# *AUDIO ENCRYPTION AND DECRYPTION TOOL*

# USER MANUAL

## The Project Team

This product was developed by a team of 3rd-year B.Tech Computer Science students at **VIT Chennai**, comprising **Shyam Sundar D, Sagar GV and DhanushKodi** under the guidance and supervision of **Dr. Subbulakshmi T**.

# Table of Content

# Chapter 1 - Introduction

## 1.1 Product Description

Introducing our advanced Audio Encryption and Decryption system that ensures the integrity of your audio files. Audio encryption and decryption is a powerful tool designed to secure audio data by encrypting it using advanced cryptographic algorithms. This tool provides a secure way to protect sensitive audio information such as voice recordings, audio files, and conversations from unauthorized access, theft, or eavesdropping.

Using this tool, audio data can be encrypted with a unique key that only authorized users can access. This ensures that the audio remains confidential and secure, and prevents it from being intercepted or tampered with by hackers or cybercriminals.

In addition to encryption, this tool also provides a decryption feature that allows authorized users to easily access the encrypted audio data by entering the correct key. This feature ensures that authorized users can easily decrypt and access the audio data, while unauthorized users are prevented from accessing it.

Overall, audio encryption and decryption is an essential tool for anyone who wants to protect their audio data from unauthorized access and ensure that it remains confidential and secure.

## 1.2 Product Users

The tool Audio encryption and decryption can be useful for a variety of users, such as:

- **Musicians:** Musicians who want to protect their unreleased music from being leaked or stolen can use audio encryption to encrypt their audio files and decrypt them only when they need to.

- **Record labels:** Record labels can use audio encryption to protect their copyrighted audio content from unauthorized access or distribution.
- **Audio engineers:** Audio engineers can use audio encryption to ensure that the audio files they are working on are secure and cannot be accessed by unauthorized users.
- **Podcasters:** Podcasters who want to keep their audio content private or secure can use audio encryption to protect their podcasts from being accessed or distributed without their permission.
- **Voice actors:** Voice actors can use audio encryption to protect their voice recordings and ensure that they are not stolen or used without their consent.
- **Security professionals:** Security professionals can use audio encryption as a part of their overall security strategy to protect sensitive audio data.
- **Law enforcement agencies:** Law enforcement agencies can use audio encryption to protect their communication channels and ensure that sensitive information is not intercepted or accessed by unauthorized parties.

## 1.3 Purpose Statement

This manual is required for the using this product as it helps the users understand how to use the features of the Audio Encryption and Decryption system effectively. The manual provides detailed instructions on how to install and set up the product, how to add the audio file, how to download the encrypted and decrypted file. It also includes guidelines on how to configure the product to meet specific needs. The user can also have insight about the architecture and working of the product.

The users can also know about the typically intended applications of the product include encryption and decryption of audio files using AES, DES, 3DES,RSA,AES+RSA,ECC,FERNET

## 1.4 Document Usage Description

The document usage description for the Audio encryption and Decryption system provides an overview of the document's purpose and structure, as well as guidance on how to use it effectively. The document is divided into several chapters, each with a specific focus.

- Chapter 1 introduces the product and provides essential information such as the intended audience, applicability, and problem reporting instructions. This chapter provides users with a high-level understanding of the product and what to do in case of any issues.

- Chapter 2 discusses the scope of the document, outlining the areas that the document covers and what it does not. This chapter ensures that users understand the limitations of the document and the product.

- Chapter 3 describes the features of the product, providing a detailed overview of what the product can do. It serves as a reference for users to understand the capabilities of the product and how to utilize them.

- Chapter 4 is the user guide section, which contains instructions for using the product. This chapter includes critical information such as caution and warning instructions, environment compatibility, installation and setup instructions, usage instructions, and related information. The user guide section provides step-by-step guidance on how to use the product effectively and optimize its features.

- Chapter 5 is focused on errors and recovery. It outlines how to identify issues that may arise when using the product and provides guidance on how to address those issues.

- Chapter 6 is the appendix section, which includes additional information on the product's architecture. This section is beneficial for users who want a deeper understanding of how the product works.

Overall, the document usage description emphasizes that each section of the document is interconnected, and users should refer to multiple sections to ensure they have a comprehensive understanding of the product. It also highlights the importance of following the problem reporting instructions provided in Chapter 1 to report any issues or problems encountered while using the product.

## 1.5 Problem Reporting Instructions

If you encounter any problems or issues with our Audio Encryption and Decryption system, please report them to us immediately. This will allow us to investigateand address the issue promptly, ensuring that you have the best possible experience with our product.

To report a problem, please email us at gvsagar2002@gmail.com. In your email, please provide a detailed description of the issue you are facing, including any error messages or other relevant information. If possible, please include screenshots or logs that can help us to diagnose the problem more quickly.

Once we receive your email, we will investigate the issue and provide you with a solution or workaround as soon as possible. We appreciate your feedback and will work diligently to ensure that any issues are resolved promptly and to your satisfaction.

