VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE 4003 - Cyber Security

"Secured Audio Signal and Voice Message Transmission Using Hybrid Algorithm"

Group Members:

20BCI0090 : Vandit Gabani

20BCI0088 : Shantanu Patra

20BCI0206 : Zeel Lukhi

20BCI0273 : Ananya Singhal

Guided By

Prof. Navamani T M

Assosiate Professor Grade 1

1. Abstract:

In asymmetric key cryptography, also called Public Key cryptography, two different keys (which forms a key pair) are used. One key is used for encryption & only the other corresponding key must be used for decryption. No other key can decrypt the message, not even the original (i.e. the first) key used for encryption. The beauty of this scheme is that every communicating party needs just a key pair for communicating with any number of other communicating parties. Once some one obtains a key pair, he /she can communicate with any one else. RSA is a well known public key cryptography algorithm. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography.

The security of the RSA cryptosystem is based on two mathematical problems: the problem of factoring large numbers know mathematical attack and the problem of trying all possible private keys know brute force attack. So to improve the security, this scheme presents a new cryptography algorithm based on additive homomorphic properties called Modified RSA Encryption Algorithm (MREA). MREA is secure as compared to RSA as it is based on the factoring problem as well as decisional composite residuosity assumptions which is the intractability hypothesis. The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m1 and m2, one can compute the encryption of m1 + m2. This scheme also presents comparison between RSA and MREA cryptosystems in terms of security and performance.

With the increasing need for secure speech communication, data encryption protocols are extremely important for storage and transmission of sensitive and personal information. As the prevalence of Voice assistants surges, companies which collect speech data to improve their assistants, put millions of consumer's personal data at risk to attacks by hackers or crackers, who may use the data for nefarious purposes. To help combat this threat, we propose a system which converts the speech commands into text, encrypted using Modified RSA + AES.

The Google Speech API and Socket connections which emulate the communication between assistants and servers, send encrypted data over to the client in text form so that the voice data is not stored and the assistant gets the data it needs to train itself.

2. Introduction:

Voice communication is now widely used in a variety of fields, including the military, confidential void conferencing, phone banking, and education. The companies behind voice assistants admit that our voice commands are recorded for quality assurance purposes, millions of people's personal information is at risk. This enables the hackers of obtain our voice samples which can later be fabricated to sound like us. With the growing demand for safe speech communication, data encryption technologies are essential for storing and transmitting sensitive and personal data over vulnerable systems. Therefore, we suggest a system that encrypts the voice using double-key RSA and AES encryptions, and stores it anonymously. The ciphertext is obtained after audio encryption which is then encrypted again to avoid easy decryption, using a modified RSA and AES encryption system. This is done to ensure that the audio being transmitted remains confidential.

This opens the door to hackers breaking into company databases and stealing our personal information, preferences, and the structure of our voices, which can then be fabricated to sound like us. With the growing demand for safe speech communication, data encryption technologies are essential for storing and transmitting sensitive and personal data over vulnerable systems where several intrusions and hackers are waiting. To secure the safe transmission of audio, various cryptographic methods are used. Cryptography is the process of converting data into an unreadable format so that it can be sent from one person to another without compromising its confidentiality. To reduce the possible harm caused by this intrusion, we suggest a system that turns voice commands into text, encrypts it using double-key RSA and AES encryptions, hashes and salts it, and stores it anonymously such that the most hackers may get are random texts of customer preferences. This model can also assist with the transmission of encrypted voice messages from one device to another. Voice messages can be utilized for ease or for the visually handicapped by using the speech to text feature and the secure encryption and authentication system. To implement this model, we plan to use various tools to emulate the voice assistants. The obtained ciphertext is then hashed to avoid easy decryption, using a modified RSA and AES encryption system to encrypt: the text and the key to decrypt the same. This is done to ensure that the audio being transmitted remains confidential.

This section involves the work done by various researchers in the field of RSA cryptosystem. The discussion is based on the Modification of RSA algorithm through the recent past. M. Thangavel et al.proposed a modified RSA key generation algorithm which uses four primes instead of two primes and thereby increasing the time needed to find these primes. While increasing the security, the key generation time of the proposed algorithm is higher than original RSA. According to, encryption and decryption are not only dependent on “N” but also other new factors computed. The encryption and decryption technique is very complex and several factors are introduced without clearly justified. H. M. Sun et al. came up with an algorithm which is known as dual RSA having two aspects from RSA with decreasing the needs of the storage for keys. It

has two applications, first one is known as blind signature and other is known as security. Segar's introduced a ne

w idea to determine the private key without using the factoring approach using Pell's equation. Pell's RSA increases the strength that taking the private key "d" above the Wiener's possible range.

Issues In Existing System:

- When Audio messages are not encrypted, they are susceptible to hackers and third party. Man-in-the-Middle attacks are very common and easy to implement. Thus any third party can access your voice recordings and audio.
- The existing audio encryption protocols are slow and space intensive. They need a considerate amount of data to be processed and encrypted. The entire process is time intensive.
- Security is another issue. The standard protocols are not secure enough to be implemented on large scale applications without affecting the performance. We need an algorithm with stronger encryption protocols.
- For Some Algorithms The audio data is stored as coordinates which leads to data expansion.
- Elliptic curves have large modulo prime value, so for every small integer representation of a message, the expansion in cipher text is massive.

3. Literature Survey:

Title - Author	Method used	Advantages	Drawbacks
Encryption and Decryption of Audio Signal based on RSA Algorithm **By Sura F. Yousif. (International Journal of Engineering Technologies and Management Research)**	- RSA ALgorithm on Audio Signal	- The presented technique use validated that it is secure, reliable and efficient to be applied in secure audio communications. - - It performed high intelligibility of the recovered audio signal.	- They need more computations than symmetric ciphers.
Implementation of Public Key Encryption Algorithm for Speech Data Encryption and Decryption **By Md. Mijanur Rahman, Tushar Kanti Saha, Md. Al-Amin Bhuiyan (IJCSNS)**	- Different Public Key Encryption Algorithms are applied in this paper. - - Different types of data in voice / speech form is taken and encrypted.	- There is no guarantee who sent a given message in the Standard encryption methods, thus Public key encryption has rapidly grown in popularity. - - It offers a very secure encryption method that addresses these concerns.	- The integer representation of the message to be encrypted should lie within the range specified by the modulus (i.e., M lies in the range [0, n-1]), which poses a limitation on the maximum number of characters that can be encrypted at a single time.

Title - Author	Method used	Advantages	Drawbacks
AES Algorithm for real-time Audio Steganography **By Tang, S.Y. , Y.J. Jiang, L.P. Zhang, and Z.B. Zhou.** *(Scopus - 2018)*	- This system uses a VoIP steganographic technique using AES and key distribution to achieve real-time covert VoIP communication. The encryption and embedding procedures are nearly identical. In terms of statistical analysis, the suggested VoIP steganographic technique was found to be secure, effective, and robust after performance testing with state-of-the-art network equipment DSLA and security tests utilizing the M-W-W approach.	- -Different values of Audio Frequency and Time Domains are supported. - It is based on Real-time services. - The quality of encrypted speech is not destroyed. - - Highly redundant representations generally allow the inclusion of a significant amount of hidden data with easy and subtle modifications that preserve the underlying cover object's perceptual content.	- It requires large amount of data to train the model. - - The real-time requirements of VoIP communication provide basic security for the system, but they limit the number of operations that can be performed, making it difficult to add more operations (for instance, security measures) to improve security.
Ameliorated ElGamal Public Key Encryption Over Finite Field **By Khoirom, Motilal Singh, Dolendro Singh Laiphrakpam, and Themrichon Tuithung** *(Springer - 2020)*	- They achieved reasonable execution speed for a public key encryption technique. The proposed approach is a strong, trustworthy public key audio encryption strategy thanks to the analysis results, the strength of ECDLP, and the enhancements achieved with ElGamal PKE.	- With the encryption process cipher audio is generated after various analysis and comparisons to demonstrate the discriminative capability of their method. - - It gives a solution to the data expansion problem as well as additional computations for embedding integer-represented messages in specified coordinates that satisfy the elliptic curve.	- The audio data is stored as coordinates which leads to data expansion. - Elliptic curves have large modulo prime value, so for every small integer representation of a message, the expansion in cipher text is massive. - - The simulation uses 512 bits Elliptic curve parameter, so it becomes computationally very intensive to do it using Brute Force technique.

