Embedding Data Secretly in Audio file with AES Encryption

MSc Internship
Cyber Security

Aditya
Student ID: X19104987

School of Computing
National College of Ireland

Supervisor: 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**………

Word Count: ……………………5018………… **Page Count**…………..20………………..

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: ……Aditya…………………………………………………………………………………………………

Date: ……28/09/2020………………………………………………………………………………………

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):	

Embedding Data Secretly in Audio file with AES Encryption

Aditya
X19104987

Abstract

As the businesses are growing rapidly, new and advanced as well as secure ways of the data transmission should be implemented. One such method is proposed in this research where concepts of steganography, as well as cryptography, are combined to form a stronger and secure technique to transmit data from one end to another. In this paper, audio steganography by LSB is implemented with one of the most secure encryption algorithm AES Encryption. The project is made in Java GUI in which the user can easily embed and extract the hidden message. The proposal is tested and evaluated by calculating the SNR(Signal to Noise Ratio) and Mean Square Error(MSE) of different types of audio extensions like .au, .wav, .mp3.

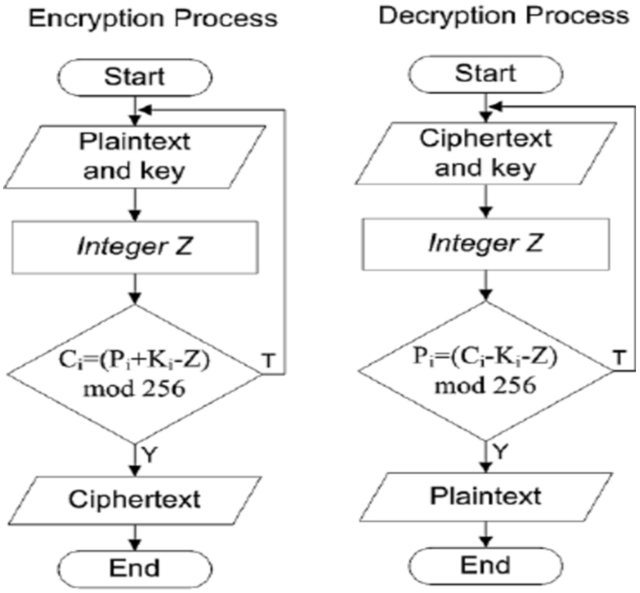
1. Introduction

As the businesses are increasing day by day, online transmission of data is increasing at an exponential rate. It has become very much important to maintain the confidentiality and integrity of the data. As many attackers or phreakers try to up their game in exploiting information security to abuse or misuse the data, the companies must secure their confidential information.

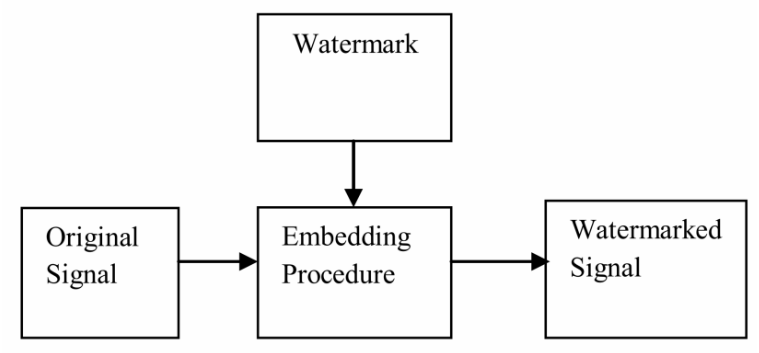
There are several ways to achieve this aim, three of the ways that are most common and effective methods are cryptography, steganography and watermarking.

Cryptography can be described as the science of secret writing. In this method, the unencrypted or the preferred message is encrypted using a certain algorithm and then the encrypted data with a key is sent to the receiver and on the receiver side, the encrypted message can be decrypted into the message using the same algorithm and the key[1]. The primary objectives of cryptography can be described as below:

- Confidentiality - No one except the intended receiver should be able to read the information.
- Integrity - The modification of the data or message can only be done by any authorised person.
- Authorisation - The identity of the receiver should be verified so that it can be verified that the user reading or getting the message is indeed the intended receiver.
- Non-repudiation - A technique or method which can prove that the message is received from a trusted source.
- Availability - The availability is whenever an authorised person requires the data, they should get it without any difficulties or hindrances.

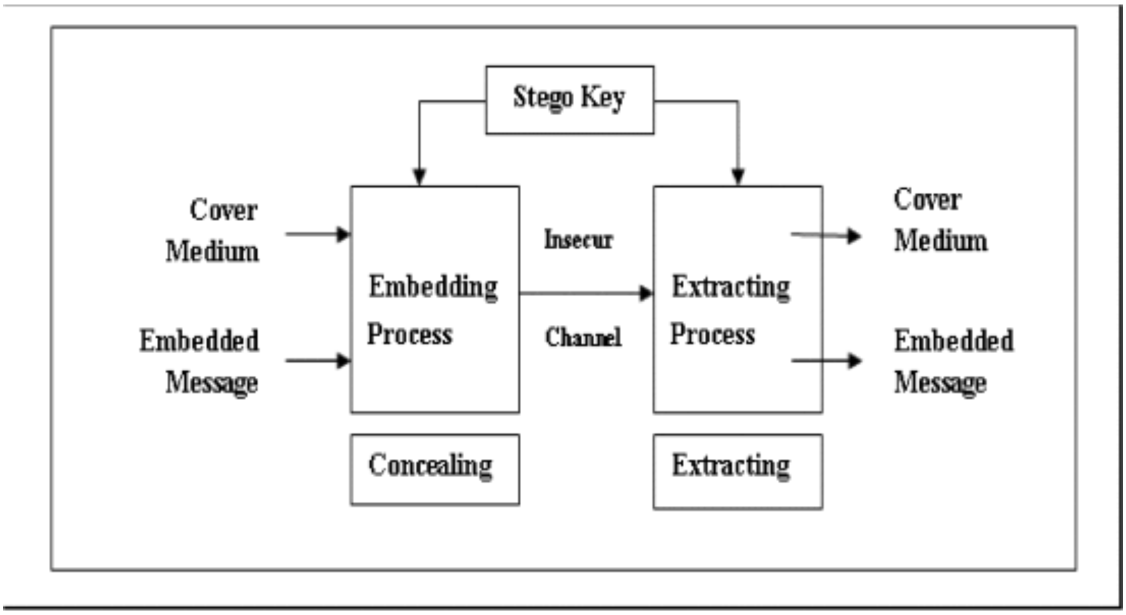


Watermarking is used to prevent the fraud of digital media. The owner or author of the media, watermarks their work establishing the ownership rights to stop unauthorized access, illegal applications[3].