# Chapter 2 - Scope of Product

The Audio encryption and decryption is a tool that can be used to secure audio files from unauthorized access or use. It involves the use of mathematical algorithms to convert audio data into a scrambled form, which can only be deciphered by someone with the appropriate decryption key.

There is a significant scope for audio encryption and decryption in various industries such as music, film, and broadcasting. These industries deal with sensitive audio data that needs to be protected from piracy or unauthorized use. Audio encryption and decryption can provide an effective way to secure such data and prevent it from falling into the wrong hands.

Moreover, audio encryption and decryption can also be used for personal use by individuals who want to secure their audio files, such as personal recordings or voice messages. It can prevent anyone from accessing their audio files without the appropriate decryption key, ensuring their privacy and confidentiality.

# Chapter 3- Features

The Audio encryption and decryption system includes several key features that allow you encrypt and decrypt audio files using many cryptic algorithms. Some of the features of the product include:

- **Encryption Algorithms**: The tool should support various encryption algorithms like AES, Blowfish, etc.
- **Key Management:** The tool should allow users to generate, store and manage encryption keys securely.
- **File Format Support**: The tool should support audio file formats like WAV,, etc.

- **Password Protection:** The tool should allow users to set a password for encrypted files to prevent unauthorized access.
- **Batch Processing:** The tool should support batch processing of multiple files at once.
- **Compression:** The tool should allow compression of encrypted files to reduce their size for easier sharing and storage.
- **Decryption Verification:** The tool should provide an option for users to verify the authenticity of decrypted files to ensure that they have not been tampered with.
- **User Interface:** The tool should have an intuitive and user-friendly interface that is easy to use for both novice and advanced users.
- **Platform Support:** The tool should be compatible with platform like Windows
- **Support for Large Files:** The tool should be capable of encrypting and decrypting large audio files quickly and efficiently upto 200MB .

These features make the Audio encryption and decryption system an effective tool for encryption and decryption of audio files.

# Chapter 4: User Guide

## 4.1 Caution and Warning

- Always keep your encryption key safe and secure. Without the key, you will not be able to decrypt your audio files.
- Do not use simple or easily guessable encryption keys, as they can compromise the security of your audio files.
- Make sure to use the latest version of Audio Encryption and Decryption to ensure optimal performance and security.

## 4.2 Environment Compatibility

- **Hardware:** minimum of 2 GB of RAM and 50 MB of free disk space
- **Operating System:** Windows 7, 8, and 10 and above
- **Software:** System admin access to view file log in the terminal.
- **Network:** No internet connection required
- **Devices:** The tool can be run on PCs (desktops and laptops).
- **Mobile:** Not compatible

### 4.3 Installation and Setup

To use the Audio Encryption and Decryption just download the executable 'GUI.py' from the github repository. The latest release of the product executable is available in the executable folder of the master branchin the github repository. Use the following link to access the github repository - https://github.com/sagar-git-23/audio-encryption-and-decryption-

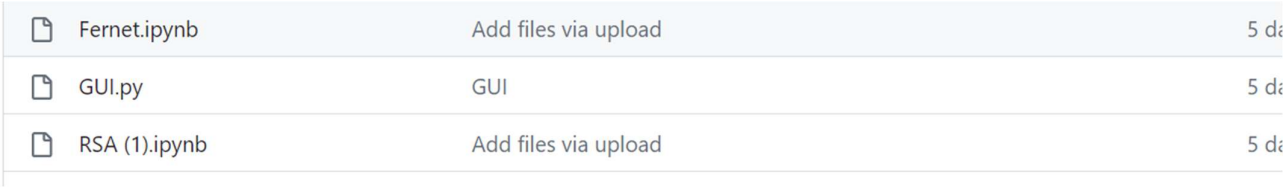



Fig.2. Python file for Audio Encryption and Decryption GUI

## 4.4 User Instructions

### 4.4.1 Running a python file

1. Open the Anaconda promt
2. Change the directory where the downloaded file is present
3. Install streamlit which is a python library
4. After the installsation, run the Gui.py , by  using the comand

