



# VIT<sup>®</sup>

**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## SIGNAL ANALYSIS AND PROCESSING

### **ENCRYPTION AND DECRYPTION OF AUDIO** **SIGNALS USING MATLAB**

#### **GROUP MEMBERS:**

19BEC0501 – Anubhav Singh

19BEC0461 – Aditya Kakkar

19BEC0463 – Utkarsh Sinha

# INTRODUCTION

In this 21st century, everyone is concerned about privacy and people are willing to pay any price to make sure their privacy is not breached. WhatsApp, Facebook, Twitter, Instagram and many other leading social networking sites which have access to millions and millions of data have invested a lot of money to protect the privacy of its users. They basically have a Multi-layered Encryption-Decryption algorithm which protect any interception between the transfer of information. Deep Neural Networks and Machine Learning algorithms have also been put into action to prevent any hack or interception. So, for that, we have tried to make a dummy model of the encryption and decryption using some simple logics.

# METHODOLOGY

In this project we are going to take a real time audio signal as the input for the project. We then use the Data Encryption Standard Algorithm to encrypt the same audio signal. We then ask the user to enter a password for encryption. The signal is then encrypted. Then the user is required to enter the decryption password. If the encryption and decryption password match, only then the signal is decrypted and the original recorded signal is plotted. Every signal is plotted against time.

# MATLAB Code

## Decryption:

```
function data_decrypt = decryption(r_encrypt, key2, r_length)
```

```
global qw stp dse dr pt A
```

```
qw = ...  
    [58 50 42 34 26 18 10 2 ...  
      60 52 44 36 28 20 12 4 ...  
      62 54 46 38 30 22 14 6 ...  
      64 56 48 40 32 24 16 8 ...  
      57 49 41 33 25 17 9 1 ...  
      59 51 43 35 27 19 11 3 ...  
      61 53 45 37 29 21 13 5 ...  
      63 55 47 39 31 23 15 7];
```

```
stp = ...  
    [40 8 48 16 56 24 64 32 ...  
      39 7 47 15 55 23 63 31 ...  
      38 6 46 14 54 22 62 30 ...  
      37 5 45 13 53 21 61 29 ...  
      36 4 44 12 52 20 60 28 ...  
      35 3 43 11 51 19 59 27 ...  
      34 2 42 10 50 18 58 26 ...  
      33 1 41 9 49 17 57 25];
```

```
dse = ...  
    [32 1 2 3 4 5 4 5 ...  
      6 7 8 9 8 9 10 11 ...  
      12 13 12 13 14 15 16 17 ...  
      16 17 18 19 20 21 20 21 ...  
      22 23 24 25 24 25 26 27 ...  
      28 29 28 29 30 31 32 1];
```

```
dr = ...  
    [57 49 41 33 25 17 9 1 ...  
      58 50 42 34 26 18 10 2 ...  
      59 51 43 35 27 19 11 3 ...  
      60 52 44 36 63 55 47 39 ...  
      31 23 15 7 62 54 46 38 ...  
      30 22 14 6 61 53 45 37 ...
```

29 21 13 5 28 20 12 4];

```

pt = ...
    [16  7 20 21 29 12 28 17 ...
      1 15 23 26  5 18 31 10 ...
      2  8 24 14 32 27  3  9 ...
      19 13 30  6 22 11  4 25];
A=[qw stp dse dr pt];

data = r_encrypt;
data_length = length(data);
adds = 64-mod(data_length,64);
if adds ~= 0
    for i = 1:adds
        data(data_length+i) = 0;
    end
    data_length1 = data_length+adds;
end

temp = data;
data = [];
for i = 1:data_length1
    data = [data,dec2bin(temp(i))];
end

key = key2;
keylength = length(key);
keytemp0 = uint8(key);
keys = [149 57 208 147 21 183 27];

if keylength < 7
    for i = keylength+1:7
        keytemp0(i) = keys(i);
    end
end
for i=1:7
    keyusetemp(i) = keytemp0(i);
end
keyuse = char(keyusetemp);

pwb = str2bin(keyuse,7,2);
pwb = rebit(pwb,A(1,177:232));
ki = gerkey(pwb);
ki = flipud(ki);

```

```

times = data_length1/64;
for i = 0:times-1
    for j = 1:64
        tempdata(j) = data(64*i+j);

    end
    decrydata = des(tempdata,ki);
    data_decrypt(i*64+1:(i+1)*64) = decrydata;
end

temp = data_decrypt;
data_decrypt = zeros(1,r_length);
for i = 1:r_length
    if temp(i) == '1'
        data_decrypt(i) = 1;
    else
        data_decrypt(i) = 0;
    end
end

function bin = str2bin(str,k,flag)
l = length(str);
bin = [];
temp = [];

for x = 1:l
    temp = [temp,dec2bin(str(x),8)];
end

if flag ~= 0
    n = ceil(l*8/k);
    rb = mod(l*8,k);
    sb = 0;
    if rb ~= 0
        sb = k-rb;
    end
    for i = 1:sb
        z(i) = '0';
    end
    temp = [temp z];
end
for x = 0:n-1
    temp1 = temp(x*k+1:(x+1)*k);
    lone = length(find(temp1 == '1'));
    if flag == 1
        if mod(lone,2) == 0

```

```

        opb = '1';
    else
        opb = '0';
    end
else if flag == 2
    if mod(lone,2) == 0
        opb = '0';
    else
        opb = '1';
    end
end

temp1 = [temp1 opb]; bin
= [bin temp1];
e
nd
els bin = temp;
e
end

```

```

function so = rebit(si,k)
lk=length(k);
for i=1:lk
    so(i)=si(k(i));
end

```

```

function ki=gerkey(k)
mt = ...
    [12  3 4 5  6 7 8  9 10 11 12 13 14 15 16;
     11  2 2 2  2 2 2  1  2  2  2  2  2  2 1];
rt = ...
    [14  17 11  24 1  5  3 28 ...
     15  6 21  10 23  19 12  4 ...
     26  8 16  7 27  20 13  2 ...
     41 52 31  37 47  55 30 40 ...
     51 45 33  48 44  49 39 56 ...
     34 53 46  42 50  36 29 32];
kl = k(1:28);
kr = k(29:56);
for i = mt(1,1):mt(1,16)
    kl = mr(kl,mt(2,i));
    kr = mr(kr,mt(2,i));
    k = [kl kr];
    for j = 1:48
        ki(i,j) = k(fix(rt(j))),
    end
end

