

## **Dynamic least significant bit technique for video steganography**

**تقنية البت الاقل اهمية الديناميكية للاخفاء الفديوي**

Wafaa hasan alwan

Computer image processing

[Wafsam2005@yahoo.com](mailto:Wafsam2005@yahoo.com)

Law college, Karbala university

### **ABSTRACT**

Video Steganography deals with hiding secret data or information within a video. In this paper, a modified or dynamic based least significant bit (LSB) technique has been proposed to hide movie in movie. A dynamic LSB is a spatial domain technique where the secret information such as images, or video is embedded in the LSB of the cover movie (i.e frames) by selecting number of bits to be embedded. The idea of the proposed method is take away the least significant pixels from one image (frame) which is in cover movie and uses them to store most significant pixels of second image (frame) which is in hidden movie. The hidden image's (frame's) values are stored in the result frame's least significant bits so they don't add greatly to the resulting combined video (movie). The proposed method is analyzed in terms of both Peak Signal to Noise Ratio (PSNR) compared to the original cover video with stegano video as well as the Mean Square Error (MSE) measured between the original and steganographic files averaged over all video frames.

**KEYWORDS** Cover video, Dynamic LSB, secret video, Steganography, Video Steganography.

### **الخلاصة**

الاخفاء الفديوي يتعامل مع اخفاء البيانات والمعلومات السرية في داخل الفديو . في هذا البحث اقترحت تقنية البت الاقل اهمية الديناميكية لاخفاء ملف فديو في ملف فديو . تقنية البت الاقل اهمية الديناميكية هي تقنية في المجال المكاني (اي تطبق على القيم الفعلية ) , حيث يتم تضمين المعلومات السرية كالصور او الفديو في البت الاقل اهمية في ملف الفديو الغطاء بتحديد عدد البت المستخدمة للتضمين (الاخفاء). الفكرة الاساسية للطريقة المقترحة هي اخذ البت الاقل اهمية من نقط الصورة (الاطار) الموجودة ضمن ملف الغطاء واستخدامها لخرن البت الاكثر اهمية من نقط الصورة الثانية (الاطار) الموجودة ضمن الملف المراد اخفائه . قيم الصور المخفيه تخرن في البت الاقل اهمية الناتج بدون اضافة تأثير كبير على ملف الفديو المجمع الناتج . تحلل الطريقة المقترحة باستخدام مقياس PSNR نسبة الضوضاء الى قمة الاشارة لمقارنة ملف الفديو الاخفاء الناتج مع ملف الغطاء الاصلي , كذلك مقياس MSE لحساب مربع متوسط الخطأ بين الملف الاصلي وملف الاخفاء الناتج لكل الاطارات .

### **1. INTRODUCTION**

Steganography is hiding private or secret data within a carrier in invisible manner. It derives from the Greek word steganos, meaning covered or secret, and graphy (writing or drawing) [1]. The medium where the secret data is hidden is called as cover medium, this can be text, image, video or an audio file. Any stego algorithm removes the redundant bits in the cover media and inserts the secret data into the space. Higher the quality of video or sound more redundant bits are available for hiding. Application of Steganography varies from military, industrial applications to copyright and Intellectual Property Rights (IPR). By using lossless steganography techniques messages can be sent and received securely [2]. Traditionally, steganography was based on hiding secret information in image files. But modern work suggests that there has been growing interest among research fraternity in applying steganographic techniques to video files as well [3], [4]. The

advantage of using video files in hiding information is the added security against the attack of hacker due to the relative complexity of the structure of video compared to image files.

Video based steganographic techniques are broadly classified into temporal domain and spatial domain. In frequency domain, images are transformed to frequency components by using FFT, DCT or DWT and then messages are embedded in some or all of the transformed coefficients.

