Embedding Data Secretly In Audio File With AES Encryption

MSc Academic
Cyber security

Aditya Jha
Student ID: X19104987

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name:	Aditya		
Student ID:	X19104987		
Programme:	MSc in Cyber Security	**Year:**	2019-2020
Module:	Academic Internship		
Supervisor:	Mr. Imran Khan		
Submission Due Date:	28/09/2020		
Project Title:	Embedding Data Secretly In Audio File With AES Encryption		

qPage Count: 11

Word Count: 806

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:

Date: 17 August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Aditya
Student ID: x19104987

1 Introduction

The configuration manual would talk about the research project, the important methods, evaluations and implementations which have taken place. It proposes a novel approach to hide the data inside the digital medium using the combination of audio steganography and AES algorithm. A Java GUI code is implemented which gives the user options to encrypt and decrypt the data over audio files. To perform the encryption the AES encryption method takes the plain text and encrypt it using either 16-byte, 24 byte or 32 byte of key size. A salt function is added with the AES encryption to provide double security to ciphertext. The audio files are broken down into audio and image frames and further LSB method is applied to insert data into the audio frames. The main advantage of having LSB as a embedding algorithm over methods is that it has the capability to embed large data with least distortion.

2 Configuration of System

2.1 Hardware Configuration

Hardware	Configurations
Processor	Intel i5
Operating System	MaOS
Ram	8 GB
Hard disk	512 Gb
Graphic Card	Intel 1536mb

2.2 Software Configuration

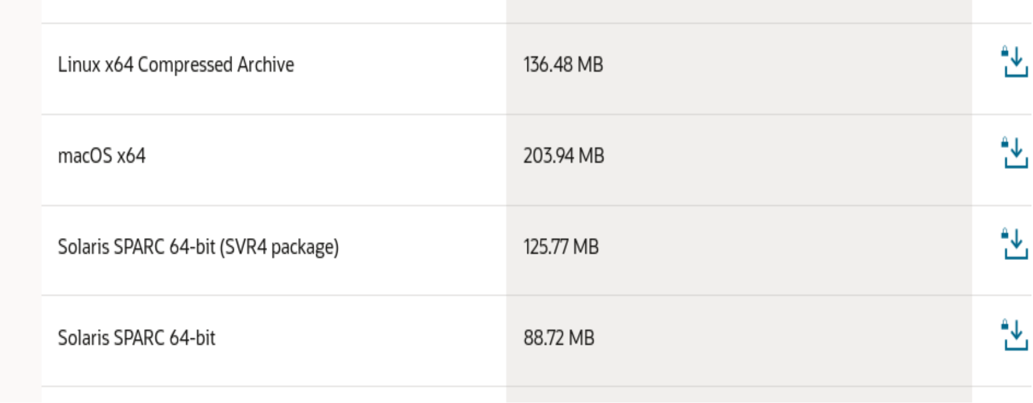
Software	Configurations
Operating System	MaOS
Scripting Language	Java
Scripting Language Version	Java 8

3 Functioning

This part gives us the detailed information about the various processes that were used in this research project and their step by step installing procure based on the requirement of the research project.

Installation of Applications

The latest version of Java can be installed from the link provided below:
https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html



Linux x64 Compressed Archive	136.48 MB
macOS x64	203.94 MB
Solaris SPARC 64-bit (SVR4 package)	125.77 MB
Solaris SPARC 64-bit	88.72 MB

Figure1: Java Executable file

3.1 Working

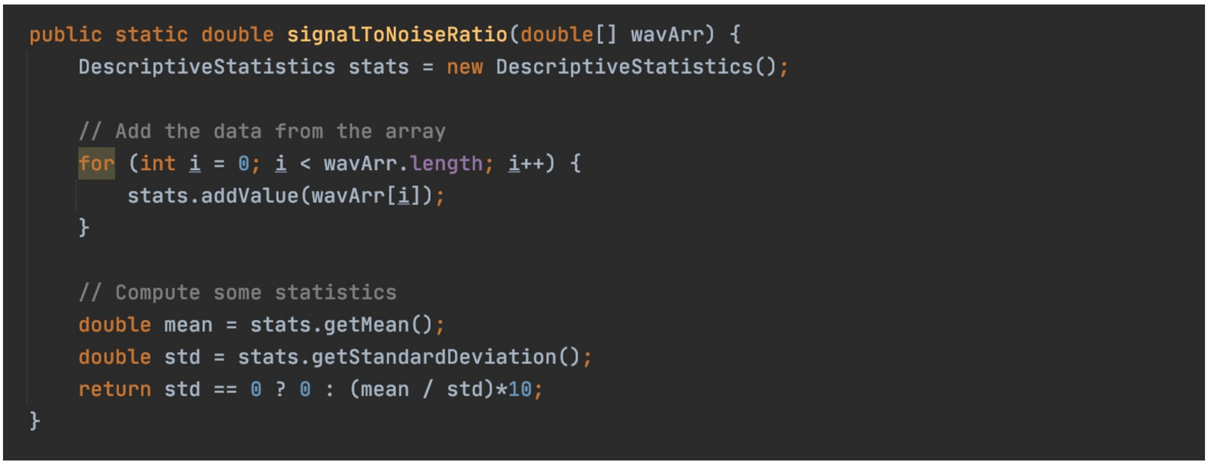
To run the Java program, different libraries were installed. These libraries are prequisite for the code as they provide the path to different modules to be inherited and run in java. The main libraries used to make this research project are the Java.security and Javax.crypto.

