Video Steganography

"Hiding in plain sight"

Rahul Maheshwari[1], Nikunj Agarwal[2], P Akshay Kumar[3], Piyush Pradeep Jain[4], Abhishek Singh Chauhan[5]
Department of Computer Science & Engineering,
Indraprastha Institute of Information Technology Delhi

Motivation

In today's era, Information Security is one of the major areas of concern, mostly because of the increasing number of users communicating all over the world. Due to the very same reason, the volume of data getting generated and exchanged is massive. The data is at the risk of being exposed to unauthorized personnel which can lead to malicious use of information and problems such as identity theft, financial frauds, etc. Steganography helps to exchange sensitive information by hiding in plain sight which makes it less prone to attacks. To implement a non-traditional method of encrypting the sensitive information, we chose video steganography as videos are the mainstream media available to the mass public and are much practical today. An increasing amount of data requires video as the base for encryption instead of traditional methods of using images for steganography.

Objective

The aim is to learn and appreciate the steganography method of hiding data and implement the algorithm that helps send data over any channel. This aligns with the various encryption techniques discussed in the class and we would like to deepen our understanding of cryptographic systems by implementing this specific technique. We aim to use video as well as audio as the means of transmitting encrypted data.

Task Performed

1. Separation of video into frames and audio for encryption.
2. Embedding of audio with information about frames which contain the encrypted concealed data.
3. Data is segmented into blocks and encrypted in frames using LSB encryption algorithm.
4. Compression is performed on the output encrypted video.
5. Decryption is performed to extract the original plain text information sent.
6. GUI (Graphical User Interface) using flask which integrates and binds all the above mentioned modules.

Sample video information

1. Video used for performing the above mentioned tasks is of length 35 secs containing a total of 842 frames.
2. Video is in .mp4 format

Pre-processing

- Extraction of frames from the video for encryption.
- Extracting audio from the video for layer 1 encryption.
- Tokenizing the plain text message into length of 8 to be embedded in frames.

Audio Steganography

Audio of the video file is used to encrypt the frame number which contains the encrypted data.

Four methods for audio steganography were implemented -

- Least significant bit method - In this method the data is encoded in the LSB bit of the audio data. First n frames of the audio data are used to encode n bit of data. Problem with this approach is that the attacker can easily extract the data.
- Parity based technique - In this method the data is encoded by breaking the audio data into n equal size sample. And data is encoded by calculating the parity of the data. It also has the problem similar to the LSB based approach as if attacker know the size of data encoded data can be extracted by breaking the data into small samples and calculating the parity value.
- Random method - This method was created by us to solve the problem of the above stated methods. For this the number of frames are encoded into the first sixteen bytes of the data. And audio frames which are used for encoding are generated are using the secret key. Only problem with this approach is that the number of frame which is encoded using LSB can be extracted.
- Complete random method - For this method secret key is used to select 16 audio frames which contains the number of frames. And using the frame number and secret key the frame used for encoding the final data is selected. This method works well because only the sender and receiver know the secret key.

Video Steganography

- LSB technique - Each word in the text is converted to binary form and the LSB of each byte is set to match the corresponding binary bit of the word.