Embedding may be bit level or in block level. Moreover in spatial domain the bits of the message can be inserted in intensity pixels of the video in LSB positions. The advantage in the method is that the amount of data (payload) that can be embedded is more in LSB techniques. However most of the LSB techniques are prone to attack as described in [5] and [6]. This makes research fraternity interested in designing new methods. Techniques other than LSB substitution also exist in literature and have been discussed in the next section.

In this paper a dynamic based LSB Technique is proposed in spatial domain. An application of the algorithm is illustrated with AVI (Audio Video Interleave) file as a hidden and cover medium. The results obtained are significant and encouraging. Effort has also been taken to study the steganalysis of the proposed scheme.

The rest of the paper is arranged as follows, section 2 does literature survey of the recent steganographic techniques. In section 3 the proposed video steganographic technique has been described with its idea and example . The algorithm is proposed in section 4 with an application of it in AVI carrier file and illustration. Section 5 gives results and performance evaluation with dynamic LSB technique with steganalysis of the technique. Conclusion and future work are presented in Section 6.

## **2. LITERATURE SURVEY**

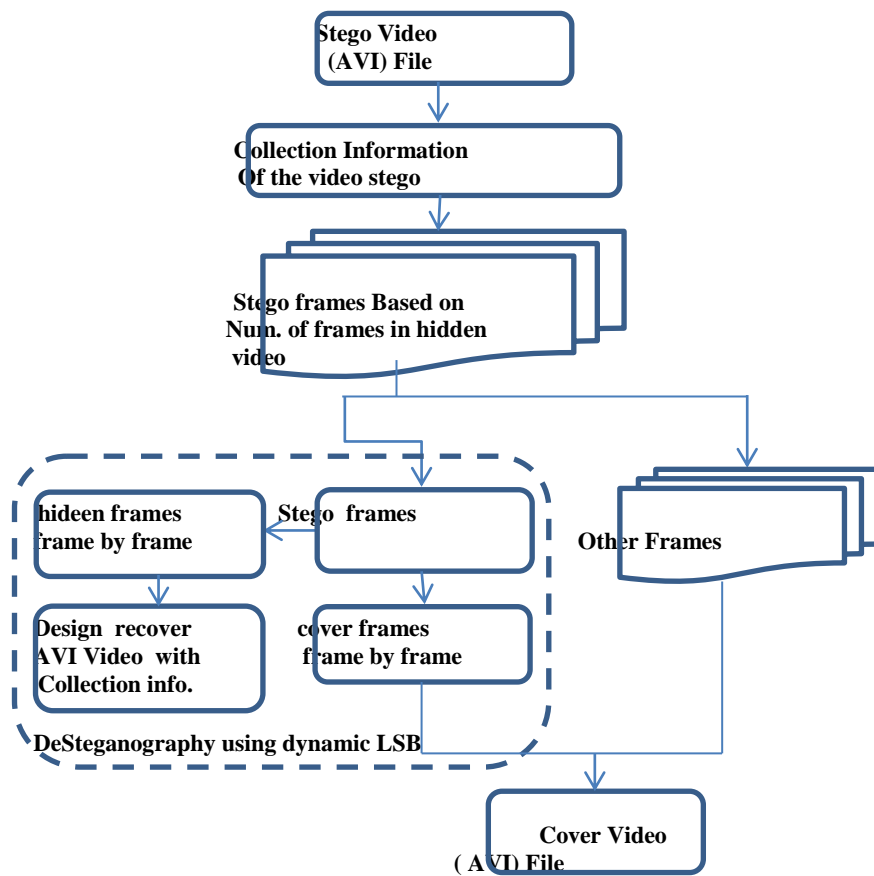
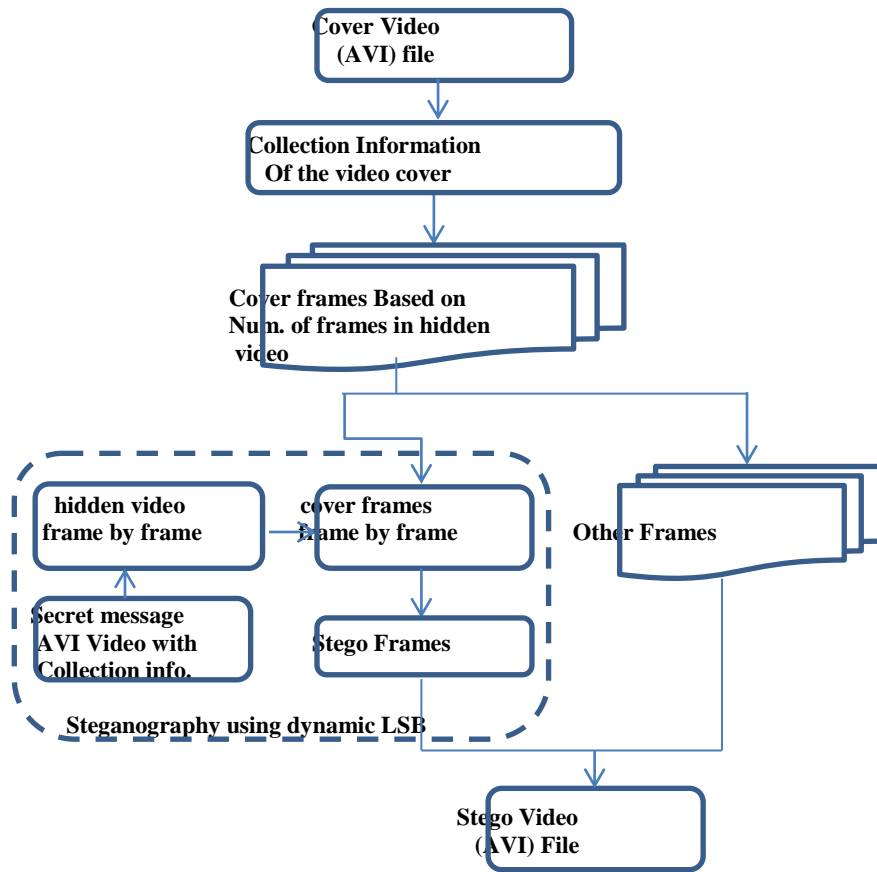
Several steganographic methods have been proposed in literature and most of which are performed in pixel domain. However major contribution is in the domain of Image Steganography. The existing methods are mainly based on LSB where LSBs of the cover file are directly changed with message bits. In [7] Kawaguchi et al proposes Bit Plane Complexity Segmentation (BPCS) method to embed information