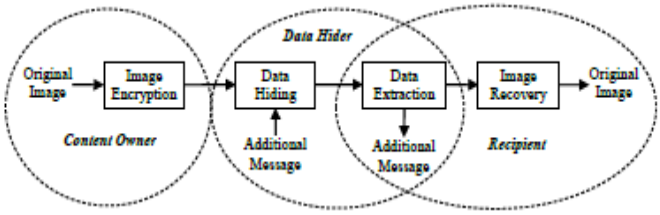

Fig.2: General framework of RDH-EI

Emerging works on RDH-EI are reviewed in Section II. While most of the related works are applicable to uncompressed images, this paper focuses on RDH in encrypted JPEG bit stream, the most popular image format, aiming at providing an RDH-EI approach with separable extraction capability, high embedding capacity, and secure encryption. We first propose an encryption scheme for enciphering JPEG bit streams. Based on JPEG encryption, a reversible data hiding method is developed for service providers to embed additional bits. Finally, we propose an iterative algorithm to recover the original image. In this work, lossless recovery is required. Although JPEG encoding itself is lossy, users always hope not to introduce further degradation to a JPEG image while uploading. That is why lossless recovery is required.

Generally, an RDH-EI framework has three parties, content owner, data hider and recipient, as shown in Fig. 2. To preserve privacy, the content owner encrypts an original image using an encryption key, and uploads the encrypted copy to a remote server. On the server side, the data hider embeds additional messages into the encrypted image using an embedding key to generate a marked version. The recipient can losslessly recover the original image using the encryption key after downloading the marked version. There are two different cases: both the data hider and recipient can extract the hidden message; and only the recipient can extract the message, hence two types of technique: *separable* RDH-EI and *joint* RDH-EI.

A. Separable RDH-EI

The word separable means separating data extraction from image recovery, i.e., additional messages can be extracted directly from the marked encrypted image without revealing the image content. Only those who have the embedding key can extract the messages from a marked encrypted image.

A separable RDH-EI method was first proposed. The data hider permutes and divides the encrypted pixels into segments, and compresses several LSB-planes of each segment to fewer bits using a pseudo-randomly generated matrix. As a result, spare room in each segment is created to accommodate additional messages. On the recipient side, LSBs of each segment are estimated using the MSBs of the neighboring pixels. After comparing the estimated bits with the extracted vectors, the recipient can recover the original contents. Since the additional message can be extracted directly from LSBs of the encrypted images, data extraction and image recovery are therefore separable. This method was by selecting appropriate bitplanes in the encrypted image, leading to a higher embedding capacity. Distributed source coding (DSC) is used to achieve separable RDH-EI. The data hider compresses some selected bits in the encrypted image to create room for the additional hidden message. In this method, the Slepian-Wolf encoder based on low density parity check (LDPC) is used. With the DSC based embedding, a much higher capacity is obtained.

B. Joint RDH-EI

In joint RDH-EI, the additional message can only be extracted by the recipient after image decryption, along with image recovery, while the data hider cannot perform extraction. In which the content owner encrypts an original image using a stream cipher, and the data hider embeds additional messages into ciphertext blocks by flipping three least significant bits (LSB) of half the pixels in each block. When extracting the additional messages, the recipient decrypts the marked encrypted image and generates two candidates for each block by flipping LSBs again. Since the original block is much

smoother than the interfered, the embedded bits can be extracted and the original image perfectly recovered. This joint RDH-EI method depends on the size of each block. As long as the block size is appropriately chosen, errors of extraction and recovery can be avoided. This method by exploiting spatial correlation between neigh boring blocks and using a side-match algorithm to achieve a higher embedding rate. The flipping based approach was further improved, in which multiple neigh boring pixels in different locations are used to reduce error rates in extraction and recovery.

Recently, a new joint RDH-EI method. Data embedding is realized through a public key modulation mechanism. On the recipient end, a two-class SVM classifier is designed to distinguish encrypted and non-encrypted image patches. Consequently, the recipient can jointly extract the additional messages and recover the original image. This method provides a higher embedding capacity.