Steganography is a study of the techniques through which the primary message can be hidden in the secondary message from the sender side and at the receiver end, via any key or password, the primary message can be filtered from the secondary message[2]. In general, the steps of the steganography can be as followed:

- Write an unhidden secondary message.
- Develop a stego-message by embedding the primary message in the secondary message using a stego-key.
- Transmit the stego-message to the receiver who will have the prefixed key.
- At the receiver end, the receiver can get the primary message from the stego-message by using the



Steganography can be achieved on various media platforms like image steganography, audio steganography, video steganography etc. This proposed idea is focused on implementing audio steganography.

As the name suggests, audio steganography is the concept of hiding the message in audio files. Any digital media can be represented in a bit sequence. So, generally, in audio steganography two actors are involved- Sender and Receiver. Sender embeds the message into audio while the receiver extracts the message from the audio.

1.1 Research Question

Que - ***How Security and Authenticity of data can be enhanced during data transmission in Audio Steganography?***

As LSB is vulnerable to Steganalysis or one lucky pause might result in knowing the presence of hidden data in audio, it is very much necessary to enhance the security data transmission between the actors to maintain the authenticity and integrity of the message. This project aims to enhance security by combining the concepts of steganography as well as cryptography. Since steganography can be defined as the hiding of data in any carrier and cryptography is the conversion of data into an unreadable string. So, why not combine these two concepts, so that if attacker or man in the middle even gets to know the presence of hidden data, they won't be able to access it or read it.

The proposed method in this paper is an LSB based Audio Steganography using AES Encryption to enhance security. The project is a GUI based application and developed in a Java environment.

In this research, the approach of combining the LSB based steganography as well as cryptography is enhanced in both the aspects of security as well as usability, balancing the basic principle CIA(Confidentiality, Integrity, Availability). The project is implemented using Java which is the one of the most commonly used and highly secured(if implemented correctly) language in enterprise. As well as by enhancing the GUI, the user experience in this project will be much better.

At the same time, more focus is given on AES Encryption by applying the feature to include 128 bit as well as 256 bit Encryption key which generally most of the other researches didn't focus on. AES encryption is considered one of the best encryption algorithms in market, but

AES Encryption with low bit encryption key is not much secure. Therefore, in this research, this fact was taken into consideration, and accordingly the AES is implemented.

2. Background

I. Literature Review

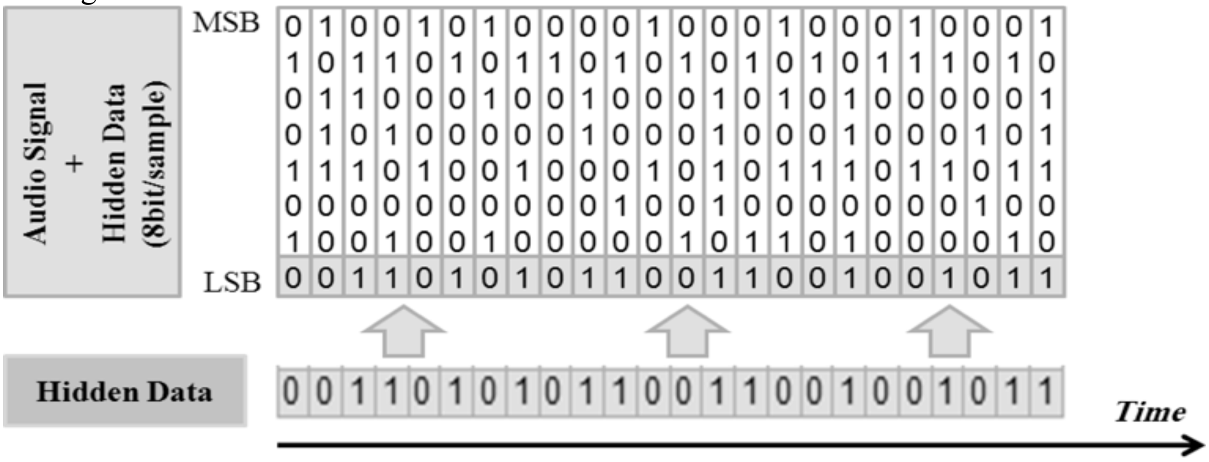
Since audio steganography is widely used and one of the most important steganographic techniques as according to research, the human auditory system is insensitive to small distortions within audio signals, which makes it perfect to hide data in audio files.

- **Low Bit Encoding**

There are various methods of audio steganography and one of the widely used is Least Signification Bit (LSB). LSB provides best un-detectability [4] which helps in the case if the attacker gets his hand on the audio, but the problem lies in it is it's not strong or robust enough and low embedding rate.

In **2016**, **Shweta V. Jadhav et al[5]** proposed an interesting way of inserting every bit of information in the selected positions of the cover audio. They selected the embedding position based on upper 3 MSB bits of cover audio. To increase security, they provided an additional secret key using Chaotic Encryption Scheme(CES). In their research, they first encrypted the private message via CES, then a WAV format audio file is used to embed the encrypted message into a 16-bit sequence. But as it is known LSB is vulnerable to steganalysis and CES is also vulnerable encryption [8], some security upgrade is required in this paper.