into the noisy areas of the image. These techniques are not limited to the LSB. Steganography techniques for compressed video stream can be found in [8], [9] and [10]. Tseng and Pan [11] presented a data hiding scheme in 2-color images, it embeds the information in any bit where at least one of the adjacent bits is the same as the original unchanged bit. Whereas in [12] selected LSB steganography algorithm is proposed. Other steganography techniques in uncompressed raw video, is illustrated [13], [14] and [15]. Another video steganography scheme based on motion vectors and linear block codes has been proposed in [16]. Various techniques of LSB exists, where [17] proposes the data is first encrypted using a key and then embedded in the carrier AVI video file in LSB keeping the key of encryption in a separate file called key file.

. Masud et al [18] has proposed a LSB technique for RGB true color image by enhancing the existing LSB substitution techniques to improve the security level of hidden information.

Existing steganographic software, such as Steganos, S-tools and Hide4PGP [19], are based on LSB. Video steganography of late has also gained quite significance for researchers. Other Examples of LSB schemes can be found in [20], [21]. Whereas EzStego developed by Machado. [22] embed information into an image in the GIF format. It sorts the palette to ensure the difference between two adjacent colors is visually indistinguishable. In [23] a robust image steganography technique based on LSB insertion and RSA encryption technique has been used.

## **3. Proposed Video Ssteganography Technique**

The technique is a dynamic based Least Significant Bit (DLSB) technique for Video Steganography has been proposed. The flow diagram of the same is given in Figure1 represent Embedding (steganography) and Extracting (De\_ steganography).



