This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2868192, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A Secure and High-capacity Data-hiding Method Using Compression, Encryption and Optimized Pixel Value Differencing

Awdhesh K. Shukla[1,2], Akanksha Singh[3], Balvinder Singh[1,4], Amod Kumar[1,2]

[1]AcSIR, CSIR-CSIO Campus, Chandigarh, India
[2]CSIR-Central Scientific Instruments Organisation, Chandigarh, India
[3]Meerut Institute of Engineering and Technology, Meerut, India
[4]CSIR-Institute of Microbial Technology, Chandigarh, India

Corresponding author: Awdhesh K. Shukla (e-mail: akshukla@csio.res.in).

ABSTRACT A high capacity data hiding method using lossless compression, Advanced Encryption Standard (AES), modified pixel value differencing (MPVD) and least significant bit (LSB) substitution is presented. Arithmetic coding was applied on secret message for lossless compression, which provided ~22% higher embedding capacity. The compressed secret message is subjected to AES encryption; this provides higher security in the cases of steganalysis attacks. After compression and encryption, LSB substitution and MPVD are applied. In MPVD, adaptive non-overlapping 3x3 pixel blocks or a combination of 3x3 and 2x2 blocks are used in raster fashion. It is experimentally established that with the proposed method, significant enhancement in embedding capacity was achieved and 214132 extra bits than existing methods could be embedded due to the use of arithmetic compression and MPVD. MPVD and arithmetic coding together resulted into 25% enhanced embedding capacity than earlier methods. The proposed method also provides high levels of visual quality with an average of 36.38 dB at 4.00 bpp. The proposed method is also proved to be secure against regular/singular (RS) steganalysis.

INDEX TERMS Advanced Encryption Standard (AES), least significant bit (LSB), modified pixel value differencing (MPVD), PVD, steganography

I. INTRODUCTION

In order to protect and secure the transmission of data over an open channel e.g. internet, an information security system must be in place. Cryptography and information hiding are the two branches of information security system. Cryptography is used for encrypting and decrypting the data into ‘ciphertext’, which is meaningless and hard to understand. In spite of very high levels of security provided by various advanced cryptographic techniques, illegible nature of ‘ciphertext’ easily draws attention of adversaries and, thus, may result in failure of communication.

Information/data hiding is a mechanism which ensures that the presence of the secret data remains undetected [1]. Data hiding can further be sub-divided into digital watermarking and steganography. In digital watermarking, noise tolerant signals such as an audio, image or video etc. are used to covertly hide a kind of signal (watermark) [2-9] which is used to establish the ownership of such signal. Steganography is used for embedding high amount of data in cover files (image, video, audio etc.). Image based covers are the most frequently used covers. Steganography is divided into spatial and frequency domains. Spatial domain involves embedding of secret message by direct modification of intensity of cover image pixels. The transformed domain coefficients are altered to embed secret data in frequency domain methods. Spatial domain methods require less computational complexity and provide higher embedding capacity than frequency domain methods.

Steganographic methods, devised to fulfill the requirements of recovery of original cover image are termed as reversible embedding [10-18]. Reversible embedding is required for applications like military, medical and legal etc. Generally, during embedding, the LSBs of cover image are overwritten by secret message bits, the original bits are lost and cannot be recovered, thus named as irreversible data embedding [19-26]. Both spatial and frequency domains have been used for reversible and irreversible embedding methods. Irreversible embedding provides higher embedding capacity than reversible embedding.

The performance evaluation of image steganography methods is based on the parameters like: hiding capacity, visual quality/imperceptibility and security/un-detectability. However, these evaluating parameters produce opposite effects with each other e.g. the steganographic methods designed to achieve higher hiding capacity result in visual distortions to the steganographed images and reduced security. Thus, proper corrective/balancing measures are required to make balance between these parameters. Qualities like high embedding capacity, un-detectability and satisfactory visual quality are required for real applications.

The simplest and most popular image steganographic technique is the LSB substitution [20]. It involves the embedding of messages into cover image by directly replacing the LSBs. The hiding capacity can be as high as 4 LSBs per pixel. A common weakness of LSB embedding is that sample value changes asymmetrically. Through LSB embedding, visual quality may decrease and become sensitive to steganalysis [27] attacks.

The pixel value differencing (PVD)[19] was introduced by Wu and Tsai for improving imperceptibility in stego images. In PVD method, data embedding is done by readjusting the difference between two pixels. The PVD based methods are vulnerable to histogram analysis and provide low embedding rates.

Enhanced hiding capacity over PVD [19] based methods was proposed [21] by combining LSB and PVD. The PVD and LSB embedding methods are applied on smooth and edge areas of the cover image respectively, after partitioning the cover image into smooth and edge areas. The hiding capacity is reported to be improved. This approach is susceptible to RS steganalysis [27] detection attacks. Wu et al. method [19] was further improved by Yang et al. [22] by reporting a readjustment strategy for lower level. This resulted in enhanced visual quality and remains secure against RS steganalysis [27]. But this method does not enhance the hiding capacity.

Adaptive steganographic method [23] was proposed for achieving high embedding capacity by integrating LSB and PVD. The cover image is partitioned into 1x3 non-overlapping pixel blocks and second pixel of each block is selected as base pixel. The base pixel is subjected to k-bit LSB and first and third pixels of the block are subjected to PVD. This results in enhanced embedding capacity than that of Yang et al. [22].

Recursive information hiding scheme was proposed by Hussain et al. [28]. The method combines PVD [22] , OPAP [20], PVD Shift along with modification of prediction errors (MPE) [11]. PVD shift was introduced to embed one extra bit in one stego pixel per block. In this method, if secret bit is 1 and LSB of stego pixel is 0 or vice-versa, then 1 is added or subtracted to the stego pixel value to satisfy that secret bit is equal to the modulus of stego-pixel value with 2. But when both the bits i.e. secret bit and LSB of stego pixel are same or equal, such case is not addressed. On implementing the method by combining all the above methods, the average PSNR falls down to ~27dB which is much lower than the other existing methods [19] [22] [23]. On implementing without PVD shift, PSNR goes up to 32dB whereas if it is implemented without MPE and PVD shift, PSNR rises up to 35dB. Thus, the method reported by Hussain et al. [28] is considered with PVD and OPAP only for evaluation in our study. The challenge remains to achieve higher embedding capacity while maintaining image quality.

In this study, we present a spatial domain, comprehensive steganographic method by combining the lossless compression, state of the art encryption, modified pixel value differencing (MPVD) and LSB substitution. Method reported by Khodaei and Faez [23] has been modified by introducing 3x3 pixel block (8-neighbour pixel block) to achieve higher embedding rate and better correlated pixels to provide higher visual quality. The motivation behind proposed method has been to enhance embedding capacity substantially without compromising quality aspects.

The structure of the paper is as follows. LSB and PVD based related steganographic approaches are discussed in section I. In Section II, proposed comprehensive steganographic scheme is described with compression, encryption, message embedding and extraction. Experimental results are presented and discussed in Section III. In Section IV, conclusions are presented.