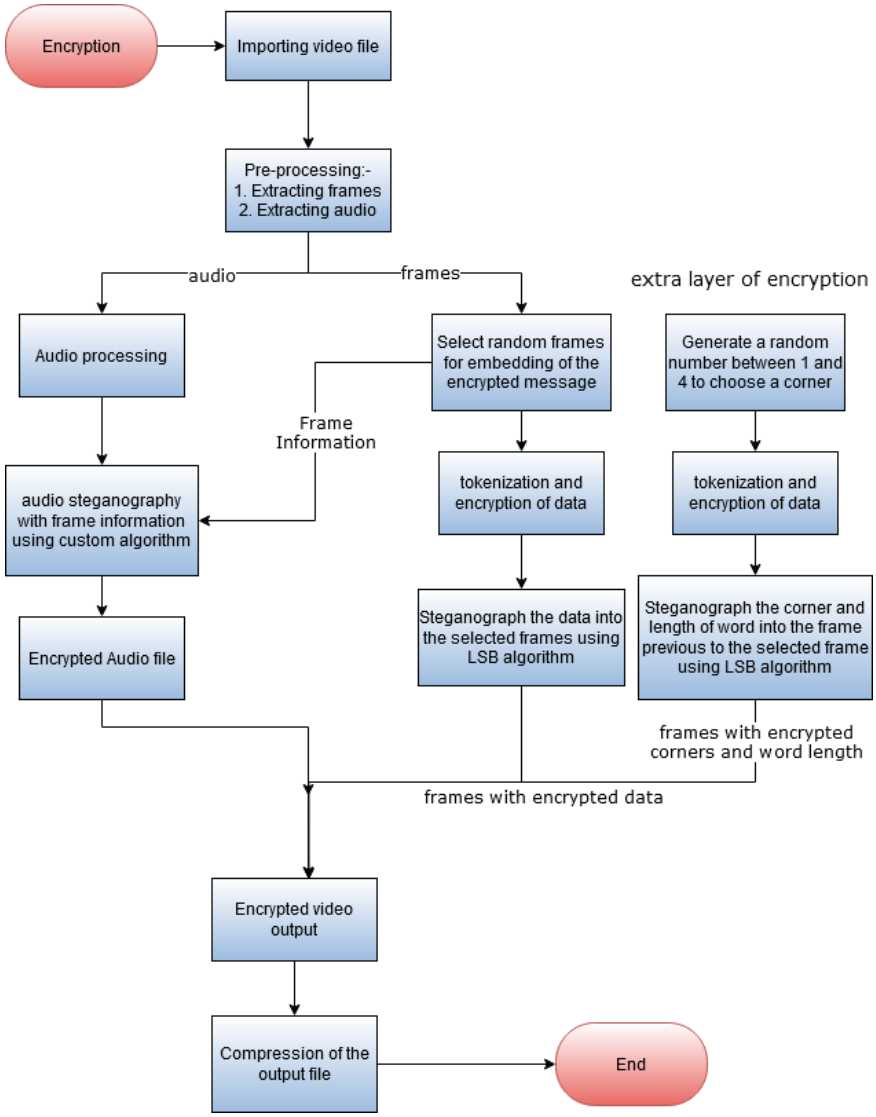


Figure 1: Flowchart for Encryption

- Encryption:
    1. Randomly generated corner number between 1 and 4 to decide at which corner the data will be encrypted. [1-Top left, 2-Top right, 3-Bottom left, 4-Bottom right]
    2. The corner information and word length are encrypted using LSB technique in the first few bytes of the frame previous to the selected frame.
    3. Each word is encrypted in the earlier randomly chosen corner using LSB technique.
    4. The frames are then encrypted into the audio using the complete random technique mentioned in the previous slide.

