A High End Capacity in Digital Image Steganography: Empowering Security by Mottling through Morphing

Sanjay Bajpai, Dr. Kanak Saxena

Abstract— One of the many techniques to provide security during data communication is digital image steganography. Embedding capacity and distortion are the two main factors that play a vital role while hiding a message in an image. A balance has to be maintained between the two so that neither does affect the other. Proposed paper discusses the techniques which comprises of two phases that leads to formulation of very secure platform with high embedding capacity. First phase mottles the arbitrary image by morphing through two parameters – text or/and image to obtain the cover image which depends upon the features of the chosen image. Second phase embeds the text message of sufficient large length incorporating multi-keys and compound operators in the algorithm to cater robust security. A bunch of images having almost the same texture are transmitted over the network to make the extraction process further complex. Our experimental results prove the horizon touching embedding capacity with extra security layer.

Index Terms— Distortion, Embedding capacity, Extraction, Morphing, Mottled-image, Multi-keys, Water-marking.

1 INTRODUCTION

DATA Security is one of the most important challenges being faced by all kinds of organizations. Many companies have explored that how critical is the information to get success in their business or operations and very few have managed to adopt effective measures to make their information secure, avoiding unauthorized access, preventing intrusions, stopping secret information disclosure, etc. Data storage in digital form has out-weighed the traditional approach because of its simplicity in the operations like storing, maintaining, updating, retrieving and very easy and cheap transmission. Since vice and virtue moves together and it is not an exception. We have to compromise with privacy and security. A lot of effort has been done and is continuous to protect our valuable data and from misuse by unethical persons. Copyrights of our data can be enforced by watermarking [1], [2], [3]. Cryptography is another science of securing data by dismantling it using some algorithms and encryption-keys [4], [5] but it gives a clue to the intruders that something important is hidden and it gives a thinking reason on that way. Steganography is evolving as a robust method for securing data as it does not give a clue to suspect. It is an art and science of hiding data in other innocuous medium [6], [7], [8].

1.1 Brief History and Applications

Historically, it is not new to the present era and has been in use since a long time but in other form. Plan designed by Alice and Bob [9] to escape from the prison through communication which was keenly observed by the warden Wendy. They used innocuous object to accomplish communication without giving any reason to suspect. Similarly, Greek soldiers engraved their message on the wood and pasted it with wax to hide it [10]. In 5th century BC, a slave's head was shaved to hide information and sent when his hair grew back [11], [12], [13]. Its application spectrum has spawned in many areas including confidential communication, secret data storage, protection of data from alteration, Medical Imaging System, biological traits, bank details, important defence information and many more [14], [15], [16], [17].

1.2 Related Terms

It uses *cover image* which is the carrier of hidden message and should have ordinary and innocuous appearance so that it does not arouse any suspicion. *Stego image* is the cover image having the message concealed in it and used for extracting the message at the receiver end [18], [19], [20]. *Stego key* is a secret key that can be either single or composite, which is used to embed message in the cover image and extract message from the stego image. The key can be generated either by performing some calculations based on the texture and size of the cover image or by a pseudo random number. Basic mechanism of steganography is shown in Fig. 1. *Embedding domains* refer to the characteristics of the cover image that will be used to hide the message in it. It can use the spatial domain techniques or transform domain techniques. Former is a case when pixels of the image are modified directly for embedding and in later case images are mathematically transformed before actually embedding the message [21]. It uses the medium like audio, video, text file and digital images to hide information in it in such a way that it looks likes natural.

