A Video Steganography approach with randomization algorithm using image and audio steganography

Dr. Geetanjali Kale
Department of Computer Engineering
Pune Insitute of Computer Technology
Pune, India
gvkale@pict.edu

Atharva Joshi*
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India
atharvaavjoshi11@gmail.com

Ishaan Shukla*
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India
ishaanshukla10@gmail.com

Abhishek Bhosale*
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India
abhisheksachinbhosale@gmail.com

***Abstract*—The practice of hiding data within other data such that it is not exposed to any entity other than the intended one is known as steganography. Video steganography is a form of steganography in which private information is embedded into digital video recordings. In the proposed method, the text information is hidden into video frames as well as the audio component of the video. The introduction of randomness in the selection of frames in which information is to be hidden, as well as the use of audio steganography to store additional information aims at boosting security of transmission of information. For additional security, the text is first preprocessed and hashed using the MD5 hashing algorithm before being hidden inside the video components.**

***Index Terms*—Video Steganography, Cybersecurity, Information Exchange, Data Hiding, Privacy, Information Security**

I. Introduction

The word steganography has roots in the Greek words “steganos” meaning “cover” or ”protected” and “grafia” meaning “writing”. It holds significant importance in today’s digital age, mainly due to the fact that there exists certain data which is confidential and secure transmission of such data is necessary. [1]. Information security issues have increased dramatically in the past few years, for example, piracy tracking, copyright infringement, digital watermarks, digital forensics and identity authentication [2]. Every year there is an increment of about 20+ percent in the total amount of data across the internet which is why data security and information hiding techniques need to evolve.

Steganography is generally confused with cryptography, its counterpart which also aims at increasing the confidentiality aspect from the CIA triad. In cryptography, the structure of the message to be passed is altered with the help of ciphers, hashes or other algorithms. However, in steganography, the message is transmitted unaltered but hidden within some cover media. Steganography, when used alone is not very secure, but when combined with cryptography, it could prove to be a good tool for covert communication.

A. *Video Steganography*

Video Steganography is the technique of concealing secret information within videos without affecting the visual or auditory quality of the video [3]. A video, which can be split into two components, a series of images also known as frames and audio signals provides a comprehensive medium that can accomodate large amount of data. This makes video a very robust cover object. Video steganography can used to embed almost all types of information, be it textual, audio/video, images,etc inside a video.

B. *Why Video Steganography?*

In comparison with other forms of steganography like image, audio, network, etc video stegaonography provides high capacity secured data transmission medium . The complexity with which a video is constructed in itself provides the necessary protection and hence proves video steganography a robust method of data hiding. Videos provide a good channel for data obfuscation and also the fact that video quality is hardware dependent makes minor variations in video quality acceptable to users over the internet.

II. Literature Review

Singh et al. [4] used one least significant bit to hide a secret image in video. As a measure to decrease its vulnerability towards steganalysis, the algorithm goes one step ahead. Each row of pixels consisting of 8 bits is hidden in the first rows of multiple frames of the host. Hence, every 1 byte of image message needs 8 bytes from the host video frame(s). The algorithm is easy to implement. Because of the use of one bit LSB mechanism, the video distortion is minimal. However it also suffers from capacity problem, that is, the data which can be stored is comparatively lesser than other algorithms. Eltahir et al. [5] made use of HVS features to modify the traditional LSB method. They utilized the fact that the human eye is more sensitive to the change in the blue level color compared to the red level and the green level colors. For hiding the data, they used the 3-3-2 approach, meaning that, 3 least significant bits from the red color and 3 from the green color were manipulated, but from the blue color, only 2 bits were modified. Their algorithm used 1/3rd of every video frame for steganography. However, the algorithm isn't tamper proof or robust. Alaa [6] made use of LSB steganography to hide information in pixels of video frames obtained after splitting the video. To choose the pixel numbers in which information is to be hidden, she used the Stella Octangula number sequence and the data was hidden using the 3-3-2 approach. For additional security, the 256-AES encryption was used for text preprocessing. However, for larger size of texts used, the pixel corresponding to the Stella Octangula number could become very large. Also as the size of text increases, it results in a slight degradation in similarity measures.