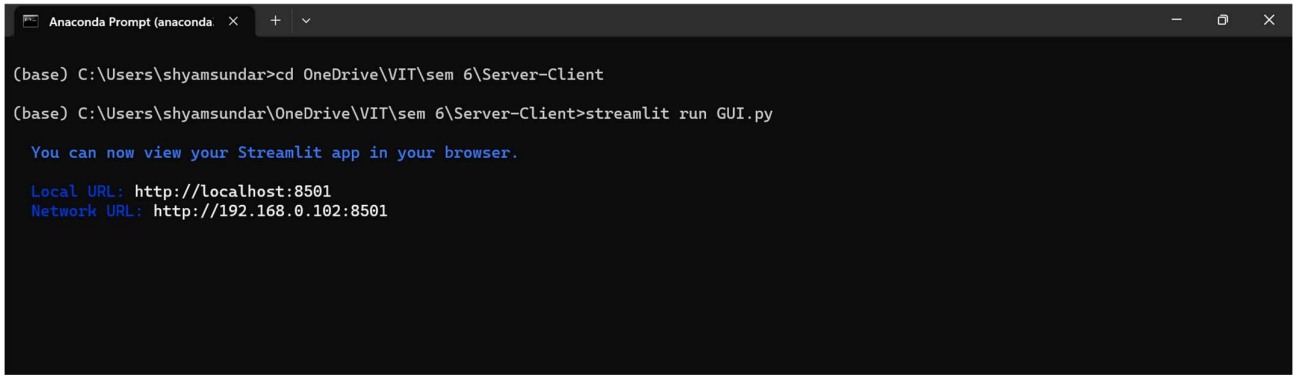



Fig.3. Entering the path of Directory and running

5. To stop the program, press ‘ctrl+c’ and terminate.

### 4.4.2 For Encryption

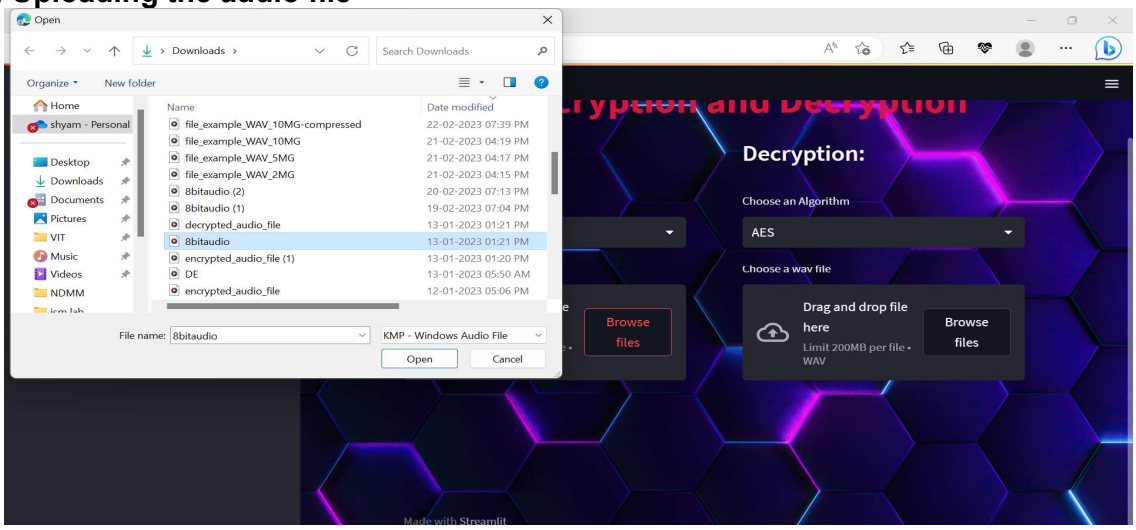
1. Enter the home page
2. Choosing the algorithm for encrypting the audio file
3. Choosing the same algorithm for decryption
4. Uploading the audio file
5. Selecting the key size from the options provided
6. Button to download encrypted audio file
7. Downloaded encrypted audio file is saved

**a) Home screen of the product**

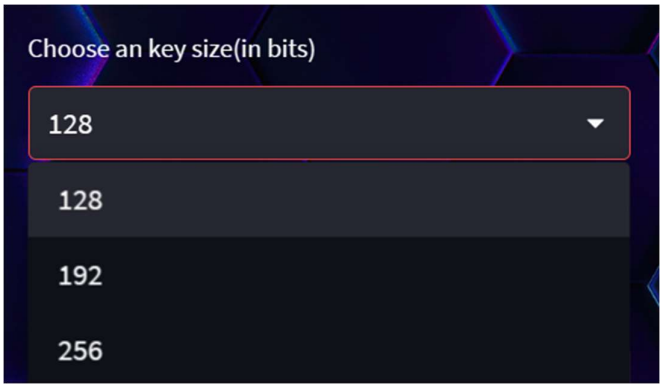
## b) Choosing the algorithm for encrypting the audio file