Title – Author	Method used	Advantages	Drawbacks
Modified RSA-based algorithm: a double secure approach ***By Israa Al_Barazanchi, Shihab A. Shawkat, Moayed H. Hameed, Khalid Saeed Lateef Al-badri*** ***(IEEE-2018)***	- This paper examines the general principles of encryption and focuses on the development of RSA and the complexity of the encryption key so that it becomes more secure in the applications used. In this project, we will work on the RSA algorithm by adding some complexity to the 3 keys (3k). - This addition will increase the security and complexity of the algorithm's speed while maintaining encryption and decryption time. - - The paper also presents an approach by means of public-key encryption to enhance cryptographic security.	- This work recommended a multilevel system for decryption and encryption for the provision of more security to storage information. - The comparison of a traditional RSA and three keys RSA showed the relatively high security and alterations in equation formation for the 3k RSA algorithm. - - All the case observations concluded that any imposter will not be successful to acquire encrypted shares of the networks.	- But that person won't be able to recover secret text while lacking private key accessibility. The creation of the longer keys would provide security for some years such as factoring them is not feasible.
Audio Encryption Optimization ***By Harsh Bijlani, Dikshant Gupta, Mayank Lovanshi*** **(International Conference 2016)**	- - Through this paper, we have proposed the model based on a comparison of different sizes of audio files by taking into account of decimation factor through which the sample rate changes and we obtained a time taken to encrypt such generated audio files and thus we provide our analysis on AES and DES algorithm for the audio file encryption and suggesting which have better performance based on our results obtained.	- The entire Encryption and Decryption process is optimized. The time taken compared to the traditional method is less. - - The Space taken has reduced substantially. The security standards have increased.	- The net computation power requirement has increased, thus we need better hardware to run the application.

Encryption Technology of Voice Transmission in Mobile Networks Using 3DES-ECC **Zhixian Chang, Marcin Wozniak- (Springer-2020)**	3DES-ECC	- This method can effectively encrypt the voice transmission process. For many experiments, the encryption time is less than 1 s, - - The encryption speed is fast, and the encryption effect is good.	- - There is high data loss compare to other algorithms.

4.Proposed System:

4.1 Overview:

- An audio file of .wav format is taken and then using hybrid algorithm (block cipher algorithm) is used for encryption. The audio file, an 8 bit .wav file is taken and converted to its numeric equivalent for it to be encrypted.
- For encrypting the audio file, it is converted into its numpy array equivalent by using the scipy library and then this array of data is encrypted using combination of this algorithm. The encrypted file is then decrypted back by removing the noise added in the first place.
- The file used is a .wav file. Waveform audio file format was developed by Microsoft and IBM and is the main audio format used over the windows operating system.
- The bitstream encoding is the linear pulse-code modulation (LPCM) format. The file encrypted is an 8 bit audio file.
- For Voice message/data we will use python speechrecognition and pyaudio modules.
- **Speech to text translation:** This is done with the help of Google Speech Recognition. This requires an active internet connection to work.

Library/modules Required:

```
from scipy.io import wavfile
import numpy as np
import matplotlib.pyplot as plt
import math
import sounddevice as sd
import random
import string
from Crypto.Cipher import AES
```

4.2 System and Functional Architecture:

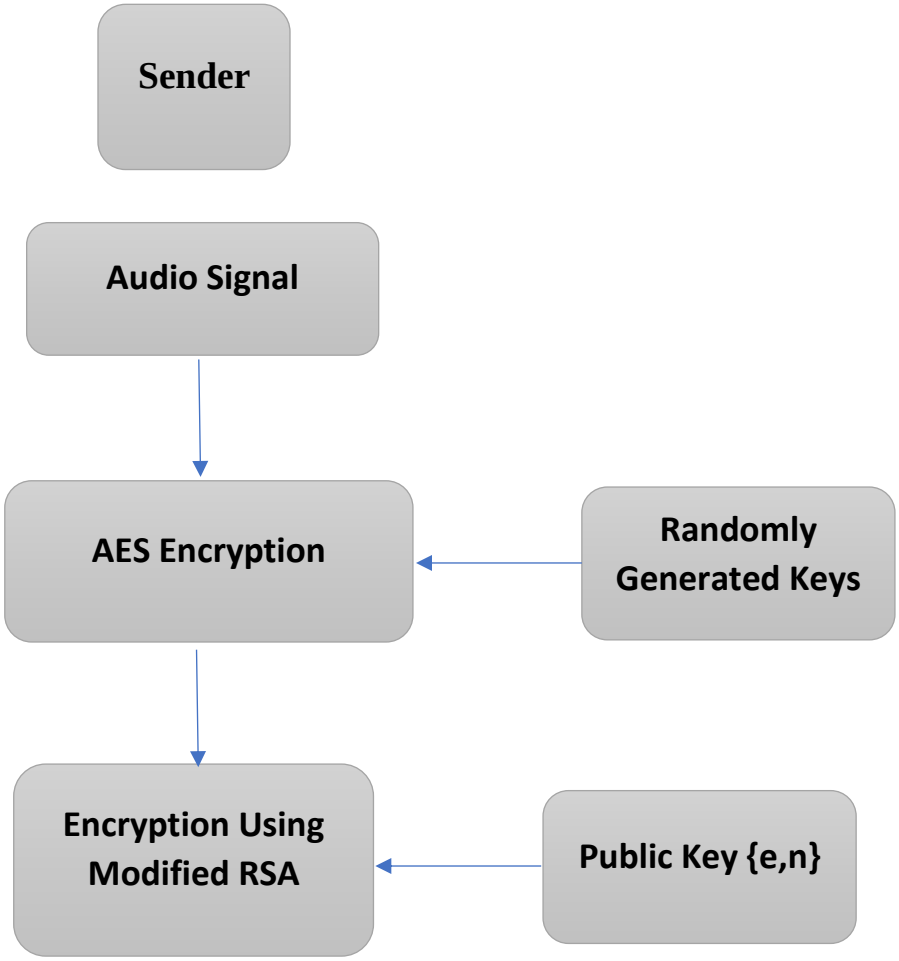


Figure.1

Figure.1 shows the overall process of encryption of audio signal using AES + modified RSA Algorithm.

Read the audio file using - wavfile.read() function.

Plot original audio file - plt.plot(Data).

1> AES encryption:

- Open audio file in 'rb' mode
- Read the file.

- Convert into numpy array.

#key Generation:

- Generate random key.
- Encrypt audio data which we stored as numpy array using AES module.
- Save generated Encrypted Audio File.

Modified RSA Encryption

- Take input of 4 prime numbers. : p1,p2,p3,p4.
- Calculate n. = p1*p2*p3*p4
- Phi : phi = (p1-1)*(p2-1)*(p3-1)*(p4-1).
- e: coprime with n. and k -random number.
- Calculate d - multiplicative inverse of e %n.