II. PROPOSED METHOD

The proposed method involves combination of Arithmetic Coding [29] to provide high embedding capacity, encryption through AES, and modified embedding scheme which includes pixel optimization. The section is divided into: i) compression through index based coding, ii) Encryption and decryption using AES, iii) embedding, and iv) extraction.

A. COMPRESSION THROUGH ARITHMETIC CODING

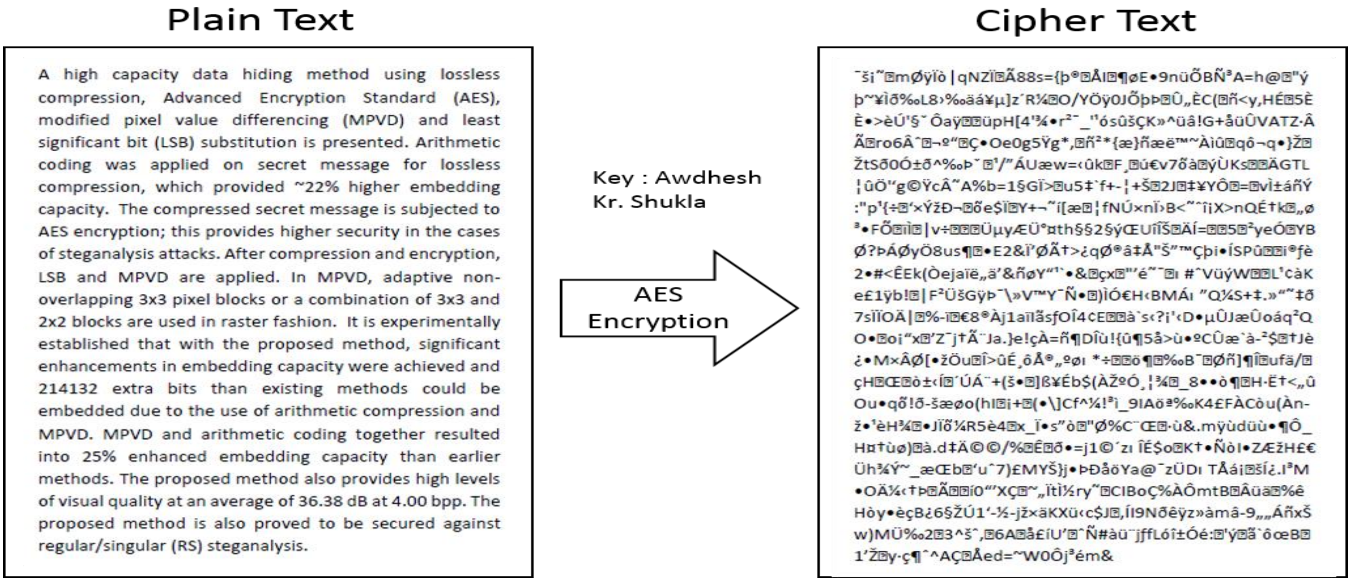
The compression is achieved through Arithmetic Coding [29]. In arithmetic coding, fewer bits are used for frequently used characters and infrequently used characters are stored with more number of bits, thus, resulting in fewer bits required for total encoding.

A pseudo random number generator (PRNG), deployed in MATLAB® was used for generating test messages of various capacities. Arithmetic coding and extraction codes were also deployed in MATLAB®. On an average, about 22% higher embedding capacity was achieved through Arithmetic Coding.

After compression, output of this step would be a bit stream of compressed secret message. Extra ‘0’ bits are added at the end, if required, to make the sequence divisible by 8. This bit stream is used as input for AES based encryption.

B. ENCRYPTION AND DECRYPTION USING AES

Compressed text file (as per section II.A above) is taken as input for encryption using AES [30]. AES encryption ensures that the message does not get disclosed even if the existence



A high capacity data hiding method using lossless compression, Advanced Encryption Standard (AES), modified pixel value differencing (MPVD) and least significant bit (LSB) substitution is presented. Arithmetic coding was applied on secret message for lossless compression, which provided ~22% higher embedding capacity. The compressed secret message is subjected to AES encryption; this provides higher security in the cases of steganalysis attacks. After compression and encryption, LSB and MPVD are applied. In MPVD, adaptive non-overlapping 3x3 pixel blocks or a combination of 3x3 and 2x2 blocks are used in raster fashion. It is experimentally established that with the proposed method, significant enhancements in embedding capacity were achieved and 214132 extra bits than existing methods could be embedded due to the use of arithmetic compression and MPVD. MPVD and arithmetic coding together resulted into 25% enhanced embedding capacity than earlier methods. The proposed method also provides high levels of visual quality at an average of 36.38 dB at 4.00 bpp. The proposed method is also proved to be secured against regular/singular (RS) steganalysis.

```
¯ši˝⍰mØÿÏò|qNZÏ⍰Ã88s={þ®⍰ÅI⍰¶øE•9nüŐBÑ³A=h@⍰"ý
þ~¥Ìð‰L8›‰ää¥µ]z´R¼⍰O/YÖÿ0JŐþÞ⍰Û„ÈC(⍰ñ<y,HÉ⍰5È
È•>èÚ'§˘Ôaÿ⍰⍰üpH[4'¾•r²¯_''ósûšÇK»^üâ!G+åüÛVATZ·Â
Ã⍰ro6Â˜⍰¬º"⍰Ç•Oe0g5Ÿg*,⍰ñ²*{æ}ñæë™~ÀìûÛ⍰qô¬q•}Ž⍰
ŽtSð0Ó±ð^‰Þ˘⍰¹/"ÁUæw=‹ûk⍰F¸⍰ú€v7őà⍰ýÙKs⍰⍰ÄGTL
¦ûÖ"g©ŸcÂ˝A%b=1§GÏ>⍰u5‡`f+-¦+Š⍰2J⍰‡¥YÔ⍰=⍰vÌ±áñÝ
:"p¹{÷⍰'×ÝžÐ¬⍰őe$Ï⍰Y+¬˝í[æ⍰¦fNÚ×nÏ›B<˝ˆî¡X>nQÉ†k⍰„ø
³•FŐ⍰ìÌ⍰|v÷⍰⍰⍰ÜµyÆÜ°¤th§§2§ýŒUîÏŠ⍰ÄÍ=⍰⍰5⍰²yeÓ⍰YB
Ø?ÞÁØyÖ8us¶⍰•E2&Ï'ØÃ†>¿qØ®â‡Å"Š"™Çþi•ÍSPû⍰⍰i®ƒè
2•#<ÊEk(Òejaïë„ä'&ñøY"¹`•&⍰çx⍰"'é˝¯⍰ı #ˆVüýW⍰⍰L¹¢àK
e£1ÿb!⍰|F²ÜšGÿÞ¯\»V™Y¯Ñ•⍰)ÌÓ€H‹BMÁı "Q¼S+‡.»"˝‡ð
7sÏÏOÄ|⍰%-ï⍰€8®Àj1aïläsƒOÎ4¢E⍰⍰à`s‹?¡'‹D•µÛJæÛoáq²Q
O•⍰oi"x⍰'Z¯j†Ã¨Ja.}e!çÀ=ñ¶DÎù!{û¶5å>ù•ºCÛæ`à-²$⍰†Jè
¿•M×ÂØ[•žÖu⍰Î>ûÉ¸ôÅ®„ºøı *÷⍰⍰ö¶⍰‰B¯⍰Øñ]¶Î⍰ufä/⍰
çH⍰Œ⍰ò±‹Í⍰´ÚÁ¨+(š•⍰]ß¥Éb$(ÀŽºÓ¸¦¾⍰_8••ò¶⍰H·Ë†<„û
Ou•qő!ð-šæøo(hI⍰i+⍰(•\]Cf^¼!³ì_9IAöª‰K4£FÀCòu(Àn-
ž•¹èH¾⍰•JÏő¼R5è4⍰x_Ï•s"ò⍰"Ø%C¨Œ⍰·ù&.mÿùdüù•¶Ô_
H¤†ùø)⍰à.d‡Ä©©/%⍰Ê⍰ð•=j1©´zı ÎÉ$o⍰K†•ÑòI•ZÆžH£€
Üh¾Ý~_æŒb⍰'uˆ7)£MYŠ}j•ÞÐåöYa@¯zÜDı TÅái⍰šÍ¿.I³M
•OÄ¼‹†Þ⍰Ã⍰⍰í0"'XÇ⍰~„ÏtÌ½ry˝⍰CIBoÇ%ÀÔmtB⍰Âüä⍰%ê
Hòy•èçB¿6§ŽÚ1'-½-jž×äKXü‹c$J⍰,ÍI9Nðêÿz»àmâ-9„„ÁñxŠ
w)MÜ‰2⍰3^šˆ,⍰6A⍰å£íU'⍰ˆÑ#àü¨jffLóî±Óé:⍰'ý⍰ã`ôœB⍰
1'Ž⍰y·ç¶ˆ^AÇ⍰Åed=˜W0Ôj³ém&
```