Then in **2017**, **Shital P. Rajput et al [6]** referenced [5]'s research and took a further step by proposing an enhanced version of applying audio steganography in which instead of following the traditional way of converting the message into binary form and then replacing each bit in a linear format, they proposed two algorithms. In Proposed Algorithm-I, two information bits of secret message or primary message is installed at once over LSB places of carrier audio depending on the 3 MSBs of carrier audio on the other hand in Proposed Algorithm-II, based on the compliment of 3 MSBs carrier audio, two bits of the secret message is embedded.



MSB	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0
	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1
	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1
	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1
	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
	1	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	1	0
LSB	0	0	1	1	0	1	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0	1	1

Hidden Data	0	0	1	1	0	1	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0	1	1

Fig. 3: LSB in 8b/sample signal is overwritten by one bit of the embedded data.

In **2019**, other researchers, **C.C Sobin et al [7]** proposed another way of using audio steganography by combining the concepts of steganography and cryptography. They proposed the idea of applying multiple levels of security. On the first level of security, they have encrypted the secret message by using RSA algorithm and on the second level, they created a powerful genetic algorithm based on LSB data hiding scheme and hides the message in audio. During the result analysis, it can be easily seen that Peak Signal to Noise Ratio (PSNR) value of the second method i.e. Genetic Algorithm is much better than the first method of RSA algorithm making the second method a better option. Also, in cases for small audio files, RSA is not a good option.

Another interesting approach was taken by **Mohamad Anwar et al [9]**, in **2019**, using audio steganography by using Lifting Wavelet Transform(LWT) and dynamic key. The researchers aimed to create an android application in which frame marking would be done and by using Dynamic Key confidential data will be encrypted and then embedded to the audio using LWT. Then during the decryption time, there will be no need to manually enter the Dynamic key value, the message could be decrypted using a marked frame. During the testing phase, it can be seen the audio loss can be up to 20% by using this approach and also as mentioned in the research, this project is under development stage, so further research would be needed.

Joydwip Mohajon et al [10] in **2018,** took the same approach as [7] of using Genetic Algorithm but applying with K-bit symmetric security key. In their research, they applied genetic algorithm-based to embed the message in the audio and for security improvement they used K-bit security key such that for comparing message bits it can generate around 8-15 Most Significant Bit(MSB) positions. Not much detail was given on whether they developed their encryption or they were using already existed ones.

In LSB, there were so many different and interest approach taken to enhance and use it effectively. Other interesting and recent approaches are discussed in **[11] , [12] , [13], [14], [15] , [16]**. Through these researches, it can be concluded that LSB on its own is not secure enough because of its low robustness and low embedding rate and is vulnerable to stego-analysis. Therefore, another approach must be taken for fail-safe situations.

- ## Discrete Wavelet Transform (DWT)

Another approach which can be taken is using DWT to embed the data in audio. In comparison to LSB based approach, this approach is quite novel and many researchers have turned their attention to make this approach more approachable and effective.

The basic functionality of DWT can be that it can divide the message into high and low-frequency parts. In the high parts, specifics of the edge components are stored while the lower part stores the signal majority part of the signal information which is again divided into two parts. For every division, the DWT algorithm is applied in the vertical direction and then horizontal direction [17].

In **2014, B.Geethavani et al[18]** proposed the idea of using audio steganography in DWT algorithm and for increasing the security the author applied the Blowfish Algorithm.

The steps taken by the author can be explained as:

Sender's end

1. Take the message M.
2. Apply the Blowfish algorithm to encrypt the message M into cypher text C.
3. By using the DWT, embed the cypher text C into sample audio of .wav extension.
4. Transmit the stego audio S to the receiver.

Receiver's end

1. To extract the stego audio S, the median noise filter is applied.
2. Inverse DWT is applied to S.
3. Blowfish decryption process is applied to decrypt the information.
4. The receiver gets the message.

Interestingly, **Rupayan Das et al[19]**, in 2017, proposed to enhance the audio steganography by applying DWT over a signal. Then the cryptoanalysis of the original, stego audio and hidden data is done. The block diagram of the author's proposed idea is shown below.

