# A STEGANOGRAPHIC APPROACH IN VIDEO WITH ATTACK DETECTION

Md. Golam Rabiul Alam, Md. Monirul Islam, Tahmina Naznen, Tasnim Niger, Shanjida Sharmin
Department of Computer Science and Engineering, International Islamic University Chittagong, Bangladesh.
E-mail: gra9710@yahoo.com, monirliton@yahoo.com, taha.cse@gmail.com,tasnimtisha@yahoo.com, kheya_iiuc@yahoo.com

**Abstract:** *Steganography is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. Studies have shown that Human Vision Systems (HVS) is unable to detect changes in uncorrelated areas of the digital media, due to the complexity of such areas, where it is easy to detect changes in correlated areas. Security and quality are two important issues of steganography. In this paper we have introduced a model of steganography with attack detection using Least Significant Bit (LSB) hiding. Here we have used a video file as a cover media to hide a message. In our algorithm error detection code is used, so that receiver will be able to detect if there is any alteration in the cover media and retrieve the message.*

***Keywords****: Cover Medium, Human Vision System, Stego Video, Attack Detection, Steganography.*

## 1. Introduction

Steganography hides information in a manner that the existence of the message is unknown. The goal of steganography is to communicate as many bits as possible without creating any detectable artifacts in the cover-object. If any suspicion about the secret communication is raised, then the goal is defeated. Steganalysis is the art of detecting the presence of covert communication between sender and receiver. A steganographic scheme is considered secure if no existing steganalysis method distinguishes cover and stego-images with a success better than random guessing. The embedding process on an object, while being perceptually transparent, leaves statistical artifacts that can be used to distinguish stego and cover-objects. The argument that data hiding methods leave telltale effects is common to all steganalysis methods [1]–[4]. In this method, we first choose a video file as cover medium. Then we select a frame from it, which will carry the secret message and determine whether it has indexed color data or true color data. According to our proposed Embedding algorithm we embed the message bits sequentially into selected frame. To extract the massage first we read the stego video (the video file that contain secret message) and apply our proposed Extract algorithm. And finally we get our secret message. If there occurs any external attack in our stego video, the proposed Extract algorithm can also detect it. Here AVI video files are used in our experiment.

## 2. Literature Review

The word *steganography* comes from the Greek *steganos* (covered or secret) and *–graphy* (writing or drawing) and that means, literally, covered writing [5].Typically there are three types of steganographic approaches [6].They are Pure Steganography, Secret Key Steganography and Public Key Steganography. Pure Steganography is defined as a steganographic system that does not require the exchange of the stego-key. In this case the sender and receiver can rely only upon the presumption that no other parties is aware of this secret message. Secret Key Steganography is defined as a steganographic system that requires the exchange of a secret key or stego-key prior to communication. Only the parties who know the secret key can reverse the process and read the secret message. The most important benefits of Secret Key Steganography is that, if the stego-image is intercepted, only parties who know the secret key can extract the secret message. Public Key Steganography is defined as a steganographic system that uses both public key and private key to secure the communication between the parties want to communicate secretly. Here public key is remained open to all but private key is kept secret.

### 2.1 Steganography in Video

The sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message.

Video steganogrpahy is very much similar to image steganography. Digital video is a stream of image frames. Video file is also subject to Bit Plane Complexity Segmentation (BPCS) steganography [9]. The BPCS technique makes use of the property of bit plane decomposition. A very popular format for video is the MPEG format [10], and there is research going on as to how to increase the payload in video steganography. An interesting topic would be to make the user capable of downloading two files as a single bigger file. This can be useful in areas where internet bandwidth is still a problem. So

suppose the user wants to download a video, so all the information concerning the video like which player it would play in the browser, the caption the video will carry (if any) and any accompanying pictures can be embedded in the single video file and then the browser should be able to demultiplex the different information. This would help save the bandwidth.

**2.2 Attack Detection**

Attack detection is the technique, which can detect external attack in the stego media if there occurred any attack in it. If intruder attacks on the stego object, our proposed extraction algorithm will help the receiver to detect it and inform the sender to change the communication channel.