```

```

        end
    end
    function nk = mr(k,n)
    l = length(k);
    k1 = k(n+1:l);
    k2 = k(1:n);
    nk = [k1 k2];

    function ef = des(pf,ki)
    global A

    pfb = rebit(pf,A(1,1:64));
    lpfb = pfb(1:32);
    rpfb = pfb(33:64);

    for i = 0:15
        templ = lpfb;
        lpfb = rpfb;
        tempr = rebit(rpfb,A(1,129:176));
        rpfb = char(xor(ki(i+1,:)-48,tempr-48)+48);
        rpfb = sbox(rpfb);
        rpfb = rebit(rpfb,A(1,233:264));
        rpfb = char(xor(templ-48,rpfb-48)+48);
    end
    pfb = [rpfb lpfb];
    pfb = rebit(pfb,A(1,65:128));
    ef = pfb;
    function so = sbox(si)
    sbox1 = ...
        [14  4 131  2 15 11  8  3 10  6 12  5  9  0  7;
         0 15  74 14  2 13  1 10  6 12 11  9  5  3  8;
         4  1 148 13  6  2 11 15 12  9  7  3 10  5  0;
        15 12  82  4  9  1  7  5 11  3 14 10  0  6 13];
    sbox2 = ...
        [15  1  814  6 11  3  4  9  7  2 13 12  0  5 10;
         3 13  4 7 15  2  8 14 12  0  1 10  6  9 11  5;
         0 14  711 10  4 13  1  5  8 12  6  9  3  2 15;
        13  8 10  1  3 15  4  2 11  6  7 12  0  5 14  9];
    sbox3 = ...
        [10  0  9 14  6  3 15  5  1 13 12  7 11  4  2  8;
         13  7  0  9  3  4  6 10  2  8  5 14 12 11  5  1;
         13  6  4  9  8 15  3  0 11  1  2 12  5 10 14  7;
         1 10 13  0  6  9  8  7  4 15 14  3 11  5  2 12];
    sbox4 = ...

```



```

        [ 7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15;
          13 8 11 5 6 5 0 3 4 7 2 12 1 10 14 9;
          10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4;
          3 15 0 6 10 1 13 8 9 4 5 11 12 7 2 14];
sbox5 = ...
        [ 2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9;
          14 11 2 12 4 7 13 1 5 0 15 10 3 9 8 6;
          4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14;
          1 8 12 7 1 14 2 13 6 15 0 9 10 4 5 3];
sbox6 = ...
        [12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11;
          10 15 4 2 7 12 9 5 6 1 13 14 0 11 3 8;
          9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6;
          4 3 2 12 9 5 15 10 11 14 1 7 6 0 8 13];
sbox7 = ...
        [ 4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1;
          13 0 11 7 4 9 1 10 14 3 5 12 2 15 8 6;
          1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2;
          6 11 13 8 1 4 10 7 9 5 0 15 14 2 3 12];
sbox8 = ...
        [13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7;
          1 15 13 8 10 3 7 4 12 5 6 11 0 14 9 2;
          7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8;
          2 1 14 7 4 10 8 13 15 12 9 0 3 5 6 11]; sbox =
[sbox1 sbox2 sbox3 sbox4 sbox5 sbox6 sbox7 sbox8];
sboxout = [ ];
for i = 0:7
    sboxin(i+1,1:6) = si(i*6+1:(i+1)*6);
    rind = bin2dec([sboxin(i+1,1),sboxin(i+1,6)])+1;
    nind = bin2dec(sboxin(i+1,2:5))+1+i*16; sboxout
    = [sboxout dec2bin(sbox(rind,nind),4)];
end
so = sboxout;

```

## Encryption Function:

```

function [data_encrypt,data_length] =Encryption(r,key1)

global qw stp dse dr pt A

```

```

qw = ...
    [58 50 42 34 26 18 10 2 ...
      60 52 44 36 28 20 12 4 ...
      62 54 46 38 30 22 14 6 ...
      64 56 48 40 32 24 16 8 ...
      57 49 41 33 25 17 9 1 ...
      59 51 43 35 27 19 11 3 ...
      61 53 45 37 29 21 13 5 ...
      63 55 47 39 31 23 15 7];

stp = ...
    [40 8 48 16 56 24 64 32 ...
      39 7 47 15 55 23 63 31 ...
      38 6 46 14 54 22 62 30 ...
      37 5 45 13 53 21 61 29 ...
      36 4 44 12 52 20 60 28 ...
      35 3 43 11 51 19 59 27 ...
      34 2 42 10 50 18 58 26 ...
      33 1 41 9 49 17 57 25];

dse = ...
    [32 1 2 3 4 5 4 5 ...
      6 7 8 9 8 9 10 11 ...
      12 13 12 13 14 15 16 17 ...
      16 17 18 19 20 21 20 21 ...
      22 23 24 25 24 25 26 27 ...
      28 29 28 29 30 31 32 1];

dr = ...
    [57 49 41 33 25 17 9 1 ...
      58 50 42 34 26 18 10 2 ...
      59 51 43 35 27 19 11 3 ...
      60 52 44 36 63 55 47 39 ...
      31 23 15 7 62 54 46 38 ...
      30 22 14 6 61 53 45 37 ...
      29 21 13 5 28 20 12 4];

pt = ...
    [16 7 20 21 29 12 28 17 ...
      1 15 23 26 5 18 31 10 ...
      2 8 24 14 32 27 3 9 ...
      19 13 30 6 22 11 4 25];

A=[qw stp dse dr pt];
data = r;
data_length = length(data);
adds = 64-mod(data_length,64);
if adds ~= 0
    for i = 1:adds

