

**Project Report**

**Audio steganography**

Digital Signal Processing

**Submitted By:**

* Muhammad Hamza (173088)
* Abdul Hadi Zahid (174442)

**Submitted To:**

* Sir Kaleem Ullah

**Dated:**

17/05/2019

**Abstract:**

In this report we are going to talk about confidential information hiding with the help of information and data security solutions expert. Reasons to hide information but most common reason is to protect it from unauthorized access and people coming to know the existence of the secret information. In steganography information is hidden in cover file such that only dispatcher and recipient know the existence of the confidential information. In order to achieve this purpose, we have made a MATLAB program which allows us to hide any secret text information into any audio file.

**Introduction:**

In the corporate world audio data hiding can be used to hide and secure confidential chemical formulas or designs of new invention. Not only in commercial sector, it can also be used in used in the non-commercial sector to hide data that someone wants to keep private.

Steganography is the science of hiding confidential information in a cover file so that only dispatcher and recipient know the existence of the confidential information. Confidential information is encoded in a way so that the very existence of the information is hidden. The main objective of steganography is to converse securely in a completely invisible mode and to shun drawing doubt to the communication of hidden information. Steganography not only prevents people from knowing about the hidden confidential information, but it also prevents others from thinking that somebody is communicating in a hidden way. If a steganography technique causes someone to believe there is hidden information in a carrier medium, then the technique has botched. The essential technique of audio steganography consists of Carrier (Audio file), Message and Password. Carrier file is also called as a cover-file, which hides the confidential information. Essentially, in steganography message is the information that the sender desires that it should remain confidential. Message can be image, plain text, audio or any type of file. Password is also called as a stego-key, and it assures that only the receiver who knows the password will be able to extract the confidential information or message from a cover-file. The cover-file with the confidential information is called as a stego-file.

**Software Requirements:**

* MATLAB

**Working:**

First, we have selected a .wav file as our cover file in which we want to encode our data. We read the file using file handling technique on MATLAB and made 16-bit fragments of our file. Then we load another file .txt file which we want to hide and covert text into binary bits. One thing which we should keep in mind is that, the size of cover file should always be greater than our (.txt) file. Then we will enter a key which works as a random sequence generator according to which .txt file binary bits will override the least significant bit of each fragment of our cover file. Since our data is only covering the least significant bits of each 16-bit fragment of cover file so there will be not much impact on the intelligence of the audio. It will seem like nothing happened to the audio file (Recover- file).

That’s how the program works and embed our confidential data into the audio file. Next in order to decode the data from segno-file we have to enter the same key (stego-key) to extract the exact text file from stego- file. Incase if someone enters the wrong key, they will only get a garbage value.

In addition to that we designed a MATLAB app for smooth and user-friendly interfacing of our project, which is shown below. It has various options of adding files and entering keys plus it indicates whether process was successful or not. We have also included options for plotting and playing of wav files.

**MATLAB code:**

**For text embedding in cover file:**

function [x] = textEmbedding(key\_char, filename\_org, pathname\_org, filename\_encode, pathname\_encode, filename\_text\_org, pathname\_text\_org)

x = 1;

if(strcmp(key\_char, ' ') || strcmp(filename\_org ,' ') || strcmp(pathname\_org ,' ') || strcmp(filename\_encode, ' ') || strcmp(pathname\_encode ,' ') || strcmp(filename\_text\_org, ' ') || strcmp(pathname\_text\_org,' '))

x=0;

return;

end

%open a wav file for hidding text

audio\_org=fopen([pathname\_org filename\_org],'r');

header=fread(audio\_org,40,'uint8=>char');

data\_size=fread(audio\_org,1,'uint32');

data=fread(audio\_org,inf,'uint16') ;

fclose(audio\_org);

lsb=1;

text\_org=fopen([pathname\_text\_org filename\_text\_org],'r');

msg=fread(text\_org,inf,'uint8');

fclose(text\_org);

msg\_bin=de2bi(msg',8); %then convert message to binary

[m,n]=size(msg\_bin); %size of message binary

msg\_bin\_re=reshape(msg\_bin,m\*n,1); %reshape the message binary in a column vector

msg\_length=length(msg\_bin\_re); %length of message binary

if msg\_length > length(data)

x=0;

return;

else

m\_bin=de2bi(m,32)';

n\_bin=de2bi(n,32)';

key\_decimal = double(key\_char);

identity=0;

for i = 1:length(key\_decimal)

identity = identity+key\_decimal(i);

end

rng(identity, 'twister');

randomizer\_array = randperm(length(data)-64,msg\_length);

length\_random = length(randomizer\_array);