**2.3 Audio Video Interleave (AVI) File**

Audio Video Interleave, known by its acronym AVI, is a multimedia container format introduced by Microsoft in November 992 as part of its Video for Windows technology. It is a derivative of the Resource Interchange File Format (RIFF), which divides a file's data into blocks, or "chunks." [7]. We know that a video file is a combination of number of frames where a frame is snapshot (pixmap) of the current axes or figure. AVI Frames have two different types of structure having two fields: color data (cdata) and colormap [8]. Here structure of the avi frames are given.

Table 1: Structure of AVI file format

| Frame's Image Type | Cdata Field | Color-map field |
|---|---|---|
| True color | Height-by-width-by-3 array | Empty |
| Indexed | Height-by-width array | m-by-3 array |

## 3. Previous Work in Video Steganography

Noda et al [9] explain video steganography by using image steganographic techniques. They use BPCS steganography combined with wavelet compression. Chae and Manjunath [11] use an embedding scheme based on texture masking and lattice structure. They use the block DCT (Discrete Cosine Transforms) in individual video frames for embedding data. Westfeld and Wolf [12] describe a steganographic technique used in avideoconferencing system. It is a DCT based lossy compression mechanism. George et al [13] analyze the spread spectrum technique when applied to images and video. The spread spectrum method has the advantage that the watermark extraction is possible without using the original unmarked image.

## 4. Proposed Steganographic Approach

In this section our proposed steganographic approach is described. At first the proposed algorithm of embedding and extracting with attack detection is mentioned. After that the technique is described elaborately.

### A. Algorithms

**A.1 Algorithm: Embedding Algorithm**

Input: The cover video (V), The secret message (M).
Output: The stego video (V').
Step 1: Convert the secret message (M) to a stream of bits, L(M) is the length of M in bits.
Step 2: Select any frame (F) from cover video for embedding message.
Step 3: Separate cdata and colormap of that frame (if the frame is indexed).
Step 4: Call Media Test Procedure to get position matrix.
Step 5: If the position matrix is not found that means the cover image is not suitable for the secret message. Choose another cover & go to step 2.
Step 6: Call the hide procedure to hide the message.
Step 7: Exit.

**A.2 Algorithm: Media Test**

Input: Selected Frame (F), L(M).
Output: Position matrix, threshold value.
Step 1: Divide cdata of F into m*n blocks.
Step 2: For each block calculate

$$\delta b = \sum_{i=1}^{p-1} |Cbi - Cb(i+1)|$$

Step 3: Set threshold value (τ).
Step 4: If τ is above a certain value, Return.
Otherwise each block with the property (δb> τ) will be used for embedding. The total number of pixels of such blocks is the MMS.
Step 5: If the size of the secret message L (M) is greater than MMS go to step 3 and decrease the value of τ.
Step 6: Construct the position matrix such that (δb>τ).
Step 7: Exit.

**A.3 Algorithm: Hide**

Input: The selected Frame (F), The secret message (M).
Output: The stego video (V').
Step 1: Repeat for i=1 to the size of secret message L(M) do step 2 to 3 Choose mi.
Step 2: If mi=most significant bit of Cp then do step 3 to 4.
Step 3: Set the exclusive-or between bit 7 in Cp & LSB of Cp to zero by changing the value of the LSB of Cp.
Set the exclusive-or between the LSB of Cp & bit of Cp to be the value of bit 2 of Cp.
Else
Set the exclusive-or between bit 7 in Cp & LSB of Cp to one by changing the value of the LSB of Cp.

Set the exclusive-or between the LSB of Cp & bit 7 of Cp to be the value of bit 2 of Cp.
Step 4: Concate the embedded cdata with its colormap to form the stego Frame (if the frame is indexed image).
Step 5: Replace the stego frame with other frames in its original position.
Step 6: Exit.

Step 8: For image message:
Line up the extracted message bits according to the correct sequence and convert them to integer number. To retrieve the secret image reshapes these numbers according to the image size that included in the secret key.
Step 9: Exit.