```

```

        data(data_length+i) = 0;
    end
    data_length1 = data_length+adds;
end
temp = data;
data = [];
for i = 1:data_length1
    data = [data,dec2bin(temp(i))];
end

key = key1;
keylength = length(key);
keytemp0 = uint8(key);
keys = [149 57 208 147 21 183 27];

if keylength < 7
    for i = keylength+1:7
        keytemp0(i) = keys(i);
    end
end
for i=1:7
    keyusetemp(i) = keytemp0(i);
end
keyuse = char(keyusetemp);

pwb = str2bin(keyuse,7,2);
pwb = rebit(pwb,A(1,177:232));
ki = gerkey(pwb);

times = data_length1/64;
for i = 0:times-1
    for j = 1:64
        tempdata(j) = data(64*i+j);
    end
    encrydata = des(tempdata,ki);
    data_encrypt(i*64+1:(i+1)*64) = encrydata;
end

temp = data_encrypt;
data_encrypt = zeros(size(temp));
for i = 1:length(data_encrypt)
    if temp(i) == '1'

```

```

        data_encrypt(i) = 1;
    else
        data_encrypt(i) = 0;
    end
end

```

```

function bin = str2bin(str,k,flag)

```

```

l = length(str);
bin = [];
temp = [];

for x = 1:l
    temp = [temp,dec2bin(str(x),8)];
end

if flag ~= 0
    n = ceil(l*8/k);
    rb = mod(l*8,k);
    sb = 0;
    if rb ~= 0
        sb = k-rb;
    for i = 1:sb
        z(i) = '0';
    end
    temp = [temp z];
end
for x = 0:n-1
    temp1 = temp(x*k+1:(x+1)*k);
    lone = length(find(temp1 == '1'));
    if flag == 1
        if mod(lone,2) == 0
            opb = '1';
        else
            opb = '0';
        end
    else if flag == 2
        if mod(lone,2) == 0

```

```

        opb = '0';
    else
        opb = '1';
    end
end

    temp1 = [temp1 opb]; bin
    = [bin temp1];
    e
end
els bin = temp;
e

end

```

```
function so = rebit(si,k)
```

```

lk=length(k);
for i=1:lk
    so(i)=si(k(i));
end

```

```
function ki=gerkey(k)
```

```

mt = ...
    [12  3 4 5  6 7 8  9 10 11 12 13 14 15 16;
     11  2 2 2  2 2 2  1  2  2  2  2  2  2 1];
rt = ...
    [14  17 11  24 1  5  3 28 ...
     15  6 21  10 23  19 12  4 ...
     26  8 16  7 27  20 13  2 ...
     41 52 31  37 47  55 30 40 ...
     51 45 33  48 44  49 39 56 ...
     34 53 46  42 50  36 29 32];
kl = k(1:28);
kr = k(29:56);
for i = mt(1,1):mt(1,16)
    kl = mr(kl,mt(2,i));
    kr = mr(kr,mt(2,i));
    k = [kl kr];
    for j = 1:48
        ki(i,j) = k(fix(rt(j)));
    end
end

```

```
end  
end
```

```
function nk = mr(k,n)
```

```
l = length(k);  
k1 = k(n+1:l);  
k2 = k(1:n);  
nk = [k1 k2];
```

```
function ef = des(pf,ki)
```

```
global A
```

```
pfb = rebit(pf,A(1,1:64));  
lpfb = pfb(1:32);  
rpfb = pfb(33:64);
```

```
for i = 0:15  
    templ = lpfb;  
    lpfb = rpfb;  
    tempr = rebit(rpfb,A(1,129:176));  
    rpfb = char(xor(ki(i+1,:)-48,tempr-48)+48);  
    rpfb = sbox(rpfb);  
    rpfb = rebit(rpfb,A(1,233:264));  
    rpfb = char(xor(templ-48,rpfb-48)+48);  
end
```

```
pfb = [rpfb lpfb];  
pfb = rebit(pfb,A(1,65:128));  
ef = pfb;
```

```
function so = sbox(si)
```

```
sbox1 = ...  
    [14  4 13  1  2 15 11  8  3 10  6 12  5  9  0  7;  
      0 15  7  4 14  2 13  1 10  6 12 11  9  5  3  8;  
      4  1 14  8 13  6  2 11 15 12  9  7  3 10  5  0;  
     15 12  8  2  4  9  1  7  5 11  3 14 10  0 6 13];
```

```

sbox2 = ...
    [15  1  8 14  6 11  3  4  9 7  2 13 12 0  5 10;
      3 13  4  7 15  2  8 14 12 0  1 10  6 9 11  5;
      0 14  7 11 10  4 13  1  5 8 12  6  9 3  2 15;
     13  8 10  1  3 15  4  2 11 6  7 12  0 5 14  9];

sbox3 = ...
    [10  0  9 14  6  3 15 5  1 13  12 7 11  4 2  8;
     13  7  0  9  3  4  6 10  2  8  5 14 12 11 5  1;
     13  6  4  9  8 15  3  0 11  1  2 12  5 10 14  7;
      1 10 13  0  6  9  8  7  4 15  14  3 11  5 2 12];

sbox4 = ...
    [ 7 13 14  3  0  6  9 10  12  8  5 11 12  4 15;
     13  8 11  5  6  5  0  3  4 7  2 12  1 10 14  9;
     10  6  9  0 12 11  7 13 15 1  3 14  5  2  8  4;
      3 15  0  6 10  1 13  8  9 4  5 11 12  7  2 14];

sbox5 = ...
    [ 2 12  4  1  7 10 11  6  8  5  3 15 13  0 14  9;
     14 11  2 12  4  7 13  1  5  0 15 10  3  9  8  6;
      4  2  1 11 10 13  7  8 15  9 12  5  6  3  0 14;
      1  8 12  7  1 14  2 13  6 15  0  9 10  4  5  3];

sbox6 = ...
    [12  1 10 15  9  2  6  8  0 13  3  4 14  7  5 11;
     10 15  4  2  7 12  9  5  6  1 13 14  0 11  3  8;
      9 14 15  5  2  8 12  3  7  0  4 10  1 13 11  6;
      4  3  2 12  9  5 15 10 11 14  1  7  6  0  8 13];

sbox7 = ...
    [ 4 11  2 14 15  0  8 13  3 12  9  7  5 10  6  1;
     13  0 11  7  4  9  1 10 14  3  5 12  2 15  8  6;
      1  4 11 13 12  3  7 14 10 15  6  8  0  5  9  2;
      6 11 13  8  1  4 10  7  9  5  0 15 14  2  3 12];

sbox8 = ...
    [13  2  8 4  6 15 11  1 10  9  3 14  5  0 12  7;
      1 15 13 8 10  3  7  4 12  5  6 11  0 14  9  2;
      7 11  4 1  9 12 14  2  0  6 10 13 15  3  5  8;
      2  1 14 7 4 10 8 13 15 12  9  0  3  5  6 11]; sbox =
[sbox1 sbox2 sbox3 sbox4 sbox5 sbox6 sbox7 sbox8];
sboxout = [ ];
for i = 0:7
    sboxin(i+1,1:6) = si(i*6+1:(i+1)*6);
    rind = bin2dec([sboxin(i+1,1),sboxin(i+1,6)])+1;
    nind = bin2dec(sboxin(i+1,2:5))+1+i*16; sboxout
    = [sboxout dec2bin(sbox(rind,nind),4)];
end
so = sboxout;