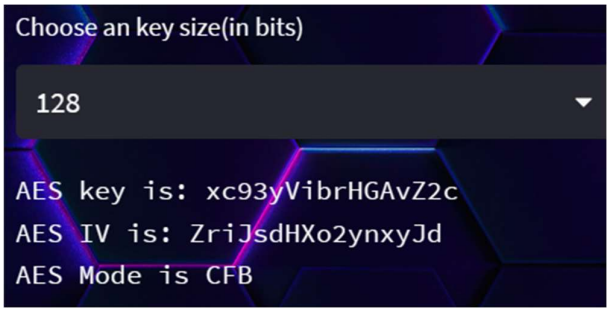

## c) Choosing the same algorithm for decryption

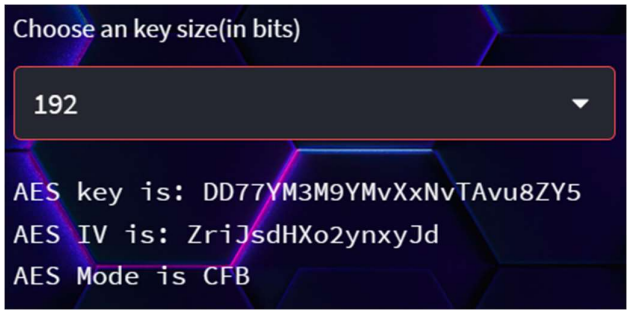

## d) Uploading the audio file

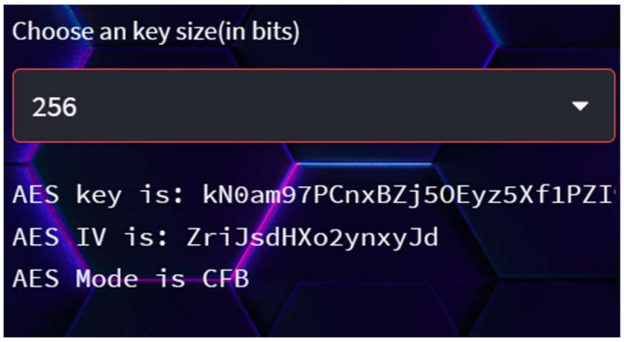


**e) Selecting the key size from the options provided**



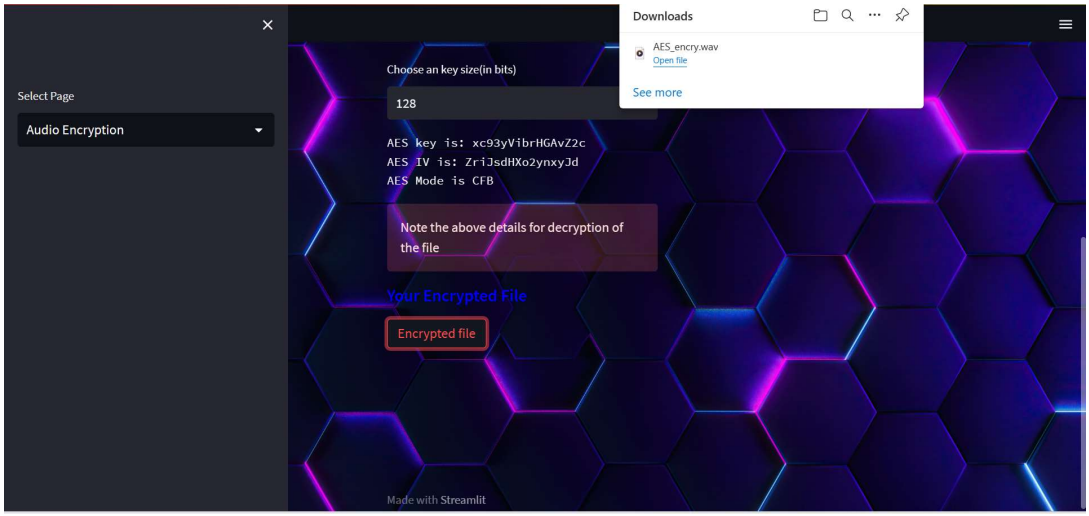




**f) Button to download encrypted audio file**

**g) Downloaded encrypted audio file is saved**



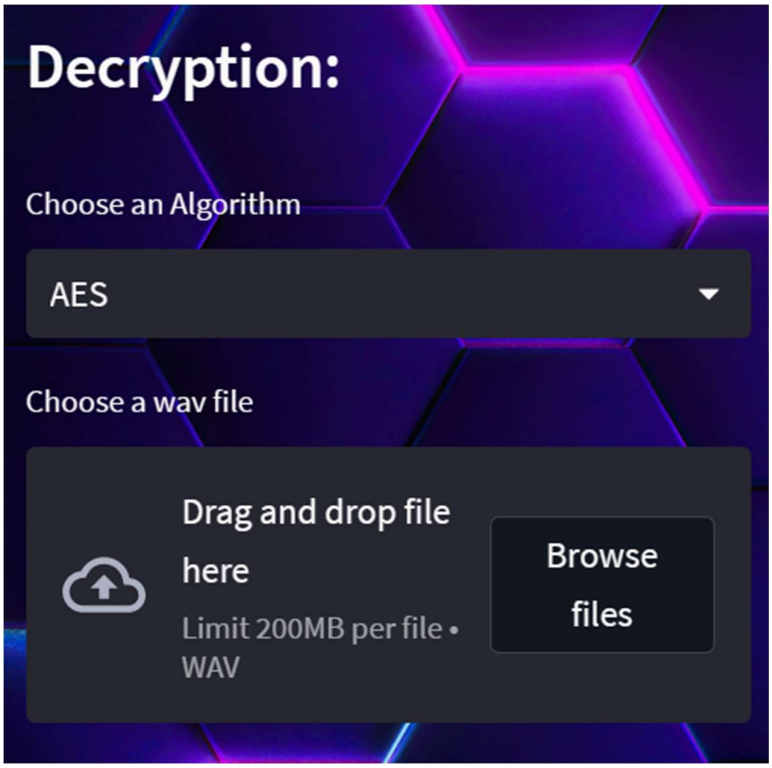

## 4.4.3For Decryption

1. Enter the home page
2. Choosing the algorithm for decrypting the audio file
3. Uploading the audio file which is already encrypted
4. Copy the key and Iv from the encryption
5. Button to download encrypted audio file
6. Downloaded encrypted audio file is saved