C. RDH-EI for JPEG Bitstream

As most RDH-EI methods are designed for uncompressed spatial-domain images, proposes an approach capable of reversely hiding messages into encrypted JPEG bit streams. This scheme aims at encrypting a JPEG bit stream into a properly organized structure and embedding additional messages into the encrypted bit stream by slight modifications. During the bit stream encryption, all appended bits of the Huffman codes are encrypted with a stream cipher, and all Huffman codes are kept unchanged. After encryption, the file size is preserved, and the format is compliant to common JPEG decoders. On the server side, the bit stream of every other block is selected as a candidate. If all AC coefficients of a candidate block are zero, the block is skipped. Additional bits are then encoded by LDPC-based error correction codes (ECC), and embedded into the useful candidate bit stream by flipping the LSBs of the encrypted appended bits of the AC coefficients in each candidate block. On the recipient side, LSBs of the appended bits of each candidate bit stream are flipped again to estimate the additional bits using a predefined blocking artifact function and an ECC decoder. Meanwhile, the original bit stream can be losslessly recovered according to the extracted bits.

In some interesting ideas of RDH were proposed for JPEG images by combining image scrambling and data embedding. By scrambling the JPEG structure, additional message is embedded into the encrypted bit stream. However, in these methods data embedding must be combined with image encryption.

The framework of the proposed method is depicted in Fig. 3. The JPEG RDH-EI workflow includes three parties: *content owner*, *data hider*, and *recipient*.

Given a JPEG bitstream and an encryption key, the content owner generates a ciphertext bitstream after syntax parsing and encryption. In the process, the file size is kept unchanged and the format is compliant to common JPEG decoders.

When a remote server receives the encrypted bitstream, the data hider parses the bitstream and hides additional messages in it using an embedding key. After the marked encrypted bitstream is constructed, the file size and format compliance are preserved. In this scheme, the server can extract additional messages from the marked encrypted bitstream using the embedding key.

On the recipient side, the additional messages can also be extracted from the received bitstream if the embedding key is available. A recipient with only the encryption key can view an approximate image by a direct decryption. If both the encryption and embedding keys are available, the recipient can losslessly recover the original bitstream after decrypting the marked encrypted JPEG bitstream.