Java.security Library provides various classes and interfaces which are used in the security framework. Some of the interfaces include Public key, private key, AlgorithmConstraints etc. Some of the classes includes Codesource, Keystore, Policy, Signature etc.

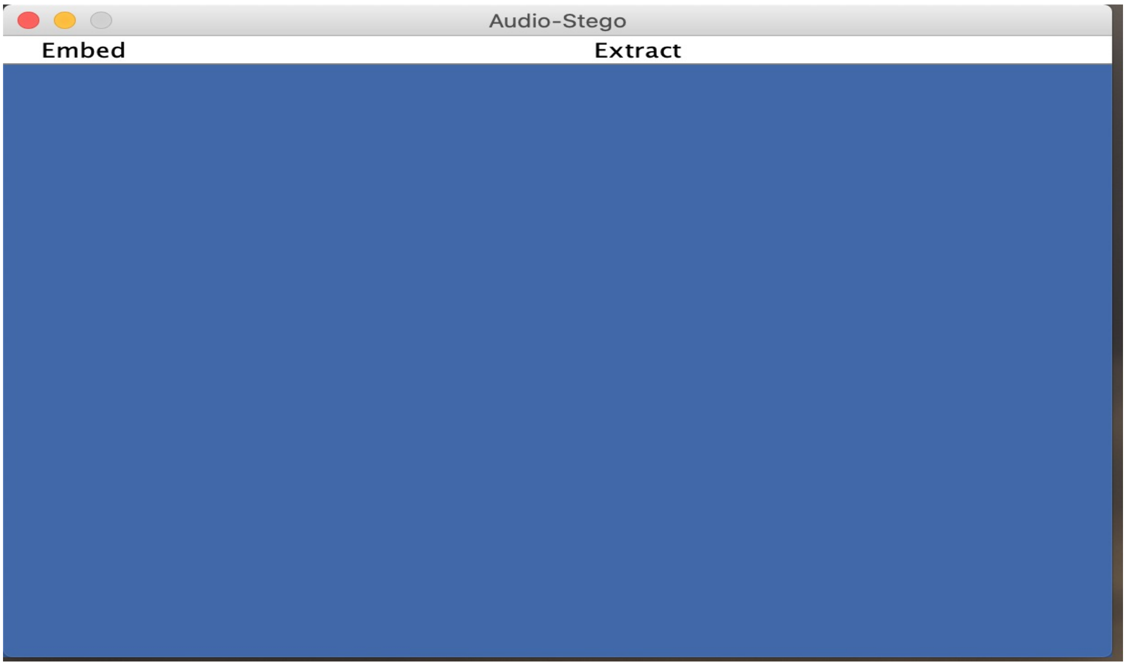
Javax.crypto is a Java library which provides classes and interfaces which are used for operations performed under cryptography. We have used an symmetric cryptographic algorithm in our research project which is AES. This library deals with classes for providing cipher to the plaintext. Apart from encryption this package provides operations for key genration and genrating Message Authentication Code (MAC) [1].