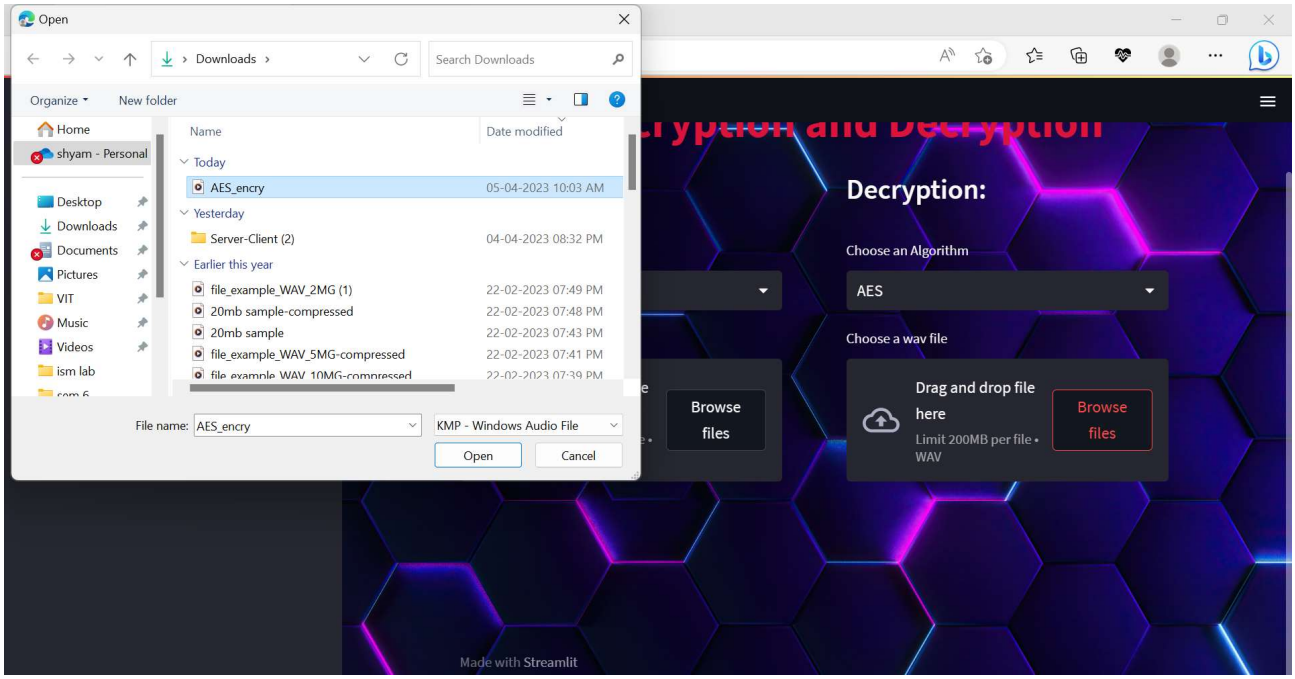
**a) Home screen of the product**

### b) Choosing the algorithm for encrypting the audio file

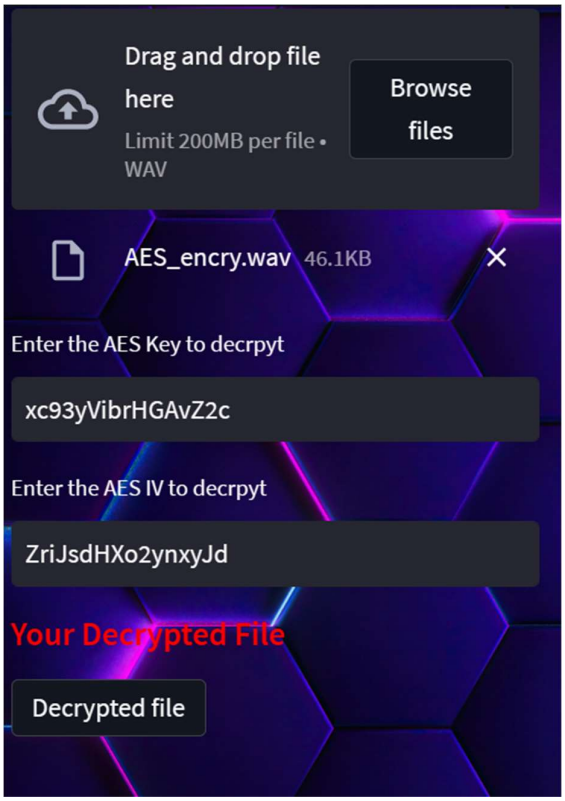

### c) Choosing the same algorithm for decryption



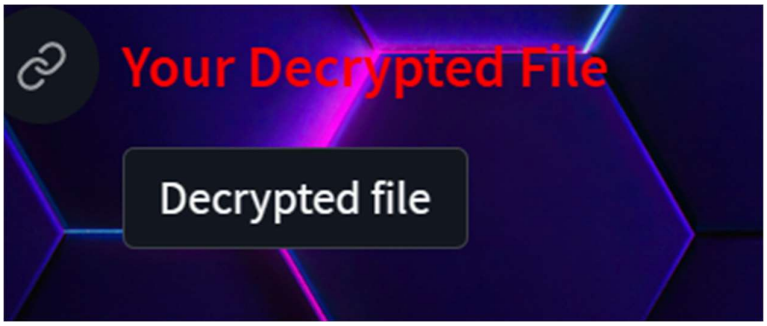
### d) Uploading the audio file



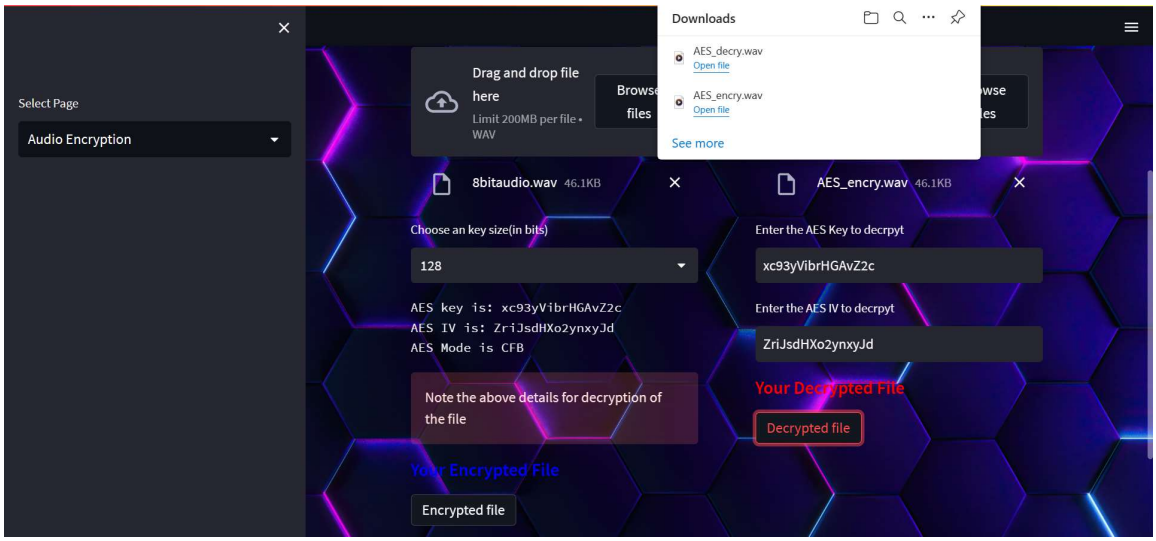

### e) Enter the key and IV from the encrypted



## f) Button to download encrypted audio file




## g) Downloaded encrypted audio file is saved



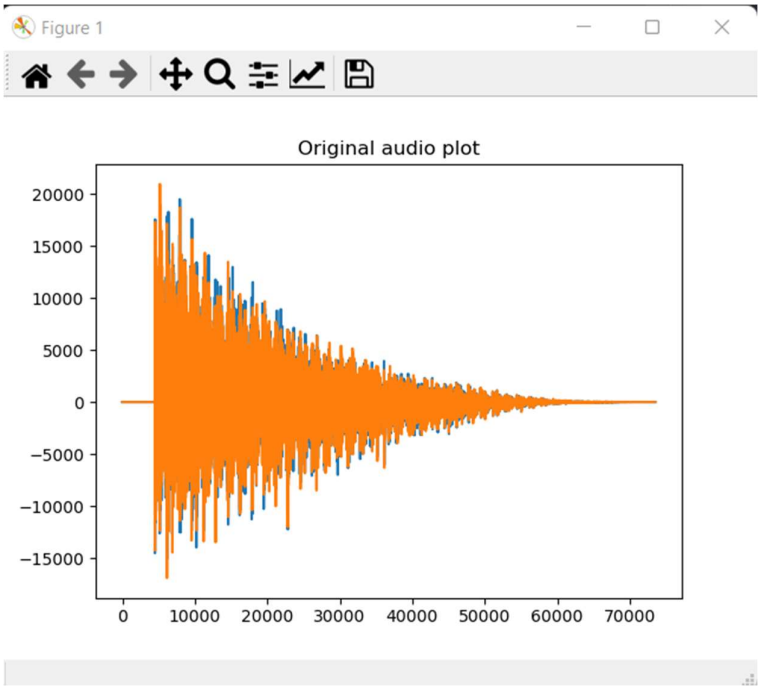
# NOISE ATTACK

- **Sender.py**

```
(base) C:\Users\Sagar>cd Downloads

(base) C:\Users\Sagar\Downloads>cd Server-Client

(base) C:\Users\Sagar\Downloads\Server-Client>python Sender.py
Sender.py:11: WavFileWarning: Chunk (non-data) not understood, skipping it.