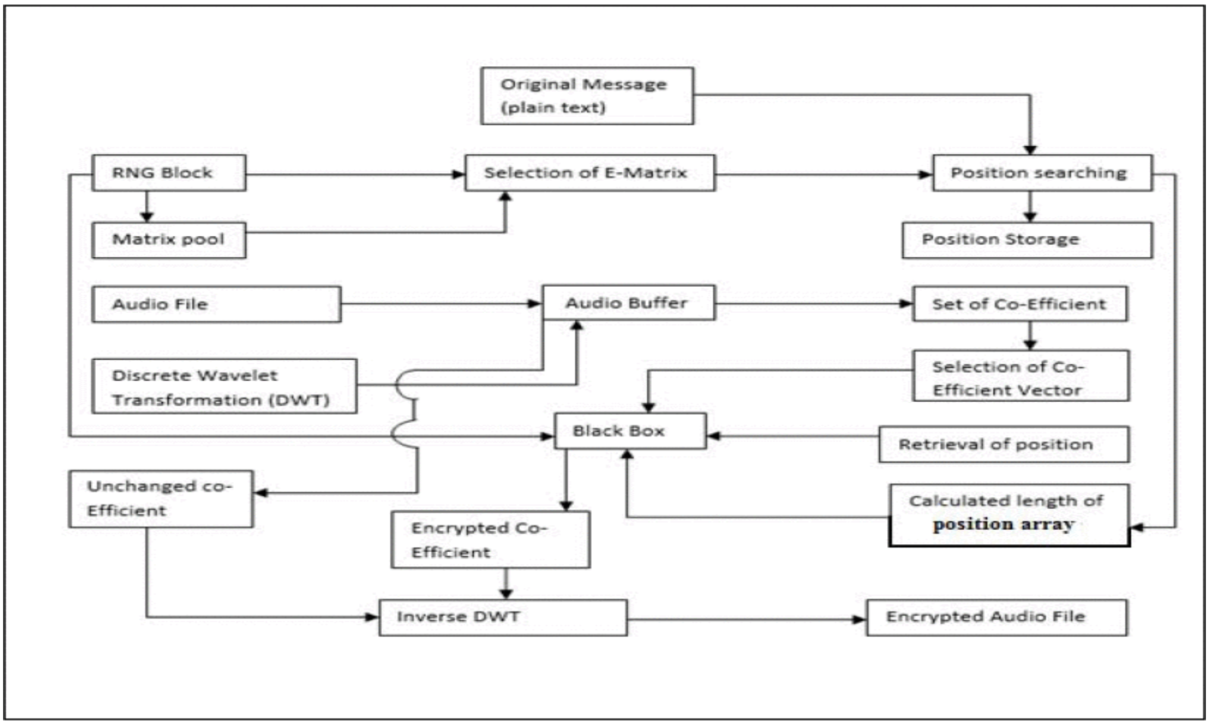


Fig 4. Block diagram

Even though several kinds of research are being done in DWT but there are some drawbacks of DWT too like the audio distortion is more in comparison to LSB leading to more inaccurate data [20]. But there are certain advantages of using DWT such as it provides time-frequency representation.

- **Echo Hiding**

Another type of audio steganography is echo hiding. The idea of using echo hiding in audio steganography was first proposed in 1996. In echo hiding, data gets embedded to audio signals via short echo to the host signal. The echo 's nature is a distortion that's applied to the host audio. Hence, the challenge of the Human Auditory System(HAS) susceptibility issue to additive noise is avoided. In this system, the data can be shielded by three echo signal perimeters: Initial amplitude, the delay (offset) and the decay rate such that the echo is not audible [21].

Methodology

Since by reviewing the other researches mentioned it can be said that Audio Steganography using LSB is convenient and easy to apply, also, it provides less data loss and audio distortion but it is vulnerable to steganalysis []. Hence in this research, we are proposing a java GUI application applying LSB based audio steganography on the secret text message and to enhance its security we are using bit AES encryption. By applying AES encryption, we are taking a security measure, just in case if any hacker or phreaker gets to know that there is data embedded in the audio and somehow they extract the data, they will still get the encrypted data unable to make sense of, which will maintain the confidentiality of the data.

I. Basic Terms Defined

1. JAVA

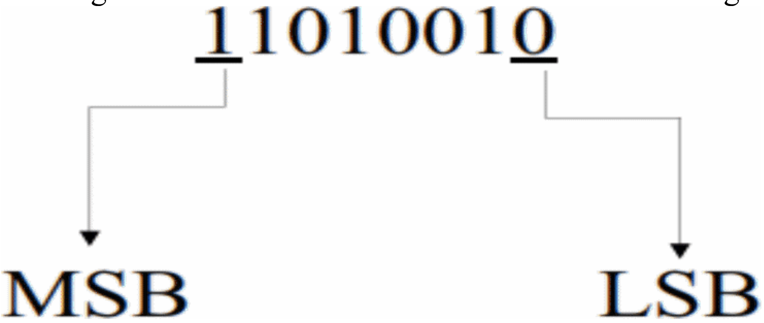
In 1995, Sun Microsystem launched Java technology. For the first time, it was applied on Netscape web browser and then in 1996, Java Development Kit(JDK) and Standard Edition(SE) was launched. One thing that made Java distinct from other programming languages of that time was that it was introduced keeping the security perspective in mind. Java offers various many security features. It provides access control like public, private and protected variables, supports automatic garbage collection and removed pointer as a datatype [22].

One of the java framework used in this project is Java Cryptographic Architecture(JCA) which is as the name suggests is a framework to access and develop cryptographic functions. JCA provides an array of APIs for cryptographic services like Digest Message Algorithms(DMA), message authentication codes (MACs), key generators and symmetric bulk encryption. JCA contain two types of package one is Java.Security Kit and another is javax.crypto. Java.Security Kit contains classes which are not related to export controls such as MessageDigest and Signature, on the other hand, javax.crypto contains the export controls like Cipher and Key Agreement [23].

JCE provides a good structure and implementation for encryption techniques such as key generation and agreement, message authentication etc. These are all the attributes which provide better security. MACs provide data integrity as well as authenticity, which can help to detect tampering of data during the transmission of data over any network. By using these codes, the user can easily implement signatures and encryption in their code development [24].

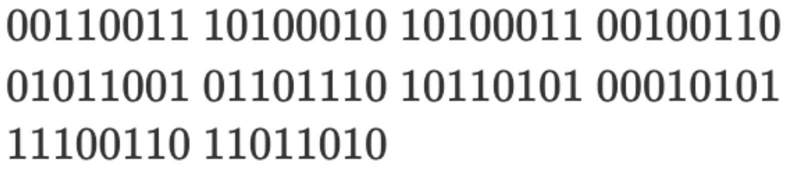
2. Least Significant Bit (LSB)

There are multiple types of methods in steganography. One of the most used methods is LSB based steganography. This method is easy to implement and messages can be hidden quite secretly. The distortion rate in LSB is very low in comparison to other approaches. The basic plan is to change the lowest bits of the audio file with the message's bits [27].