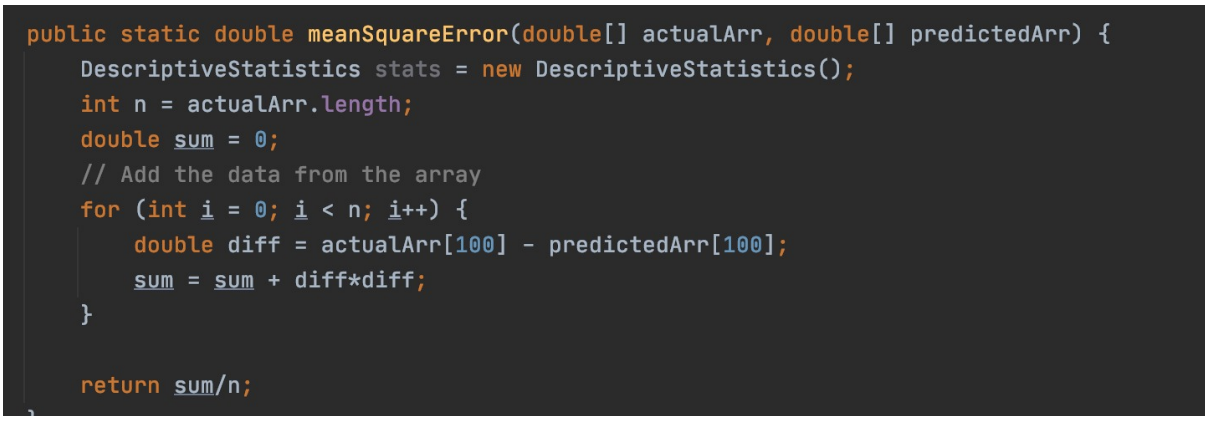
The testing is performed on the proposed research method which calculates the values of SNR and MSE.The SNR gives the signal to noise ration and MSE is the mean square error which is defined by the degradation in the stego frames. The SNR and MSE values are created for each of frame used for embedding the data, these frames are compared with the frames of orignal video and the average is taken to compute the final PSNR and MSE values.
To calculate SNR value
snr = 20 * log10(max_pixel / sqrt(mse))

File Name	File Size	SNR(Sound to noise ratio) value	MSE(Mean Square Error)
speechwav.wav(original)	117 kb	5.636014274227477	0.15479867783745405
speechwav1.wav(Stego)	117 kb	5.747632917700155	0.39344463122204887

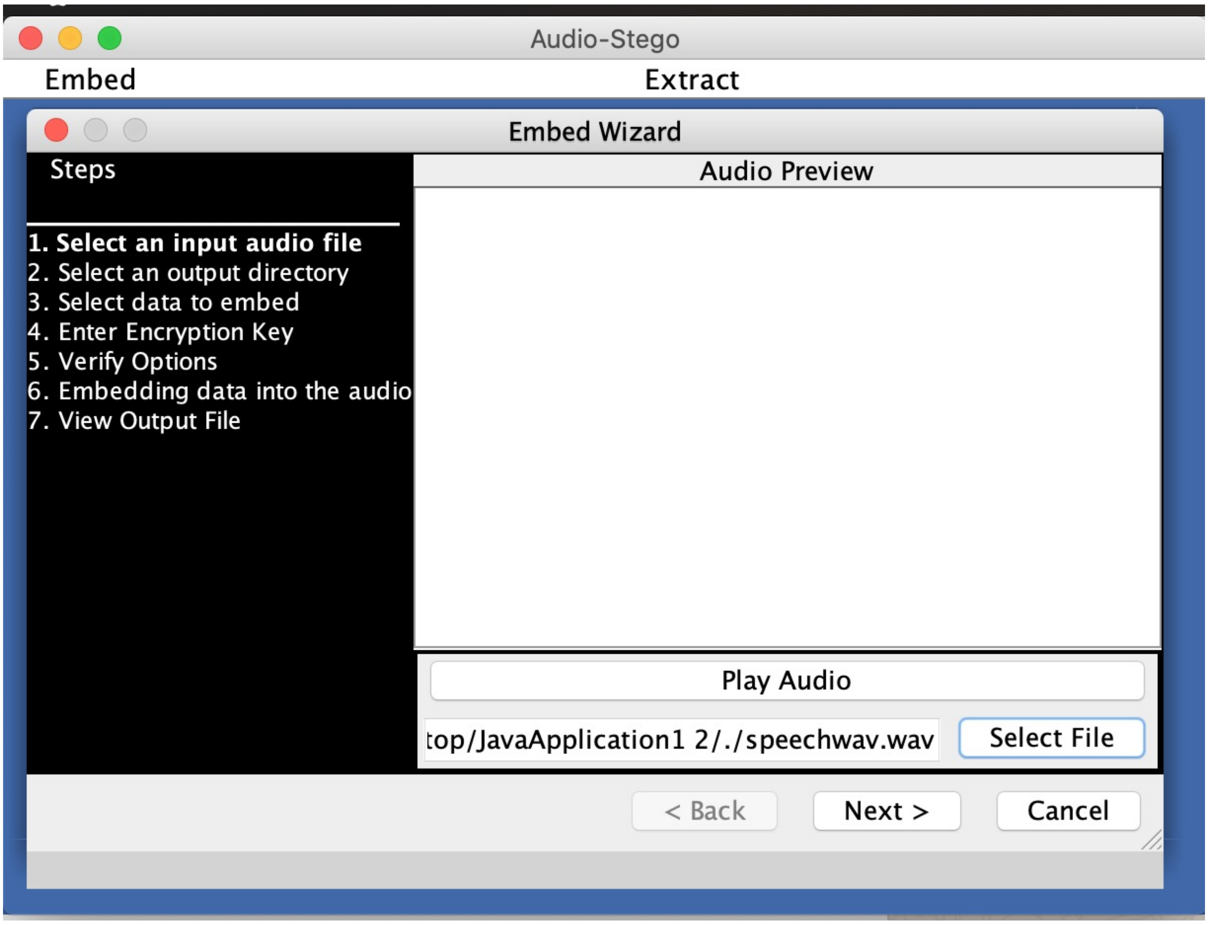


```
public static double signalToNoiseRatio(double[] wavArr) {
    DescriptiveStatistics stats = new DescriptiveStatistics();

    // Add the data from the array
    for (int i = 0; i < wavArr.length; i++) {
        stats.addValue(wavArr[i]);
    }

    // Compute some statistics
    double mean = stats.getMean();
    double std = stats.getStandardDeviation();
    return std == 0 ? 0 : (mean / std)*10;
}
```



