International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-07, Issue-06, SEP 2021

Video Steganography Using LSB Technique by Unique Frame Selection Method to Increase the Security

Md Firoj Ali, Assistant Professor, Ramakrishna Mission Residential College(Autonomous), Narendrapur, University of Calcutta, Kolkata, India. 700103. firojali.mca@gmail.com

Abstract Steganography is a state-of-the-art technique to hide a piece of information within another such that the change in the cover information cannot be detectable. In this paper, we have proposed security for an image using the concept of video steganography where the secret information is to be hidden in the randomly chosen unique video frame. We have used the LSB method to hide that image into multiple video frames which are generated by Linear Congruential Generator, a pseudo random number generator that generates random numbers defined by a recurrence relation. Since we are first resizing the frame equal to the cover frame along with the bit-slicing method, exactly 8 cover frames are needed to embed the secret image. As the cover frames are selected on a random basis in our proposed algorithm, it is very hard to get the secret image for the eavesdroppers.

Keywords —Bit-plane slicing, Pseudo-random number generator, LSB, MSB, Steganography

I. INTRODUCTION

Steganography is the method where a message can be identified only by the sender and the receiver. It emerges from the combination of two different words such as "Stego" and "graphy" where "Stego" means "Covered or concealed" and "graphy" means "writing and drawing". It has been started in various forms from the past 2500 years. It is vastly used in World War 2 to secretly pass one's information to the others by hiding from the opponent. Later it has been done in microfilm, micro-drops, and some chips. So the concept of steganography is very old and effective. The method is used to block the survival of communication by introducing the concept that the message should be hidden in some other files. The supported formats can be image, audio, video and the hidden data may be any important text, image, audio, or even a video [1,2,3,4,5,9]. There are two methods for hiding data in another one. The first well-known method is Cryptography which encodes the private data in a way so that it can be unlocked only at the sender side. This method includes a very effective, costly, and secured pipeline for the decryption key. The second method Steganography works on hiding the actual message instead of encrypting. The method proposes cover writing while cryptography relies on secret writing and is used only for data protection. So by steganography, secret communication can be easily done and also refers to concealing the existence of the message. In the later stages of Steganography, encryption can be added to it for the best communication pipeline using a mixture of steganography and cryptography.

Though Steganography is less popular than cryptography, it has more benefits as outsiders cannot notice the secret data. According to the cover media, steganography has different names corresponding to the use cases such as audio steganography, image steganography, text steganography, video steganography.

Video steganography can hide an image, text, audio also even another video within a mask video. It is more prominent as videos are large and consist of audio and many running images. In video steganography, image and audio steganography techniques can also be introduced by imposing high security. In video steganography, the original video file is known as cover video while the video after embedding the secret data is known as stego-video. In this paper, we are interested in the manipulation of selected video frames of the cover video file to hide secret data or messages.

II. LITERATURE SURVEY

Nowadays, the main motto of steganography[1] is to hide intellectual protection of the data from hackers. It has been often seen that people emphasize the digital steganography system rather than doing micro concepts. So there is a high concern regarding steganography. Since it is a very old method, several methods have been used to update it in recent scenarios and use cases. Mainly they are pure steganography, public-key steganography, private-key steganography. As there are many steganography types, there should be some different approaches. They are mainly divided into two domain categories: spatial domain or substitution domain technique and transform or frequency domain technique.

The spatial domain mainly substitutes the redundant part of the cover image with that secret information by operating pixel or block wise. On the other hand, in the transformation domain, steganography is done by embedding the secret

 DOI : 10.35291/2454-9150.2021.0541

information in the transformed space of the signal. So, messages are also embedded in that transform coefficient.

Munasinghe et al [2] proposed a method to hide a video in a visual file in which the LSB of each byte of the cover file is changed to embed the secret data. This method only changes the least significant bits by keeping the size of the ultimate cover file constant. Hence the existence of the data cannot be detected.

Begum et al [3]. proposed LSB-based video steganography which uses visual cryptography for secure data hiding and transmission over networks. This approach uses the genetic algorithm to shuffle the pixel location of the stego image and visual cryptography to create the shares. Since visual cryptography is used, it suffers from the requirement to represent every pixel within the original frame by multiple pixels in every share, leading to a contrast deterioration problem called pixel expansion.