```

## Encryption:

```
clc
clear all
close all

objt=audiorecorder

recordblocking(objt,5)

play(objt)

aa=getaudiodata(objt);

figure,plot(aa)

blocksize=8;

inp=reshape(aa,200,200);
% % read host
host=inp+0.8;

% imwrite(watermarked_image_uint,'watermarked_image.bmp')

tic
% % % encryption
x=uint8(host+128);
figure,imshow(x,[])
% xx=double(host)
[a b c]=size(x);
N=a*b;

m(1)=0.8;

for i=1:N-1
    m(i+1)=4*m(i)-4*m(i)^2;
end
m=mod(1000*m,200);
m=uint8(m);
n=1;
for i=1:a
```



```

        for j=1:b
            I1= x;

            e(i,j)=bitxor(m(n),x(i,j));

        end
    end

    figure,imshow(e)

    imwrite(e,'e.jpg');

    %bkjbjb
    % clear;
    % clc;
    e=imread('e.jpg')
    x1=uint8(e);

    [a1 b1 c1]=size(x1);
    N1=a1*b1;
    m1(1)=0.8;
    % disp('???');
    for i=1:N1-1
        m1(i+1)=4*m1(i)-4*m1(i)^2;
    end
    m1=mod(1000*m1,256);
    m1=uint8(m1); n1=1;
    for i=1:a1 for
        j=1:b1

            e1(i,j)=bitxor(m1(n1),x1(i,j));

        end
    end
end

```

n=r

n1=

```

figure,imshow(e1)
s ='MATLAB'
val=double(s)
nume=reshape(val, 3, 5)
m = [1 5 3; 2 11 8; 4 24
21] %inv(m)
nume=nume-32 ncoded=mod(m*(nume),95)+32
scoded=reshape(char(ncoded),1,15) sdecoded =
reshape(double(scoded),3,5) ndecoded = mod
(inv(m)*(sdecoded-32), 95) + 32 sdecoded =
reshape(char(ndecoded), 1, 15)

```

## Main:

```

clc
clear all
close all

inp=input('PLS ENTER TO START RECORDING : ')

objectt=audiorecorder

recordblocking(objectt,10)

play(objectt)

git=getaudiodata(objectt);

figure,plot(git)
title('audio speech recorded')

N =length(git)
r =git;
for i = 1:N
    if r(i) >= -0.1
        r(i) = 1;
    else
        r(i) = 0;
    end
end
end