**Figure1: a - Embedding b - Extracting**

A video stream (AVI) consists of collection of frames and the secret data such as text, image or video stream (AVI) consist of collection of frames which is embedded in these frames as payload. The information of the cover video and hidden video (AVI) such as number of frames (n), frame speed (fp/sec), frame height (H) and width (W) are extracted from the header. The cover video and hidden are then broken down into frames. Now the proposed DLSB based technique has been applied to conceal the data in the carrier frames. The size of the message does not matter in video only when stego movie in movie, number of frames for the hidden movie must be small or equal to number of frames for the cover movie, steganography as the message or image can be embedded in multiple frames but in movie frame by frame.

The idea is that small changes to the least significant bits of a color won't be noticeable. For example, if a pixel's red color component is 254 instead of 255, no one will notice. Fact, you can make a whole lot more changes without seriously altering an image to see why, notice that the most significant bit in an 8-bit binary value contributes fully half of the total value. If you change that bit, you change the value by 128, half of the possible number of values 256.

The next most significant bit accounts for half of the value that you can create with the remaining 7 bits. Changing that bit alters the value by 64, half of the number of possible values with 7 bits 128. Together the two most significant bits account for 3/4 of the total value. If you continue this line of reasoning, the 3 most significant bits account for roughly 7/8 of the total, the 4 most significant bits account for 15/16 of the total, and so forth. Even if you take away 4 of a pixel's 8 bits of red, green, and blue color information, the resulting color is pretty close to the original color. Instead of storing only 1 bit of information in pixels scattered around the image, you can store several bits in every red, green, and blue color value throughout the image. This example does just that. It takes away the least significant pixels from one image and uses them to store the most significant pixels of a second hidden image. The hidden image's values are stored in the result image's least significant bits so they don't add greatly to the resulting combined image i.e. stego image.

For example, suppose you want to use 3 bits to hide one image inside another and consider the red component of a particular pixel. Suppose the cover image's red component for that pixel in binary is 10110101 and the hidden image's red component for that pixel is 01010011. To use 3 bits to hide the second value inside the value, we remove the 3 **least** significant bits of the first value and replace them with the **most** significant bits of the second value. In this example,

$$10110\del{101} + 010\del{10011} = 10110010 .$$

To see that the change is small, note that the original pixel's value was  $10110101 = 181$  and the final value is  $10110010 = 178$ . We stored 3 bits from the hidden value but only changed the original value by a small amount. To recover the hidden image, you extract the stego image's 3 least significant bits and use them for the hidden image's most significant bits. In this example,  $10110010$  gives the original image's value as  $10110000$  and the hidden image's value is  $01000000$ . These values are slightly different from the original values but they're close enough to be useful.

#### **4. ALGORITHM OF DLSB WITH AN APPLICATION**

The proposed algorithm, both for encoding and decoding along with application are given in this section. Embedding technique is given in section 4.1 whereas Extracting technique is given in section 4.2. The proposed technique is illustrated with an example in section 4.3.

#### **4.1 Algorithm of Embedding**

Input: cover video (AVI), hidden video (AVI).

Output: Stego video (AVI) .

Step 1: Input cover video file or stream.

Step 2: Read required information of the cover video.

Step 3: Break the cover video into frames.

Step 4: Input hidden video file or stream.

Step 5: Read required information of the hidden video.

Step 6: Break the hidden video into frames.

Step 7: Find (user selected) LSB bits of each RGB pixels of the cover frame.

Step 8: Find (user selected) MSB bits of each RGB pixels of the hidden frame.

Step 9: Obtain the position for embedding the secret data using dynamic LSB in cover frame.

Step 10: Embed the modified eight bits of the secret image pixel into (user selected) bits of LSB of RGB pixels of the cover frame respectively red by red, green by green blue by blue layers using the position obtained from step 9.

Step 11: Regenerate video frames (stego video).

#### **4.2 Algorithm of Extracting**

Input: Stego video (AVI) .