Figure 1. Example of AES encryption

of the message is disclosed, thus, ensuring higher security level.

AES uses 128-bit data and variable length (128/192/256-bit) keys. AES is an iterative cipher, the number of rounds in AES vary with the length of the key. Different 128-bit round keys are used in each of these rounds, which are computed out of original AES key. All the computations in AES are performed on bytes rather than bits. 128 bits of a data block is treated as 16 bytes and is arranged as a 4x4 matrix, referred to as state array in AES. An example of AES encryption is shown in Fig.1.

1) ENCRYPTION USING AES

Input: 128-bit data, 128-bit key
Output: Cipher text

Steps involved:

For each round, repeat the following steps:

a. SubBytes()
b. ShiftRows()
c. MixColumns()
d. AddRoundkey()

Step c is not required in the final round.

Step 1: SubBytes (state)

All the input bytes of the block are taken as independent entities and are substituted by their substituent from the S_Box, with single S_Box used for entire data.

Step 2: ShiftRows (state)

Each row of the data matrix is shifted to the left and any elements that 'drop off' are re-inserted from the right side of the row. The number of shifts a row encounters varies for different rows.

Step 3: MixColumns (state)

A mathematical function is used to transform each four-byte column. Four bytes of one column are taken as input by this function and four new output bytes are generated and original column bytes are replaced with these new output bytes. This step is not executed in the last round.

Step 4: AddRoundKey (state, Key[i])

XOR is performed on received round key and state.

2) DECRYPTION USING AES

For each round, with the state and key as inputs (except for last round), following steps are repeated:

Step 1: Inverse_SubBytes()
Step 2: Inverse_ShiftRows()
Step 3: Inverse_MixColumns()
Step 4: Inverse_AddRoundkey()

Step 3 is not performed for last round. All the encryption steps should be performed in reverse to achieve decryption.

C. EMBEDDING PROCEDURE

After compression and encryption of secret message, modified approach of Khodaei and Faez's LSB+PVD method [23] is applied to embed the message in cover image. In the proposed embedding method, the cover image is divided into non-overlapping pixel blocks of 3x3 or 3x3 + 2x2 pixel blocks as per the cardinality of the cover image. That is, if the dimensions of the cover image are not divisible by 3x3 pixel blocks, the remaining pixels are taken 2x2 pixel blocks. The embedding process is presented below:

Step 1: Read the cover image I and divide it into 3x3 non-overlapping blocks. However, if the height and width of the image are not divisible by 3x3 blocks, the image is divided into 3x3 plus 2x2 blocks. For example, a 512x512 pixel image

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2868192, IEEE Access

is partitioned into 28900 blocks of 3x3 pixels and 511 blocks of 2x2 pixels.

Step 2: Read block $\mathrm{B_i}$ and name the pixels as a_1, a_2, a_3, a_4, a_c, a_6, a_7, a_8, a_9 as shown in Fig. 2(a). For 2x2 pixel blocks, the pixels would be named as a_c, a_2, a_3, and a_4, shown in Fig. 2(c).

Step 3: Select a_c as base pixel or reference pixel.

Step 4: For reference pixel, embed 3 bits directly into 3 LSBs of a_c to get a_c' and apply optimization to obtain a_c'' given in Table 1.

Step 5: Calculate the difference $\mathrm{d_i}$ for all pixel values except for a_c.

$$\mathrm{d_i} = |\mathrm{a_c}\text{-}\mathrm{a_i}| \tag{1}$$

Step 6: Compute the ranges $\mathrm{R_i}$ to which $\mathrm{d_i}$ belongs. As stated in the range table, compute $\mathrm{C_i}$ i.e. number of bits to be hidden.

$$\mathrm{t_i} = (\lceil \log_2 |\mathrm{u_i}| \rceil)\text{-}1 \quad \{\mathrm{t_i} = 5, \text{if } \mathrm{t_i} > 5 \tag{2}$$

For lower level t = 3, 4 for R_1, R_2 respectively; t = 5 for higher level i.e. R_3, R_4 and R_5.

Step 7: Read the $\mathrm{C_i}$ bits continuously from S secret message as $\mathrm{S_i}$.

Step 8: Replace $\mathrm{C_i}$ bits of a_i with $\mathrm{S_i}$ to obtain a_i'.

Step 9: Now, apply optimization to all a_i' as given in table to obtain a_i'' as final stego pixel.

Thus, we obtain final stego block shown in Fig. 2(b) and Fig. 2(d).

OPTIMIZATION FOR CENTRAL PIXEL

Step 1: Calculate the new difference $\mathrm{d_c}$ by (3).