Apply Modified RSA encryption on AES encrypted file:

```
encrypted=[]
for i in data:
   encrypted.append((i**e)%n)
#encrypted = (data**e)%n
#print(encrypted)
plt.plot(encrypted)
plt.title("Encrypted Audio Plot")
```

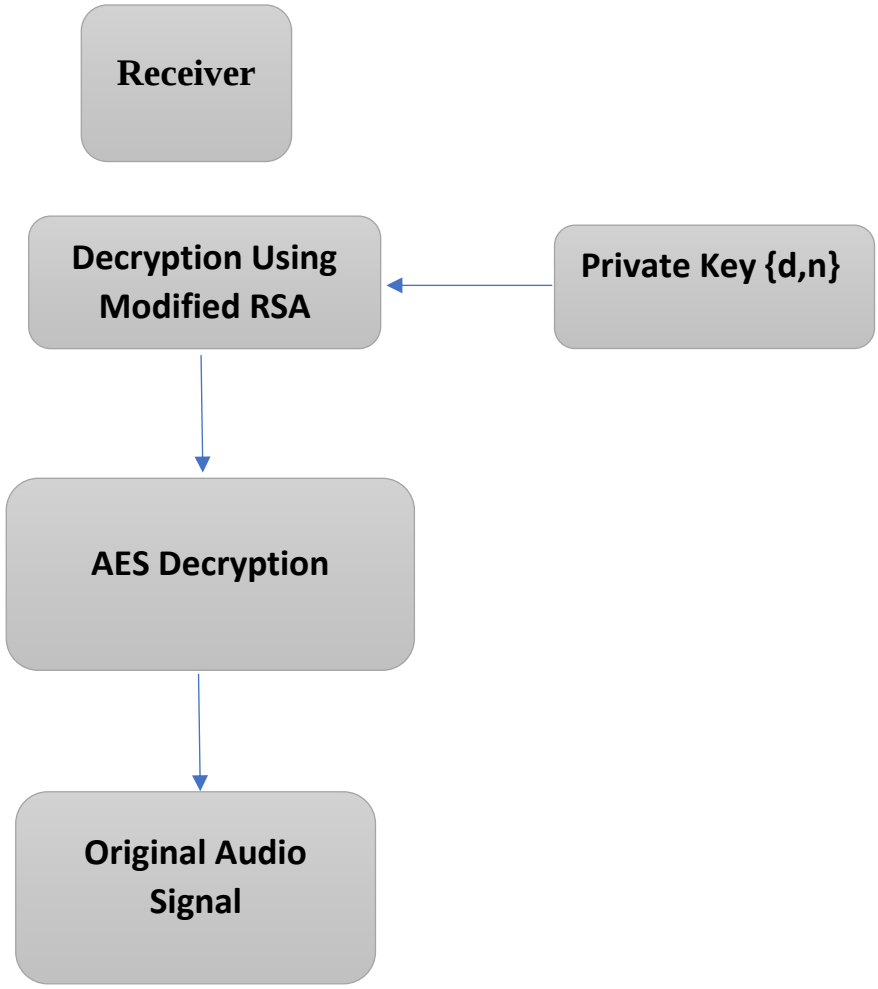


Figure.2

Figure.2 shows the overall process of Decryption of audio signal using AES + modified RSA Algorithm.and shows overall flow of Decryption.

RSA Decryption:

- Open and read last generated encrypted audio file.
- Here power modulo will give big numbers so define function for it's calculation.

decrypted = (data**d)%n

```
plt.plot(decrypted)
print(decrypted)
plt.title('Decrypted Audio Plot')
```

AES Decryption:

- Open and read last generated decrypted audio file.
- Read and decrypt it.

```
with open('encrypted_audio_file.wav', 'rb') as fd:
   contents = fd.read()
decryptor = AES.new(AES_KEY.encode("utf-8"), AES.MODE_CFB,
AES_IV.encode("utf-8"))
decrypted_audio = decryptor.decrypt(contents)
```

4.3 Modular Design:

Modified RSA Algorithm:

Key generation

1. Select FOUR PRIME NUMBERS P,Q,R,S.
2. Calculate n=p*q*r*s.
3. Calculate f(n)=(p-1)*(q-1) *(r-1)*(s-1)
4. Select integers e; gcd(f(n),e)=1;1<e<f(n).
5. Calculate d; d=e-1 mod f(n)
6. Public key KU ={e,n}

7. Private Key KR ={d,N}.

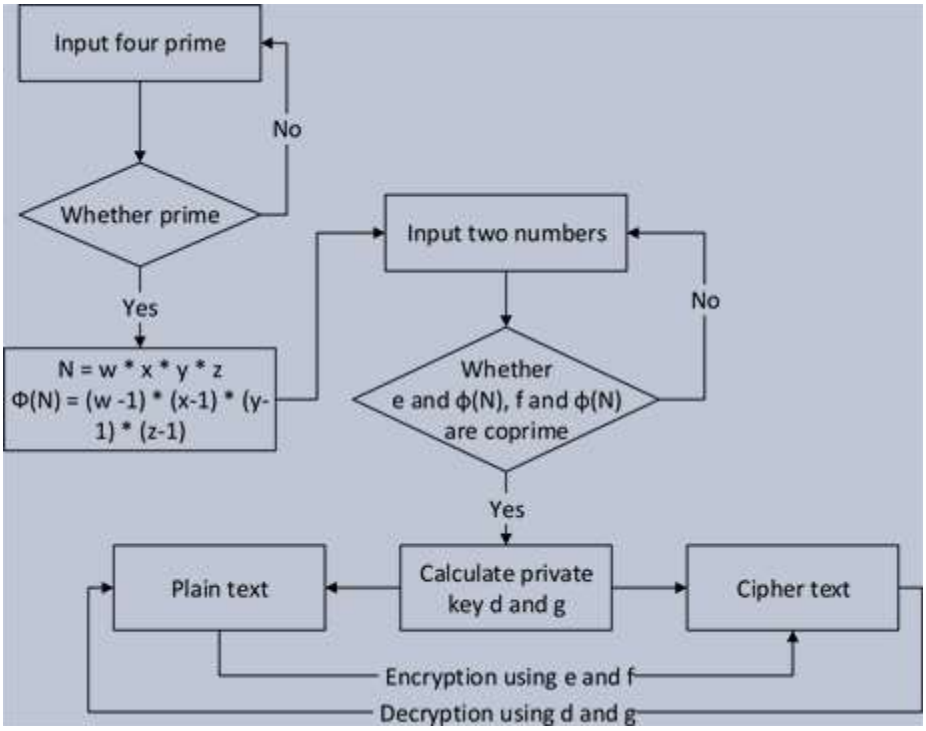


Figure.3

Figure.3 Shows the overall process and flow of modifies RSA Algorithm Encryption and Decryption. It give information about Key Generation process and how many keys will be used and what types of public and private key will be genrated.

Encryption:

Plain text : M < n

Cipher text : C= M**e mod n.

Decryption:

Cipher text to Plain text: M=C**d mod n.

Here encryption and decryption using division operation instead of using that use of successive subtraction which is reduce the mathematical steps. And also to achieve the high computational speed.