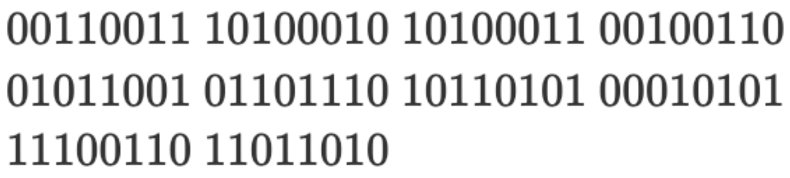
As shown above, the 1 bit is MSB(Most Significant Bit) and 0 is the LSB(Least Significant Bit)

Let us assume the message is 0f 10 bits, which will make several bytes used = 10 bytes



00110011 10100010 10100011 00100110
01011001 01101110 10110101 00010101
11100110 11011010

Suppose the message value in binary is 1110101011 then the new binary will be



00110011 10100010 10100011 00100110
01011001 01101110 10110101 00010101
11100110 11011010

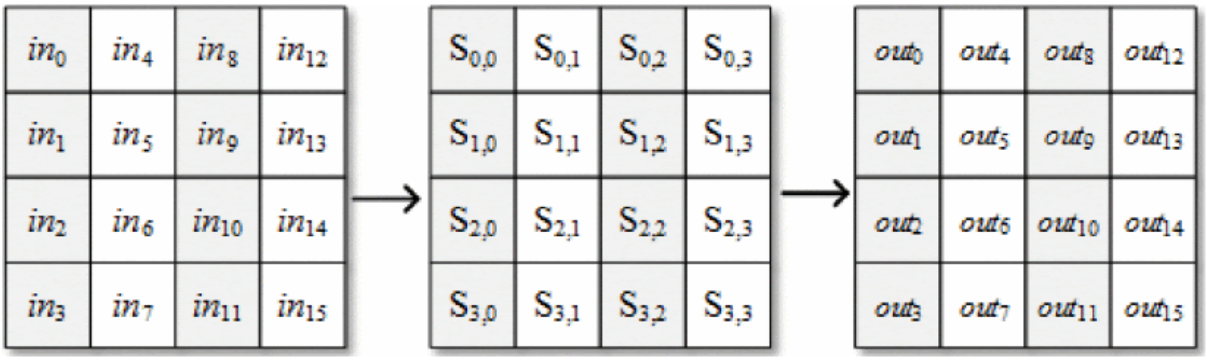
Based on the theory stated above the whole message gets embedded into the audio.

3. Advanced Encryption Standard (AES)

Officially, NIST confirmed the development of AES on September 12, 1997. It was originated from the Rijndael algorithm and declared as the new encryption algorithm by the Federal Information Processing Standards(FIPS), and replaced the previous encryption algorithm i.e Data Encryption Standard(DES). AES encryption can use encryption keys of 128, 192 and 256 bit to encode and decode the data.

In general, AES is a symmetric block cypher which enciphers and deciphers data in a block such as for 128 bits $N_b = 4$, where N_b is the number of blocks. Similarly, $N_k = 4,6$ and 8 for 128, 192 and 256 respectively bits where N_k is key length and number of the rounds is depended on the key length. For $N_k = 4,6$ and 8, number of rounds are $N_r = 10,12$ and 14 respectively.

When the message or encrypted message stored in the input array is arranged orderly bit by bit in an array, then the encoding or decoding process starts. The last step is a copy of an array of bites via input array to output array(which is either the message or encrypted message depending on what the input is). The algorithm can be presented as



in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

out_0	out_4	out_8	out_{12}
out_1	out_5	out_9	out_{13}
out_2	out_6	out_{10}	out_{14}
out_3	out_7	out_{11}	out_{15}

Fig 5. AES algorithm

Via substitution, table functions independently over every byte and that's how non-linear byte type substitution takes place. Then every point in the box is represented in its hexadecimal form calculated using the intersection of row and column index.

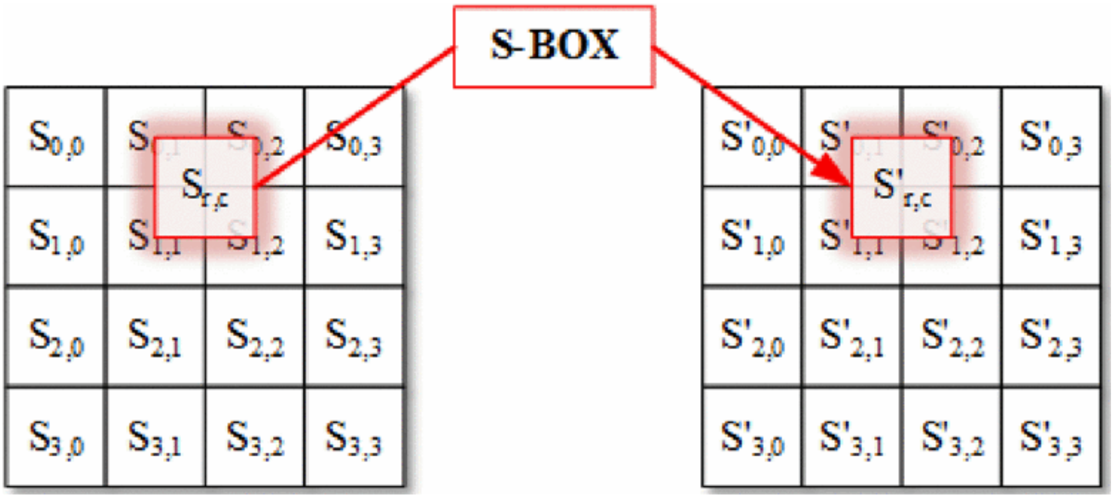

S-BOX
$S_{r,c}$
$S'_{r,c}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig 6. Sub-bytes Conversion

The Shift rows start shifting cyclically.