- *Sanjay Bajpai is research scholar and currently pursuing Ph.D. in computer science from Barkatullah University, Bhopal, INDIA, M: +91 9425642238. E-mail: sbajpai31@gmail.com*
- *Dr. Kanak Saxena is professor and head in Computer & Applications department of Samrat Ashok Techonological Institute, Vidisha, INDIA, E-mail: ks.pub.2011@mail.com*

ISSN 2229-5518

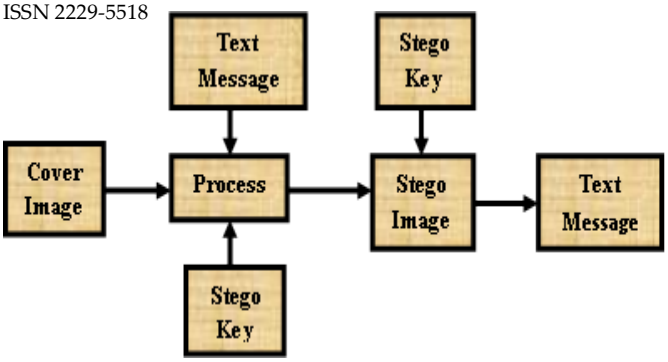


Fig 1. General Skeleton of Steganography

Remainder of the paper is organized as follows. In section 2, we shall briefly discuss about the existing methods. Section 3 describes the proposed method with analysis. Section 4 includes experimental results and comparisons with previous works. Finally, the conclusion, constraints and enhancement is presented in section 5.

2 EXISTING METHODS

Many algorithms have been developed in recent past to embed data in the cover medium. Basic principle for hiding data in the image is to replace the bits of the pixel by the bits of the text message and most common is the LSB substitution and for more detail about the LSB methods, readers are referred to [22], [23], [24], [25]. LSB substitution methods works on the principle of replacing LSBs of the pixel selected either sequentially or randomly. It can be illustrated by the following relation

$$X_i * K + Mi = Yi \quad (1)$$

where X_i is the i^{th} pixel of the cover image, K is the secret key to embed the message, M_i is the i^{th} message bit and Y_i is the i^{th} pixel of the stego image containing the message bit [26], [27].

Optimal Pixel Adjustment Procedure (OPAP) proposed by Chi-Kwon Chan and L. M. Cheng [28] modify the embedded bits to improve the visibility of the stego image. Adjustments are done on the basis of the pixel differences between the cover image and the stego image. Pixel Indicator Technique [29] is employed in the RGB images where two LSBs of one channel mark the existence of message in other two channels. Pixel Value Differencing [30] method emphasize the embedding in those areas of the image which are less susceptible to the human eyes. Since the difference between the pixels in smooth area is less than those of edge area so less distortion occurs in edge area and hence more bits can be embedded in this part.

Arithmetic coding [31] can also be employed for data compression prior to embedding to increase security and embedding capacity. But it fails in case of embedding capacity as its basic principle is based on the difference between the lower bound and upper bound of each new symbol in the text message. Initially, the probability of occurrence of each distinct symbol is calculated and its range is recorded. Upper bound and lower bound of the symbols are calculated by following formulas.

$$UB = LB + (CR \times UB \text{ of new symbol}) \quad (2)$$

$$LB = LB + (CR \times LB \text{ of new symbol}) \quad (3)$$

$$CR = UB - LB \quad (4)$$

where UB, LB and CR stands for upper bound, lower bound and current range respectively. Initially UB is set to 1 and LB is set to 0. As the length of the text message increases, both of its bounds converges to same value and hence cannot be applied. In our previous paper "submitted for publication" [32], we embedded a message of very large length (almost equal to the number of pixels in the image) by compartmentalizing the pixels into its components by employing multi keys without compromising with security and distortion.Mahmud Hasan and et al. [33] devised a process which they named *block processing mechanism* and in that they divided the complete cover image into non-overlapping blocks of dimension 4×4. They identified the Most Frequent Pixels (MFPs) and Second Most Frequent Pixels (SMFPs) of each block and deleted their occurrences. Then they encoded the secret message in these remaining pixels. We will compare their results in section 4. Sanjay Bajpai and Kanak Saxena [34] incorporated Huffman coding algorithm in multi-key LSB substitution method to enhance both security and embedding capacity. Message length increased to about 1.5 times the number of pixels in the image and this number varies to both sides depending upon the domain to which this message belongs.