Output: Recover video (AVI), Cover video (AVI)

Step 1: Input stego video file or stream.

Step 2: Read required information from the stego video.

Step 3: Break the video into frames.

Step 4: Find user selected LSB bits of each RGB pixels of the stego frame. .

Step 5: Find number of frames for hidden movie and Obtain the position of embedded bits of the secret data using dynamic LSB in stego frame.

Step 6: Retrieve the LSB embedded bits from stego frame respectively red by red, green by green , blue by blue layers.

Step 7: Reconstruct the layers by modified the embedded bits in to eight bits again respectively ,red , green ,blue and save as frames .

Step 8: Regenerate hidden video frames which obtained from step 7.

Step 9: Regenerate cover video frames again.

#### **4.3. Illustration of DLSB technique**

Consider a RGB pixel value of the cover frame as below

R: 10110111 = 183

G: 10010100 = 148

B: 11001001 = 201

and a RGB pixel value of message hidden frame to be inserted in cover frame:

R: 10000000 = 128

G: 000 10111 = 23

B: 01111111 = 127

Selected DLSB for example (4 ) bit in a series of binary numbers, so that the solution of DLSB is

R: 1011~~0111~~ (183) + 1000~~0000~~ (128) = 10111000 (184)

G: 1001~~0100~~ (184) + 000 10~~111~~ ( 23) =10010001 (145).

B: 1100~~1001~~ ( 201) + 0111~~1111~~ (127) = 11000111(199).

We can expand above example to contain the whole image and then to all frames and make stego video as follows:

Resume cover video contain cover frames which contain three layers, the hidden video contain hidden frames which contain three layers, and dynamic algorithm work as follows for every frame :

cover red + hidden red  $\xrightarrow{\text{DLSB}}$  Stego red.  
 cover green + hidden green  $\xrightarrow{\text{DLSB}}$  Stego green.  
 cover blue + hidden blue  $\xrightarrow{\text{DLSB}}$  Stego blue .

This code can loop for all frames of hidden video and the rest of cover video put in AVI Stego video without do any things, then reconstruct Stego video.

The recover of the hidden video from stego video can be obtain by inverse Dynamic LSB algorithm which work as follows for every frame:

Stego red  $\xrightarrow{\text{IDLSB}}$  hidden red .  
 Stego green  $\xrightarrow{\text{IDLSB}}$  hidden green.  
 Stego blue  $\xrightarrow{\text{IDLSB}}$  hidden blue.

This code can loop for all frames of hidden video, and then reconstruct recover(hidden) video.

## 5. RESULTS AND PERFORMANCE EVALUATION

Any Steganography technique is characterized mainly by two attributes, imperceptibility and capacity. Imperceptibility means the embedded data must be imperceptible to the observer (perceptual invisibility) and computer analysis (statistical invisibility) [24]. The performance of the proposed technique is evaluated using three different video streams (train.avi, boot.avi and basketball.avi) and three secret video (count.avi, train.avi, boot .avi).

we use an objective measure, the Mean squared Error (MSE), Peak Signal to Noise Ratio (PSNR) to compare between cover video and stego video ,also between hidden video and recover video calculated below

$$\text{MSE} = \frac{1}{H*W} \sum_{i=1}^H \sum_{j=1}^W (p(i,j) - s(i,j))^2 \quad (1)$$

Where , MSE is Mean Square error, H and W are height width and P(i,j) represents original frame and S(i,j) represents corresponding stego frame.

$$\text{PSNR} = 10 \log_{10} l^2 / \text{MSE} \quad (2)$$

where, PSNR is peak signal to noise ratio, l is peak signal level for a grey scale image it is taken as 255.





