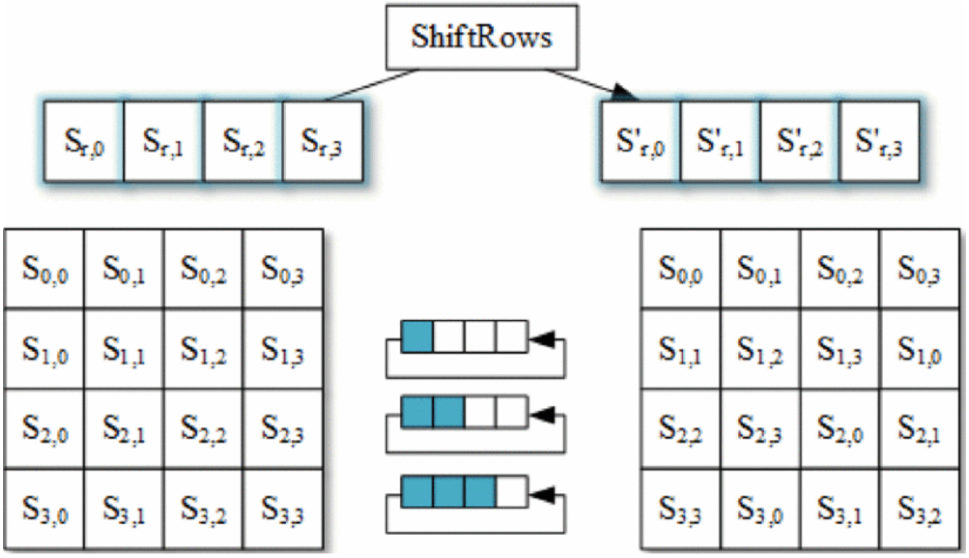

ShiftRows
$S_{r,0}$ $S_{r,1}$ $S_{r,2}$ $S_{r,3}$
$S'_{r,0}$ $S'_{r,1}$ $S'_{r,2}$ $S'_{r,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,1}$ $S_{1,2}$ $S_{1,3}$ $S_{1,0}$
$S_{2,2}$ $S_{2,3}$ $S_{2,0}$ $S_{2,1}$
$S_{3,3}$ $S_{3,0}$ $S_{3,1}$ $S_{3,2}$

Fig 7. The shift row conversion

Then the output of shifted rows is mixed with columns by the Mix column matrix

$$\begin{bmatrix} S'0,c \\ S'1,c \\ S'2,c \\ S'3,c \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S0,c \\ S1,c \\ S2,c \\ S3,c \end{bmatrix}, for\ 0 \leq c \leq 1$$

Fig 8. Conversion Matrix

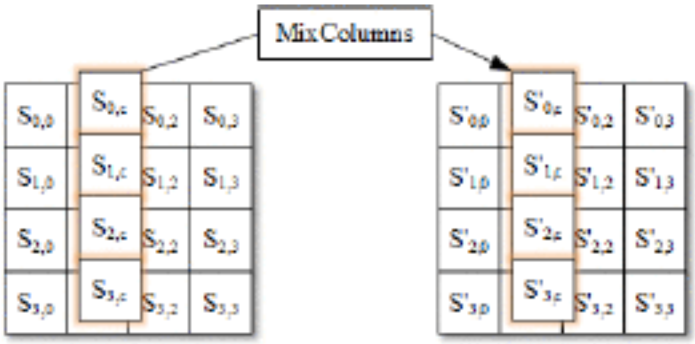

Mix Columns
$S_{0,0}$ $S_{0,c}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,c}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,c}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,c}$ $S_{3,2}$ $S_{3,3}$
$S'_{0,0}$ $S'_{0,c}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,c}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,c}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,c}$ $S'_{3,2}$ $S'_{3,3}$

Fig 9. Mix Coloumns Alteration

The last step is called Addround Key Conversion. In this step, the round key is added using the XOR gate.

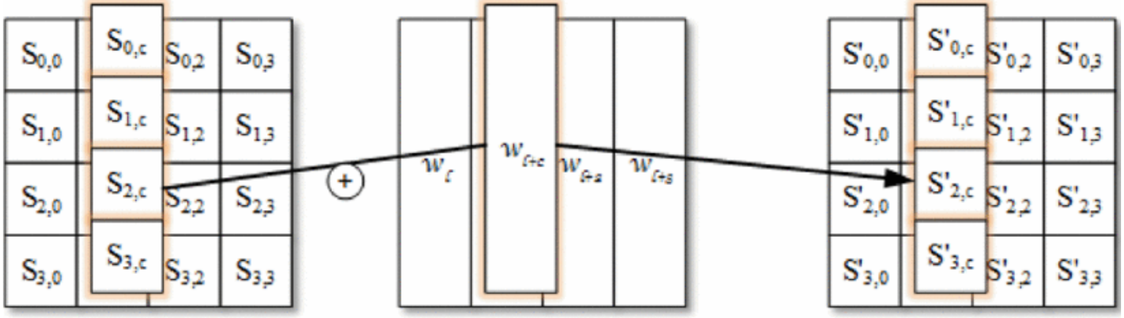

$S_{0,0}$ $S_{0,c}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,c}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,c}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,c}$ $S_{3,2}$ $S_{3,3}$
$\oplus$
w_l w_{l+c} w_{l+2} w_{l+3}
$S'_{0,0}$ $S'_{0,c}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,c}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,c}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,c}$ $S'_{3,2}$ $S'_{3,3}$

Fig 10. Addround Key Conversion

By using the reverse of this process AES decryption takes place.
So, while doing this research the main question we came over was

Why AES Encryption and how safe it is?

Some encryption algorithms are quite effective in some scenarios but in some scenarios, they are not. For eg. RSA is very effective for large data but for small data, it's not much effective. Hence, we found out that for the most number of scenarios AES Encryption is more suitable. Second, it is easy to implement. According to OWASP organisation, many encryptions are not easy to implement which in result causes vulnerabilities, which later on hackers exploit [25].

In the matter of safety, it can be said on AES, the brute force attack cannot be implemented as it will take approximately $1100 * 10^{75}$ combinations to fully break the encryption, which is next to impossible. That is the reason which makes AES most widely used encryption algorithm in the world [26].

II Flow Diagram

- Embedding Process
  The embedding process of this proposed project is done in Java GUI. The working of the embedding process is explained in the steps below:
  Step 1: The user will select an audio file of .wav or .au extension in which the message needs to be embedded. Select the next button.
  Step 2: Then the output directory will be selected in which the stego-audio will be saved. Select the next button.
  Step 3: Enter the message or select a .txt file which needs to be encrypted and embedded. Select the next button.
  Step 4: In this step, the user needs to set a key of a minimum of 4 characters. It's better to set a long key for strong security. Select the encrypt button. An AES-256 bit encrypted message will be shown. Select ok and then next.
  Step 5: Verify all the details and select next.