3 PROPOSED METHOD

3.1 Outlook

All the algorithms that are proposed so far like JSteg, Outguess [35], F5 [36], Singular Value Decomposition [37] which are known as transform domain techniques and spatial domain techniques which are discussed in the previous section, all focus only at point, that is, to embed the message in the image without causing any distortion. All these techniques manipulate the LSBs of the cover image. Number of LSBs generally varies from 1 to 3. Different regions of the image (smooth or edge etc.) and different pattern of pixels selected for embedding plays a vital role in stipulating the security and embedding capacity.

3.2 Inception

Embedding capacity of stated methods vary from a few characters [31] to lakhs of characters [34]. The proposed approach is leaving this figure far behind. An arbitrary image is selected and mottled by masking all of its bits either by the bits of the text or by the bits of another image having different texture. This mottled image does not convey any specific meaning and has the look of modern art and is used as the cover image. All the bits of the cover image can be replaced by the bits of the secret message. Message length approximately equal to three times the number of pixels in the cover image can be used for hiding. This will lead to the generation of another image having almost the same look as the cover image, a new modern art, called the stego image. Recipient is flooded with many such images in a bunch to make the process further complex. Recipient will be informed secretly later on by other means about the cover image, stego image and the keys. Both of these images will be used to extract the hidden message using the selected operation. General skeleton of the proposed process for embedding is shown in Fig. 2 and for extraction in Fig. 3.

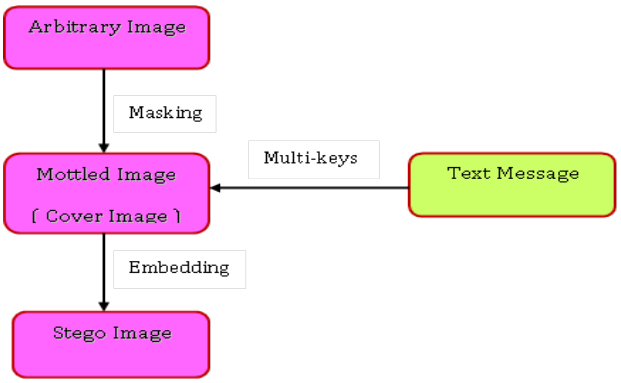


Fig 2. Embedding in the Proposed Process

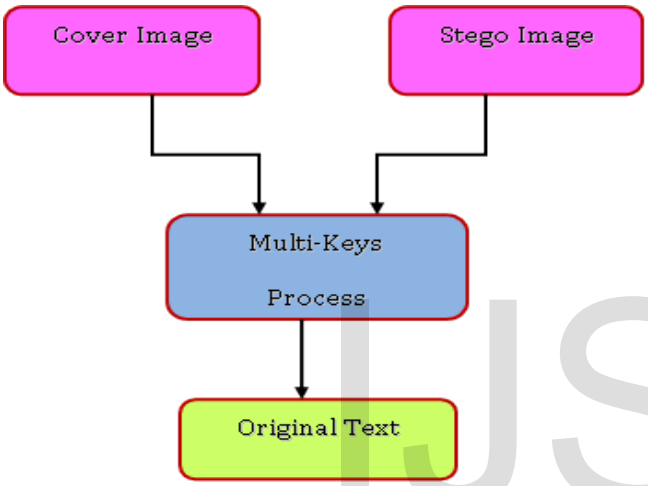


Fig 3. Extraction in the Proposed Process

3.3 Embedding Process

We used bitwise operators to embed data in the cover image. Bits of the pixel components are masked by the bits of the secret message using these operators. Experimental results reveal that it is impossible to extract the original message if bits are masked by either OR or AND operators. Message can be extracted if masking is done by either *Exclusive OR* (XOR) or *Exclusive NOR* (XNOR) operators. To increase the security and make the extraction process tough, we used both the (XOR and XNOR) operators for masking by varying the number of pixels.