```

```

figure,plot(r)
title('digital signal')

key1 = char(inputdlg('encrypt key'));
key2 = char(inputdlg('decrypt key'));

if (key1==key2)
    out=1;
else
    out=git(1:length(git)).*git;
end

[r_encrypt,r_length] = Encryption(r,key1);
r_decrypt = Decryption(r_encrypt,key2,r_length);

count = 100;
R = zeros(1,length(r)*count);
R_encrypt = zeros(1,length(r_encrypt)*count);
R_decrypt = zeros(1,length(r_decrypt)*count);
for i = 1:length(r)*count
    R(i) = r(((i-1)-mod((i-1),count))/count+1);
end
for i = 1:length(r_encrypt)*count
    R_encrypt(i) = r_encrypt(((i-1)-mod((i-1),count))/count+1);
end
for i = 1:length(r_decrypt)*count
    R_decrypt(i) = r_decrypt(((i-1)-mod((i-1),count))/count+1);
end
figure
plot(0:1/count:length(r)-1/count,R,'b','LineWidth',2)
% axis([0,500,-1,2])
title('digital')
grid on

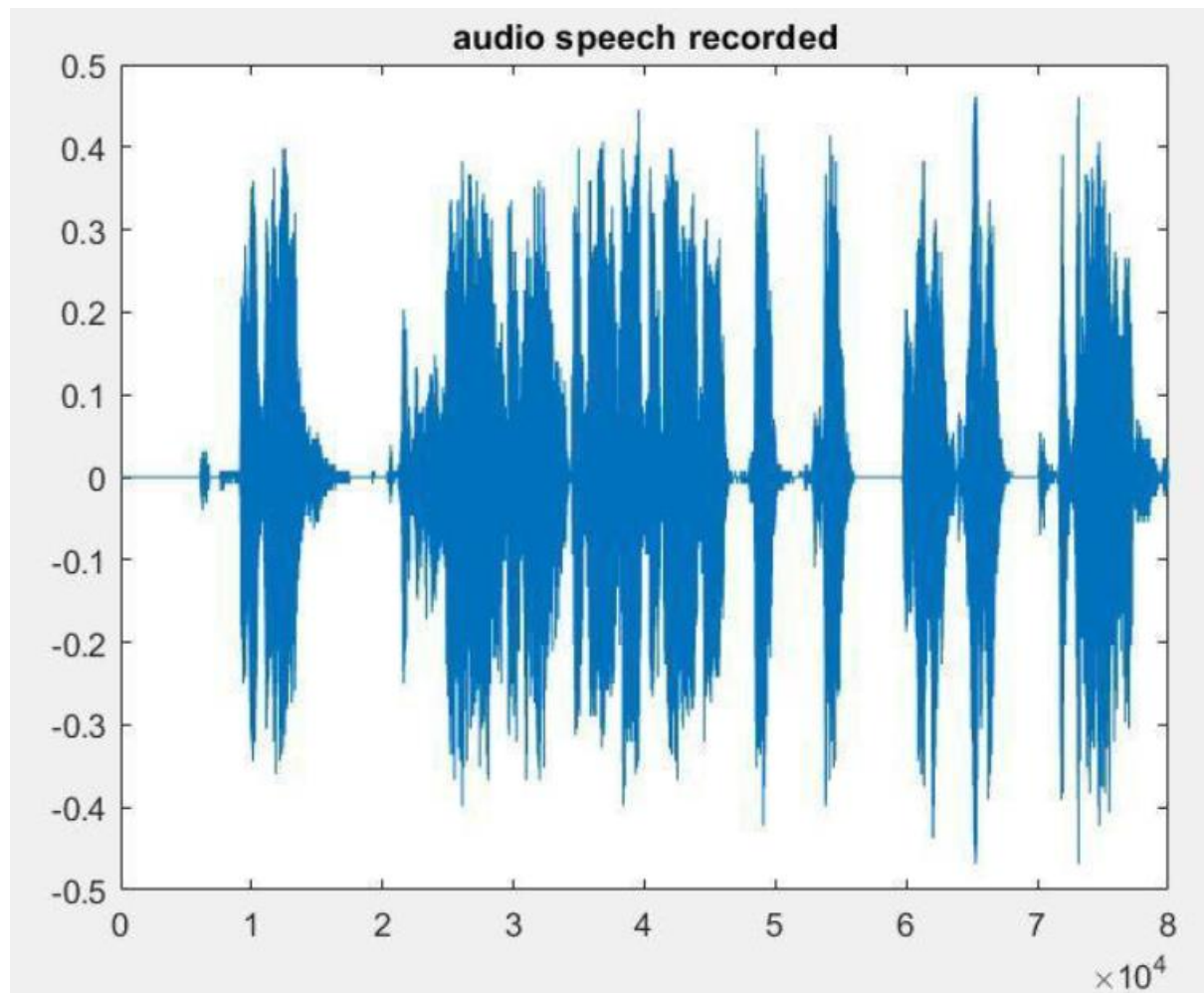
figure,
plot(0:1/count:length(r_encrypt)-1/count,R_encrypt,'b','LineWidth',2)
% axis([0,50,-1,2])
title('encrypted')
figure
plot(0:1/count:length(r_decrypt)-1/count,R_decrypt,'b','LineWidth',2)

```

```
2)
% axis([0,50,-1,2])
title('decrypted') grid on

figure
plot(git.*out,'b','LineWidth',
2)
% axis([0,500,-1,2])
title('recovered
speech') grid on
```