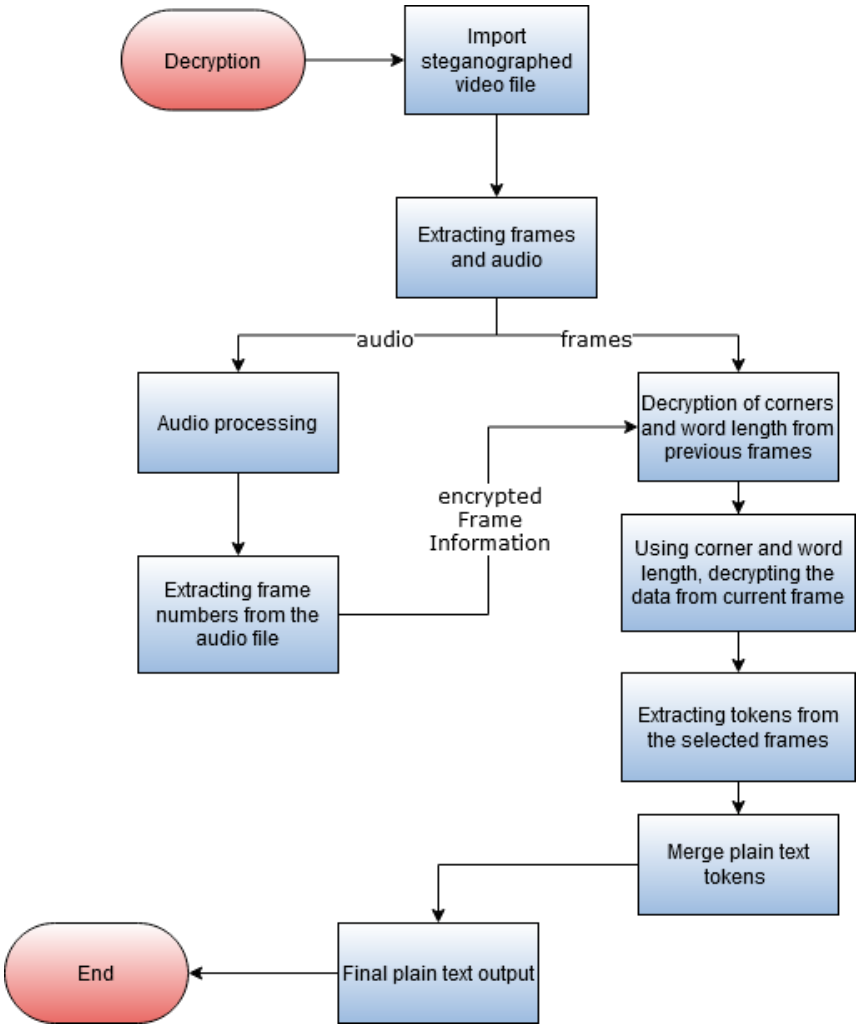


Figure 2: Flowchart for Decryption

- Decryption:
    1. The frames selected for encryption are decrypted from the audio.
    2. Firstly, the frame previous to the selected frame is used to extract the corner and word length of the word encrypted in the current frame.
    3. Using the corner and word length information, the location of word is identified and the message is decrypted by extracting the LSB of bytes from the location.
    4. The obtained binary is then converted to character which gives the text given as input by the user.

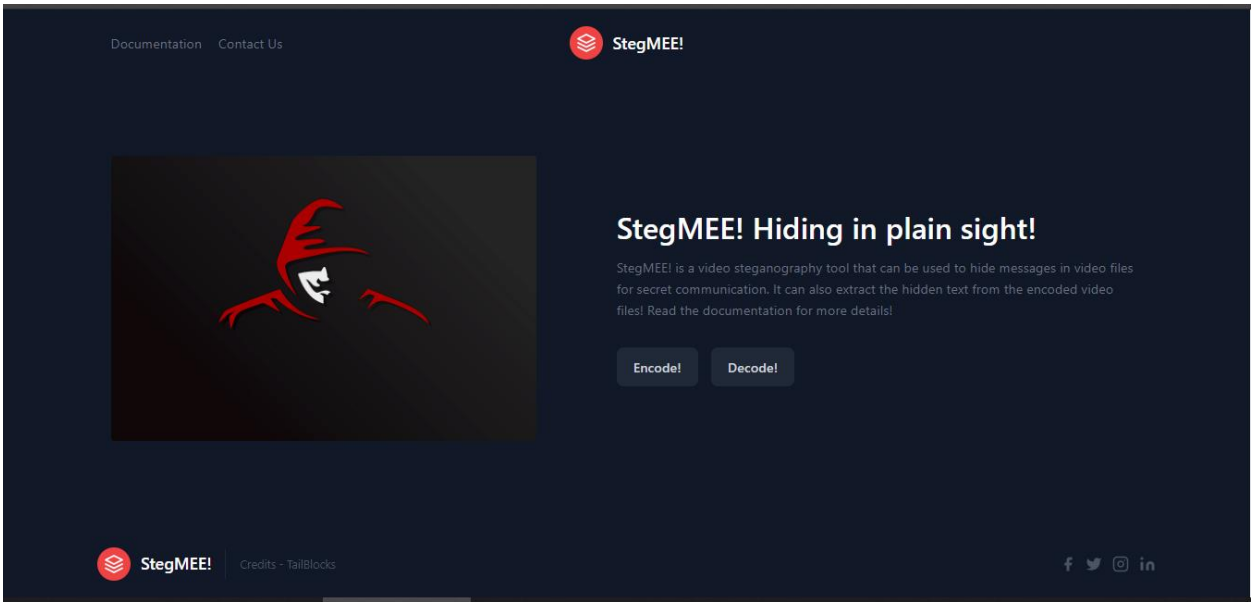
Additional Task

- Compression is performed on the encrypted output video file to reduce the size in order to optimize the process and reduce the bandwidth required for sending the file to someone.