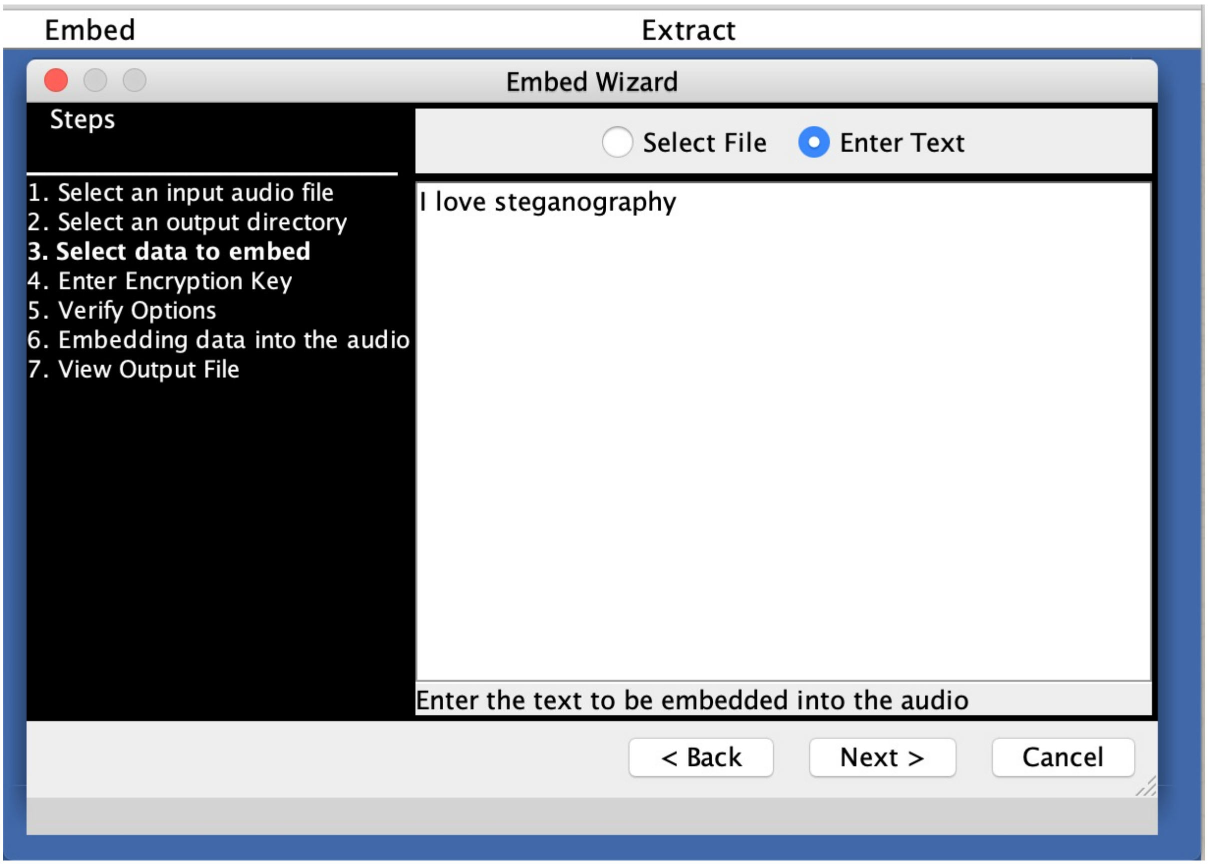


```
public static double meanSquareError(double[] actualArr, double[] predictedArr) {
    DescriptiveStatistics stats = new DescriptiveStatistics();
    int n = actualArr.length;
    double sum = 0;
    // Add the data from the array
    for (int i = 0; i < n; i++) {
        double diff = actualArr[100] - predictedArr[100];
        sum = sum + diff*diff;
    }

    return sum/n;
```