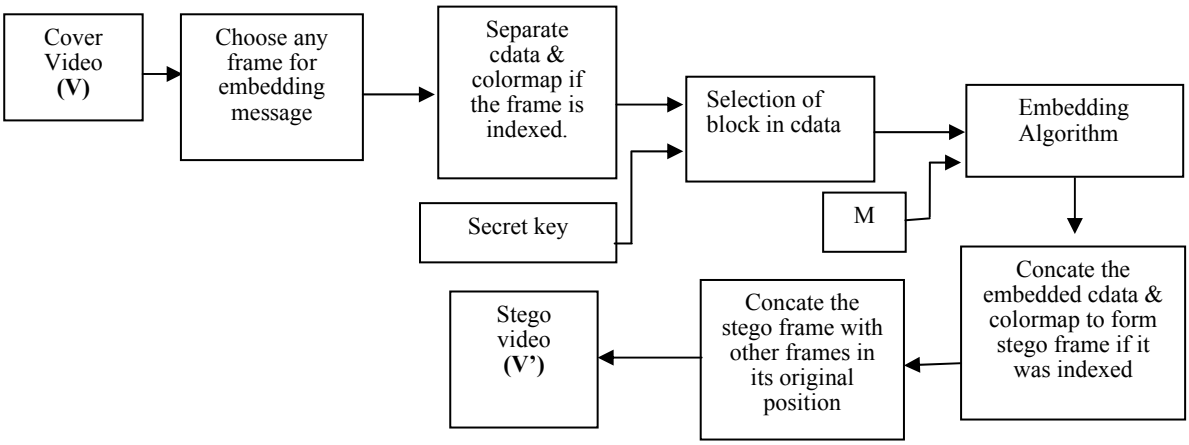


Fig 1: Proposed Embedding Process

### A.4 Algorithm: Extract

Input: The stego video (V'), the secret key (K)
Output: Embedded message, Attacked (true, false)
Step 1: Identify the Stego frame (F), threshold value τ, the sequence of the secret message and block size. (And separate cdata and colormap of the stego frame if it is indexed).
Step 2: Divide the cdata of F into m*n block.
Step 3: For each block calculate

$$\delta b = \sum_{i=1}^{p-1} |Cbi - Cb(i+1)|$$

Step 4: Using the threshold τ and δb determine the blocks containing the secret message.
Step 5: Construct CLSB matrix using the equation
$C1' = C2 \oplus C7$ & construct ALSB matrix.
Step 6: Calculate the result matrix R using equation Ri = | CLSBij – ALSBij |
Step 7: If result =0 for, then retrieve the message bits using the CLSB & equation

$$m = \begin{cases} c8 & \text{if } C1' \oplus C7 = 0 \\ \overline{c8} & \text{otherwise} \end{cases}$$

& go to step 8.
Else
result =1 then do
set attacked=true. & output that: The message can not be retrieved.
Step 8: For text message:
Line up the extracted message bits according to the correct sequence and convert them to character in order to form the secret message.
Or

### B. Details Technique

In our technique we introduce an algorithm that uses the two LSBs of the cover video frame to hide a message and makes use of the Error Detection Code to detect attacks. Our proposed method takes as an input a cover video (V) that will be used to hide the secret message (M), (In our method the cover object is a video file while the secret message could be of any kind). This system produces as an output a stego video that will be used later to extract the message from it.
The implementation consists of an Embedding Phase and an Extracting Phase.

### B.1 Embedding Phase

Firstly we read the message that will be hidden in the video is read from a file and converts into a stream of bits. In the embedding process the cover object passes through four major steps. In the first step we have to choose any frame from the cover video for embedding message. In the second step we separate cdata and colormap from that frame it is indexed otherwise it is not necessary. Then we select the block by using stego key and lastly embed the message in the selected frame using our embedding technique.
Here Fig. 1 shows the overall embedding technique. After blocking the pixels of the frame we will calculate statistical information according to the following equation:

$$\delta b = \sum_{i=1}^{p-1} |Cbi - Cb(i+1)|$$

| | | |
|---|---|---|
| 108 | 210 | 80 |
| 40 | 50 | 220 |
| 60 | 120 | 110 |

Block(a)

| | | |
|---|---|---|
| 210 | 212 | 213 |
| 209 | 210 | 209 |
| 210 | 213 | 212 |

Block(b)

Where: P = n*m, this is the number pixels in block.
Cbi : is the pixel value *i* at a cover block.

Example:
We have two 3x3 blocks named as block(a) and block(b)
Now δb for two blocks are 758 and 16 respectively, so block(a) represents an inhomogeneous block with large value, while block(b) is a highly correlated block with small value. Using a predefined threshold (τ) we choose all blocks having δb greater than τ to construct the Position Matrix (a matrix contains the coordinates of the blocks used for hiding the message). Once these blocks are known, the number of these blocks determines the Maximum Message Size (MMS). At the end of this step the MMS is returned along with the Position Matrix that contains only (δb > τ).In the second step the size of the secret message is checked to be less than or equal to the MMS. If it is not, then a different cover object will be chosen, or the MMS will be increased by decreasing τ.