  fs,data = wavfile.read('audio.wav') #specify path  #fs-freq of given wav file data-freq of wav file in array
Sender.py:11: WavFileWarning: Incomplete chunk ID: b'\x00', ignoring it.
  fs,data = wavfile.read('audio.wav') #specify path  #fs-freq of given wav file data-freq of wav file in array
Warning: QT_DEVICE_PIXEL_RATIO is deprecated. Instead use:
   QT_AUTO_SCREEN_SCALE_FACTOR to enable platform plugin controlled per-screen factors.
   QT_SCREEN_SCALE_FACTORS to set per-screen factors.
   QT_SCALE_FACTOR to set the application global scale factor.
```

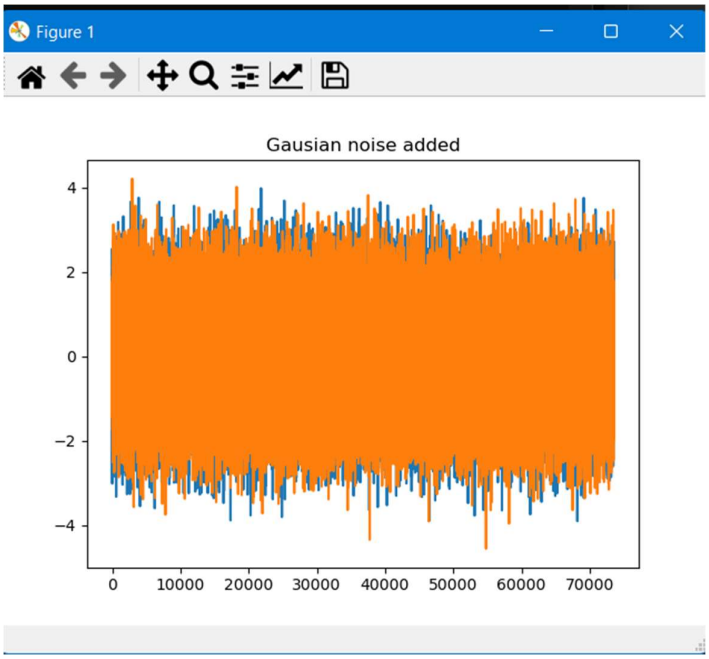



- **From the graph, we can inference that, it is the plot of the original audio file.**
- **And a socket connection is opened, and will be waiting for the Attacker to intercept the audio file**

- **Attacker.py**

```
(base) C:\Users\Sagar\Downloads\Server-Client>python Attacker.py
Traceback (most recent call last):
  File "Attacker.py", line 10, in <module>
    s.connect(("127.0.0.1",65432))
ConnectionRefusedError: [WinError 10061] No connection could be made because the target machine actively refused it

(base) C:\Users\Sagar\Downloads\Server-Client>python Attacker.py
Attacker.py:21: WavFileWarning: Chunk (non-data) not understood, skipping it.
  fs,data=wavfile.read("decrypt.wav")
Attacker.py:21: WavFileWarning: Incomplete chunk ID: b'\x00', ignoring it.
  fs,data=wavfile.read("decrypt.wav")
Warning: QT_DEVICE_PIXEL_RATIO is deprecated. Instead use:
   QT_AUTO_SCREEN_SCALE_FACTOR to enable platform plugin controlled per-screen factors.
   QT_SCREEN_SCALE_FACTORS to set per-screen factors.
   QT_SCALE_FACTOR to set the application global scale factor.
```