1. Select an input audio file
2. Select an output directory
3. Select data to embed
4. Enter Encryption Key
5. Verify Options
6. Embedding data into the audio
7. View Output File

top/JavaApplication1 2/./speechwav.wav

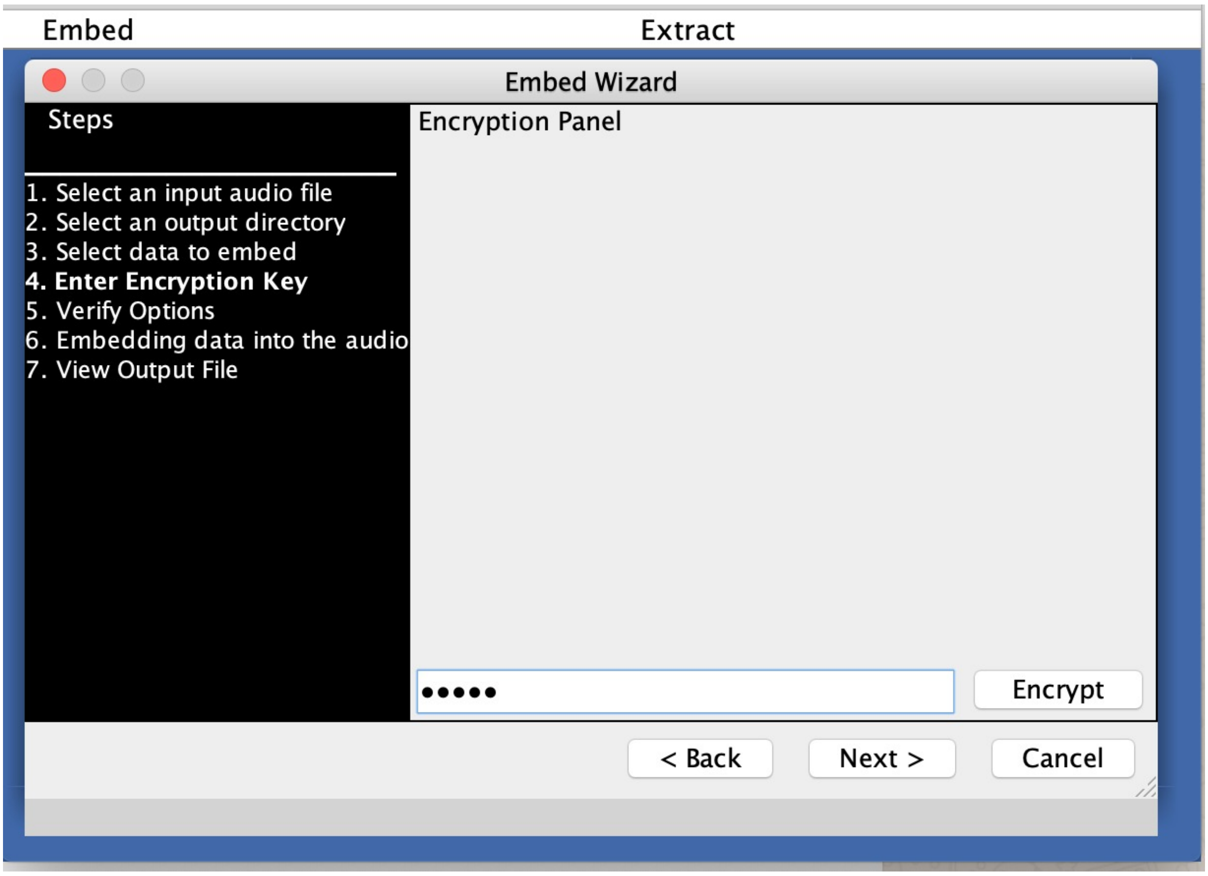


1. Select an input audio file
2. Select an output directory
3. **Select data to embed**
4. Enter Encryption Key
5. Verify Options
6. Embedding data into the audio
7. View Output File