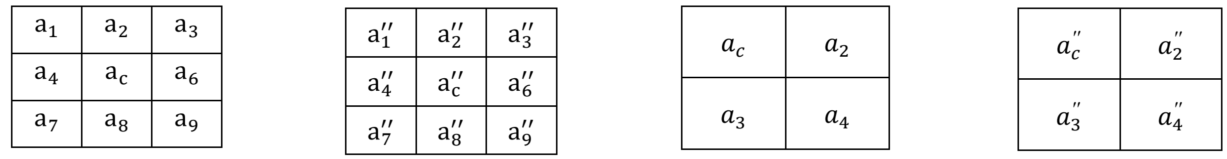
$$d_c = a_c - a_c' \tag{3}$$

Step 2: Perform optimization of central pixel by (4).

$$a_c'' = \begin{cases} a_c' + 8\ , & if\ d_c \geq 5 \\ a_c' - 8\ , & if\ d_c \leq -5 \\ a_c'\ , & otherwise \end{cases} \tag{4}$$

Table 1 : Range Table

	Lower level		Higher level		
Range	R1	R2	R3	R4	R5
Lower-Upper bound	[0-15]	[16-31]	[32-63]	[64-127]	[128-255]
No of bits ($\mathrm{t_i}$)	3 bits	4 bits	5 bits	5 bits	5 bits



(a) Original pixel block (b) Stego-pixel block
3x3 Pixel block

(c) Original pixel block (d) Stego-pixel block
2x2 Pixel block

Figure 2. Pixel Schematic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2868192, IEEE Access

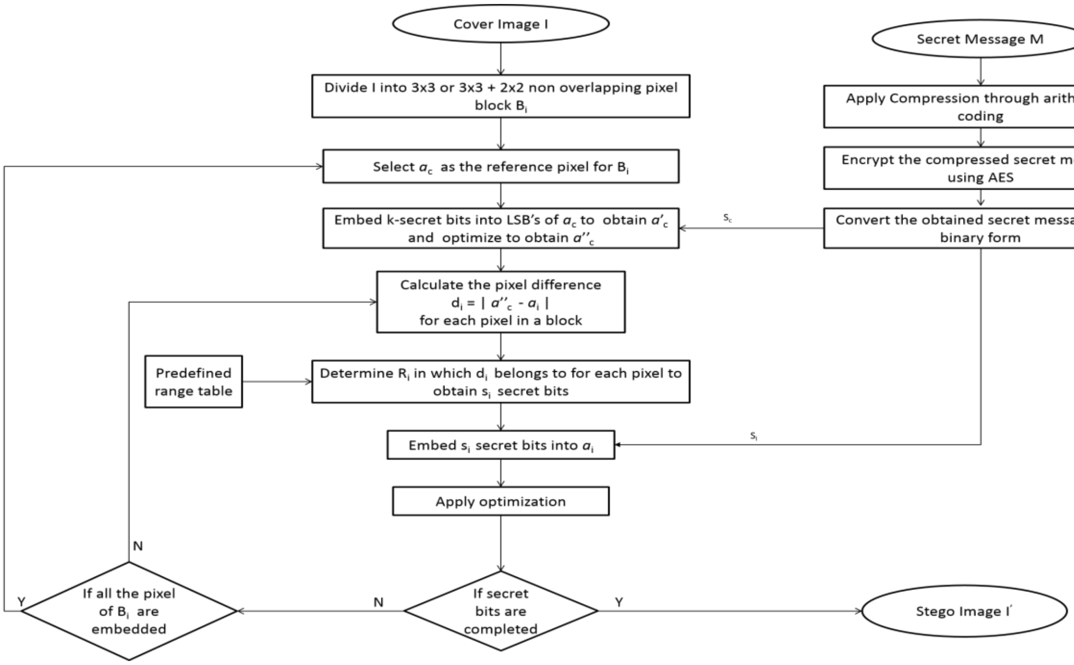

Cover Image I
Secret Message M
Divide I into 3x3 or 3x3 + 2x2 non overlapping pixel block B$_i$
Apply Compression through arithmetic coding
Select a_c as the reference pixel for B$_i$
Encrypt the compressed secret message using AES
Embed k-secret bits into LSB's of a_c to obtain a'_c and optimize to obtain a''_c
S$_c$
Convert the obtained secret message into binary form
Calculate the pixel difference d$_i$ = | a''_c - a_i | for each pixel in a block
Predefined range table
Determine R$_i$ in which d$_i$ belongs to for each pixel to obtain s$_i$ secret bits
S$_i$
Embed s$_i$ secret bits into a_i
Apply optimization
N
Y
If all the pixel of B$_i$ are embedded
N
If secret bits are completed
Y
Stego Image I$'$

Figure 3. Flowchart of proposed method

OPTIMIZATION FOR OTHER BLOCK PIXELS

Step 1: Find $2^{k\,th}$ bit of a_i' and name it as a_k

Step 2: Now, compute

$$a_i'' = \begin{cases} a_i' + 2^k, & if\ a_k = 0\ and\ 0 \le a_i' + 2^k \le 255 \\ a_i' - 2^k, & if\ a_k = 1\ and\ 0 \le a_i' + 2^k \le 255 \end{cases} \tag{5}$$

Step 3: Compute a_i'''

$$a_i''' = \begin{cases} a_i', & if\ \ |a_i - a_i'| < |a_i - a_i''|\ and\ 0 \le a_i' \le 255 \\ a_c'', & otherwise \end{cases} \tag{6}$$

D. EXTRACTION PROCEDURE

Step 1: Read the cover image I and divide it into 3x3 non overlapping blocks. However, if the height and width of the image are not divisible by 3x3 blocks, the image is partitioned into 3x3 plus 2x2 blocks. For example, a 512x512 pixel image is partitioned into 28900 blocks of 3x3 pixels and 511 blocks of 2x2 pixels.

Step 2: Read block B_i and name the pixels as a_1, a_2, a_3, a_4, a_c, a_6, a_7, a_8, a_9 as shown in Fig.2(b). For 2x2 pixel blocks, the pixels would be named as a_c, a_2, a_3, a_4, shown in Fig.2(d).

Step 3: Extract k- secret bits from k- LSB's of a_c' and name it as S_c.

Step 4: Calculate the difference d_i'

$$d_i' = |a_c'' - a_i'| \tag{7}$$

Step 5: Compute the ranges R_i to which d_i' belongs. As stated in range table obtain the C_i'.

Step 6: Then extract C_ith rightmost LSB's of a_i' and name it as S_i for all the pixels of B_i.

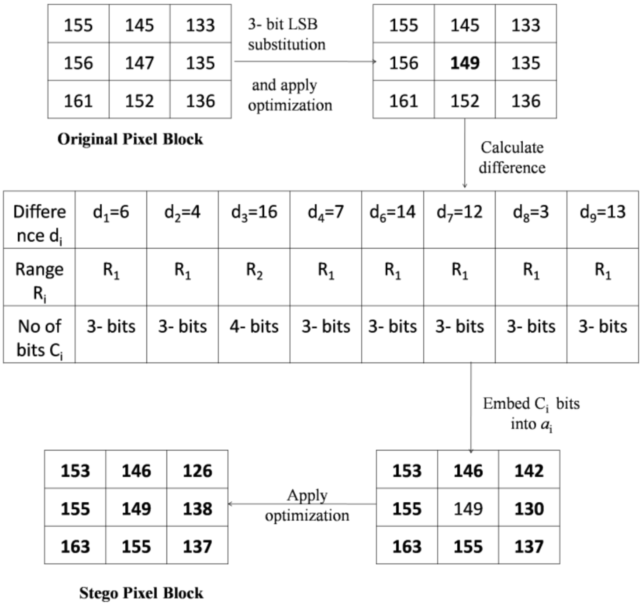
Step 7: Now, concatenate S_c and all S_i to obtain secret message.