# OUTCOMES:



## ENCRYPT AND DECRYPT KEY

encrypt key

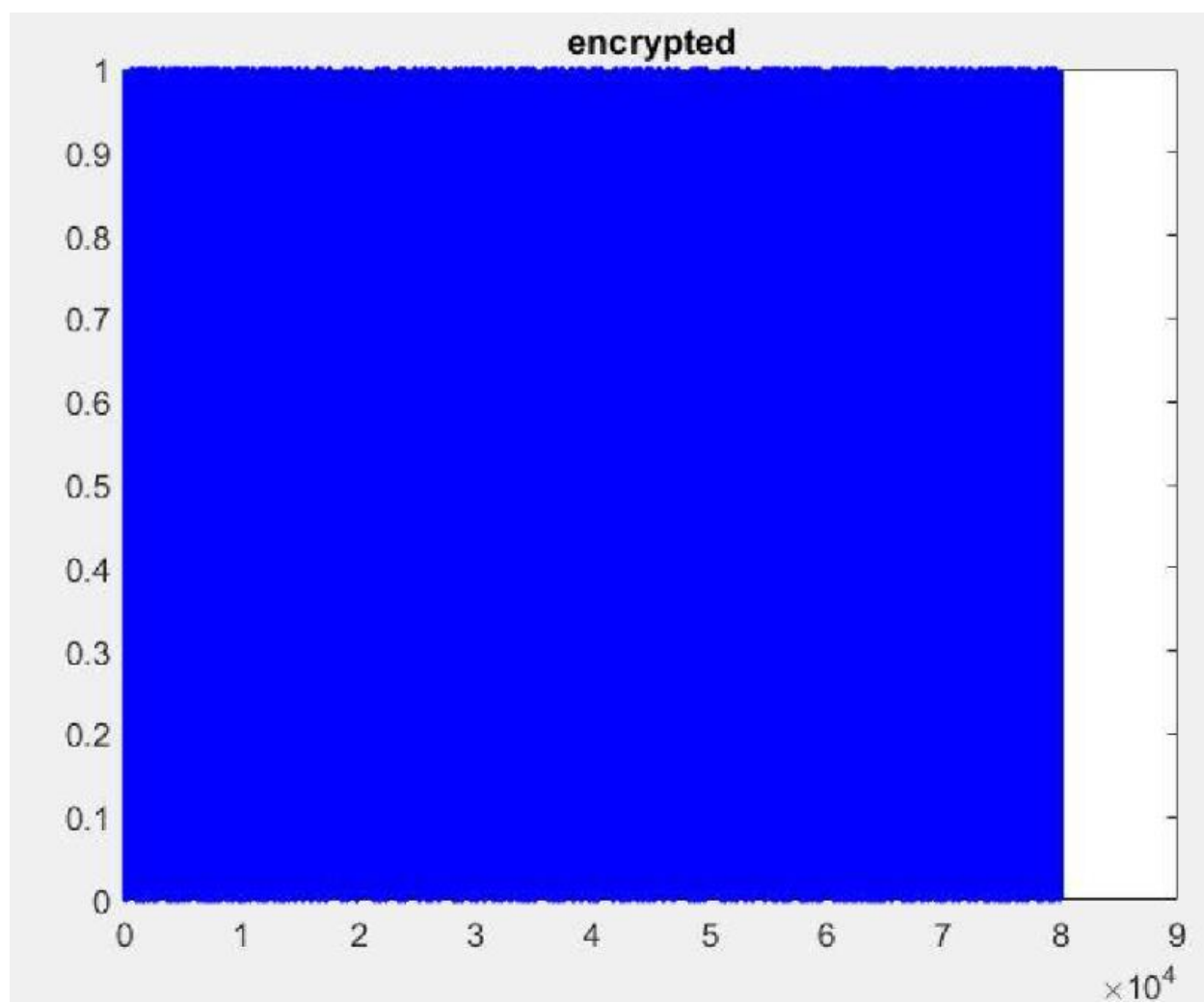
12345

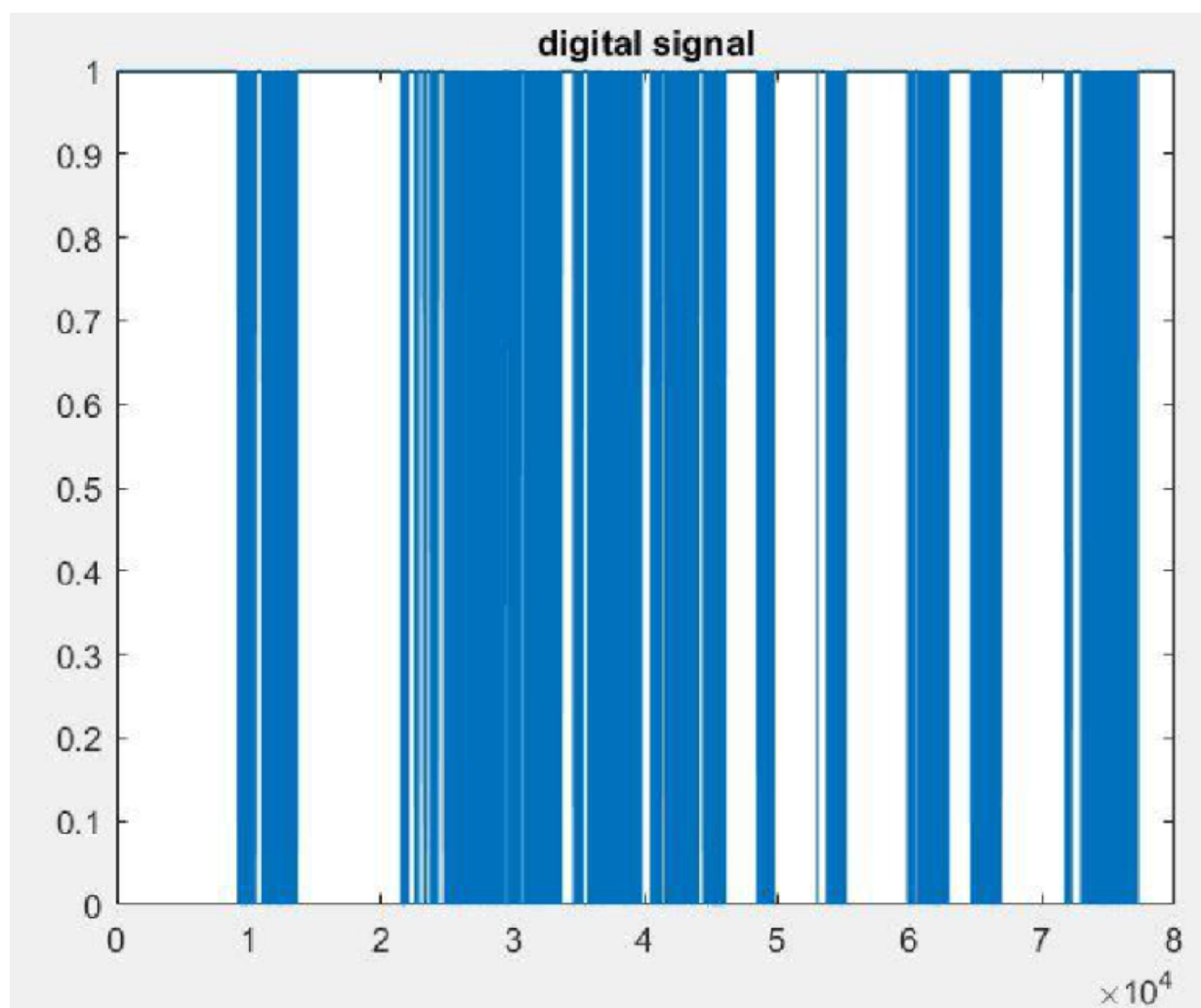
OK Cancel