III. Methodology

Our project works in 2 phases:

A. Encoding

In this phase, we take the piece of text to be hidden (stego text) and video in which the text is to be hidden (cover video) as inputs. The video is now split into two components, namely, video frames which are a series of images in .jpg/.png format, and the audio file which is in .wav format.

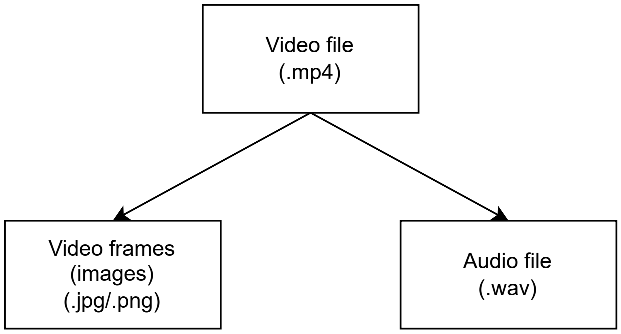

Fig. 1. Splitting of video

The video is split into corresponding frames depending on its fps (frames per second). We also split the text into 5 parts. Two cases arise during splitting of input text: One in which the length of text is divisible by 4 and the other in which the length of text is not divisible by 4. In the first case, the text is split into 4 equal parts and the 5th part is just an empty string. In the other case, the closest number divisible by 4 but less than the length of text would be found and the text would be split such that the first 4 parts are of equal length and the remainder (5th part) would be the remaining substring. Now we run our randomisation algorithm, the Fisher-Yates shuffling algorithm [7]. In this algorithm, an array of numbers is taken as an input. This array is now rearranged according to the seed provided (which is also generated randomly in this case). Then we choose first 5 indexes from this shuffled array which would correspond to the frame numbers in which we will encode the 5 parts of the text. Now one by one we run our image steganography algorithm five times to hide the data within corresponding frames chosen for them [8].

1) Image Steganography: We hide the data in the video frames using the 2-bit LSB steganography mechanism. For each of the 5 parts of the texts obtained, we use the MD5 hashing algorithm for additional security. Image steganography algorithm is now implemented on this hashed text. For this purpose, we use the 2-2-2-2 split strategy. This means that, the data is hidden only in the R and G components from the RGB color model. Every character of the input text is first represented in its ASCII form (integer). Then this integer value (which ranges from 0 to 255 as per IBM standard) is converted into its corresponding 8-bit binary form. Then these 8 bits are split into 4 pairs of 2 bits each. The final result is that we obtain these 4 pairs for every character in the MD5 hashed text [9]. Every component of every pixel in the image is also represented in the form of an 8-bit binary value. Now every 2-bit pair is hidden in the image by replacing the least significant 2 bits from the red and green component values. For every i^{th} character in the text, we require two pixels, namely $(2i - 1)^{th}$ and $(2i)^{th}$ pixels.

2) Audio Steganography: The information about the random frames which were chosen during the image steganography phase also need to be provided to the receiver for decoding. For this purpose, we use the extracted audio file. The audio signal is transformed into a vector representation and then converted into a 2-dimensional matrix. This 2-D matrix is then represented as a product of orthogonal and upper triangular matrices using QR decomposition. The length of the message containing the frame locations is embedded in the first 15 samples and actual message is embedded starting from the 16th frame. Then inverse QR decomposition is applied and final audio file is reconstructed with the hidden message [10].

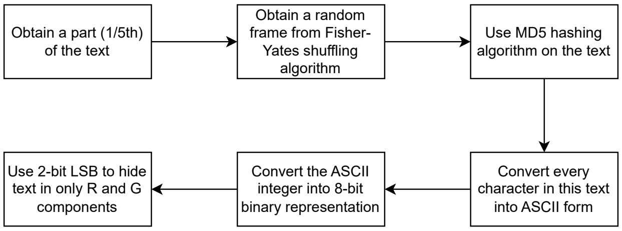
Obtain a part (1/5th) of the text
Obtain a random frame from Fisher-Yates shuffling algorithm
Use MD5 hashing algorithm on the text
Convert every character in this text into ASCII form
Convert the ASCII integer into 8-bit binary representation
Use 2-bit LSB to hide text in only R and G components

Fig. 2. Encoding- Image Steganography

After both of these steps are performed, an output video is formed by stitching back reconstructed video frames and the reconstructed audio file obtained by the image and audio steganography respectively.