Step 6: The embedding process is completed. Play the input or output audio to verify. Then select finish.

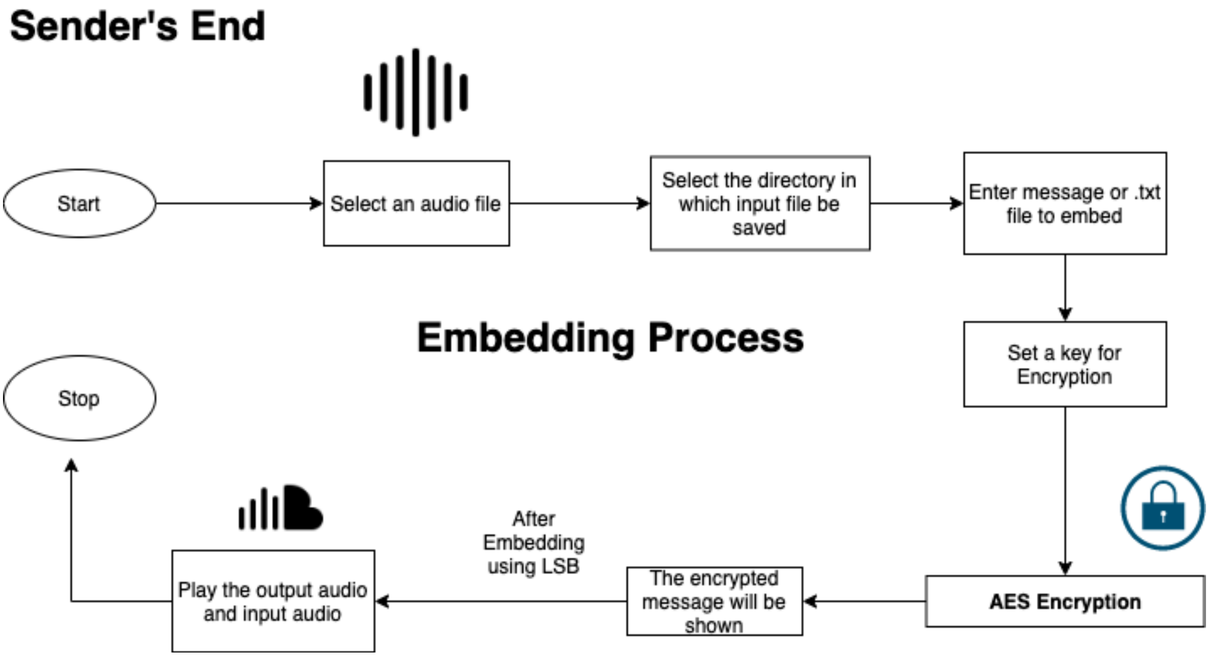

Sender's End
Start
Select an audio file
Select the directory in which input file be saved
Enter message or .txt file to embed
Embedding Process
Set a key for Encryption
Stop
AES Encryption
The encrypted message will be shown
After Embedding using LSB
Play the output audio and input audio

Fig 11. Flow diagram of embedding

- Extraction Process
  The extraction process is the reverse process of the embedding process. The steps were as followed:
  Step 1: Select a stego audio file from which message needs to be extracted. Then, select next.
  Step 2: Select the output directory in which the secret message fill will be extracted. Then, select next.
  Step 3: Enter the AED decryption key which was already decided between the sender and receiver. Then, select next.
  Step 4: If the encryption key entered is correct, then on the fourth step verify the contents and click next. If the key is not correct the process will stop.
  Step 5: The message will be extracted then decrypted. The user will be shown the output and a file with name Secret.txt will also be saved in the directory selected in step 2.