E. AN EXAMPLE OF PROPOSED METHOD

An example of proposed method is illustrated in Fig. 3. The $(a_1, a_2, a_3, a_4, a_c, a_6, a_7, a_8, a_9)$ are (155, 145, 133, 156, 147, 135, 161, 152, 136) respectively. The secret bits S = $(101001010111001101001101 1001)_2$. Embed 3-bits i.e. $(101)_2$ directly into $a_c = 147$ by LSB substitution to obtain

$a_c' = 149$ and apply optimization $a_c'' = 149$. Now, compute the difference $(d_1 = 6, d_2 = 4, d_3 = 16, d_4 = 7, d_6 = 14, d_7 = 12, d_8 = 3, d_9 = 13)$. Find the Ri from range table and obtain the number of bits Ci to embed the secret bits. Now, embed secret message bits into a_i and we obtain a_i' as $(a_1' = 153, a_2' = 146, a_3' = 142, a_4' = 155, a_6' = 130,\ a_7' = 163, a_8' = 155, a_9' = 137)$. Then, apply optimization to obtain the final stego pixel as $(a_1'' = 153, a''_2 = 146, a''_3 = 126, a''_4 = 155, a''_6 = 138, a_7'' = 163, a_8'' = 155, a_9'' = 137)$.

To recover the embedded bits, consider $a_c'' = 149$. Then, extract 3-bits from rightmost LSB's of a_c' as $S_1 = (101)_2$. Calculate the difference $d_i' = |a_c'' - a_i''|$. Find Ci from range table to which d_i' belongs. Now, extract Ci bits from rightmost LSB's of a_i'' as S_i. Finally, concatenate S_1 and S_i to obtain the secret message bits S.



155	145	133
156	147	135
161	152	136

155	145	133
156	**149**	135
161	152	136

Difference d_i	d_1=6	d_2=4	d_3=16	d_4=7	d_6=14	d_7=12	d_8=3	d_9=13
Range R_i	R_1	R_1	R_2	R_1	R_1	R_1	R_1	R_1
No of bits C_i	3- bits	3- bits	4- bits	3- bits	3- bits	3- bits	3- bits	3- bits

153	146	142
155	149	130
163	155	137

153	146	126
155	149	138
163	155	137

Figure 4. Example of proposed method

III. EXPERIMENTAL RESULTS

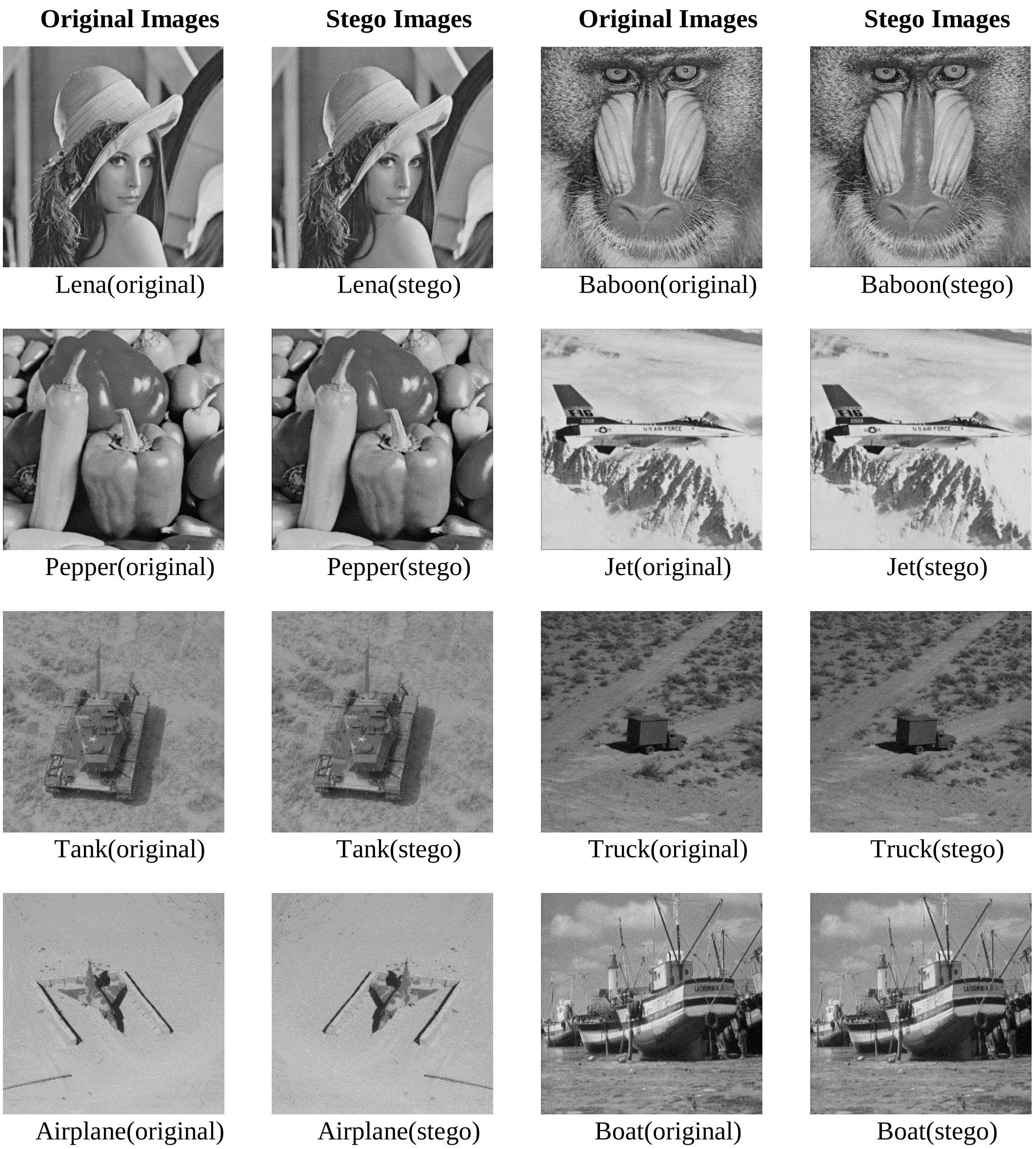
In this section, the implementation of proposed high capacity data hiding method is described. MATLAB® was used to implement and evaluate the proposed method. Test images were taken from standard image database USC-SIPI (University of South Carolina-Signal and Image Processing Institute with URL: http://sipi.usc.edu/database/), http://www.imageprocessingplace.com/root_files_V3/image_databases.htm and other online resources. Standard grayscale images of 512x512 pixels were used for extensive experimentation. Original and stego images are shown in Fig. 5. The secret message was generated by a pseudorandom number generator implemented in MATLAB. Comparison of hiding capacity and visual quality of proposed method with existing methods is shown in Table 1. The performance comparison of our proposed method with existing methods Wu et al. [10], Yang [11], Khodaei [12] and Hussain [28] is done based on hiding capacity, PSNR, bpp, and security by RS-steganalysis. Experimental results are analyzed and presented in following sub-sections.