Dasgupta et al [4]. proposed a hash-based least significant technique for video steganography. The proposed technique takes 8-bits of secret data at a time and conceals them in LSB of RGB pixel value of the cover frames in 3, 3, 2 order respectively. Such that out of 8-bits of message 6-bits are inserted in the R & G pixel and remaining 2 bits are inserted in B pixel. The embedding positions of the eight bits out of 4 available bits of LSB are obtained using a hash function of the form, k= p%n where k is LSB bit position within the pixel, p represents the position of each hidden image pixel and n is the number of bits of LSB.

Video steganography in recent days has also gained quite significance for researchers. Various techniques of LSB exist, where [5] proposed the data is first encrypted using a key and then embedded in their career AVI video file in LSB keeping the key of encryption in another file called key file. Another video steganography scheme based on motion vectors and linear block codes has been proposed in [6, 7].

III. Proposed Method

In this section, we proposed an algorithm that consists of a random selection of frames, the LSB method, and resizing an image. The architecture is shown in Figure 3. As previously, we also know video steganography can be done by audio or image steganography method, here we choose image steganography logic to implement video steganography. Here we take the video path, the hidden image path as the input, and the key of the hidden technique method.

A. Bit Slicing

In this section, the information that needs to be hidden is taken for processing. This process includes image bit slicing and division of that image into 8 different bit-planes. The pixel values of each image are converted to its corresponding 8-bit binary values. Every i-th bit has been taken from each byte of pixel values to form the i-th bit-plane image as in Figure 1.

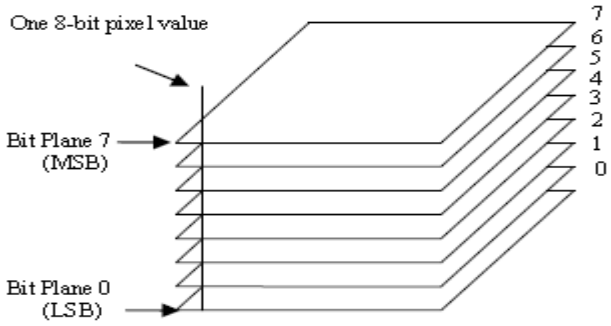

Figure 1 i-th Bit Plane

B. LSB Technique

. The video file should be converted into frames. Generally, in a small video file (AVI, MPEG, MP4), at least 20-25 frames can be generated per second (FPS). Frame in the video is an image consisting of a collection of pixel values (color and intensity) that can be represented in a matrix form or a list formation within a single instance. The 24-bitmap RGB images have 24 bits values for each pixel, 8-bits for each 3 color channel. The RGB is most suitable as there is lots of information where we can hide secret messages, with a one-bit change for each byte as in Figure 2.

Each component is of one byte i.e. 8-bits in which the first one is the most significant bit. The LSB technique (Least Significant Bit) is used for hiding secret information resulting in the change in the last bit of each byte of the component. Substitution of the least significant bit (LSB) results in human imperceptibility. We all know the first and foremost requirement of steganography is the invisibility of the changes in the cover frames. The strength of steganography lies in its ability to be unnoticed by the human eye. For, hiding three bits of the data in every pixel's color, we used a 24-bit image. The human eye cannot easily differentiate between 21-bit color and 24-bit color.

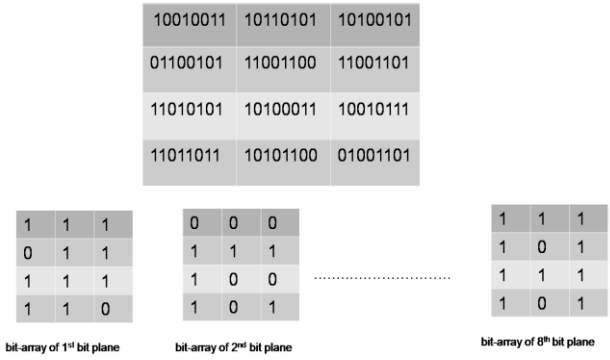

Figure 2 Visual Representation of MSB, LSB Plane

C. Frame Selection

The embedding process of this algorithm involves selecting frames randomly from the cover video in which the data is to be hidden. Here we used Linear Congruential Generator (LCG), a pseudo random number generator that

 DOI : 10.35291/2454-9150.2021.0541

generates random numbers defined by a recurrence relation. We used a seed-value which is also the key for retrieving the data, given by the user. By using this seed value, cover frames can be selected and those frames are unique for that particular seed value.