%hide binary length of message in the last 64 bits of the data

data(length(data)-31:length(data))=bitset(data(length(data)-31:length(data)),lsb,n\_bin(1:32));

data(length(data)-63:length(data)-32)=bitset(data(length(data)-63:length(data)-32),lsb,m\_bin(1:32));

data(randomizer\_array(1:length\_random))=bitset(data(randomizer\_array(1:length\_random)),lsb,msg\_bin(1:msg\_length)');

%open a new wav file in write mode and copy the original header and text data

audio\_emb=fopen([pathname\_encode filename\_encode],'w');

fwrite(audio\_emb,header,'uint8');

fwrite(audio\_emb,data\_size,'uint32');

fwrite(audio\_emb,data,'uint16');

fclose(audio\_emb);

end

end

**for text recovering:**

function [x] = textRecover(key\_decrypt,filename\_encode, pathname\_encode, filename\_text\_recover, pathname\_text\_recover)

x = 1;

if(strcmp(key\_decrypt,' ') || strcmp(filename\_encode,' ') || strcmp(pathname\_encode, ' ') || strcmp(filename\_text\_recover, ' ') || strcmp(pathname\_text\_recover, ' '))

x=0;

return;

end

%open the file with hidden text

audio\_emb=fopen([pathname\_encode filename\_encode],'r');

header=fread(audio\_emb,40,'uint8=>char');

data\_size=fread(audio\_emb,1,'uint32');

data=fread(audio\_emb,inf,'uint16');

fclose(audio\_emb);

lsb=1;

%extract the length of text

m\_bin=zeros(32,1);

n\_bin=zeros(32,1);

n\_bin(1:32)=bitget(data(length(data)-31:length(data)),lsb);

m\_bin(1:32)=bitget(data(length(data)-63:length(data)-32),lsb);

msg\_length=bi2de(m\_bin')\*bi2de(n\_bin');

msg\_bin=zeros(msg\_length,1);

key\_decimal = double(key\_decrypt);

identity=0;

for i = 1:length(key\_decimal)

identity = identity+key\_decimal(i);

end

rng(identity, 'twister');

randomizer\_array = randperm(length(data)-64,msg\_length);

length\_random = length(randomizer\_array);

%extract the lsb from wave data sample

msg\_bin(1:msg\_length)=bitget(data(randomizer\_array(1:length\_random)),lsb);

msg\_bin\_re=reshape(msg\_bin,msg\_length/8,8);

msg\_dec=bi2de(msg\_bin\_re); %convert it to decimal

msg=char(msg\_dec)'; %convert to char(ASCII)

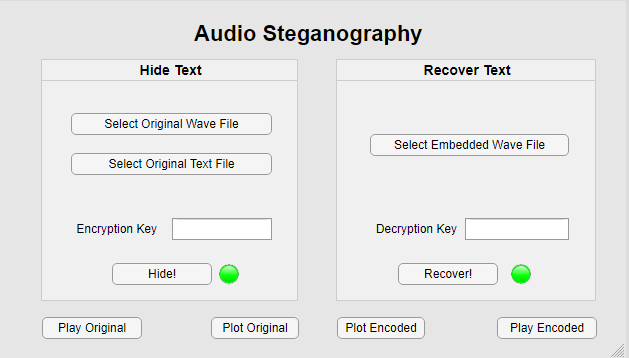
text\_rec=fopen([pathname\_text\_recover filename\_text\_recover],'w');

fprintf(text\_rec,msg,'uint8');

fclose(text\_rec);

end

**Program Interface:**



**Conclusion:**

We have made the technique of audio steganography using MATLAB. We loaded a cover audio file using file handling and encoded the text data bit wise, so that no one except the knows the key can decode it. We have used many MATLAB functions such as file handling, app designing to further enhance the efficiency and interfacing of our project. We can further enhance the working of this project by introducing other types of files as stego-file and improve the versatility of our project.