AES Algorithm:

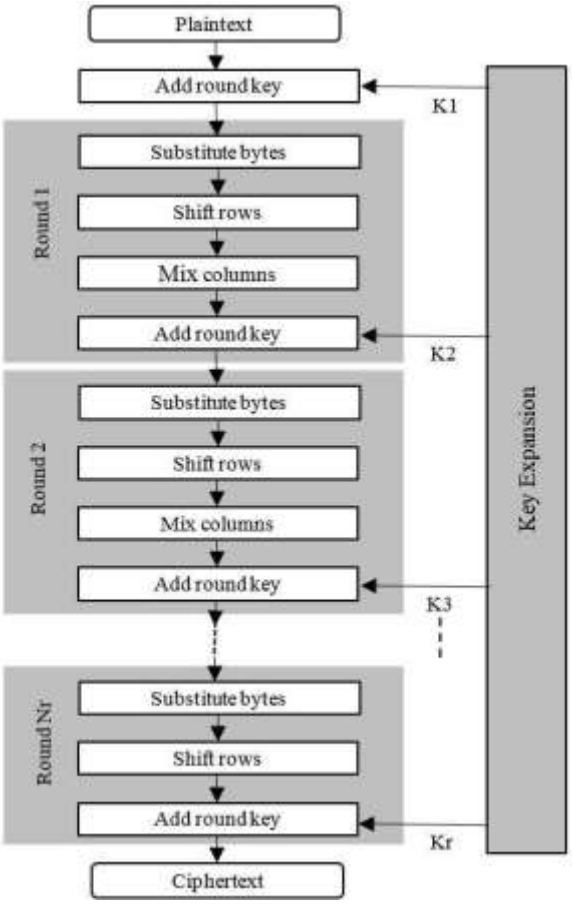


Figure.4

Figure.4 This is overall diagram for AES Algorithm which consist of n round and each round has 4 steps.this shows what are all steps are performed in each round of AES Algorithm.

Encryption :

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

[b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15]

Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes:

This step implements the substitution.
In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

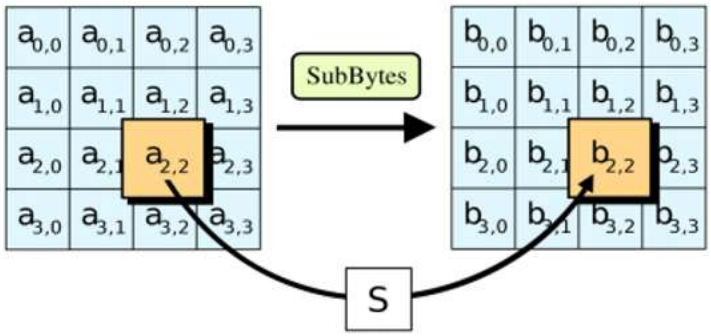


Figure.5

ShiftRows :

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

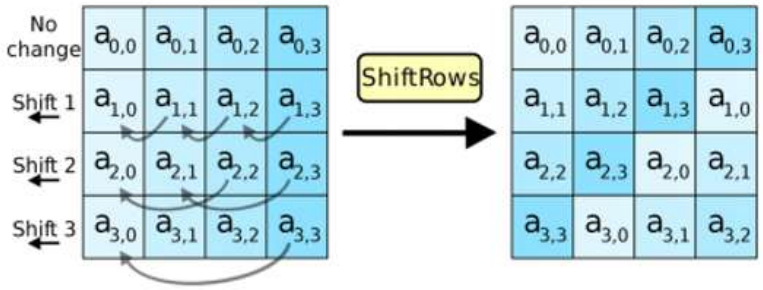


Figure.6

MixColumns:

This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.
This step is skipped in the last round.

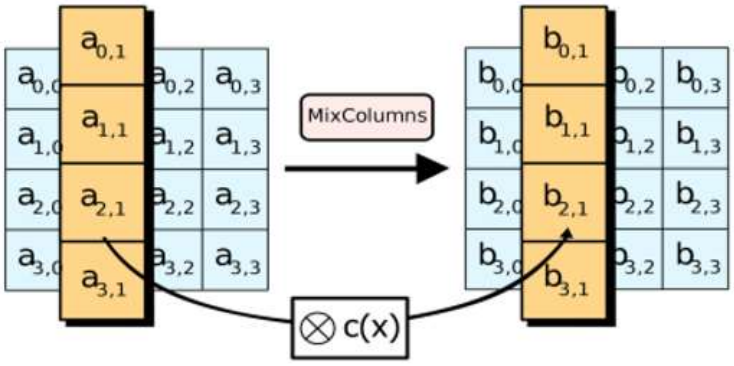


Figure.7

Add Round Keys :
Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.

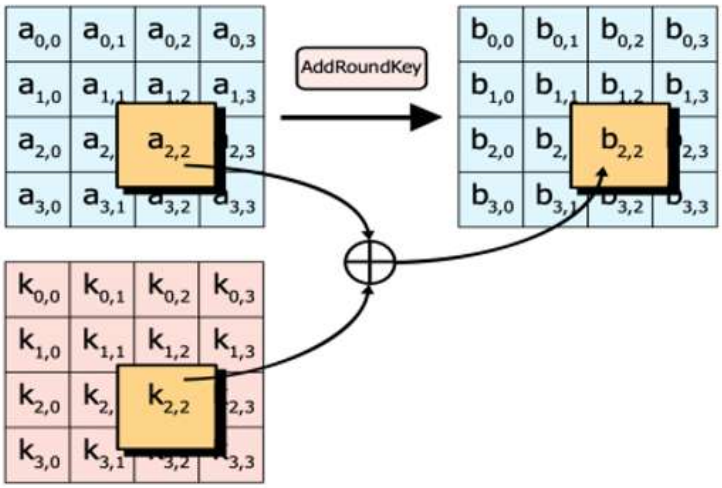


Figure.8

Cipher Feedback Mode(CFB):

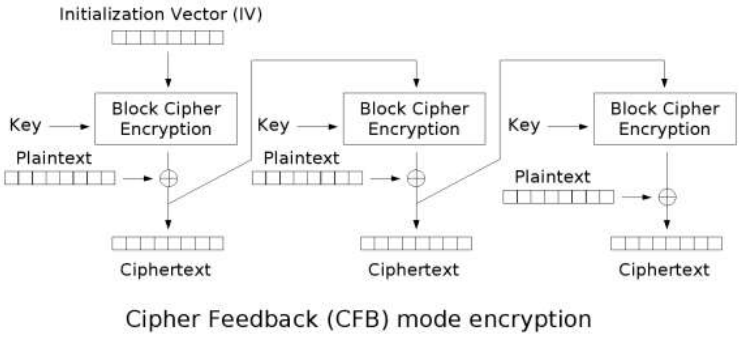


Figure.9

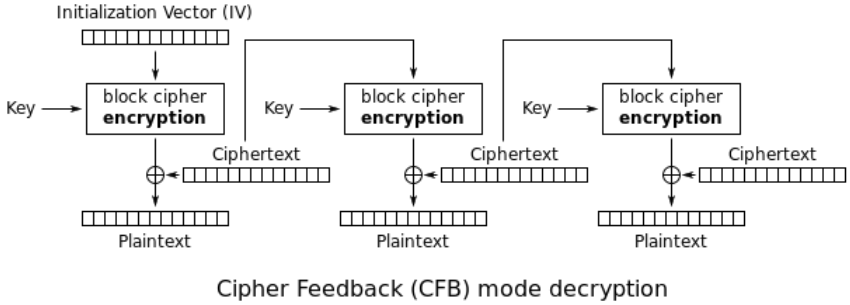


Figure.10

4.4 Innovative Idea:

- Accessibility, reliability, confidentiality and secrecy of data are the main aspects that should be maintained in audio security.
- The use of four prime number will give the ability to the modified encryption technique to provide more security in accessing. Many experiments have been done under this proving Modified RSA encryption Algorithm using four keys to be faster and efficient than the original encryption and decryption process.
- By applying this approach we can achieve the high computational speed and reduce the complexity of the mathematical steps.
- This can help bolster the security of the cryptosystem and can confuse hackers who are already familiar with the RSA encryption system.
- Hybrid algorithms do not have a defined decryption method, since the data is itself encrypted many times it is impossible to crack the key. Thus the user data will be secure.
- Our Encryption algorithm is more secure than the existing algorithms.
- The comparison of a traditional RSA and Four keys RSA showed the relatively high security and alterations in equation formation for 4k RSA algorithm. All the case observations concluded that any imposter might get successful to acquire encrypted shares of the networks. But that person won't be able to recover secret text while lacking a private key accessibility.