Let N be the total number of pixels in the cover image which are equal to width × height (dimensions of the image). M slots (S) of varying capacity K are constructed to accommodate N pixels. S_i, $1 \le i \le M$ (M is the total number of slots) represent the i^{th} slot of capacity K_i and K_i is the number of pixels in the slot S_i which satisfies the relation:

$$N = \sum_{i=1}^{M} K_i \quad (5)$$

Capacity of each slot S_i is calculated by the formula:

$$K_i = C \times K_{i-1} \quad (6)$$

where C is any constant (real number) such that $1 < C < 2$. Initially K_1 is set to a value between 500 and 1000. Number of slots M depends on the constant C and

$$K_m = N - \sum_{i=1}^{M-1} K_i \quad (7)$$

We applied XOR and XNOR operators alternatively on each successive slot to mask the bits of the cover image to increase the robustness of the process. It will be almost impossible to guess that how many bits are masked by XOR operator and how many bits by XNOR since number of bits in each slot are constantly changing and governed by the constant C as mentioned in (6).

3.4 Algorithm-1 (Embedding)

1. Generate the mottled image from a selected image by morphing it.
2. Calculate the number of pixels, N, of the cover image by multiplying its dimensions (width×height).
3. Set the initial value for the capacity K_1 in the range (500 – 1000) and assign a value to the constant C such that $1 < C < 2$ (such as 1.2, 1.45, 1.82 etc.).
4. Generate $K_i = C \times K_{i-1}$; (i = 2 to M-1) and assign K_i pixels to slot S_i until K_i pixels are grabbed from the image and K_M = remaining pixels of the image assigned to S_M.
5. Read $3 \times K_i$ characters from the file and mask bits of these characters into the bits of the pixels of slot S_i (each pixel is fragmented into RGB components) by applying XOR and XNOR bitwise operators alternatively in the specified order.

3.5 Algorithm-2 (Extraction)

1. Receive and decode the initial value of K, value of constant C and order of operations.
2. Decode the names of cover image and stego image.
3. Frame the slots $S^{(1)}$ and $S^{(2)}$ of the cover image and stego image respectively and assign pixels to these slots by the equation (6).
4. Perform XOR and XNOR operations on the bits of the slot $S^{(1)}$ and $S^{(2)}$ applying the order of operations alternatively and store the resultant bits into an temporary array T.
5. Transform the bits of array T into string to obtain the original message.

4 EXPERIMENTAL RESULTS

Our results show that a message of very large length (3 times the number of pixels) is embedded successfully in the image. We tested our results on 20 different images and extraction is also accomplished successfully in all the cases. We achieved a high order of security by employing many parameters at the time of embedding and transmission. We constructed a set of 20 different images for transmission out of which there is only one pair of cover and stego images. To find out this pair, we have to check $^{20}C_2$ combination which itself is a big deal and after this, internal parameters play a vital role and it is almost impossible to determine which of the bits are masked by which operator. Generally, a medium size image has 4 lakh to 6 lakh pixels and each pixel is fragmented into its RGB components.

For testing purpose, we reduced the size of image whose dimension varies from 160×120 pixels to 230×170 pixels. We

embedded the message of length 57600 characters in one of the image whose dimension is 160×120 which is equal to 160×120×3. Same pattern is adopted for images of other dimensions. Some of the images are shown for illustration which comprises of cover images, stego images and fake images. Task is to find a pair of cover and stego images to crack the code if internal parameters for embedding the messages are known. In real application, this number will be more and will increase the complexity. Images in Fig. 4 to Fig. 6 are the arbitrary selected images and images in Fig. 7 to Fig. 11 comprise the cover image, stego image and fake images.