Selecting the frames, the user can retrieve the data i.e. bit-planes embedded in those frames. The Python3 random.seed(x) method [7] is used, where x is the seed value for which the function generates a unique sequence of random numbers. So there will be no repetition of frames as we used the seed value in the random function.

Since we divided the original image into 8-bit planes, 8 frames are sufficient to embed those bit-planes. The LCG generates the residues of successive powers of a number i.e. pseudo-random numbers having good randomness properties by equation (1):

$$x_n = a_n \ mod \ m \quad (1)$$

which is equivalently

$$x_n = a x_{n-1} \ mod \ m \quad [7] \quad (2)$$

Here we used a = multiplier and m = 2k. As the above modulus function is cyclic after certain periods, here the maximum possible period is 2k-2. So our function to generate 8 sufficient frames without any repetition is

$$x_n = 5 x_{n-1} \ mod \ 2^5 \quad (3)$$

x0, be the initial seed value which is an odd integer. Thus the random sequence of frames is generated from taking the first random number of the random sequences corresponding to the respective residues.

For instance, if we take the initial seed value 1 then from the above equation (3), the generated sequence is 5,25,29,17,21,9,13,1 and their corresponding pseudo random numbers are 34,145,16,65,30,126,115,120 which are also the unique frame numbers.

D.Embedding

A simple LSB insertion process is used for the embedding image in the cover video file frame. We replaced the last bits of each 24-bit pixel of the selected cover frames (target frames) with each bit-plane image generated by the bit slicing method. Here we embedded every (8-i)-th bit plane image into the ith frame of the randomly generated sequence.

E. Extraction

In this section, we propose retrieval of the hidden image from the embedded video which is called stego-video. We used a very simple extraction algorithm to extract the secret image from the stego-video.

Step1: Firstly, we extract frames all frames from the video

Step2: Select the stego frames using the above-mentioned frame selection method

Step3: As the secret image is embedded into the least significant bits of every stego-frames, so we simply do the logical AND operation between 1 and every byte of the carrier frames to get the bit-array of LSBs which is the bit-array of one of the bit-plane images. At this stage, we have 8 bit-plane images of the original image. To recover the original image those 8 bit-plane images must be merged according to the order by which they are inserted into the cover frame sequences.

IV. ALGORITHM

A. Algorithm for Image Embedding

Step 1: Read the image file
Step 2: Resize that image into the size of the cover video
Step 3: Divide the images into bit-slicing method [8]
Step 4: Input the cover video
Step 5: Split that video into frames
Step 6: Select 8 frames using our recurrence relation where a bit sliced image will be embedded
Step 7: Find that LSB bit of the cover frames
Step 8: Embed those bit planes into that image frames and return those frames to their previous position
Step 9: Regenerate video frames

B. Algorithm for Extraction

Step 1: Input the stego video
Step 2: Extract frames from the video
Step 3: Find the frames by the specified key
Step 4: Find the LSB of the frames
Step 5: Extract the bits from that stego frame
Step 6: Merge the bits to form the final image