A wide variety of attacks are possible on RSA which includes brute force attack, timing attack etc.. The time needed to break an RSA system is equivalent to the time needed for finding the prime numbers used.This introduces the requirement of factoring the product "N". Elliptic Curve factorization Method (ECM)and General Number Field Sieves (GNFS) is used commonly for factoring "N". These are the fastest known factoring methods. Even though an attacker can factorize "N" by using those methods but it is still not sufficient enough in the computation of two arbitrary component "e" and "f". Above factorization technique can be used to find "w", "x", "y", "z" but "e" and "f" can only be found by an exhaustive brute force attack. In other words,

$\Omega system = \Omega w,x,y,z + \Omega bruteforce$
Here,
Ω_{system} = Time needed to break the system
$\Omega_{w,x,y,z}$ = Time needed to find w, x, y, z using GNFS or ECM
$\Omega_{brute\ force}$ = Time needed for brute force attack for finding e, f

The important observation is, Modified RSA (MRSA) involves four primes "w", "x", "y", "z" and two random numbers "e", "f" for encryption whereas the original RSA involves only two prime numbers "w", "x" and only one random number "e" for encryption. So, the time needed to break Modified RSA (MRSA) algorithm will be greater than the time needed to break the original RSA at least by a factor of 2. And this will make the proposed Modified RSA (MRSA) algorithm more secure than the original RSA algorithm.

5.Implementation Details and Analysis

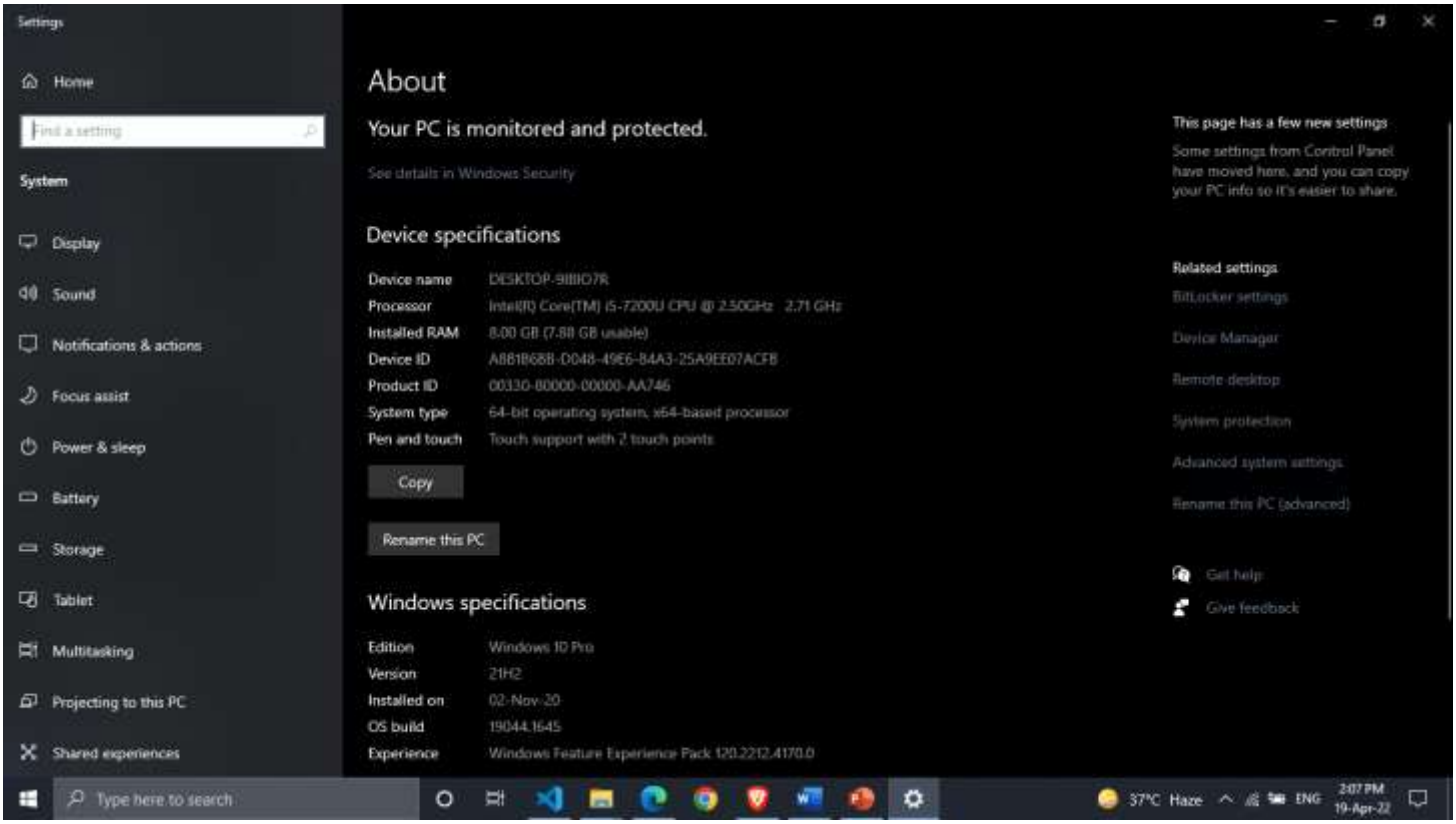
5.1 Software details and Screen shots with respective Description

Hardware and software requirements:

- Hardware Requirements:
- Ram : 4GB (minimum)
- Internet connectivity

Software Requirements:

- Windows7 and above or Mac OS X and above or Linux ,Ubuntu .
- Python Environment
- Python IDE (VS Code)
- Python jupyter Book



Modules Required:

You should install this all modules in your VS code or other IDE which you are using.

Command : pip install module/library name

```
from scipy.io import wavfile
import numpy as np
import matplotlib.pyplot as plt
import math
import sounddevice as sd
import random
import string
from Crypto.Cipher import AES
```