Result Images

Fig 4. Scene-1

Fig 5. Scene-2

Fig 6. Scene-3

Fig 7. Scene-4

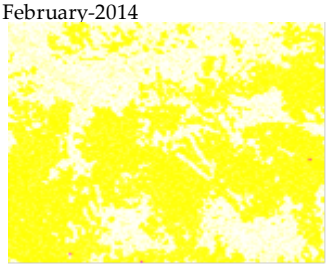


Fig 8. Scene-5

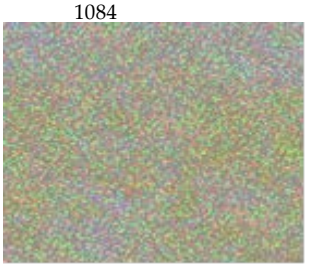


Fig 9. Scene-6

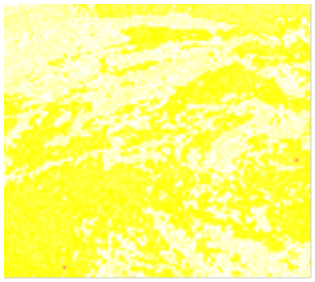


Fig 10. Scene-7

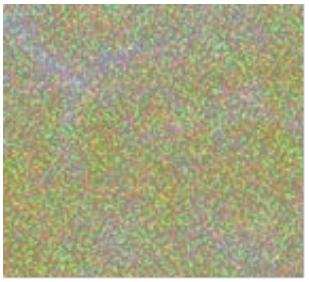


Fig 11. Scene-8

We compared the performance obtained by proposed algorithm with the algorithms proposed by Mahmud Hasan et al. [33], our previous designed algorithms of [32] and [34] which clearly show that our results have been improved as shown in Table 1. Mahmud Hasan et al. [33] utilized only 32 bits out of 128 bits after performing calculations in their *block processing mechanism* and embedded 7 bits of a character. On an average they used 3.50 pixels per character. In [32], we fragmented the pixel into RGB components and stored one character per pixel and in [34] that is *embedding through Huffman coding*, we could store 1.5 characters per pixel. This figure fluctuates to both sides depending upon the type of message as discussed in [34]. Our proposed algorithm is able to embed 3 characters per pixel enhancing both the security and embedding capacity.

Table 1

Comparative Performance Measurements

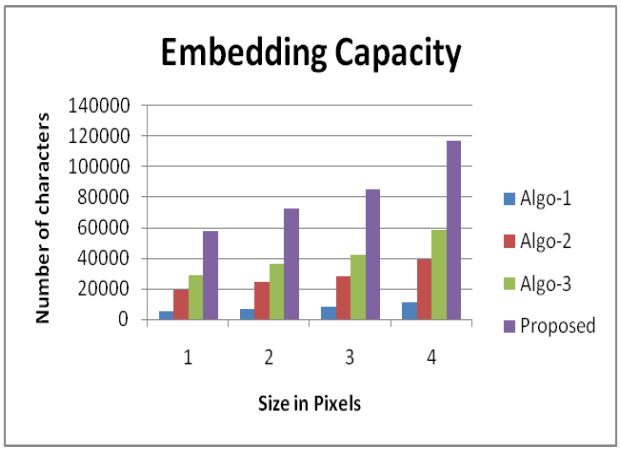
Cover Image			Number of Characters Embedded			
Size in KB	**Type**	**Dimension in pixels**	**Block Processing Mechanism**	**Multi-key Embedding Algorithm**	**Embedding through Huffman Coding**	**Proposed Method**
7	jpg	160×120	5485	19200	28800	57600
15	jpg	180×135	6942	24300	36450	72900
28	jpg	195×145	8078	28275	42412	84825
41	jpg	230×170	11171	39100	58650	117300

5 CONCLUSION

We can say with commitment that we have improved security mechanisms and embedding capacity over the previous algorithms which is clearly visible in Table-1. There are other approaches also proposed by many authors but because of very low embedding capacity something around one or two bits per pixel, we have not included in the comparison process.