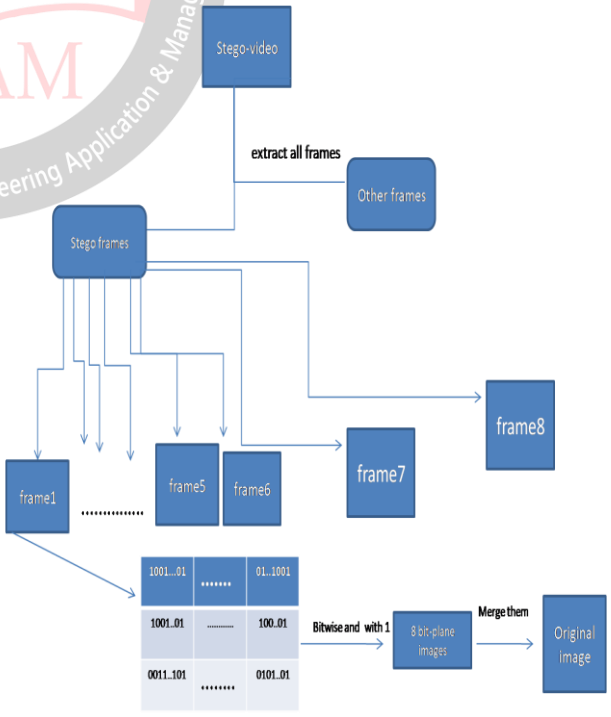

Figure 3 Block Diagram of Extraction Algorithm

 DOI : 10.35291/2454-9150.2021.0541

Figure 4 Generation of STEGO Frames

V. RESULTS AND DISCUSSION

The proposed method has been implemented in OpenCV-Python. A steganography technique is mainly characterized by its imperceptibility. The performance of the proposed method is evaluated using video streams (drop.avi) and the secret image (cameraman.tif) as in Figure 4. The perceptual imperceptibility of the embedded data is indicated by comparing the original video to its stego video so that their visual differences can be determined. An additional measurement MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) can be done to check the imperceptibility between the stego frame and corresponding cover frame.

$$MSE = \frac{1}{H*W}\sum_{i=0}^{H}\sum_{j=0}^{W}(P(i,j) - S(i,j))^2 \quad (4)$$

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

$$PSNR = 10log_{10}\frac{L^2}{MSE} \quad (5)$$

Where *PSNR* is the peak signal to noise ratio, *L* is the peak signal level for an image that is taken as 255.

Table 1 Image Parameters of the Cover Video

Video Name	Resolution	Frames /sec.	No. of Frames	Secret Image Size
drop.avi	240x256	30	182	512x512

Table 2 Average PSNR and MSE Values of Cover Frames for Different Secret Images

Cover Frame	Secret Image	Size of Secret Images	MSE	PSNR(dB)
240x256	cameraman	512x512	1.6238	46.0255
240x256	lenna	512x512	1.6166	46.0447
240x256	lenna	256x256	0.4041	52.0659
240x256	peppers	512x512	1.6364	45.0516

VI. CONCLUSIONS

In this paper, we have proposed security for an image using the concept of video steganography where the secret information is to be hidden in the randomly chosen unique video frames. We developed the LSB method to hide that image into multiple video frames which are generated by some pseudo-random number generator using eight bits of secret images divided into 8 bit-planes. We focused on the frame selection method, which is very unique and played an important role in LSB steganography or any other techniques for steganography. The LSB technique utilizes selected cover video files in the spatial domain to hide the presence of sensitive data regardless of its format. The proposed technique is applied to several types of files (.avi,.mpeg,.mp4). Table 2 shows that the values of *MSE* and *PSNR(dB)* are good enough and also better than Dasgupta et al [4] and Akbar et al [8]. As the cover frames are selected on a random basis in our proposed algorithm, it is very hard to get the secret image for the hackers.

Our proposed method can be applied in other Steganography techniques for better security.

REFERENCES

[1] K Panchal and Patel, F N Patel "Steganography : A brief survey", International Journal of Modern Trends in Engineering and Research. (2015); 2, 747-750.

[2] A Munasinghe, A Dharmaratne and K D Zoysa, "Video Steganography. International Conference on Advances in ICT for Emerging Regions", (2014).

[3] R Begum, and S Pradeep. Best Approach for LSB based video steganography using genetic Algorithm and visual Cryptography for secured data hiding and transmission over networks. International Journal of Advanced Research in

 DOI : 10.35291/2454-9150.2021.0541

Computer Science and Software Engineering. (2014); 4.

[4] K Dasgupta, J K Mandal and P Dutta, "Hash-based Least Significant Bit Technique for Video Steganography(HLSB)". International Journal of Security, Privacy and Trust Management. (2012); 1, 37-45.

[5] M Ramalingan, "Stego Machine Video Steganography using Modified LSB Algorithm", World Academy of Science, Engineering, and Technology International Journal of Information and Communication Engineering. (2011); 50, 170-173.

[6] F Pan et al, "Video first resize pseudo-random, steganography using the motion vector and linear block codes", Proceedings of The International Conference on Image Processing. (2002)

[7] R Jain, "Random Number Generation Washington University", Available: https://www.cse.wustl.edu/~jain/cse567-11

[8] S Akbar, N Rao, and T Anand, "Bit Plane Slicing Algorithm for data security using Fusion Technologies", International Journal of Recent Technology and Engineering (IJRTE). (2019); 7, 323-325

[9] R Jain and J Boaddh, "Advances in digital image steganography", International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH). (2016); 163-171.

 DOI : 10.35291/2454-9150.2021.0541