○ Select File ● Enter Text

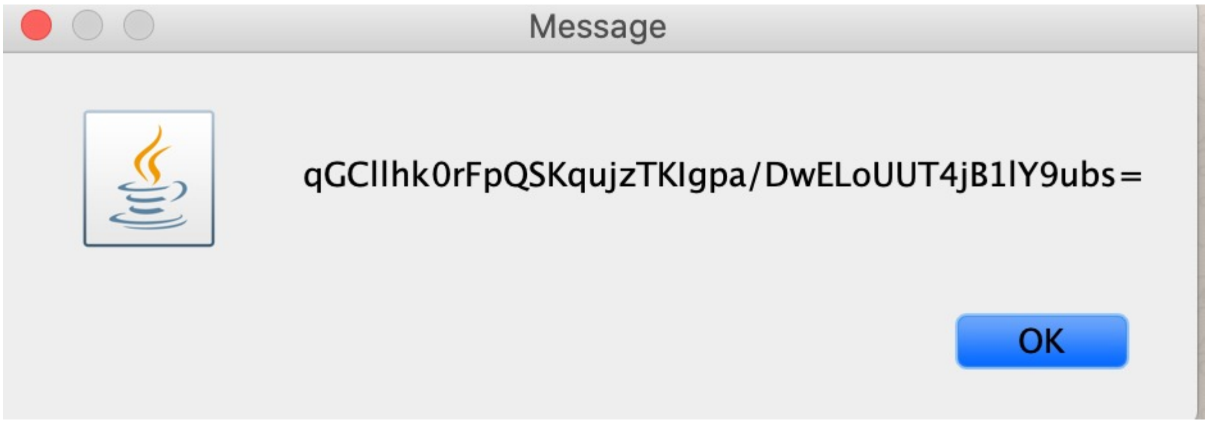
I love steganography

Enter the text to be embedded into the audio



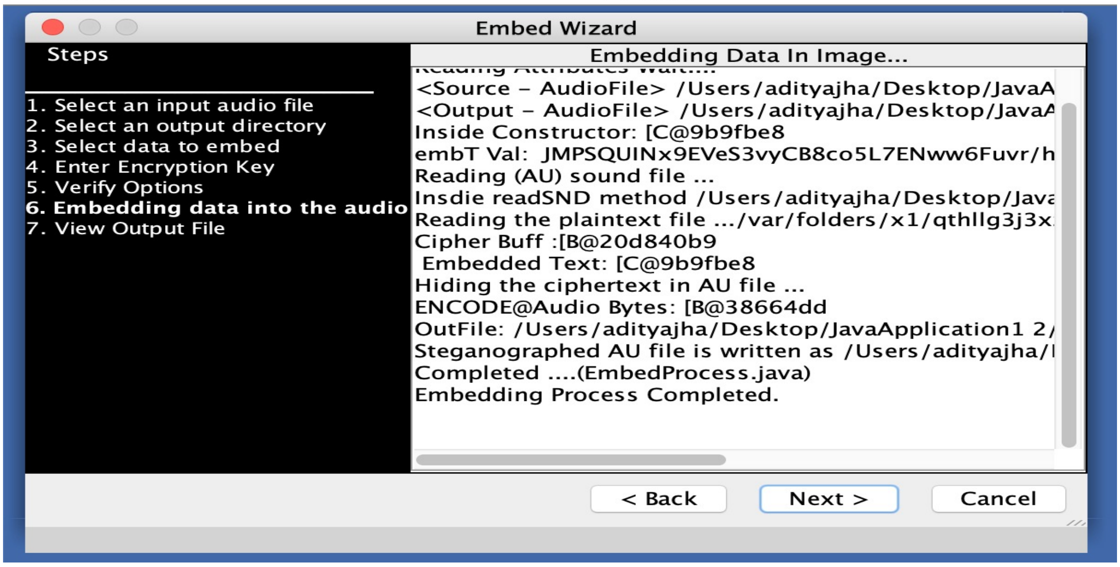
1. Select an input audio file
2. Select an output directory
3. Select data to embed
4. Enter Encryption Key
5. Verify Options
6. Embedding data into the audio
7. View Output File