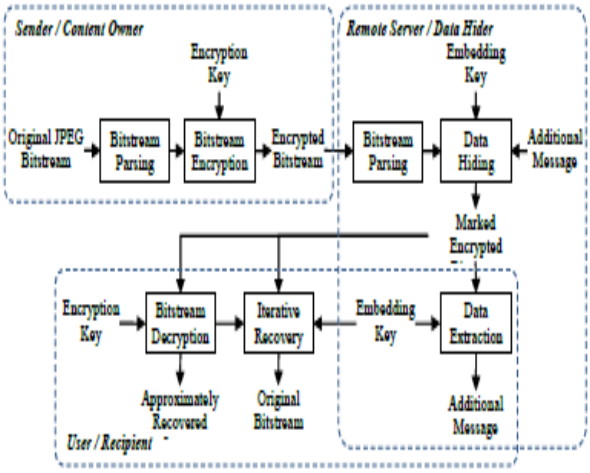

Fig.3: Framework of the proposed method

V. EXPERIMENTAL RESULTS

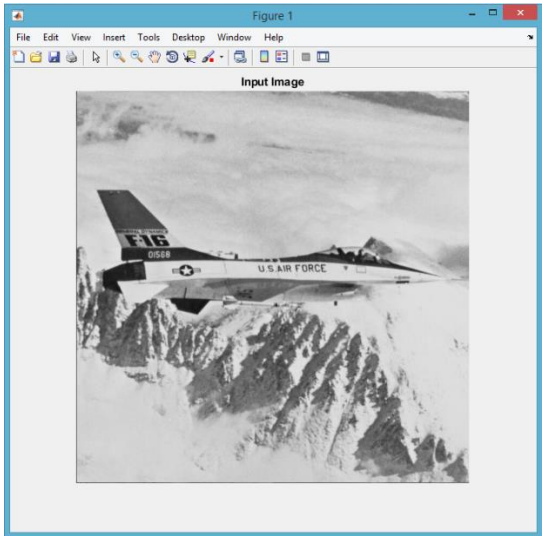

Fig.4: Input Image

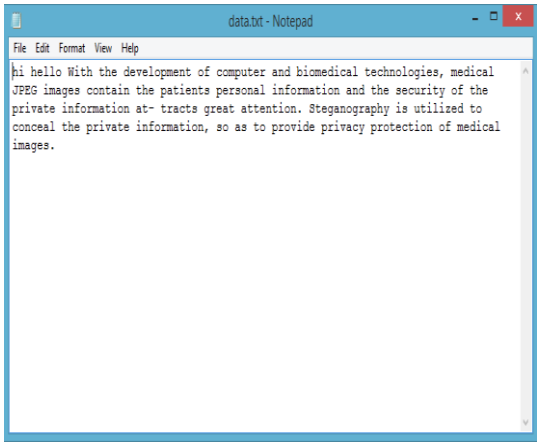

Fig.5: Hided data

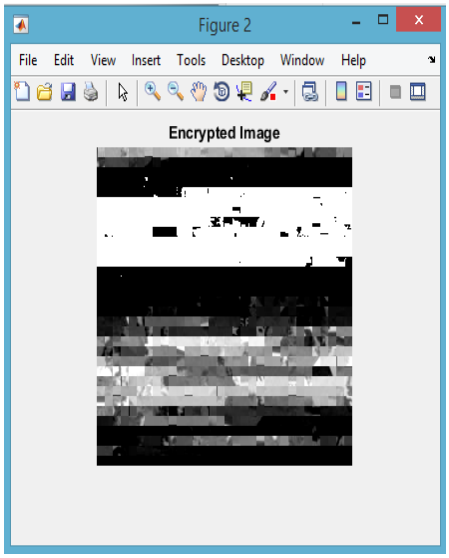

Fig.6: Encrypted image

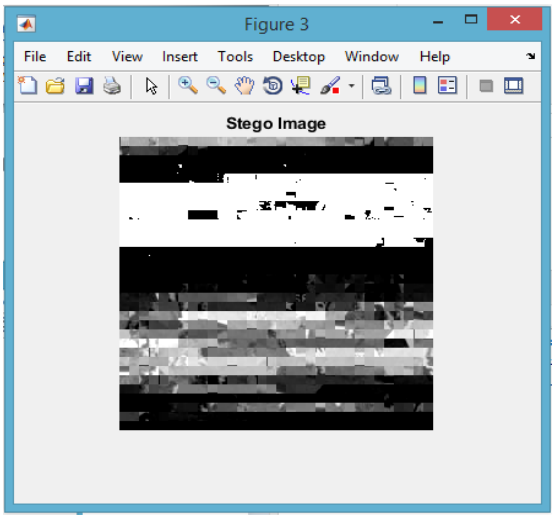

Fig.7: Stego Image

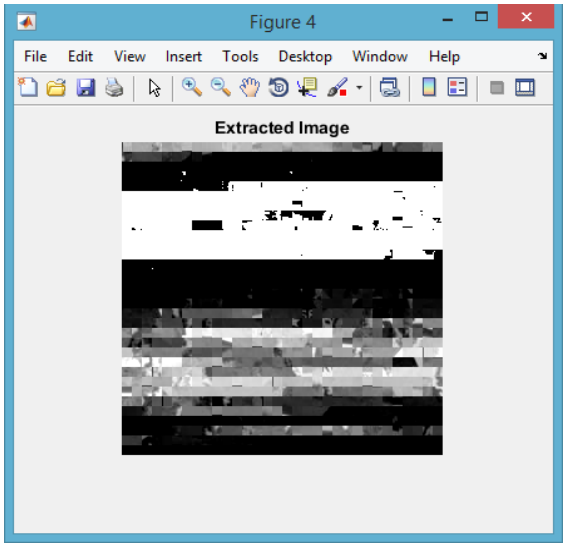

Fig.8: Extracted Image

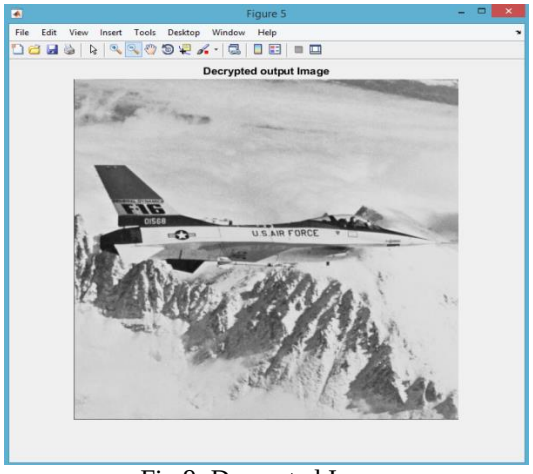

Fig.9: Decrypted Image

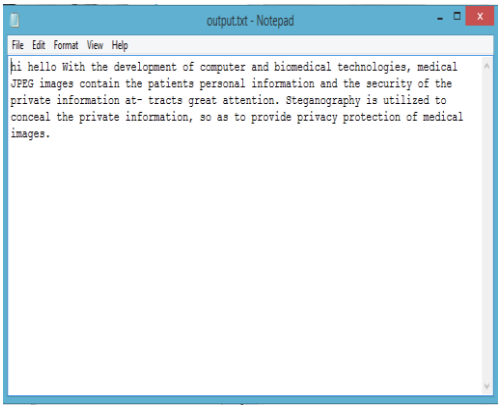

Fig.10: Decrypted Data

VI. CONCLUSION

This paper proposes a separable reversible data hiding scheme for the encrypted JPEG bitstream. A JPEG encryption and decryption algorithm is developed to hide the content of an original image. When the server receives the enciphered

bitstream, a data hider can embed additional messages into the encrypted copy by compressing the padding bits of the bitstream. With an iterative recovery method based on blocking artifacts, the recipient can losslessly recover the original bitstream. The proposed method provides larger embedding capacity than the previous approach. It is separable because anyone who has the embedding key can extract the additional message from the marked encrypted bitstream without revealing the original content of the JPEG image.

The proposed method also offers better security than the previous work. A new JPEG bitstream corresponding to a smaller sized image is constructed. Therefore, information leakage of the original content, *e.g.*, contour of an image, can be avoided. The procedure is realized by rearranging some entropy-coded segments to generate the padding bits. These bits are embedded into the reserved segments labeled by APPn in the JPEG header. The encrypted bitstream can still be decoded by the commonly-used decoders, *e.g.*, the decoder incorporated in the Windows operating system. Meanwhile, the amount of data of the bitstream is unchanged.