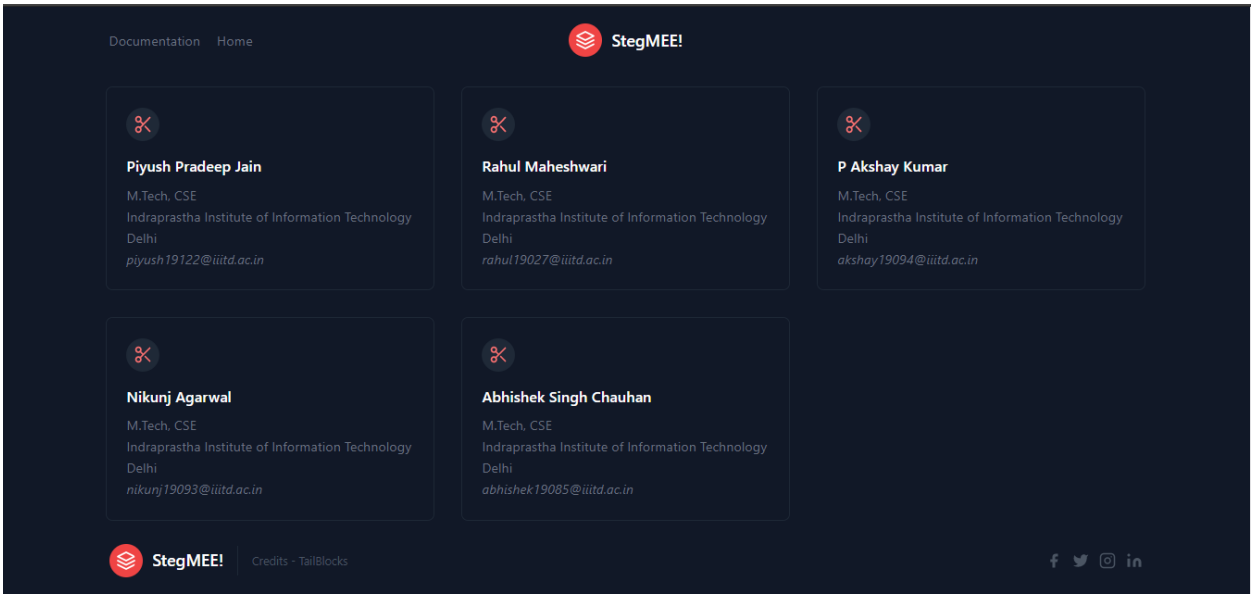
- Compression involved usage of ffv1 intra frame lossless codec and conversion of output file to mkv format.
- Compression reduced the size upto 3.5x-4x.

Demonstration

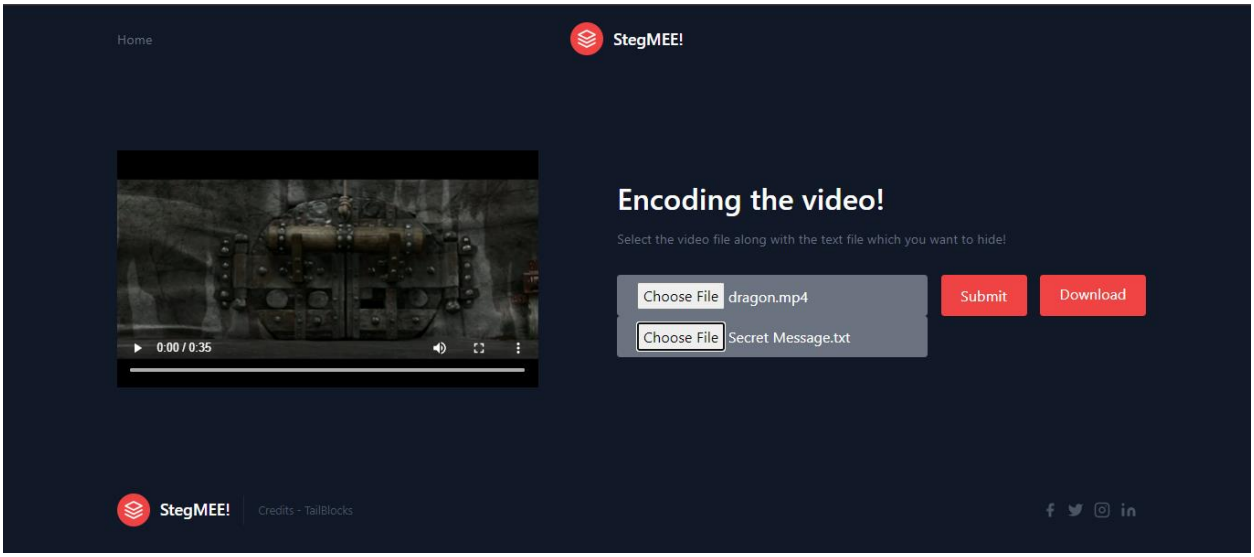
1) Open the link (http://127.0.0.1:5000/). Homepage appears.