A. ANALYSIS OF HIDING CAPACITY AND VISUAL QUALITY

Visual quality and hiding capacity of the proposed method with respect to earlier methods is analyzed in this section. The hiding capacity is estimated as the total number of message bits embedded in the stego-image. The embedding rate (bpp) is measured using (8), where M and N are the cardinality of the cover image.

Table 2 : Comparison of hiding capacity (bits) and visual quality of proposed method with existing methods

Image	Wu et al.[21]			Yang et al. [22]			Khodaei et al. [23]			Hussain et al. [28]			Proposed method				
													Un-compressed		Compressed		
	Capacity (bits)	Bits/ pixel (bpp)	PSNR (dB)	Capacity (bits)	Bits/ pixel (bpp)	PSNR (dB)	Capacity (bits)	Bits/ pixel (bpp)	PSNR (dB)	Capacity (bits)	Bits/ pixel (bpp)	PSNR (dB)	Capacity (bits)	Bits/ pixel (bpp)	Capacity (bits)	Bits/ pixel (bpp)	PSNR (dB)
Lena	409811	1.56	41.53	765905	2.92	34.63	806948	3.07	36.20	800673	3.05	35.76	815085	3.11	994403	3.79	37.32
Baboon	457170	1.74	37.44	717849	2.73	30.53	851311	3.25	32.70	825881	3.15	33.57	927623	3.54	1131700	4.32	33.18
Pepper	407257	1.55	41.39	770272	2.94	33.87	803184	3.06	34.03	798636	3.04	35.64	813268	3.10	992187	3.78	37.47
Jet	409819	1.56	40.70	770464	2.94	34.02	805809	3.07	35.69	795304	3.03	35.99	822622	3.14	1003598	3.83	36.83
Tank	459565	1.75	38.13	668986	2.55	36.34	847945	3.23	35.04	865093	3.30	35.84	877591	3.35	1070661	4.08	36.70
Truck	456768	1.74	38.47	669060	2.55	36.66	848025	3.23	35.14	865094	3.30	35.92	878300	3.35	1071526	4.09	36.65
Airplane	454130	1.73	38.55	672230	2.56	36.69	836866	3.19	35.62	865093	3.30	34.07	851034	3.25	1038261	3.96	37.38
Boat	477836	1.82	36.33	661223	2.52	32.76	851309	3.25	31.77	865094	3.30	34.03	894744	3.41	1091587	4.16	35.51
Average	**441544**	**1.68**	**39.06**	**711998**	**2.71**	**34.44**	**831424**	**3.17**	**34.52**	**835108**	**3.18**	**35.10**	**860033**	**3.28**	**1049240**	**4.00**	**36.38**



Lena(original) Lena(stego) Baboon(original) Baboon(stego)

Pepper(original) Pepper(stego) Jet(original) Jet(stego)

Tank(original) Tank(stego) Truck(original) Truck(stego)

Airplane(original) Airplane(stego) Boat(original) Boat(stego)

Figure 5 : Images taken for experiments and corresponding stego images

The visual quality (PSNR) is measured using (9).

$$\text{bpp} = \frac{\text{Embedding capacity}}{\text{MxN}} \qquad (8)$$

$$PSNR = 10log_{10}\frac{255^2}{MSE} \qquad (9)$$

Mean squared error (MSE) is defined in (10), where a_i *and* a_i' are the cover and stego-pixel values respectively.

$$MSE = \sum_{i=1}^{MxN}\frac{(a_i' - a_i)^2}{MxN} \qquad (10)$$

Comparison of hiding capacity and image quality of the proposed method with existing methods Wu et al.[21], Yang et al. [22], Khodaei et al. [23] and Hussain et al. [28] is presented in Table 2. The average hiding capacity achieved from our proposed method is higher than existing methods [10]-[12] and [28]. Our method could embed from 994403 bits (at 3.79 bpp) to 1091587 bits (4.16 bpp) with PSNR ranging from 33.18 dB to 37.47 dB with average embedding capacity of 1040240 bits, and average PSNR of 36.38 dB at 4.09 bpp. Wu et al. [21], Yang et al. [22], Khodaei et al. [23] and Hussain et al. [28] had average payload of 441544 bits, 711998 bits, 831424 bits and 835108 bits respectively. We have achieved the highest embedding capacity of 1049240 bits and the highest embedding rate of 4.00 bpp.

An average PSNR of 36.38 dB was achieved through our proposed method, which is 1.86 dB higher than Khodaei et al.'s method [23], 1.28 dB higher than Hussain et al.'s method

[28], and 1.94 dB higher than Yang et al. [22]. Since, Wu et al.'s [21] method could embed only 42% of proposed method, the PSNR is higher than proposed method. Average hiding capacity of 1049240 bits was achieved through proposed method. Hiding capacity achieved through proposed method is 214132 bits higher than Hussain et al.'s method [28].

B. ANALYSIS OF EMBEDDING RATE AND PSNR

Comparison of embedding rate versus PSNR of the proposed method was carried out with existing methods for the images Lena, Baboon, Pepper and Jet Fig. 6.

For the image Lena, PSNR evaluated for the proposed method and other existing method [8], [11], [12], [28] at 0.3bpp are 49.92, 50.80, 46.77, 48.92 and 46.80 respectively. For the proposed method, PSNR at maximum bpp i.e 3.79 is 37.32dB. The maximum embedding rate and PSNR of previous existing methods are evaluated as 41.53dB at 1.56 bpp [8], 34.63dB at 2.92 bpp [11], 36.20dB at 3.07 bpp[12] and 35.76dB at 3.05 bpp [28]. For the image Baboon, PSNR evaluated for the proposed method and other existing methods [8], [11], [12], [28] at 0.3bpp are 42.78, 42.18, 40.25, 36.74 and 42.64 respectively. For the proposed method, PSNR at maximum bpp i.e 4.32 is 33.18dB. The maximum embedding rate and PSNR of previous existing methods are evaluated as 37.44dB at 1.74 bpp [8], 30.53dB at 2.73 bpp [11], 32.70dB at 3.25 bpp[12] and 33.57dB at 3.15 bpp [28]. For the image Pepper, PSNR evaluated for the proposed method and other existing method [8], [11], [12], [28] at 0.3bpp are 48.26, 48.81, 42.88, 44.57 and 45.83 respectively. For the proposed method, PSNR at maximum bpp i.e 3.78 is 37.47dB. The maximum embedding rate and PSNR of previous existing methods are evaluated as 41.39dB at 1.5 bpp [8], 33.86dB at 2.94 bpp [11], 34.03dB at 3.06 bpp [12] and 35.64dB at 3.04 bpp [28]. For the image Jet, PSNR evaluated for the proposed method and other existing method [8], [11], [12], [28] at 0.3bpp are 49.57, 50.13, 45.46, 48.03 and 46.71 respectively. For the proposed method, PSNR at maximum bpp i.e 3.83 is 36.83dB. The maximum embedding rate and PSNR of previous existing methods are evaluated as 40.70dB at 1.56 bpp [8], 34.02dB at 2.94 bpp [11], 35.69dB at 3.07 bpp [12] and 35.99dB at 3.03 bpp [28].

Therefore, the experimental results show that the proposed method provides the highest PSNR and highest embedding capacity as compared to earlier methods.