VII. REFERENCES

[1]. Naskar PK, Chaudhuri A (2014)A secure symmetric image encryptionbased on bit-wise operation. Int J Image Graph Sig Process 2:30–38

[2]. El Abbadi NK, Mohamad A, Abdul-Hameed M (2014) ImageEncryption based on singular value decomposition. J Comput Sci10(7):1222–1230

[3]. Sun Y, Chen L, Xu R, Kong R (2014) An image encryption algorithmutilizing Julia Sets and Hilbert Curves journals.plos.org/plosone/article?id=10.1371/journal.pone.0084655, January 3, 2014

[4]. Ye R, Zeng S, Lun P, Ma J, Lai C (2014) An Image EncryptionScheme Based on Bit Circular Shift and Bi-directional Diffusion.Int J Inf Tech Comput Sci 6(1):82–92

[5]. Naskar PK, Chaudhuri A, Chaudhuri A (2014) A secure symmetricimage encryption based on linear geometry. Appl Innov MobComput (AIMoC):67-74

[6]. Harram IM, Bakura MUM, Gwoma ZM(2014) Simulation of AESbased data encryption in Vb.NET. Int J Recent Dev Eng Technol(IJRDET) 2(4):5–9.

[7]. Chen J-x, Zhu Z-l, Liu Z, Chong F, Zhang L-b, Yu H (2014) Anovel double-image encryption scheme based on cross-image pixelscrambling in gyrator domains. Opt Soc Am 22(6):7349–7361

[8]. Gaddam SVK, Lal M (2011) Development of bio-crypto key fromfingerprints using cancelable templates. Int J Comput Sci Eng(IJCSE) 3(2):775–783.

[9]. Alghamdi AS, Ullah H (2010) A secure iris image encryption techniqueusing bio- chaotic algorithm. (IJCNS) Int J Comput NetwSecur 2(4):78–84.

[10]. Uludag U, Pankanti S, Prabhakar S, Jain AK (2004) Biometriccryptosystems:issues and challenges. Proc IEEE 92(6):948–960

[11]. Teoh ABJ , Ling DNC, Goh A (2004) Biohashing: two factor authenticationfeaturing fingerprint data and tokenized random number.J Pattern Recognition Soc, Elsevier

[12]. Abdullah AS, Hanif U, Maqsood M, Muhammad KK (2009) Biochaotic stream cipher-based iris image encryption, cse, vol. 2, pp. 739-744, International Conference on Computational Science and Engineering, Canada