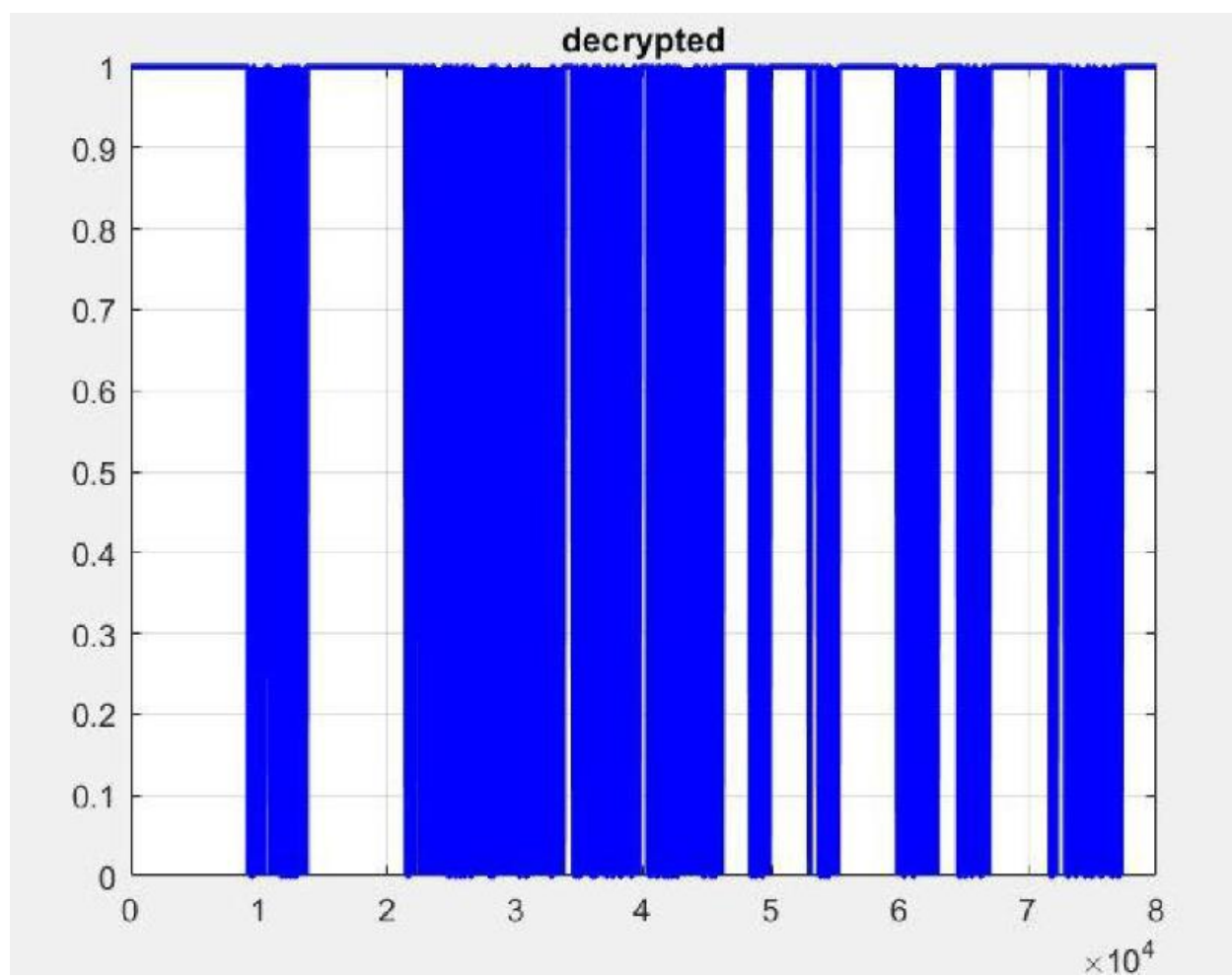
decrypt key

12345

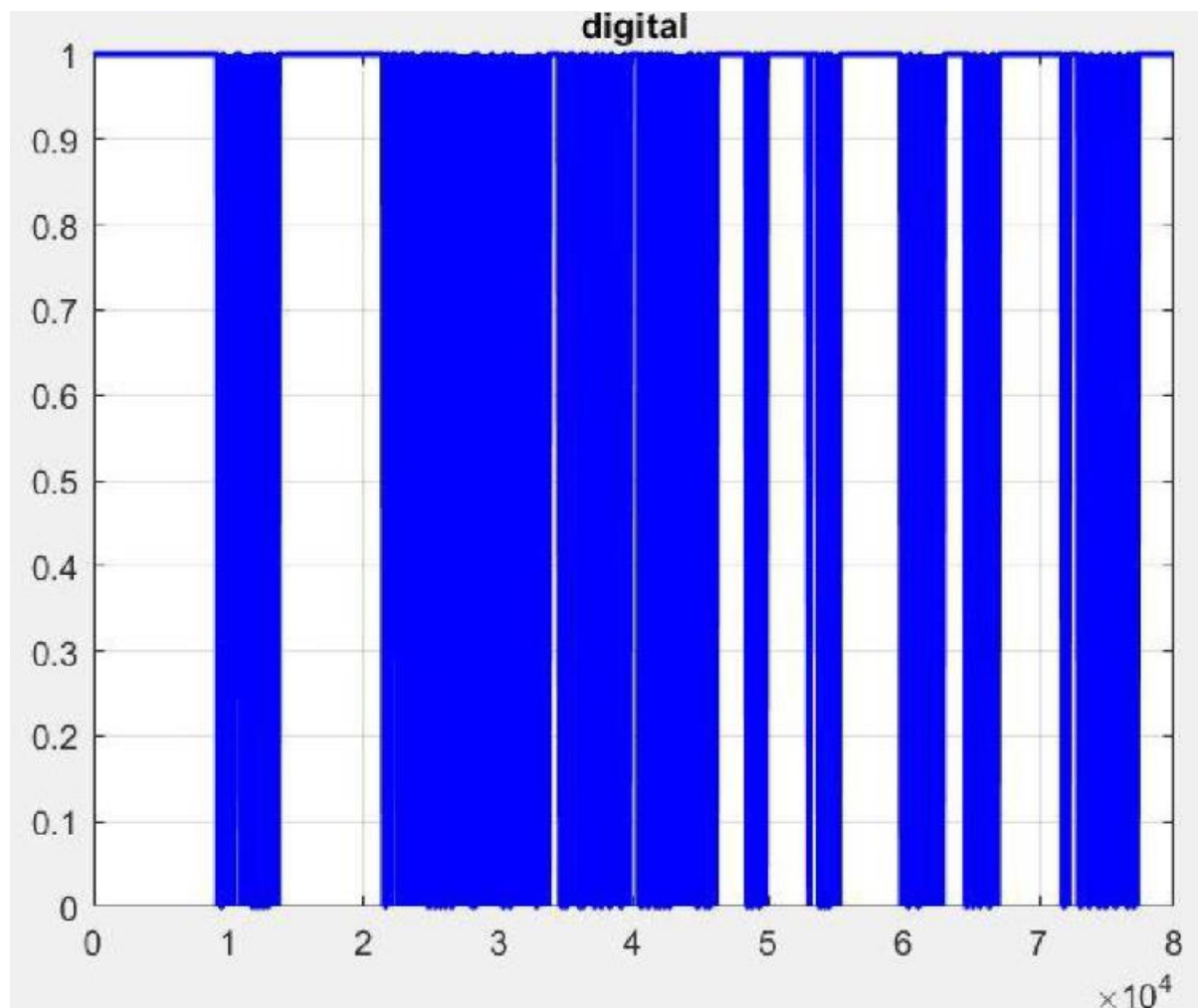
OK Cancel











## COMMAND WINDOW

Command Window

PLS ENTER TO START RECORDING :