C. SECURITY ANALYSIS

The proposed method was tested against well-known RS steganalysis detection attacks. The results of RS steganalysis attack are shown in Fig. 7.

In RS steganalysis, the image is partitioned into groups of n adjacent pixels and "smoothness" of each group is calculated using flipping masks M and –M respectively

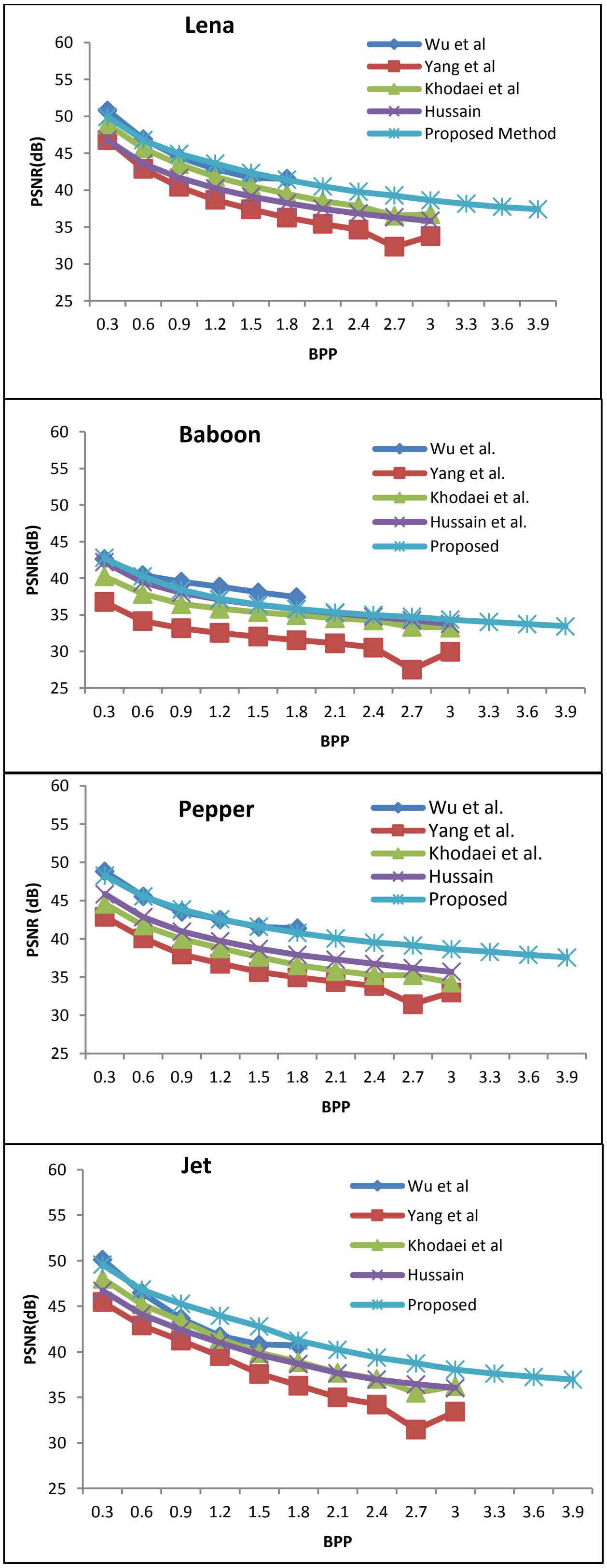
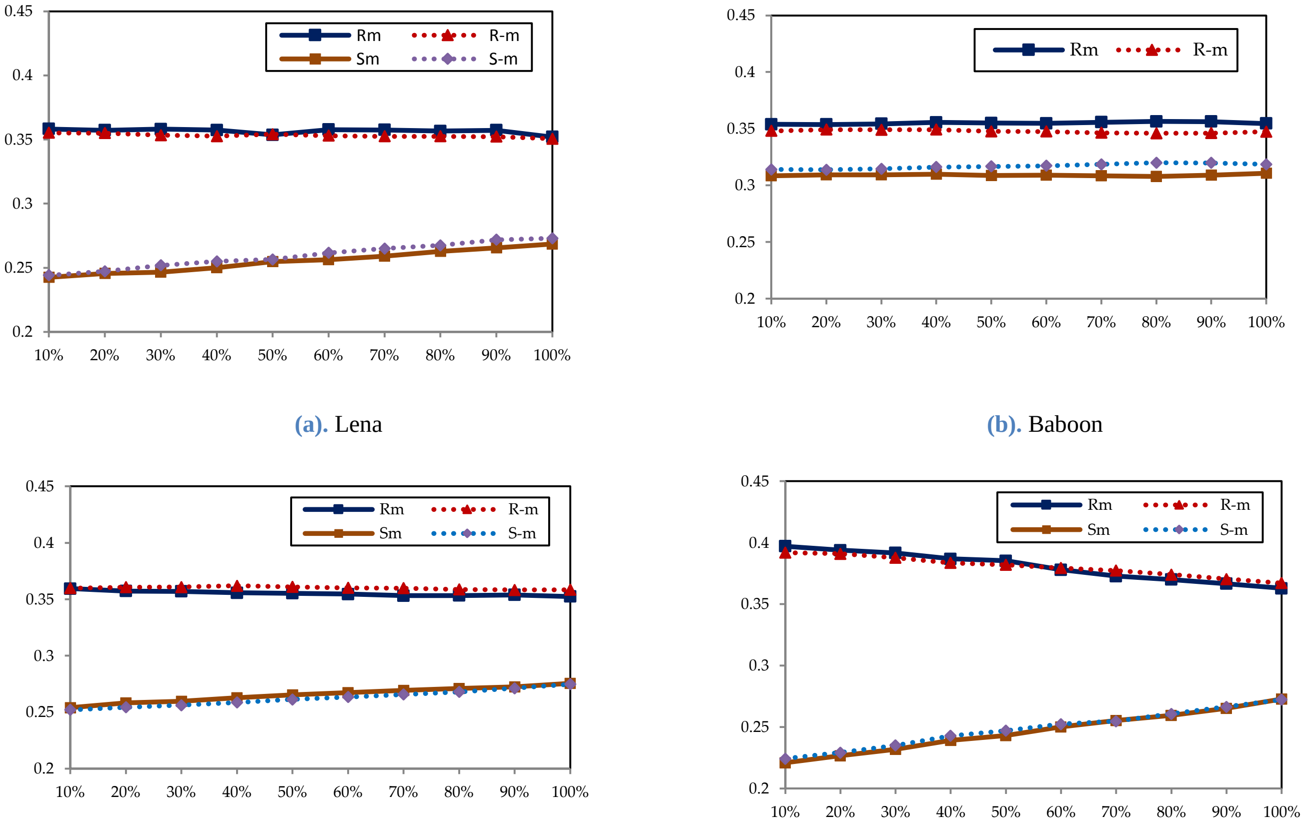


Figure 6. Hiding Capacity versus visual quality

Table 3 : Experimental results of SSIM and standard deviation for average bpp=4

	Lena	Baboon	Pepper	Jet	Tank	Truck	Airplane	Boat
SSIM	0.9403	0.9688	0.9433	0.9400	0.9552	0.9575	0.9422	0.9582
Standard Deviation	0.1465	0.3682	0.099	0.1535	0.2559	0.0067	0.0157	0.4125



(a). Lena

(b). Baboon

(b). Pepper

(d). Jet

Figure 7. Test result of RS steganalysis

corresponding to $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$ and $\begin{bmatrix}0 & -1\\-1 & 0\end{bmatrix}$ of the discrimination function D. The proportion of increase or decrease of D in each block is indicated by parameters RM, SM, R-M, and S-M. An image is considered as normal and not containing any hidden message if these parameters satisfy the condition RM≈ R-M> SM≈ S-M. The image is considered to be having hidden data if RM and S-M decrease or R-M and SM increase.

