Indian Institute of Technology

Bhubaneswar

Security & Forensics Lab II
Experiment No.2

Submitted by :
Satendra
21CS06012
M.Tech(CSE)

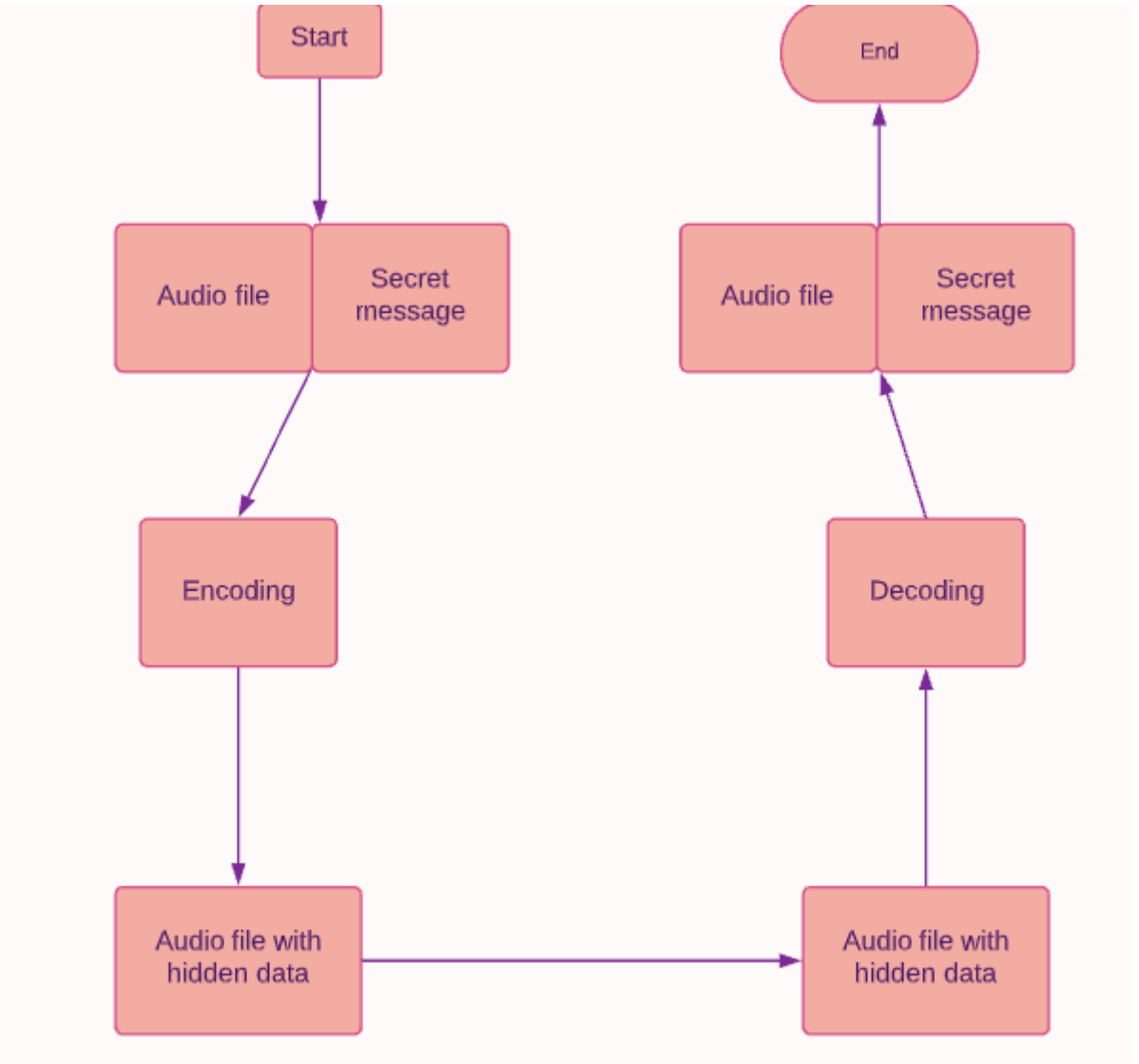
The aim of the experiment is :

Design and implement an efficient steganography algorithm that will hide and retrieve a secret text message inside a cover audio file (.wav format). Any kind of substitution is allowed to hide the secret message within the cover audio. Your objective should be to intelligently select the portions of the audio such that the original voice does not get distorted (or try to minimize the distortion). The quality of the audio after the message embedding will be checked manually by listening to the modified audio. However, the input secret message will vary in length. Thus, the quality degradation will be checked with small as well as large messages. The secret text message will be given by the user during execution. Successful extraction of the message is also needed.

Working:

1. Import all necessary library –wav for audio format
2. Read audio file
3. Convert audio file to byte array.
4. Enter the secret text message
5. Convert text to binary
6. Replace the LSB of each byte of the audio file by one bit from the text bit array
7. Write modified bytes to new wav audio file
8. Open a new audio file
9. Convert audio to byte array
10. Divide strings into blocks of eight binary strings convert them and join them back to a string.
11. Return the extracted text.

Flow chart :


Start
Audio file
Secret message
Encoding
Audio file with hidden data
Audio file with hidden data
Decoding
Audio file
Secret message
End

Algorithm

Encoding:

We have been given a Cover_audio.wav file and we want to hide a secret message("Satendra kaurav(iitbbs)") in it.

1. we will be performing LSB bit manipulation on the carrier Audio file. We perform AND operation between each byte of cover audio and a bitmask that resets the LSB of the cover byte.

Example

Suppose the audio frame is $A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1\ A_0$ and the bit mask is 11111110 (254) so we simply perform AND operation between them.

$A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1\ A_0$ & 11111110 = ($A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1$ 0)(modified bytes**)**

2. Then we will perform a simple logical OR operation between the modified carrier byte and the next bit (0 or 1) from the secret message.

modified bytes is ($A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1$ 0) and then we convert the secret message into binary and perform OR operation. Secret bits(0/1)

$A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1$ 0 | 0000000(0/1) = $A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1$ (0/1) this is encoded file with hidden data.

Decoding :

1. convert encoded audio to byte array then I extract the LSB of each byte.

2. append to result string.

3. return the extracted text.

Output :

```
print("Enter the secret message")
text=input()
print("audio file path")
file_name=input();
Encoding_Audio(text,file_name)
print("Secret message is : ",end=' ')
print(Decoding())
```

```
Enter the secret message
satendra kaurav(iitbbs)
audio file path
/content/cover_audio.wav
Secret message is :  satendra kaurav(iitbbs)
```

Code:

https://colab.research.google.com/drive/1IGA_MbDNiR7Eo7NfMsT5RMZxW5oQpBwk#scrollTo=jbGq233sVglj