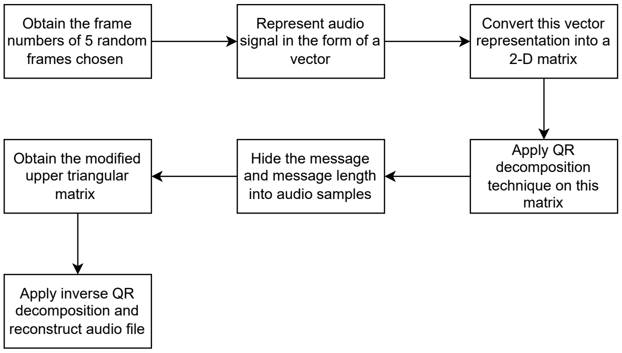
Obtain the frame numbers of 5 random frames chosen
Represent audio signal in the form of a vector
Convert this vector representation into a 2-D matrix
Apply QR decomposition technique on this matrix
Hide the message and message length into audio samples
Obtain the modified upper triangular matrix
Apply inverse QR decomposition and reconstruct audio file

Fig. 3. Encoding- Audio Steganography

B. Decoding

The video that is obtained as a result of the encoding process is split again into its two constituents: Series of Images and audio signal. The first step in decoding is to find the exact index of the image frames inside which actual input text was hidden. This is done using QR decomposition of the extracted audio signal. After this, the information of the 5 frames is obtained and the image steganography technique is used on those exact frames for decoding the message similar to the encoding mechanism. For this purpose, the images are represented according to intensity values of the RGB value. Then the integer values are converted to 8-bit binary values and then the last 2-bits of every pixel are taken according to the length of the hashed text. Because MD5 hash produces a series of alphanumerical characters of length 32 characters, the decoding extracts last 2-bits from first 64 pixels only. Doing so, 5 pieces of text are obtained. These pieces are not in their raw form, but in MD5 hashed form. Every piece is then decrypted using MD5 hash decoding to obtain the pieces as they were from original message. Finally these pieces are just concatenated to obtain the original message which was supposed to be transmitted.

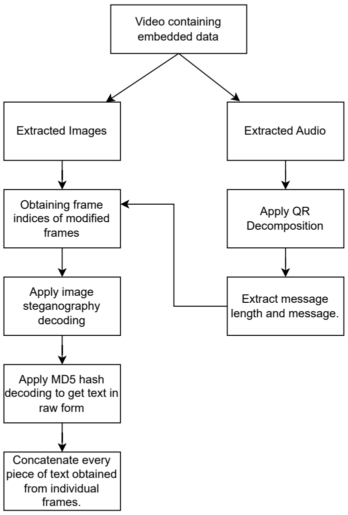
Video containing embedded data
Extracted Images
Extracted Audio
Obtaining frame indices of modified frames
Apply QR Decomposition
Apply image steganography decoding
Extract message length and message.
Apply MD5 hash decoding to get text in raw form
Concatenate every piece of text obtained from individual frames.

Fig. 4. Decoding Phase

IV. Experimentation and Results

A. Similarity Measures

Fig. 5. test-1 original frames

Fig. 6. test-1 reconstructed frames

Fig. 7. test-2 original frames

Fig. 8. test-2 reconstructed frames

Fig. 9. test-3 original frames

Fig. 10. test-3 reconstructed frames

1) For images: Physically we can see that there is no visible difference between the images. Mathematically, this can be confirmed by looking at similarity measures like Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE).