These objective measures is calculated for every layer (red, green, blue) and then the average is calculated of each frame by adding them and diving on three , and then adding the average for all frames and dividing by number of hidden frames to obtain the average MSE and average PSNR for all video as a result summarized in table (5-1):

### 5.1. Result of steganography movie in movie with objective measure:

In table (5-1) show the fourth first columns are represent cover , hidden, stego, recover video respectively ,the next two column represent number of frames for cover and hidden column , and the fourth last columns represent average mean square error and average peak signal to noise ratio between cover video with stego video and between hidden video with recover video.

If you look closely at the movie (stego ) show here, you will see that stego movie looks very much like the cover movie and the recovered hidden movie looks very much like its original value. By using 4 Bits, you get a pretty remarkable result. If you experiment with other numbers of bits such as 1 to 7, you will see lots of degradation in one movie or in other.

**Table (5-1) show cover, hidden, stego, and recover video with average MSE & average PSNR .**

<i>Cover video</i>	<i>Hidden video</i>	<i>Stego video</i>	<i>Recover video</i>	<i>No. of frame cover</i>	<i>No. of frame hidde</i>	<i>AMSE cover&amp; stego</i>	<i>APSNR cover&amp; stego</i>	<i>AMSE hidden&amp; recover</i>	<i>APSNR hidden &amp; recover</i>
				69	16	20.26	25.33	14.56	28.62
				250	69	15.13	28.24	20.56	25.19
				259	250	12.85	29.39	15.39	28.70
				259	69	11.35	31.16	21.87	24.57
				250	16	15.05	28.29	14.56	28.62

## **5.2. Graphical User Interface (GUI)**

This GUI is created as a user friendly wizard and does not need any previous training to operate it. It helps user to do steganography and construct recover movie again .This GUI is designed by using vb6 as show in figure 2.

This will help user with a wizard to

- load at first stage cover AVI movie and hidden AVI movie files .
- Detect number of bits to be embedded by entering .
- Hide image (frame) from hidden movie in image (frame ) of cover movie by DLSB.
- Construct (write ) stego movie the hidden message in a stego'd video
- Retrieve the hidden movie from a stego'd video by extracting images (frames ) in Visible image .
- Construct (write ) Recover movie the hidden message .

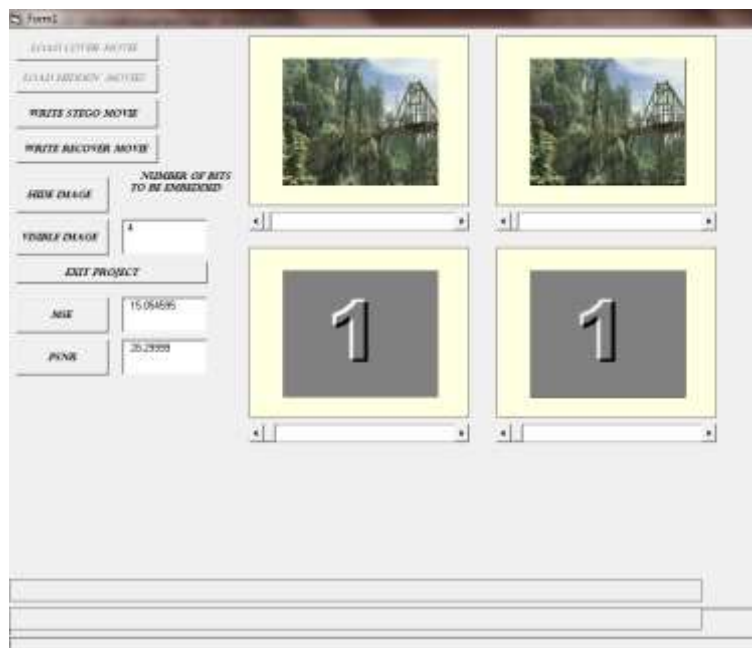


Figure 2 the Graphical User Inter Face (GUI).