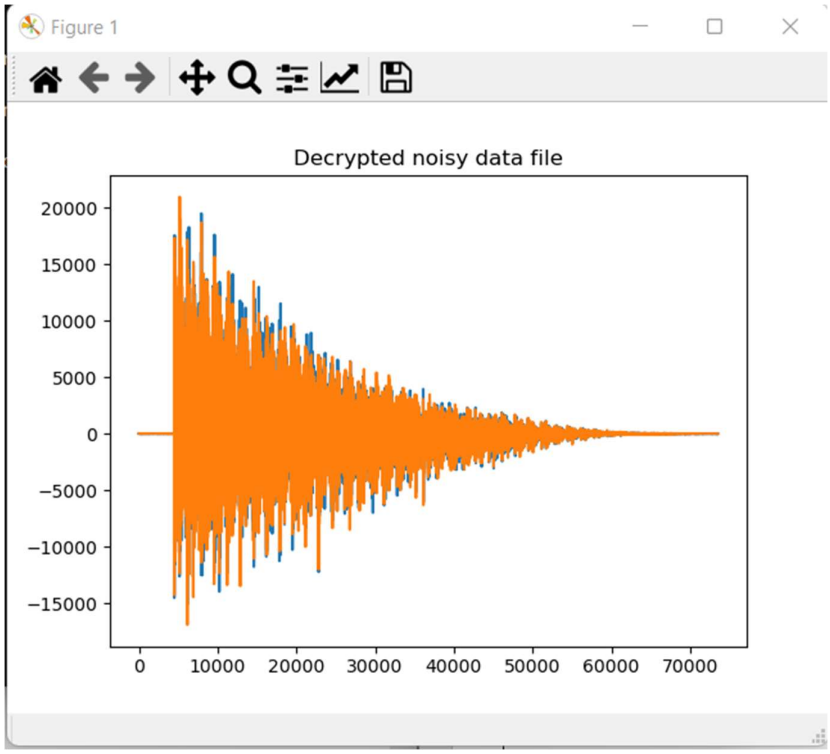

- **From the graph, we can inference that, it is the plot of the encrypted audio file transmitted from the sender.**
- **Now as soon as this session opened, the sender will send the audio file and close the socket connection**
- **And another socket connection is opened, and will be waiting for the receiver to intercept the audio file**

- **Receiver.py**

```
(base) C:\Users\Sagar>cd Downloads

(base) C:\Users\Sagar\Downloads>cd Server-Client

(base) C:\Users\Sagar\Downloads\Server-Client>python Reciver.py
Warning: QT_DEVICE_PIXEL_RATIO is deprecated. Instead use:
   QT_AUTO_SCREEN_SCALE_FACTOR to enable platform plugin controlled per-screen factors.
   QT_SCREEN_SCALE_FACTORS to set per-screen factors.
   QT_SCALE_FACTOR to set the application global scale factor.
```




- **From the graph, we can inference that, it is the plot of the decrypted audio file transmitted from the attacker.**
- **Now as soon as this session opened, the attacker will send the audio file and close the socket connection**
- **We can see that the audio file is unaffected by this noise attack and attacker can't do anything. To the audio file.**

# Chapter 5: Errors And Recovery

## 1. Wrong upload

**Error -** This error occurs when the audio file format is wrong like .mp3,.mp4,etc

**Recovery -** Check if the audio file format for .wav, as the tool only recognize .wav file

## 2. Wrong Key size -

**Error –** The audio file will be decrypted wrong and couldn't produce the results of the encrypted file

**Recovery -** Check if the encrypted key size and decrypted key size are size, and if different it could not produce any results

## 3. Wrong Key and IV -

**Error -** The user may enter different key or IV for encryption and decryption resulting in corrupted audio file

**Recovery –** check whether the entered key and IV are same in both columns to ensure smooth decryption

# Chapter 6 - Appendix

## 6.1. Architecture

**ENCRYPTION** **DECRYPTION**

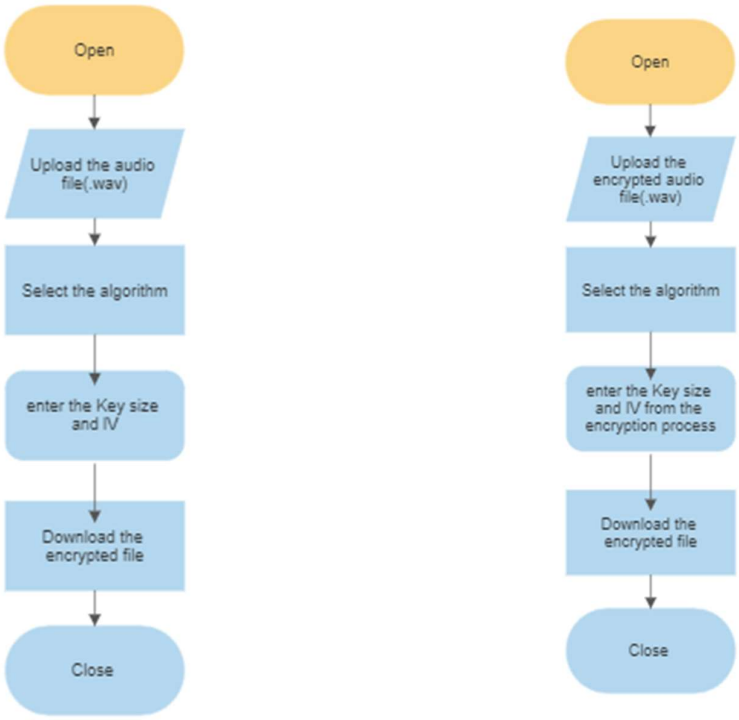



Fig.8. Architecture Diagram