●●●●●



```
qGCllhk0rFpQSKqujzTKIgpa/DwELoUUT4jB1lY9ubs=
```

Embed Extract



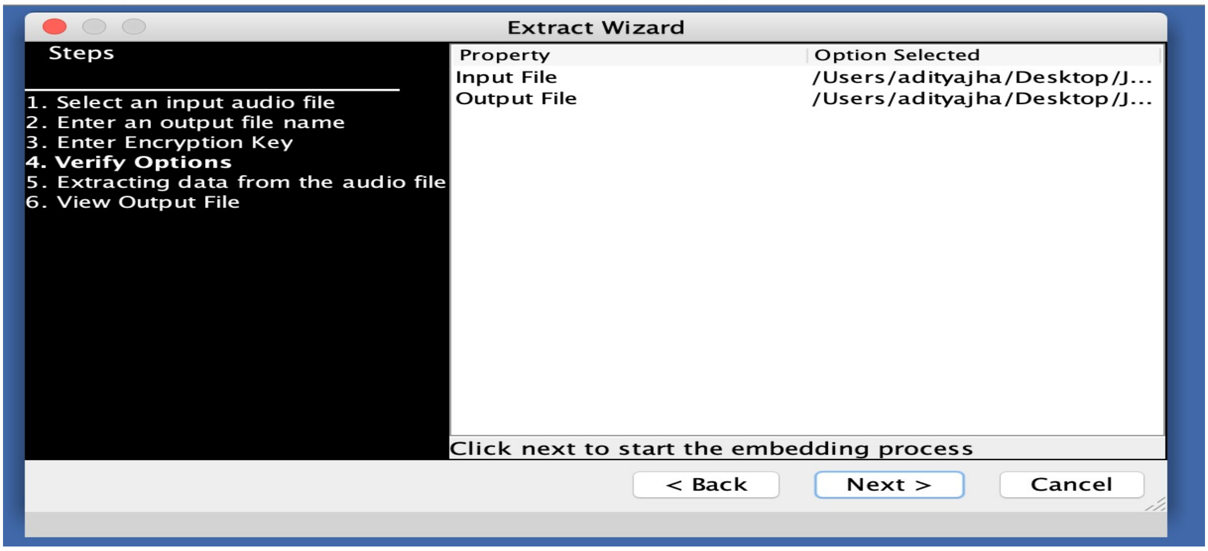
Steps

1. Select an input audio file
2. Select an output directory
3. Select data to embed
4. Enter Encryption Key
5. Verify Options
6. **Embedding data into the audio**
7. View Output File

Embedding Data In Image...

```
Reading Attributes Wait....
<Source – AudioFile> /Users/adityajha/Desktop/JavaA
<Output – AudioFile> /Users/adityajha/Desktop/JavaA
Inside Constructor: [C@9b9fbe8
embT Val:  JMPSQUINx9EVeS3vyCB8co5L7ENww6Fuvr/h
Reading (AU) sound file ...
Insdie readSND method /Users/adityajha/Desktop/Java
Reading the plaintext file .../var/folders/x1/qthllg3j3x
Cipher Buff :[B@20d840b9
 Embedded Text: [C@9b9fbe8
Hiding the ciphertext in AU file ...
ENCODE@Audio Bytes: [B@38664dd
OutFile: /Users/adityajha/Desktop/JavaApplication1 2/
Steganographed AU file is written as /Users/adityajha/I
Completed ....(EmbedProcess.java)
Embedding Process Completed.
```