Hide the message:
In this step, the secret message will be embedded inside the cover video. The block locations that will be used for embedding the secret message are located in the position matrix. The message embedding process is continued until the message length.
The embedding is done according to the given formula:

$$C1=C\{1,0\}\begin{cases} C1 \oplus C7 = 0 & \text{if } m=c8 \\ C1 \oplus C7 = 1 & \text{if } m<>c8 \end{cases} \quad \dots\dots (2)$$

$$C2 = C1 \oplus C7 \quad \text{for all pixels} \quad \dots\dots\dots \quad (3)$$

In (2) we compare the message bit with the most significant bit of cover pixel. If they are match we set the exclusive-or of c1 and c7 to 0 by changing only the LSB c1 if necessary. And if they are not match then set exclusive-or of c1 and c7 to 1 by changing only the LSB c1 if necessary. In (3) we set c2 to hold the logical exclusive-or between c1 and c7. This way we are embedding one message bit by changing at most two bits, namely c1 and c2. In the stego key we also send the selected frame number, which is used for embedding message and length of message bits if the message is text or send image size if the message is an image.

### B.2 Extracting Phase

In the Extracting Phase we use the same threshold (τ) that was used in the Embedding Phase -along with the stego key- to specify the locations being

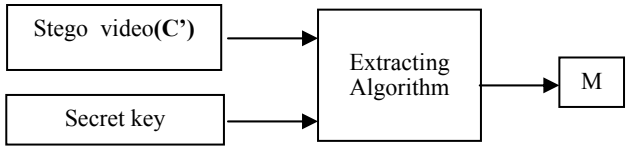


Fig 2: Proposed Extracting Technique

used to store the message and then extracting the message bits. In this Phase a reverse process is used. From the secret key we will first identify the threshold value (τ)-which will be used to identify the blocks containing the secret message- then the correct sequence of the message bits and the targeted pixels at those blocks will be identified.

Here Fig. 2 shows the overall extracting technique.

$$C1' = C2 \oplus C7 \dots\dots\dots (4)$$

$$m=\begin{cases} c8 & \text{if } C1' \oplus C7 = 0 \\ \overline{c8} & \text{otherwise} \end{cases} \dots\dots (5)$$

From receiving the stego video, the receiver firstly choose the stego frame then separate colormap and cdata if it is indexed. Then construct Position matrix in cdata using **τ** and Equation (1), this will identify the blocks used for embedding the message. Using (4), we compute the LSBs of the Stego cover to construct the computed LSB matrix (CLSB), The actual LSBs of the stego cover conforms the ALSB matrix, It will be of the same size as CLSB.

In(5) we compare the Least significant bit and c7 of

$$\mathbf{CLSB} = \begin{pmatrix} R11 & R12 & R13..R18 \\ R21 & R22 & R23..R28 \\ & . & \\ & . & \\ Rn1 & Rn2 & Rn3..Rn8 \end{pmatrix} \qquad \mathbf{ALSB} = \begin{pmatrix} A11 & A12 & A13..A18 \\ A21 & A22 & A23..A28 \\ & . & \\ & . & \\ An1 & An2 & An3..An8 \end{pmatrix}$$

Fig 3: CLSB & ALSB matrix where n = L(M) the number of message bits.

the CLSB. If they are matched we set the most significant bit(c8) in the message bit m. And if they are not matched then set inverse bit of the most significant bit in the message bit m.

**Ri = | CLSBij – ALSBij |… … … (6)**

Using Equation 6 we get the result matrix R that has the following properties:

- If R = 0 then ALSB and CLSB are identical, that means the stego cover hasn't been attacked and the extracted message will be correct and 100% similar to the original message. CLSB will be used along with Equation 5 to retrieve the message.
- If R< > 0 then ALSB and CLSB are not identical; in this case we might suspect that an attack changes some or all of the message bits.