Mean Squared Error is the average of squared deviations of the predictions of an experiment from the true value. Moreover, it is a measure that can be used in the steganography mechanism to draw out the minute differences that exist between the stego image as the end product and the cover image [11]. The mathematical formula for the MSE can be expressed as follows:

$$\mathrm{MSE}_{\,(\mathrm{x,y})} = \frac{1}{N}\sum\nolimits_{i=1}^{N}(x_{\mathrm{i}} - y_{\mathrm{i}})^2$$

The invisibility of the hidden text can be measured with the help of PSNR. To analyze the quality of the stego image with respect to the original one, PSNR is helpful [12]. PSNR can be computed with the help of the following mathematical expression:

$$\text{PSNR} = 10\log_{10} \frac{L^2}{MSE}$$

The calculations for PSNR and MSE are as in the following table:

Video name	Frame	PSNR	MSE
test-1	1	82.0205014	0.0004083478009
test-1	2	81.0923691	0.0005056423611
test-1	3	81.43481568	0.0004673032407
test-1	4	81.42474313	0.0004683883102
test-1	5	91.40800729	0.0004701967593
test-2	1	84.86023782	0.0002123521091
test-2	2	84.6516337	0.0002228009259
test-2	3	85.10016587	0.000200938786
test-2	4	85.43593241	0.0001859889403
test-2	5	94.54638255	0.0002282664609
test-5	1	81.25054189	0.0004875578704
test-5	2	81.09858663	0.0005049189815
test-5	3	81.38136338	0.0004730902778
test-5	4	81.42809805	0.0004680266204
test-5	5	90.58844429	0.0005678530093

2) Audio: In signal processing, cross-correlation is a measure of similarity of two series as a function of the displacement of one relative to the other [13]. Following are the similarity percentages between every pair of audio file before and after encoding the data.

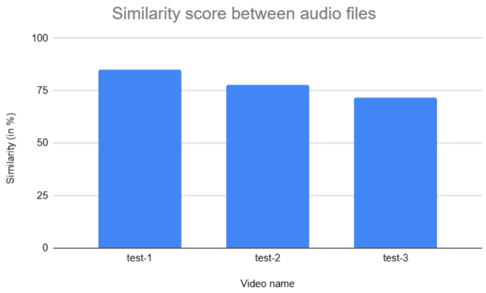

Fig. 11. Similarity score of audio before and after steganography

Spectrograms for audio extracted from individual videos are as follows:

B. Comparison

Following is the tabular representation of the time required for the entire encoding process as a function of the size of the input video file in megabytes:

Video Name	Video Size (in mb)	Time (in seconds)
test-1.mp4	1	72.25831783
test-3.mp4	5.6	281.8233782
test-2.mp4	7.18	249.9797471