Embed Extract

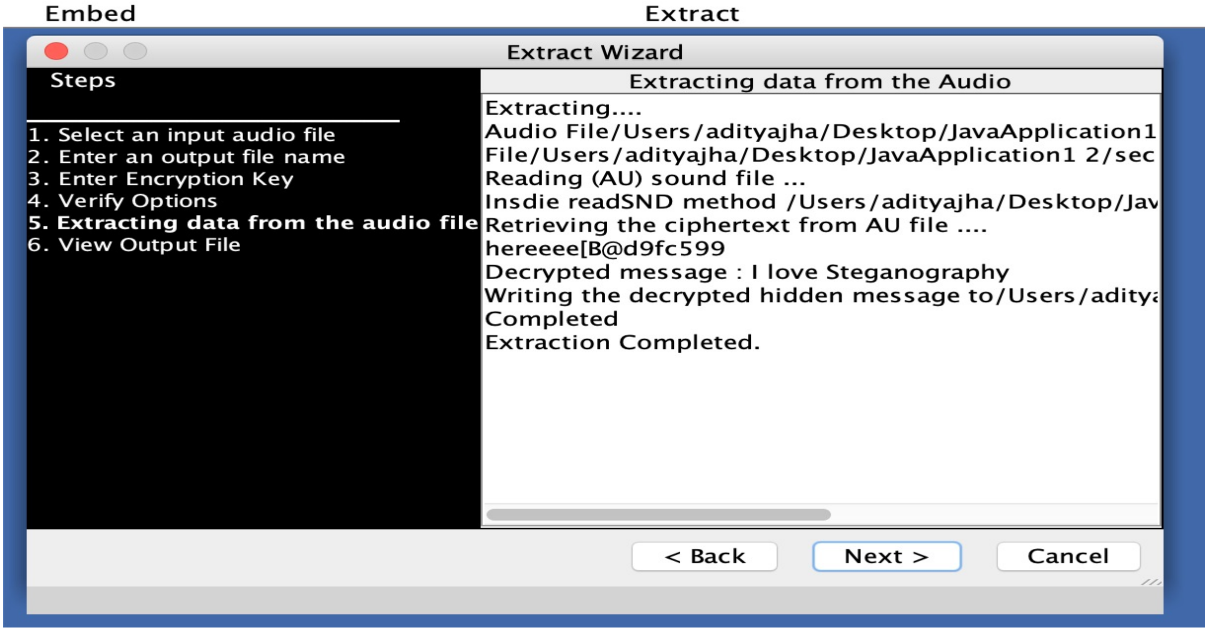


Steps

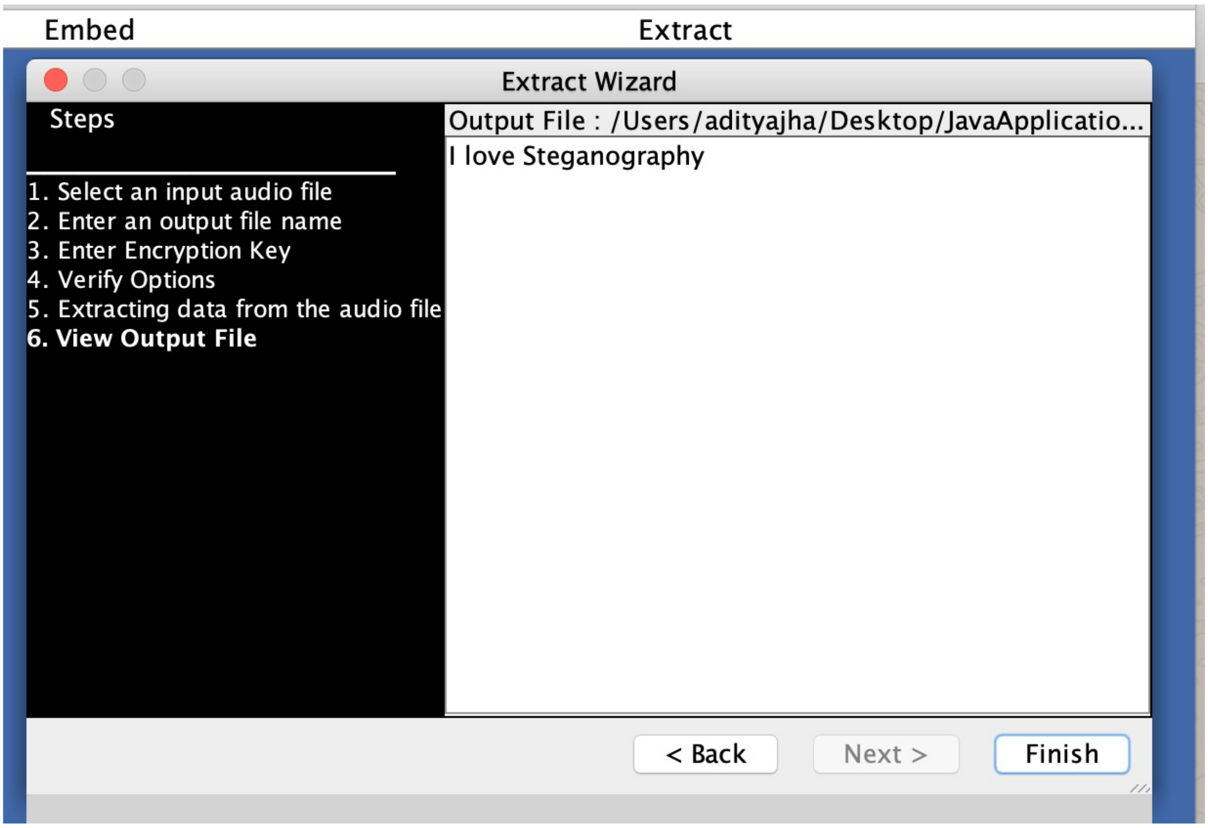
1. Select an input audio file
2. Enter an output file name
3. Enter Encryption Key
4. **Verify Options**
5. Extracting data from the audio file
6. View Output File

Property	Option Selected
Input File	/Users/adityajha/Desktop/J...
Output File	/Users/adityajha/Desktop/J...

Click next to start the embedding process


Embed
Extract
Extract Wizard
Steps
1. Select an input audio file
2. Enter an output file name
3. Enter Encryption Key
4. Verify Options
5. Extracting data from the audio file
6. View Output File
Extracting data from the Audio
< Back
Next >
Cancel

```
Extracting....
Audio File/Users/adityajha/Desktop/JavaApplication1
File/Users/adityajha/Desktop/JavaApplication1 2/sec
Reading (AU) sound file ...
Insdie readSND method /Users/adityajha/Desktop/Jav
Retrieving the ciphertext from AU file ....
hereeee[B@d9fc599
Decrypted message : I love Steganography
Writing the decrypted hidden message to/Users/adityа
Completed
Extraction Completed.
```



```
Output File : /Users/adityajha/Desktop/JavaApplicatio...
I love Steganography
```

References:

[1] Docs.oracle.com. 2020. *Javax.Crypto (Java Platform SE 7)*. [online] Available at: <https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html> [Accessed 16 August 2020].