5.2 Test cases and Analysis in terms of Table of Graph

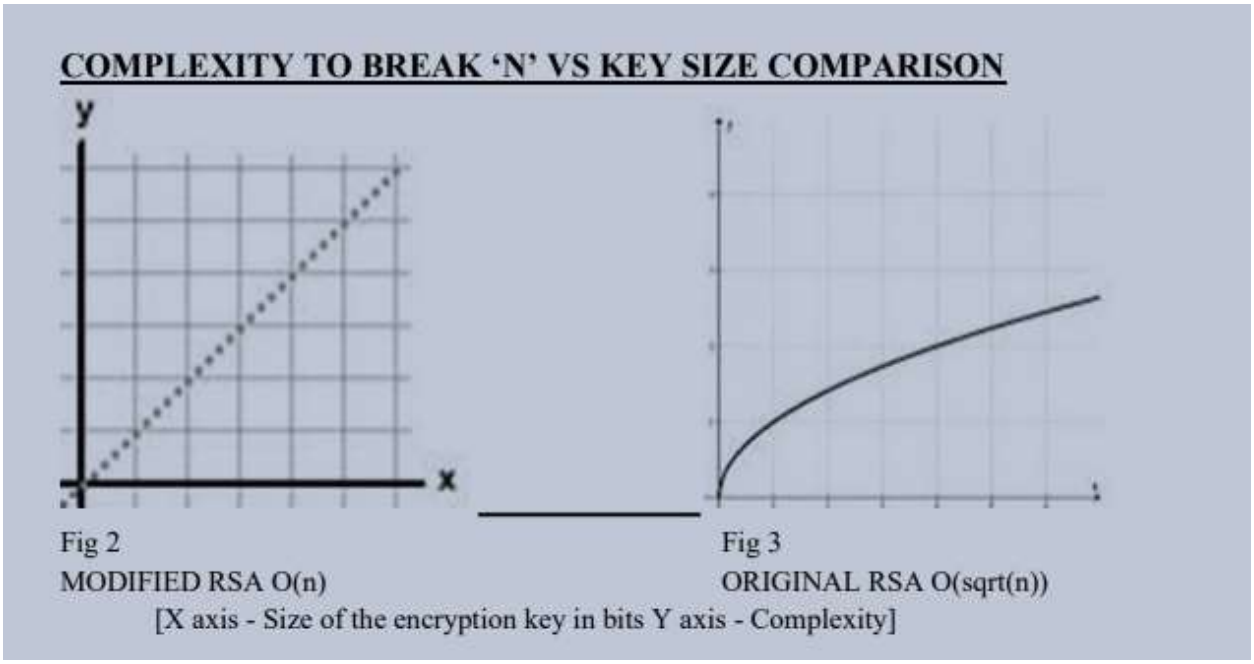


Figure.11

Figure.11 This figure shows the time or time complexity required to break the key / n vs key size comparision.

5.3 Performance Analysis (if applicable compare to existing system)

The algorithm (RSA and MRSA) have different important parameter affecting its level of security and speed. Increasing the modulus length invoke complexity of decomposing it into factor. Thus, also increase the length of the private key and hence difficult to detect the key. The RSA and Modified RSA (MRSA) parameter changes depend on time and others remain fixed to study the relative s.

Length of w, x (in bits)	Analyzing time for RSA algorithm		
	Key generation time (in ms)	Encryption time (in ms)	Decryption time (in ms)
100	76.63	0.16	0.25
128	90.46	0.17	0.28
256	94.96	0.35	0.96
512	177.47	0.56	5.2
1024	570.90	1.69	26.18
2048	4201.47	3.32	130.83
4096	54,368	11.17	1116.24

Table 1. Performance of RSA.

Length of w, x, and z (in bits)	Analyzing time for Modified RSA (MRSA) algorithm		
	Key generation time (in ms)	Encryption time (in ms)	Decryption time (in ms)
100	244	0.28	1.59
128	252.33	0.66	2.89
256	257.8	1.46	14.26
512	386.8	3.00	87.94
1024	1268.6	7.79	446.32
2048	7098.6	21.90	2472.70
4096	161,913	56.87	19,983.37

Table 2. Performance of Modified RSA (MRSA).

Comparing the above tables, it can be concluded that the time of key generation of Modified RSA (MRSA) is higher than that of RSA. The higher key generation time of Modified RSA (MRSA) can be seen as an advantage by the fact that the time to break the system is high because of the extra complexity added.

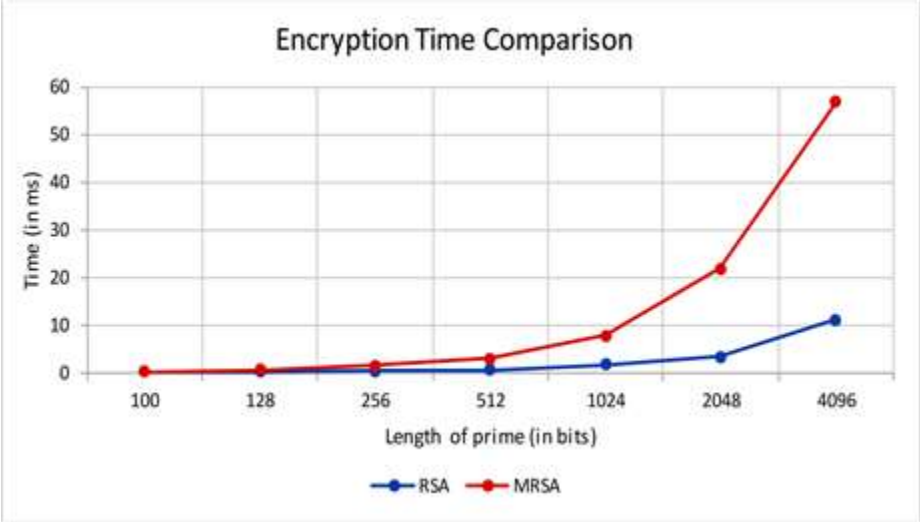


Figure.12

Figure 12 depicts the encryption time comparison between RSA and proposed Modified RSA (MRSA) scheme. It illustrates that, for the lower bit length of prime numbers, two algorithms consume the almost identical amount of time. But with the increase of bit length, the difference between curves rises rapidly.

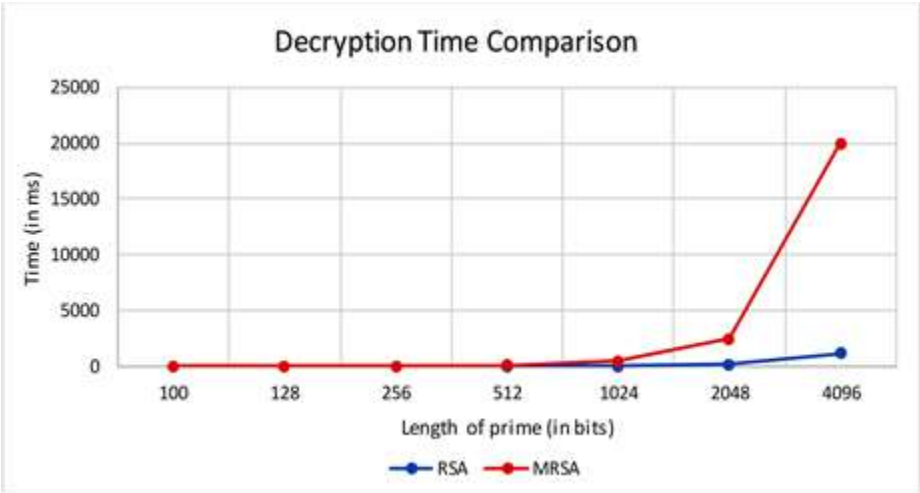


Figure.13

Figure 13 shows the decryption time comparison between RSA and proposed Modified RSA (MRSA) scheme. It demonstrates the almost identical amount of time consumed by RSA and Modified RSA (MRSA) for the lower bit length of prime numbers. With the increase of bit length, the difference between curves elevates rapidly. From the above graphs, it can be easily seen that encryption and decryption times are higher than RSA. The increase in time is adaptable because it increases the security to a great extent in the proposed Modified RSA (MRSA) method.

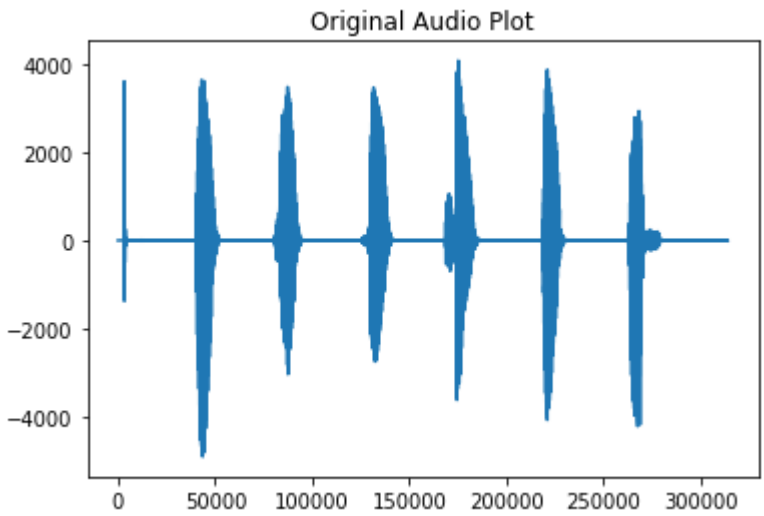
Getting Dependencies

In [ ]:
```
from scipy.io import wavfile
import numpy as np
import matplotlib.pyplot as plt
import math
import sounddevice as sd
import random
import string
from Crypto.Cipher import AES
```

Input for AES.