Here, we have to take care of that each bit (of all the pixels) of the selected image must be masked to get the cover image

in mottled form. A secret message of length approximately equal to three times the number of pixels in the image can be embedded in the cover image. We observed while obtaining the stego image that there is not much difference between the cover image and stego image and both look of the same pattern. The embedding capacity can also be further enhanced by incorporating any suitable data compression technique.



Graph – Showing the Performance Measurement

REFERENCES

[1] C. C. Chang, K. F. Hwang, M. S. Hwang, "Robust authentication scheme for protecting copyrights of images and graphics", IEE Proceedings on Vision, Image and Signal Processing 149(1) (2002), pp. 43–50.

[2] J. J. K. O' Ruanaidh, W. J. Dowling, F. M. Boland, "Watermarking digital images for copyright protection", IEE Proceedings on Vision, Image and Signal Processing 143(4) (1996), pp. 250–256.

[3] F. Y. Shih, Y.-T. Wu, "Robust watermarking and compression for medical images based on genetic algorithms", Information Sciences 175(3) (2005), pp. 200–216.

[4] H. J. Highland, "Data encryption: a non-mathematical approach", Comput. Secur. 16(1997), pp. 369–386.

[5] William Stallings, "Cryptography and Network Security: Principles and Practices." 3rd edition, 2003, pp. 259-281.

[6] D. Artz, "Digital Steganography: Hiding Data within Data", IEEE Internet Computing Magazine, Vol. 5, Issue 3, August 2002, pp. 75-80.

[7] H. Wang, S. Wang, "Cyber warfare: steganography vs. steganalysis", Commun. ACM 47(10) (2004), pp. 76–82.

[8] Alaa A. Jabbar Altaay, Shahrin bin Sahib, & Mazdak Zamani, "An Introduction to Image Steganography Techniques", *2012 International Conference on Advanced Computer Science Applications and Technologies, IEEE*, 2012, pp. 122-126.

[9] Simmons G. J, "The Prisoners Problem and the Subliminal Channel", Proceedings of crypto '83, Plenum Press, 1983, pp. 51-67.

[10] Zahra Rafinezhad, Karim Faez, "Estimating PQ Algorithm Message Length Using First- and Second-order Statistics", *978-1-4673-2181-5/12/$31.00 ©2012 IEEE*, 2012, pp. 1-5.

[11] N. F. Johnson, S. Jajodia, "Exploring steganography: seeing the unseen", IEEE Computer 31(2) (1998), pp. 26–34.

[12] N. Provos, P. Honeyman, "Hide and seek: an introduction to steganography", IEEE Security and Privacy 1(3) (2003), pp. 32–44.

[13] S. Changder, N. C. Debnath, & D. Ghosh, "A Greedy Approach to Text Steganography using Properties of Sentences", *Eighth International Conference on Information Technology: New Generations, IEEE*, 2011, pp. 30-35.

[14] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. J. Paiz, S. Pogreb, "Applications for data hiding", IBM Systems Journal 39 (3&4) (2000), pp. 547–568.

[15] S. Miaou, C. Hsu, Y. Tsai, H. Chao, "A secure data hiding technique with heterogeneous data-combining capability for electronic patient records", in: Proceedings of the IEEE 22nd Annual EMBS International Conference, Chicago, USA, July 23–28, 2000, pp.280–283.

[16] B. Dunbar, "A detailed look at steganographic techniques and their use in an open-systems environment", Sans Info Sec Reading Room, 2002, [Online] Available: http://www.sans.org/rr/whitepapers/covert/677.phpi.

[17] Abbas Cheddad, Joan Condell, Kevin Curran, & Paul Mc Kevitt, "Digital image steganography: Survey and analysis of current methods", Signal processing 90 (2010) Elsevier, 18 August, 2009, pp. 727-752.

[18] Jamil, T., "Steganography: The art of hiding information is plain sight", IEEE Potentials, 1999.

[19] C. Munuera, "Steganography and error-correcting codes", Signal Processing Elsevier 87 (2007), pp. 1528–1533.