### 5.3 Steganalysis of video

Several image Steganalysis technique exist in literature [25], [26], [27]. However these techniques do not work very well with video and yield very low performance. In recent times researchers have developed some video steganalysis techniques. In [28] a technique for video steganalysis by using the redundant information present in the temporal domain as a deterrent against secret messages embedded by spread spectrum steganography has been proposed. Kancherla et. al. [29] has proposed a video steganalysis method using neural networks and support vector machines to detect hidden information by exploring the spatial and temporal redundancies. Literature survey suggests that when temporal redundancies are used as video steganalysis then performance is more satisfactory than in spatial domain. Where as in [30] a steganalysis algorithm has been proposed that uses the correlation between adjacent frames to detect a special distribution mode across the frames. This is considered to work well with AVI file formats. However every carrier media is supposed to have its own special characteristics and thus it behaves differently when a hidden movie(message) is embedded in it. To summarise, existing video steganalysis technique may not work very well to large geometric shapes of constant color in hidden movie using the proposed DLSB technique.

### 6. CONCLUSION

A secured Dynamic LSB technique for video steganography has been presented in this paper. This technique utilizes cover video files in spatial domain to conceal the presence of sensitive data regardless of its format. The proposed technique is applied to AVI files, however it can work with any other formats with minor procedural modification. For compressed video files like MPEG the video needs to first decompress then the technique can be applied to the uncompressed video. Whereas for Flash Video FLV files the technique can be applied without modification. hide movie into video images (AVI) that provides a robust and secure way of data transmission. It implements steganography in video image and reveal process without restarting the application or starting a different application. Also this system is a Platform-independent application with high portability and high Consistency. Software based Stenographic Engine for video steganography is the future scope of the technique.



**REFERENCES**

- [1] E. Cole and R.D. Krutz, *Hiding in Plain Sight: Steganography and the Art of Covert Communication*, Wiley Publishing, Inc., ISBN 0-471-44449-9, 2003.
- [2] S.K. beisser and A. P. F. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Books, ISBN 1-58053-035-4, 1999.
- [3] D. Stanescu, M. Stratulat, B. Ciubotaru, D. Chiciudean, R. Cioarga and M. Micea, *Embedding Data in Video Stream using Steganography*, in 4th International Symposium on Applied Computational Intelligence and Informatics, SACI-2001, pp. 241-244, IEEE, 2007.
- [4] F. Pan, Li Xiang, X . Yang and Y. Guo, *Video Steganography using Motion Vector and Linear Block Codes*, in IEEE 978-1-4244-6055-7/10/, pp. 592-595,2010.
- [5] A. Westfield, and A. Pfitzmann, *Attacks on Steganographic Systems*, in Proceedings of 3rd Info. Hiding Workshop, Dresden, Germany, Sept. 28–Oct. 1, pp. 61-75, 1999.
- [6] J. Fridrich, R. Du, and L. Meng, *Steganalysis of LSB Encoding in Color Images*, in Proceedings of ICME 2000, Jul.-Aug. 2000, N.Y., USA.
- [7] E. Kawaguchi and R. O. Eason, *Principle and applications of BPCS-Steganography*, in Proceedings of SPIE Int'l Symp. on Voice, Video, and Data Communications, pp. 464-473,1998.
- [8] G. Caccia, R. Lancini, *Data Hiding in MPEG2 Bit Stream Domain*, in Proceedings of International Conference on Trends in Communications, pp.363-364, 2001.
- [9] J. Zhang, J. Li, L. Zhang, *Video Watermark Technique in Motion Vector*, in Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing, pp.179-182, 2001.
- [10] A. Giannoula,D. Hatzinakos, “Compressive Data Hiding for Video Signals”, in Proceedings of International Conference on Image Processing, pp. I529- I532, 2003.
- [11] Y. C Tseng and H. K Pan, *Data Hiding in 2-color Image in IEEE Transactions on computers*, Vol. 51, No. 7, pp. 873-878, July 2002.
- [12] J. J. Roque and J. M. Minguet, *SLSB: Improving the Steganographic Algorithm LSB*, in the 7th International Workshop on Security in Information Systems (WOSIS 2009), Milan, Italy, pp.1-11, 2009.
- [13] J. J. Chae, B. S. Manjunath, *Data Hiding in Video*, Proceedings of the 6th IEEE International Conference on Image Processing, pp.311-315, 1999.
- [14] M. Pazarci, V. Dipcin, *Data Embedding in Scrambled Digital Video*, in Proceedings of the 8<sup>th</sup> IEEE International Symposium on Computers and Communication, pp. 498-503, 2003.
- [15] A.K. Bhaumik, M. Choi, R.J. Robles and M.O. Balitanas, *Data Hiding in Video in International Journal of Database Theory and Application* Vol. 2, No. 2, pp. 9-16, June 2009.
- [16] F. Pan, L. Xiang, X. Yang and Y. Guo, *Video steganography using motion vector and linear block codes*, in Proceedings of IEEE International Conference on Software Engineering and Service Sciences (ICSESS- 20100), pp. 592-595, 2010.
- [17] M. Ramalingam, *Stego Machine Video Steganography using Modified LSB Algorithm*, in World Academy of Science, Engineering and Technology 74 2011, pp. 502-505, 2011.
- [18] K. S.M. Masud, R.Hossain, M.L., *A new approach for LSB based image steganography using secret key*, in Proceedings of 14th International Conference on Computer and Information Technology (ICCIT-2011), pp.-286-291, Dec. 2011.
- [19] *Steganographic software*, <http://www.jjtc.com/Steganography/toolmatrix.html> [last accessed on 16-04- 2012]
- [20] H. Ajetrao, Dr. P.J.Kulkarni and Navanath Gaikwad, *A Novel Scheme of Data Hiding in Binary Images*, in International Conference on Computational Intelligence and Multimedia Applications, Vol.4, pp. 70-77, Dec. 2007.
- [21] S. Sachdeva and A. Kumar, *Colour Image Steganography Based on Modified Quantization Table*, in Proceedings of Second International Conference on Advanced Computing & Communication Technologies (ACCT-2012), pp. 309-313, 2012.
- [22] R. Machado, <http://www.securityfocus.com/tools/586/scoreit>, .EzStego., Nov. 1996. [last accessed on 16-04-2012]