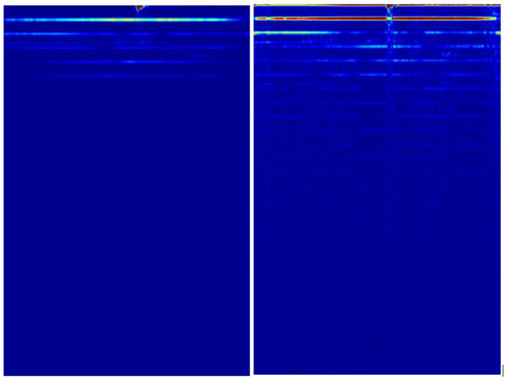

Fig. 12. test-1 before and after

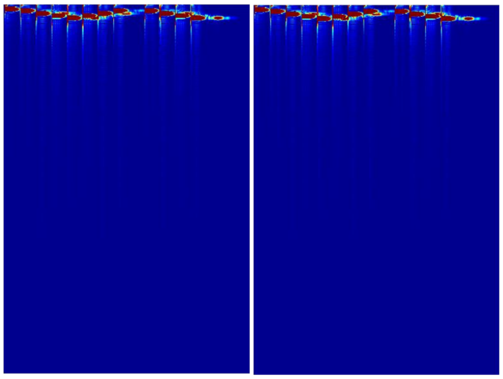

Fig. 13. test-2 before and after

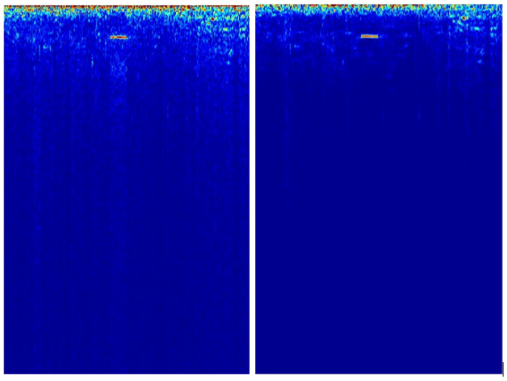

Fig. 14. test-3 before and after

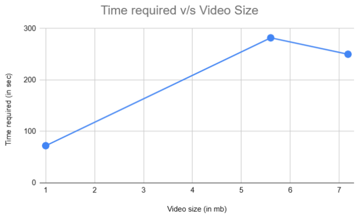

Fig. 15. Time required v/s Video Size

V. Conclusion

The proposed work provides a robust and comprehensive approach to hide textual data within videos. The concept of embedding certain part of data inside the video frames and some part within audio files makes it difficult for intruders who eavesdrop on the transmitted video file to crack the embedded message. Also the use of randomization algorithm and MD5 hash encoding adds an additional layer of security to the data. Future work will focus on using Generative Adversarial Neural Networks(GANs) for implementing generative steganography.

References

[1] ALabaichi, Ashwak, Maisa'A. Abid Ali K. Al-Dabbas, and Adnan Salih. ”Image steganography using least significant bit and secret map techniques.” International journal of electrical & computer engineering (2088-8708) 10.1 (2020).

[2] Mstafa, Ramadhan J., et al. ”A new video steganography scheme based on Shi-Tomasi corner detector.” IEEE Access 8 (2020): 161825-161837.

[3] Sadek, Mennatallah M., Amal S. Khalifa, and Mostafa GM Mostafa. ”Video steganography: a comprehensive review.” Multimedia tools and applications 74 (2015): 7063-7094.

[4] Singh, Saurabh, and Gaurav Agarwal. ”Hiding image to video: A new approach of LSB replacement.” International Journal of Engineering Science and Technology 2.12 (2010)

[5] Eltahir ME, Kiah LM, Zaidan BB, Zaidan AA (2009) High rate video streaming steganography. In: International Conference on Future Computer and Communication (ICFCC 2009)

[6] Alaa, Aly. ”Digital Rights Management and LSB Embedding Using Stella Octangula Number Sequence.” (2023).

[7] Ade-Ibijola, Abejide Olu. ”A simulated enhancement of Fisher-Yates algorithm for shuffling in virtual card games using domain-specific data structures.” International Journal of Computer Applications 54.11 (2012).

[8] Tarun, M. V. S., et al. ”Digital video steganography using LSB technique.” Red 100111.11101000 (2020): 11001001.

[9] Ali, Ammar Mohammed, and Alaa Kadhim Farhan. ”A novel improvement with an effective expansion to enhance the MD5 hash function for verification of a secure E-document.” IEEE Access 8 (2020): 80290-80304.

[10] Bilal, Ifra, and Rajiv Kumar. ”Audio steganography using QR decomposition and fast Fourier transform.” Indian Journal of Science and Technology 8.34 (2015): 1-7.

[11] Wang, Zhou, and Alan C. Bovik. ”Mean squared error: Love it or leave it? A new look at signal fidelity measures.” IEEE signal processing magazine 26.1 (2009): 98-117.

[12] Carvajal-Gamez , B.E., Gallegos-Funes, F. J. and Lopez-Bonilla, J. L. (2009), “ Scaling Factor for RGB Images to Steganography Applications”, Journal of Vectorial Relativity, Vol. 4, no. 3, pp.55-65.

[13] Yoo, Jae-Chern, and Tae Hee Han. ”Fast normalized cross-correlation.” Circuits, systems and signal processing 28 (2009): 819-843.