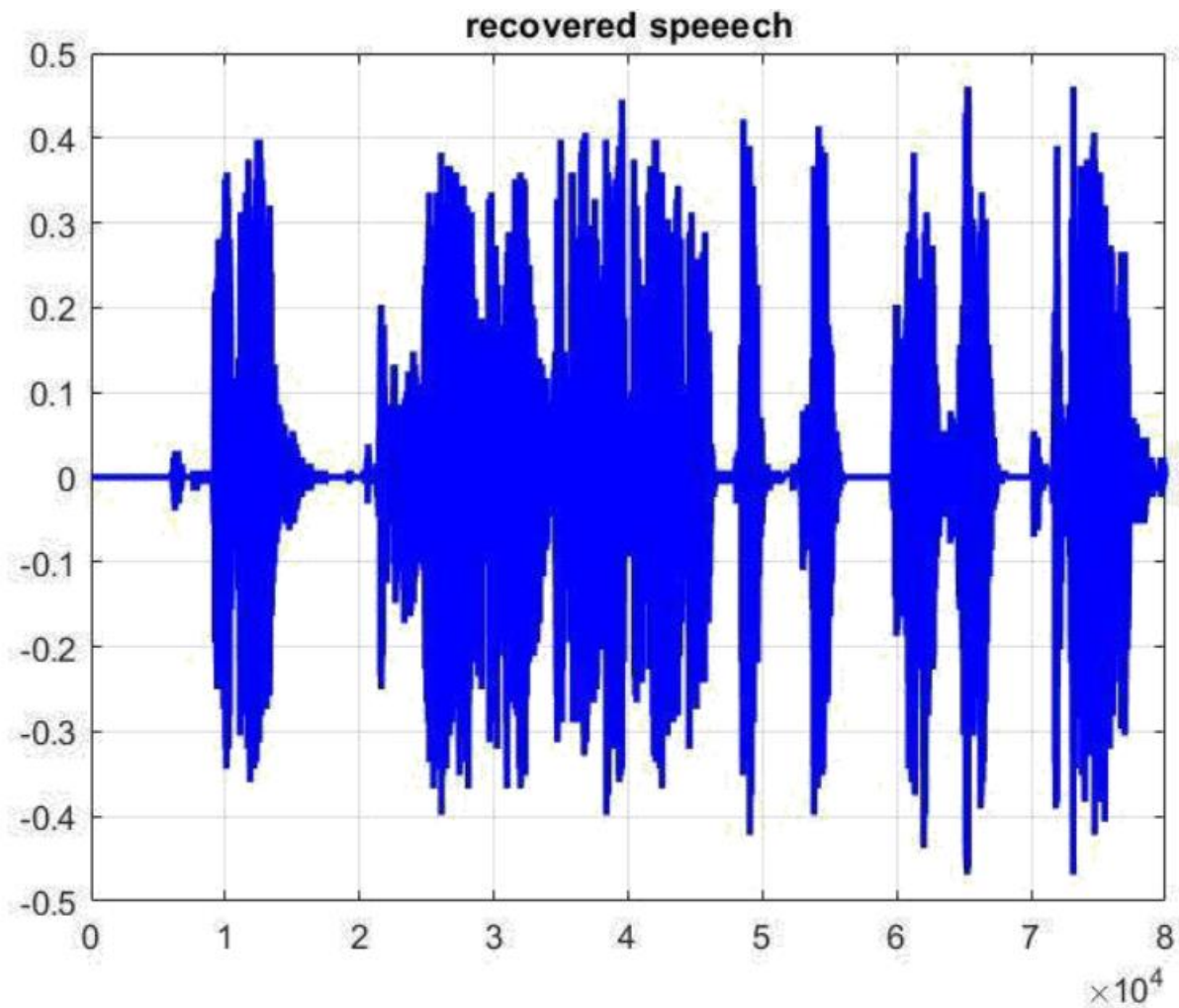
```
inp =
    []
```

objectt =

**audiorecorder** with properties:

```

    SampleRate: 8000
    BitsPerSample: 8
    NumChannels: 1
    DeviceID: -1
    CurrentSample: 1
    TotalSamples: 0
    Running: 'off'
    StartFcn: []
    StopFcn: []
    TimerFcn: []
    TimerPeriod: 0.0500
    Tag: ''
    UserData: []
    Type: 'audiorecorder'
```



**OUTPUT:-**

```
ans =  
  
    audioplayer with properties:  
  
    SampleRate: 8000  
    BitsPerSample: 8  
    NumChannels: 1  
    DeviceID: -1  
    CurrentSample: 257  
    TotalSamples: 32000  
    Running: 'on'  
    StartFcn: []  
    StopFcn: []  
    TimerFcn: []  
    TimerPeriod: 0.0500  
    Tag: ''  
    UserData: []  
    Type: 'audioplayer'  
  
N =  
  
    32000
```

# CONCLUSION

Encryption of all message signals is extremely important in today's world as data is becoming increasingly vulnerable. So, there is a need to protect the data from hackers and unwanted corruptions. If the signal contains some data of national importance then it is very dangerous to send it to someone without any encryption as it could endanger the safety of the entire nation if it goes into the wrong hand, so to prevent this from happening we created this project. Through this project we have aimed to make audio signal transfer lot more safe and secure by encrypting it using a password. Only a user who knows the password can decrypt the audio signal. This means the signal becomes a lot more safe and secure.