The test results of RS steganalysis are shown in Figure 7(a-d). The percentage of hidden message is represented by x-axis and the y-axis represents the relative percentage of regular (R_M, R_{-M}) and singular (S_M, S_{-M}) groups with the application of the masks M and -M.

In the Figure 7 (a-d), the differences between regular and singular groups remain close to each other and constant, even in the case when the hiding capacity is increased using proposed method. Thus, the results demonstrate that our proposed method withstands RS steganalysis attacks well.

The algorithm was also verified against structural similarity (SSIM) index and standard deviation. The SSIM index method is used for estimating similarity between two images. Standard deviation (SD) is used to estimate the variation between original image and the stego image. From Table 3, it is observed that SSIM value for all the test images is close to 1 which indicates that the stego images are similar to original images. It is also observed that SD for all the images is negligible which again indicates that the original images and stego images are similar in nature.

IV. CONCLUSION

In this paper, a secure and high capacity image steganography method is proposed in which arithmetic coding is used for high embedding capacity; AES for additional security of hidden contents; MPVD, LSB and pixel optimization for enhanced capacity and improved visual quality. In MPVD, lower embedding rate was opted at higher ranges. An enhanced embedding capacity of ~3% more than earlier methods has been achieved using MPVD. MPVD and arithmetic coding together resulted in 25% higher embedding. Also, the proposed method is secure against RS steganalysis. Thus, proposed scheme promises significant advancement over existing methods.

ACKNOWLEDGMENT

Authors are grateful to Prof. R.K. Sinha, Director, CSIR-CSIO, Chandigarh for allowing to carry out this study and providing necessary facilities.

REFERENCES

[1] A. Cheddad, et al., "Digital image steganography: Survey and analysis of current methods," Signal processing, vol. 90, pp. 727-752, 2010.

[2] S. Atawneh, et al., "Steganography in digital images: Common approaches and tools," IETE Technical Review, vol. 30, pp. 344-358, 2013.

[3] A. K. Singh, et al., Medical image watermarking: techniques and applications: Springer, 2017.

[4] C. Kumar, et al., "Improved wavelet-based image watermarking through SPIHT," Multimedia Tools and Applications, pp. 1-14, 2018.

[5] D. Chauhan, et al., "Combining Mexican hat wavelet and spread spectrum for adaptive watermarking and its statistical detection using medical images," Multimedia Tools and Applications, pp. 1-15, 2017.

[6] R. Srivastava, et al., "Computationally efficient joint imperceptible image watermarking and JPEG compression: a green computing approach," Multimedia Tools and Applications, pp. 1-13, 2017.

[7] C. Kumar, et al., "A recent survey on image watermarking techniques and its application in e-governance," Multimedia Tools and Applications, vol. 77, pp. 3597-3622, 2018.

[8] A. K. Singh, et al., "Hybrid technique for robust and imperceptible multiple watermarking using medical images," Multimedia Tools and Applications, vol. 75, pp. 8381-8401, 2016.

[9] S. Thakur, et al., "Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications," Multimedia Tools and Applications, pp. 1-14, 2018.

[10] J. Tian, "Reversible data embedding using a difference expansion," IEEE transactions on circuits and systems for video technology, vol. 13, pp. 890-896, 2003.

[11] W. Hong, et al., "Reversible data hiding for high quality images using modification of prediction errors," Journal of Systems and Software, vol. 82, pp. 1833-1842, 2009.

[12] Z. Zhao, et al., "Reversible data hiding based on multilevel histogram modification and sequential recovery," AEU-International Journal of Electronics and Communications, vol. 65, pp. 814-826, 2011.

[13] C. Qin, et al., "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," IEEE transactions on circuits and systems for video technology, vol. 23, pp. 1109-1118, 2013.

[14] C. Qin, et al., "Reversible data hiding scheme based on exploiting modification direction with two steganographic images," Multimedia Tools and Applications, vol. 74, pp. 5861-5872, 2015.

[15] B. Jana, et al., "Dual-Image Based Reversible Data Hiding Scheme Using Pixel Value Difference Expansion," IJ Network Security, vol. 18, pp. 633-643, 2016.

[16] P. Pal, et al., "Reversible watermarking scheme using PVD-DE," in International Conference on Computational Intelligence, Communications, and Business Analytics, 2017, pp. 511-524.

[17] A. Banerjee and B. Jana, "High-Capacity Reversible Data Hiding Scheme Using Dual Color Image Through (7, 4) Hamming Code," in Communication, Devices, and Computing, ed: Springer, 2017, pp. 127-139.

[18] B. Jana, "High payload reversible data hiding scheme using weighted matrix," Optik-International Journal for Light and Electron Optics, vol. 127, pp. 3347-3358, 2016.

[19] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," Pattern Recognition Letters, vol. 24, pp. 1613-1626, 2003.

[20] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple LSB substitution," Pattern recognition, vol. 37, pp. 469-474, 2004.

[21] H.-C. Wu, et al., "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," IEE Proceedings-Vision, Image and Signal Processing, vol. 152, pp. 611-615, 2005.

[22] C.-H. Yang, et al., "Varied PVD+ LSB evading detection programs to spatial domain in data embedding systems," Journal of Systems and Software, vol. 83, pp. 1635-1643, 2010.

[23] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," IET Image processing, vol. 6, pp. 677-686, 2012.

[24] Y.-P. Lee, et al., "High-payload image hiding with quality recovery using tri-way pixel-value differencing," Information Sciences, vol. 191, pp. 214-225, 2012.

[25] C. Balasubramanian, et al., "High payload image steganography with reduced distortion using octonary pixel pairing scheme," Multimedia Tools and Applications, vol. 73, pp. 2223-2245, 2014.

[26] S.-Y. Shen and L.-H. Huang, "A data hiding scheme using pixel value differencing and improving exploiting modification directions," Computers & Security, vol. 48, pp. 131-141, 2015.

[27] J. Fridrich, et al., "Detecting LSB steganography in color, and gray-scale images," IEEE multimedia, vol. 8, pp. 22-28, 2001.

[28] M. Hussain, et al., "Recursive information hiding scheme through LSB, PVD shift, and MPE," IETE Technical Review, vol. 35, pp. 53-63, 2018.

[29] I. H. Witten, et al., "Arithmetic coding for data compression," Communications of the ACM, vol. 30, pp. 520-540, 1987.

[30] M. J. Dworkin, et al., "Advanced Encryption Standard (AES)," 2001.