In [ ]:
```
fs, data = wavfile.read('audio.wav')
plt.plot(data)            # fs = sampling frequency = 44.1kHz
plt.title("Original Audio Plot")
```

Out[ ]:
```
Text(0.5, 1.0, 'Original Audio Plot')
```



In [ ]:
```
with open('audio.wav', 'rb') as fd:
    contents = fd.read()
```

Getting ready with AES

In [ ]:
```
AES_KEY = ''.join(random.choice(string.ascii_uppercase + string.ascii_lowercase + strin

AES_IV = ''.join(random.choice(string.ascii_uppercase + string.ascii_lowercase + string
```

In [ ]:
```
print("AES Key is ", AES_KEY)
print("AES Initialization vector is ", AES_IV)
```

```
AES Key is  aDmxEVMgZM1OSkXTJ4X0XGjn7UFkN5gu
AES Initialization vector is  3AlBZgC3LkdWvJli
```

Encryption using AES

In [ ]:
```
encryptor = AES.new(AES_KEY.encode("utf-8"), AES.MODE_CFB, AES_IV.encode("utf-8"))
encrypted_audio = encryptor.encrypt(contents)
```

In [ ]:
```
with open('encrypted_audio_file.wav', 'wb') as fd:
    fd.write(encrypted_audio)
print("A file titled 'encrypted_audio_file.wav' is generated which is the encrypted aud
```

```
A file titled 'encrypted_audio_file.wav' is generated which is the encrypted audio to be
communicated
```

In [ ]:
```
with open('encrypted_audio_file.wav', 'rb') as fd:
    contents = fd.read()
```

In [ ]:
```
fs, data = wavfile.read('audio.wav')
k = np.asarray(data, dtype = np.int32)
#print(k)
```

Generate Public and Private Key

In [ ]:
```
p1 = int(input("Enter first prime number: "))
p2 = int(input("Enter second prime number: "))
p3 = int(input("Enter third prime number: "))
p4 = int(input("Enter forth prime number: "))

n = p1*p2*p3*p4
print("n = p1*p2 = ",n)

e = int(input("Enter a small, odd number, co-prime with n: "))
k = int(input("Enter value of k:"))

phi = (p1-1)*(p2-1)*(p3-1)*(p4-1)
print("phi = ",phi)

d = int((k*phi+1)/e)

print("d= ",d)

public_key = n,e
private_key = n,d

print("Public Key = ", public_key)
print("Private Key = ",private_key)
```

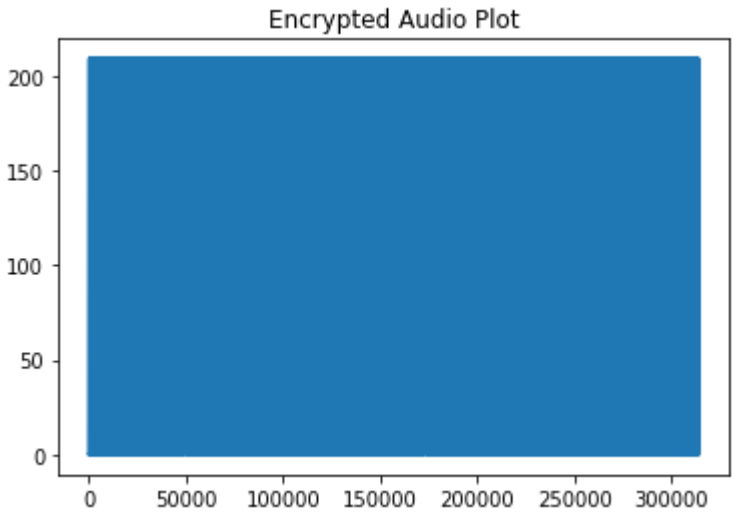
```
n = p1*p2 =  210
phi =  48
d=  80
```

```
Public Key =  (210, 3)
Private Key =  (210, 80)
```

Encrypt message using public key

In [ ]:
```
encrypted=[]
for i in data:
    encrypted.append((i**e)%n)
#encrypted = (data**e)%n
#print(encrypted)
plt.plot(encrypted)
plt.title("Encrypted Audio Plot")
```

Out[ ]:
```
Text(0.5, 1.0, 'Encrypted Audio Plot')
```



Write the Encrpyted File into an audio file

In [ ]:
```
with open('encrypted_audio_file.wav', 'wb') as fd:
    fd.write(encrypted_audio)
print("A file titled 'encrypted_audio_file.wav' is generated which is the encrypted aud
```

```
A file titled 'encrypted_audio_file.wav' is generated which is the encrypted audio to be
communicated
```

Decryption

In [ ]:
```
#RSA Decryption:XXX
```

In [ ]:
```
'''fs, Data = wavfile.read('encrypted_audio_file.wav')
plt.plot(Data)
print(Data)
ke = np.asarray(Data, dtype = np.int32)'''
```

Out[ ]:
```
"fs, Data = wavfile.read('encrypted_audio_file.wav')\nplt.plot(Data)\nprint(Data)\nke =
np.asarray(Data, dtype = np.int32)"
```

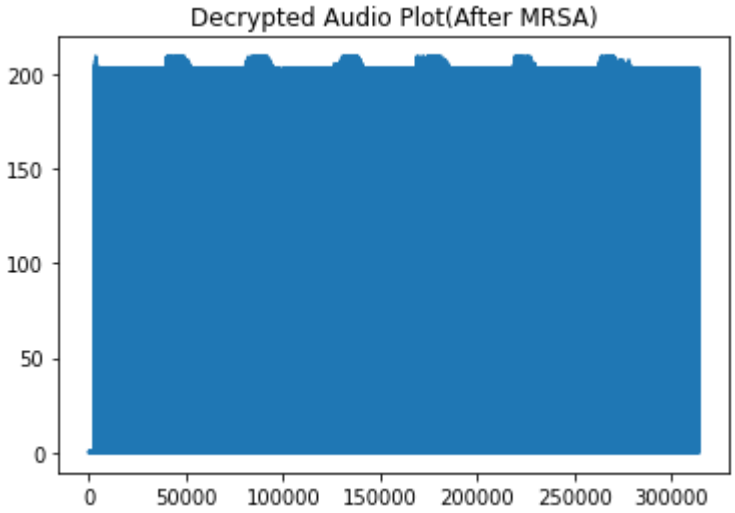
In [ ]:

```
# Python program to compute
# factorial of big numbers

# Maximum number of digits in
# output
MAX=100000

# This function multiplies x
# with the number represented by res[].
# res_size is size of res[] or
# number of digits in the number
# represented by res[]. This function
# uses simple school mathematics
# for multiplication.
# This function may value of res_size
# and returns the new value of res_size
def multiply(x, res, res_size):

        # Initialize carry
        carry = 0

        # One by one multiply n with
        # individual digits of res[]
        for i in range(res_size):
                prod = res[i] * x + carry

                # Store last digit of
                # 'prod' in res[]
                res[i] = prod % 10

                # Put rest in carry
                carry = prod // 10

        # Put carry in res and
        # increase result size
        while (carry):
                res[res_size] = carry % 10
                carry = carry // 10
                res_size+=1

        return res_size


# This function finds
# power of a number x
def power(x,n):

        # printing value "1" for power = 0
        if (n == 0) :
                print("1")
                return

        res=[0 for i in range(MAX)]
        res_size = 0
        temp = x

        # Initialize result
        while (temp != 0):
```

```
                res[res_size] = temp % 10;
                res_size+=1
                temp = temp // 10


        # Multiply x n times
        # (x^n = x*x*x....n times)
        for i in range(2, n + 1):
                res_size = multiply(x, res, res_size)

        print(x , "^" , n , " = ",end="")
        for i in range(res_size - 1, -1, -1):
                print(res[i], end="")

#exponent = 100
#base = 2
#power(base, exponent)

decrypted = (data**d)%n
plt.plot(decrypted)
print(decrypted)
plt.title('Decrypted Audio Plot(After MRSA)')
```