- [23] L. Fillatre, Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption, IEEE Transactions on Signal Processing, Volume 60, Issue:2, pp. 556-569, Feb, 2012
- [24] A. Almohammad, Steganography-Based Secret and Reliable Communications: Improving Steganographic Capacity and Imperceptibility, phd thesis Brunel University August, 2010.
- [25] N. F. Johnson and S. Jajodia, Steganalysis of Images Created using Current Steganography Software, in Lecture Notes in Computer Science, vol. 1525, pp. 32 – 47, Springer Verlag, 1998.
- [26] S. Dumitrescu, X. Wu and N. Memon, On Steganalysis of Random LSB Embedding in Continuous tone Images, in Proceedings of the International Conference on Image Processing, vol. 3, pp. 641 – 644, June 2002.
- [27] J. Fridrich, M. Goljan, D. Hoge and D. Soukal, Quantitative Steganalysis of Digital Images: Estimating the Secret Message Length,” in ACM Multimedia Systems Journal, Special issue on Multimedia Security, vol. 9, no. 3, pp. 288 – 302, 2003.
- [28] U. Budia, D. Kundur and T. Zourntos, Digital Video Steganalysis Exploiting Statistical Visibility in the Temporal Domain, in IEEE Transactions on Information Forensics and Security, vol. 1, no. 4, pp. 502 – 516, December 2006.
- [29] K. Kancherla and S. Mukkamala, Video Steganalysis using Spatial and Temporal Redundancies, in Proceedings of International Conference on High Performance Computing and Simulation, pp. 200– 207, June 2009.
- [30] Y. Su, C. Zhang, L. Wang and C. Zhang, A New Video Steganalysis based on Mode Detection, Proceedings of the International Conference on Audio, Language and Image Processing, pp. 1507– 1510, Shanghai, China, July 2008.