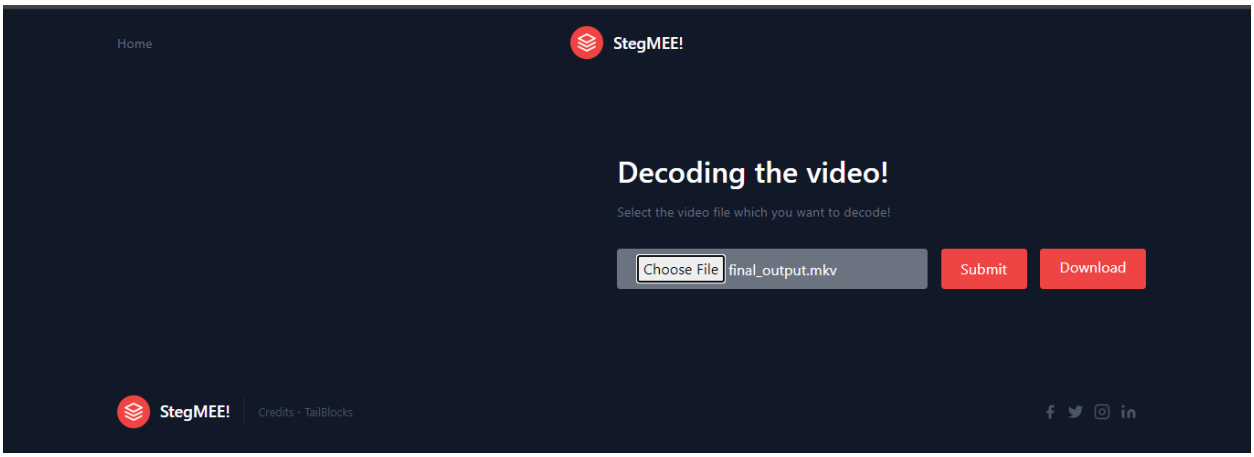
2) To view details of the web app, click on **Documentation**
3) To contact the authors, click on **Contact Us**



4) To perform encryption, click on **Encode!** button on homepage. On the landing webpage, upload a video and a text file that you want to hide. Click on **Submit** to start encoding.



5) Once done, click on **Download** to download the encoded video file.
6) To perform decryption, click on **Decode!** button on homepage. On the landing webpage, upload the encoded video. Click on **Submit** to start decoding.



7) Once done, click on **Download** button to download the decrypted plaintext.

References:

- Reddy, V. Lokeswara, A. Subramanyam, and P. Chenna Reddy. "Implementation of LSB steganography and its evaluation for various file formats." Int. J. Advanced Networking and Applications 2, no. 05 (2011): 868-872.
- Yadav, Pooja, Nishchol Mishra, and Sanjeev Sharma. "A secure video steganography with encryption based on LSB technique." In 2013 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-5. IEEE, 2013.
- https://mertjf.github.io/tailblocks/

- https://tailwindcss.com/
- https://sumit-arora.medium.com/audio-steganography-the-art-of-hiding-secrets-within-earshot-part-2-of-2-c76b1be719b3
- https://www.geeksforgeeks.org/image-based-steganography-using-python/