Out[ ]:

```
[  0   0   1 ... 141 141   0]
Text(0.5, 1.0, 'Decrypted Audio Plot(After MRSA)')
```



Write the Decrpyted File into an audio file

In [ ]:

```
encrypted = np.asarray(encrypted,dtype=np.int16)
wavfile.write('decrypted.wav',fs,encrypted)
print("A file titled 'decrypted.wav' is generated which is analog of the audio")
```

```
A file titled 'decrypted.wav' is generated which is analog of the audio
```

Loading

In [ ]:

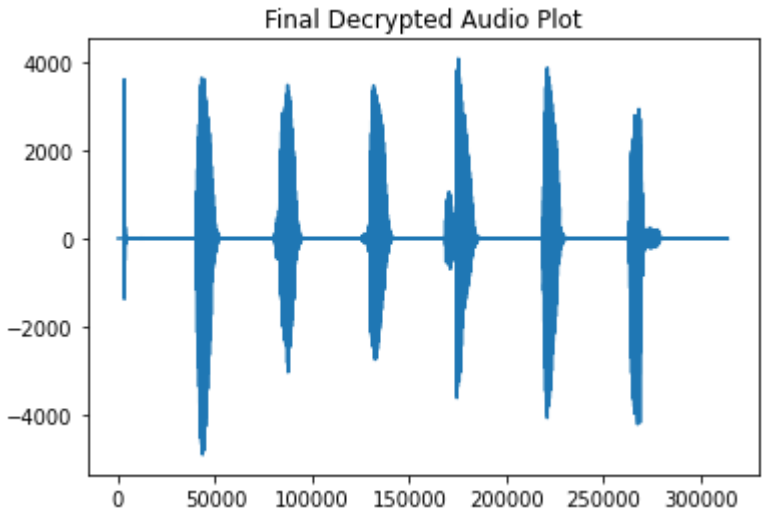
```
with open('encrypted_audio_file.wav', 'rb') as fd:
    contents = fd.read()
```

Decryption of data

In [ ]:
```
decryptor = AES.new(AES_KEY.encode("utf-8"), AES.MODE_CFB, AES_IV.encode("utf-8"))
decrypted_audio = decryptor.decrypt(contents)
```

In [ ]:
```
with open('decrypted_audio_file.wav', 'wb') as fd:
    fd.write(decrypted_audio)
```

In [ ]:
```
fs, data = wavfile.read('decrypted_audio_file.wav')
plt.plot(data)            # fs = sampling frequency = 44.1kHz
plt.title("Final Decrypted Audio Plot")
data_1 = np.asarray(data, dtype = np.int32)
```



In [ ]:
```
sd.play(data, fs)
```

Conclusion and Future Work:

Accessibility, reliability, confidentiality and secrecy of data are the main aspects that should be maintained in audio security.The use of four prime number will give the ability to the modified encryption technique to provide more security in accessing. Many experiments have been done under this proving Modified RSA encryption Algorithm using four keys to be faster and efficient than the original encryption and decryption process.By applying this approach we can achieve the high computational speed and reduce the complexity of the mathematical steps.

This can help bolster the security of the cryptosystem and can confuse hackers who are already familiar with the RSA encryption system.Hybrid algorithms do not have a defined decryption method, since the data is itself encrypted many times it is impossible to crack the key. Thus the user data will be secure.Our Encryption algorithm is more secure than the existing algorithms.The comparison of a traditional RSA and Four keys RSA showed the relatively high security and alterations in equation formation for 4k RSA algorithm. All the case observations concluded that any imposter might get successful to acquire encrypted shares of the networks. But that person won't be able to recover secret text while lacking a private key accessibility.

Audio Transmission can be done more securely in different communication apps or in communication networks , Client - Server Networks .

In voice Assistant when you speak something then your voice message(data) which is being stored can be access by hacker or unauthorised person so to secure that we can apply this method on voice data to store them securely.

Voice Enabled E-mail : Voice-enabled e -mail uses voice recognition and speech synthesis technologies to enable users to access their email from any telephone.

Telecommunication and Multimedia : Vocal information can be accessed over the telephone with the help of speech-to-text systems.

Games and Education : A speech-to-text system can make tedious jobs streamlined and simplified. In the field of study and sports, synthesized speech can be used.

IoT devices: Audio is an integral factor in IoT devices which need user input for working, audio encryption will be integral for the security.

The same implementation could be used in a network to encrypt the files travelling through it for example, attachments travelling through an email could be secured using the hybrid encryption along with the existing email security provided by the mail server or using it alone. The security of RSA depends on factoring the large number. This research works based on "n" distinct prime numbers instead of two prime numbers and it increases the attacking time to find the large

prime number. The key generation of Modified RSA (MRSA) depends on large factor value “N” thus it needs higher key generation time. The higher the key generation time increases the time need to break the system which makes the system stronger. The double encryption and decryption procedure of Modified RSA (MRSA) is simple compared to the RSA algorithm thus it is not overhead on the system. Encryption and decryption also take more time than RSA algorithm. The accomplishment of the algorithm is measured with reference to time taken for brute force attack. Limitation of this proposed schema is it will not work properly unless “n” distinct prime numbers are considered. To enhance the security of RSA algorithm by adding some extra factors in encryption and decryption process can be a good future work.

References:

Audio Encryption Using Ameliorated ElGamal Public Key Encryption Over Finite Field by Motilal Singh Khoirom,Dolendro Singh ,Laiphrakpam,Themrichon Tuithung

- Multilayer security using RSA cryptography and dual audio steganography | IEEE Conference Publication | IEEE Xplore
- https://www.researchgate.net/publication/271917701_Audio_steganography_with_AES_for_real-time_covert_voice_over_internet_protocol_communications
- https://www.researchgate.net/publication/266228471_Implementation_of_RSA_Algorithm_for_Speech_Data_Encryption_and_Decryption
- https://doi.org/10.29121/ijetmr.v5.i7.2018.259

Khalil, M.I. Real-Time Encryption/Decryption of Audio Signal, I. J. Computer Network and Information Security, 2016, 25-31.

Tang, S.Y. (1), Y.J. (1) Jiang, L.P. (1) Zhang, and Z.B. (2) Zhou. “Audio Steganography with AES for Real-Time Covert Voice over Internet Protocol Communications.” Science China Information Sciences 57, no. 3: 1–14. Accessed January 31, 2022. doi:10.1007/s11432-014-5063-2.

Khoirom, Motilal Singh, Dolendro Singh Laiphrakpam, and Themrichon Tuithung. “Audio Encryption Using Ameliorated ElGamal Public Key Encryption Over Finite Field.” Wireless Personal Communications 117, no. 2 (March 15, 2021): 809–23. https://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=149049591&site=eds-live.

Modules:

AES--pycryptodome:

https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html

Numpy:

https://numpy.org/

https://www.w3schools.com/python/numpy/numpy_intro.asp

Soundevice:

https://python-sounddevice.readthedocs.io/en/0.4.4/

Matplotlib:

https://matplotlib.org/#:~:text=Matplotlib%20is%20a%20comprehensive%20library,can%20zoom%2C%20pan%2C%20update.