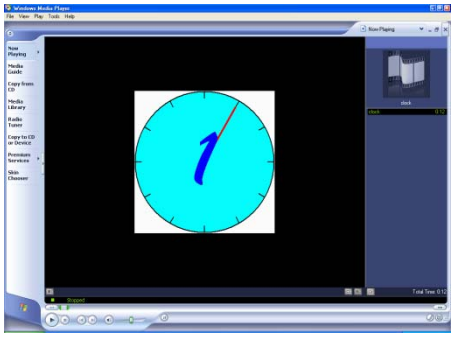

Fig 3.a: Clock (Original video that played in MATLAB during experiment)

In the last case above the receiver of the message will be able to know that there is an attack on the stego cover, he can inform the sender that their communication channel is being watched and its better to be changed.

## 5. Result of Experiment

Our proposed model makes use of Human Vision System (HVS) properties and here HVS is unable to detect any changes in digital media after embedding the secret messages (text or image) into the cover image. As for example, Fig 3.a shows the original video 'Clock' that has been played in windows media player. This video file has total 12 frames and we hide secret message in its first frame. Fig. 3.b shows the same video after embedding the message. Here the message is " Steganography is the art and science of writing hidden messages ".

So from above figures we can conclude that our model produces high image quality with no visual difference between the original and stego video.

## 6. Conclusion

Finally the proposed steganography model produces high quality stego media, robust and more secure comparable to the well-known usual LSB scheme. In addition to that the concept of error detection code was introduced. This model makes use of Human Vision System (HVS) properties and embeds the message in the most important areas of the video frame. Experimental results show that this method is efficient and effective and that it produces high quality of stego videos. Future work may focus on applying this method for other types of steganography such as audio and can be done to improve the security by using public-key steganography to protect intellectual property like copyright, license information etc.

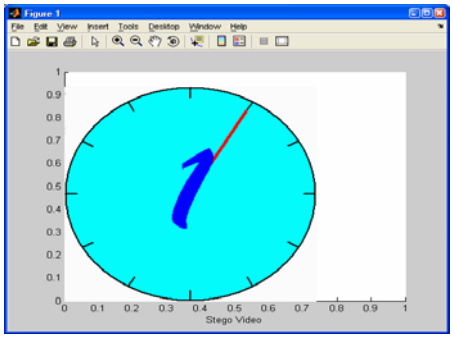


Fig 3.b: Clock (With secret message that played in windows media player)

## Reference

[1] J. Fridrich, " Feature based steganalysis for JPEG images and its implications for future design of steganographic schemes," in Proc. 6th Information Hiding Workshop, Toronto, ON, Canada, May 23–35, 2004.

[2] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in Information Hiding 5th International Workshop, F. A. P. Petitcolas, Ed. New York: Springer-Verlag, 2002, vol. 2578, Lecture Notes in Computer Science, pp. 340–354.

[3] I. Avcýbas¸, M. Kharrazi, N. Memon, and B. Sankur, "Image steganalysis with binary similarity measures," EURASIP J. Appl. Signal Process., vol. 2005, no. 17, pp. 2749–2757, Sep. 2005.

[4] J. J. Harmsen and W. A. Pearlman, "Steganalysis of additive noise modelable information hiding," Proc. SPIE/IS&T Electron. Imag., vol. 5022, Jan. 2003.

[5] Petitcolas, R. Anderson and M. Kuhn,1999.Information hiding – A Survey, Preceeding of the IEEE, 87: 1062-78.

[6] Md. Manzoor Murshed, "Steganography using LSB hiding", Upper Iowa University, Fayette, Iowa, USA.

[7] "AVI File Format", available at: http://en.wikipedia.org/wiki/AVI

[8] "AVI File Format", available at: http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/aviread.html

[9] Hideki Noda, Tomofumi Furuta, Michiharu Ni-imi, and Eiji Kawaguchi. "Video steganography based on bit-plane decomposition of wavelet-transformed video." Proceedings of SPIE-Volume- 5306,June2004

[10] Akonian, David., D'Sousa, Sunil., Agaian, Sos,"Wireless steganography." SPIE - The international society for Optical Engineer, 2006

[11] J. J. Chae and B. S. Manjunath, "Data hiding in video," IEEE International Conference on Image Processing, vol. 1, pp. 311-315, 1999.

[12] A. Westfeld and G. Wolf, "Steganography in a video conferencing system," Lecture Notes in Computer Science, 1525 ed, 1998, pp. 32.

[13] M. George, J.-Y. Chouinard, and N. Georganas, "Digital watermarking of images and video using direct sequence spread spectrum techniques," Canadian Conference on Electrical and Computer Engineering, vol. 1,pp.116-21,1999