[20] Sanjay Bajpai, & Kanak Saxena, "Techniques of Steganography for Securing Information: A Survey", International Journal on Emerging Technologies 3(1), ISSN No. (Print): 0975-8364, 15 April, 2012, pp. 48-54.

[21] Ratankirti Roy, Suvamoy Changder, Anirban Sarkar, & Narayan C Debnath, "Evaluating Image Steganography Techniques: Future Research Challenges", 978-1-4673-2088-7/13/$31.00 ©2013 IEEE, 2013, pp. 309-314.

[22] C. Kurak, J. McHugh, "A cautionary note on image downgrading", in: Proceedings of the IEEE 8th Annual Computer Security Applications Conference, 30 November – 4 December, 1992, pp.153–159.

[23] P. Alvarez, "Using extended file information (EXIF) file headers in digital evidence analysis", International Journal of Digital Evidence, Economic Crime Institute (ECI) 2(3) (2004), pp. 1–5.

[24] D. Neeta, K. Snehal, D. Jacobs, "Implementation of LSB Steganography and Its Evaluation for Various Bits", IEEE International Conference on Digital Information Management, June 2007, pp. 173-178.

[25] S. M. M. Karim, M. S. Rahman, and M.I. Hossain, "A New Approach for LSB Based Image Steganography using Secret Key", Appeared in 14th International Conference on Computer and Information Technology (ICCIT), March 2012, pp. 286-291.

[26] Bin Li, Junhui He, Jiwu Huang, & Yun Qing Shi, "A Survey on Image Steganography and Steganalysis", Journal of Information Hiding and Multimedia Signal Processing, Vol.2, Issue 2, April 2011, pp. 142-172.

[27] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding", IBM System Journal, vol. 35, no. 3, 1996, pp. 313-336.

[28] Chi-Kwong Chan, & L.M. Cheng, "Improved hiding data in images by optimal moderately significant-bit replacement", IEE Electron Lett. 37 (16), 2001, pp. 1017-1018.

[29] Adnan Gutub and et al., "Pixel indicator high capacity technique for RGB image based Steganography", *WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications*, *University of Sharjah*, *Sharjah*, *U.A.E.* 18 – 20 March 2008.

[30] D. C. Wu, & W. H. Tsai, "A steganographic method for images by pixel-value differencing", *Pattern Recognition Letters*, *vol. 24*, *no. 9-10*, 2003, pp. 1613-1626.

[31] Paul G. Howard, & Jeffrey Scott Vitter, "Practical Implementations of Arithmetic Coding", *International Conferences on Advances in Communication and Control (COMCON 3)*, *British Columbia*, *Canada*, 16-18 October, 1991, pp. 1-34.

[32] Sanjay Bajpai, & Kanak Saxena, **"**Enhancing Embedding Capacity by Compartmentalizing Pixels using LSB Techniques in Steganography", *International Journal of Computers and Applications*, ACTA Press, paper-id 202-3762, submitted – 03 April, 2013.

[33] Mahmud Hasan, Kamruddin Md. Nur, & Tanzeem Bin Noor, "A Novel Compressed Domain Technique of Reversible Steganography", *Internation-*

al Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X, 03-March, 2012, pp. 1-6.

[34] Sanjay Bajpai, & Kanak Saxena, **"**Enhancement of Security and Embedding Capacity through Huffman Coding in Steganography", *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 4, ISSN 2278-6856*, July – August, 2013, pp. 73-78.

[35] N. Provos, "Defending against statistical steganalysis", *In: Proceedings of tenth USENIX security symposium'01*, 2001, pp. 323-335.

[36] Andreas Westfeld, "F5—A Steganographic Algorithm, High Capacity Despite Better Steganalysis", *Springer-Verlag Berlin Heidelberg*, 2001, pp. 289-302.

[37] K S. Babu and et al., "Robust and High Capacity Image Steganography using SVD", *IET-UK International Conference on Information and Communication Technology in Electrical Sciences*, 2007, pp. 718-723.