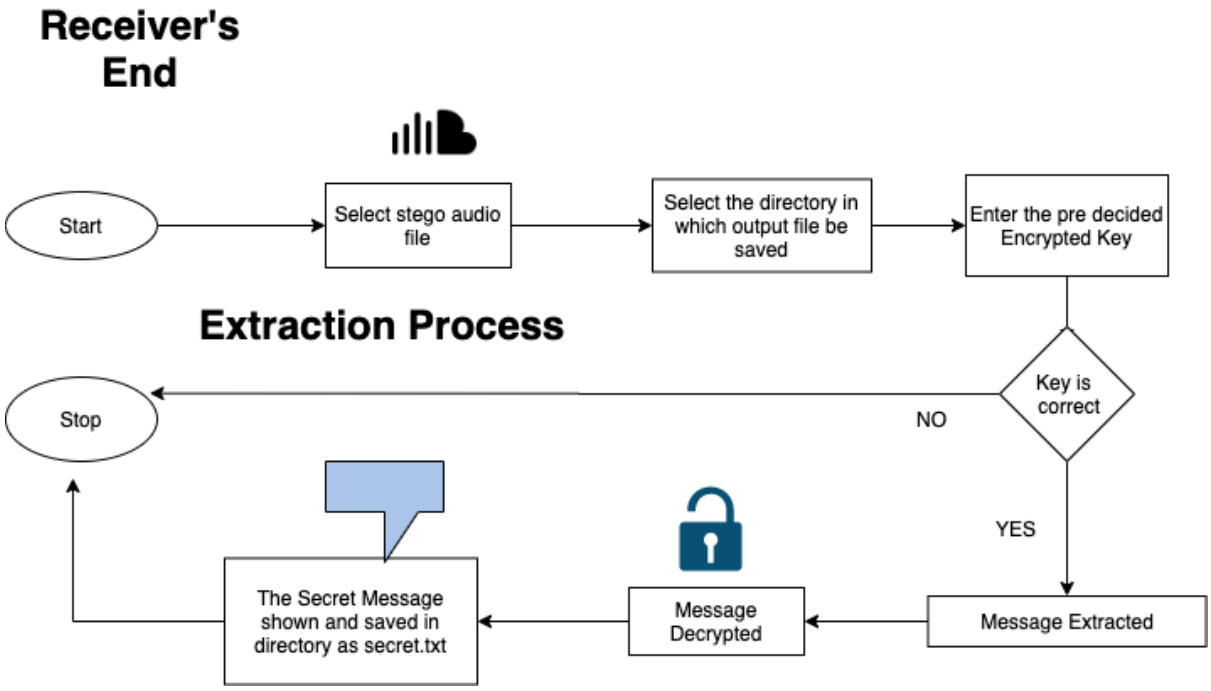
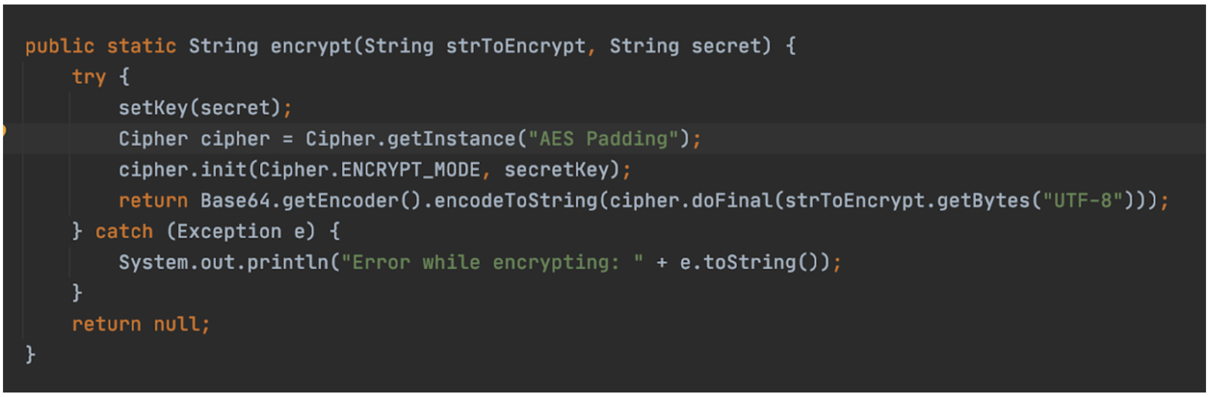

Receiver's End
Start
Select stego audio file
Select the directory in which output file be saved
Enter the pre decided Encrypted Key
Extraction Process
Key is correct
NO
Stop
YES
Message Extracted
Message Decrypted
The Secret Message shown and saved in directory as secret.txt

Fig 12. Flow diagram of the Extraction process

Implementation

The proposed software is implemented in the MAC OSX operating system using JAVA 8 in IntelliJ IDEA IDE(Integrated Development Environment).

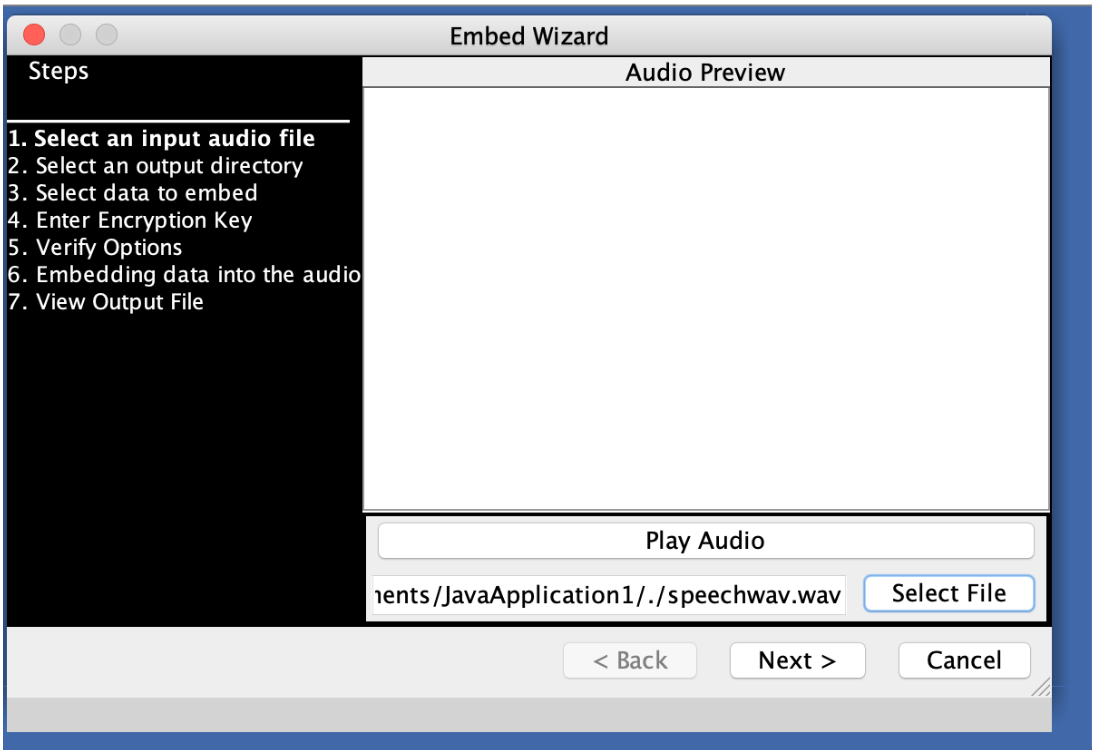
In the method below we have used AES encryption in which BASE 64 is used for salting and the secret key is taken from the sender.



```
public static String encrypt(String strToEncrypt, String secret) {
    try {
        setKey(secret);
        Cipher cipher = Cipher.getInstance("AES Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));
    } catch (Exception e) {
        System.out.println("Error while encrypting: " + e.toString());
    }
    return null;
}
```

Fig 13. AES Encryption Method

In the implementation, first of all, we have taken a sample audio named speechwav.wav which is of 117 kb. This is the audio in which the message will be embedded.

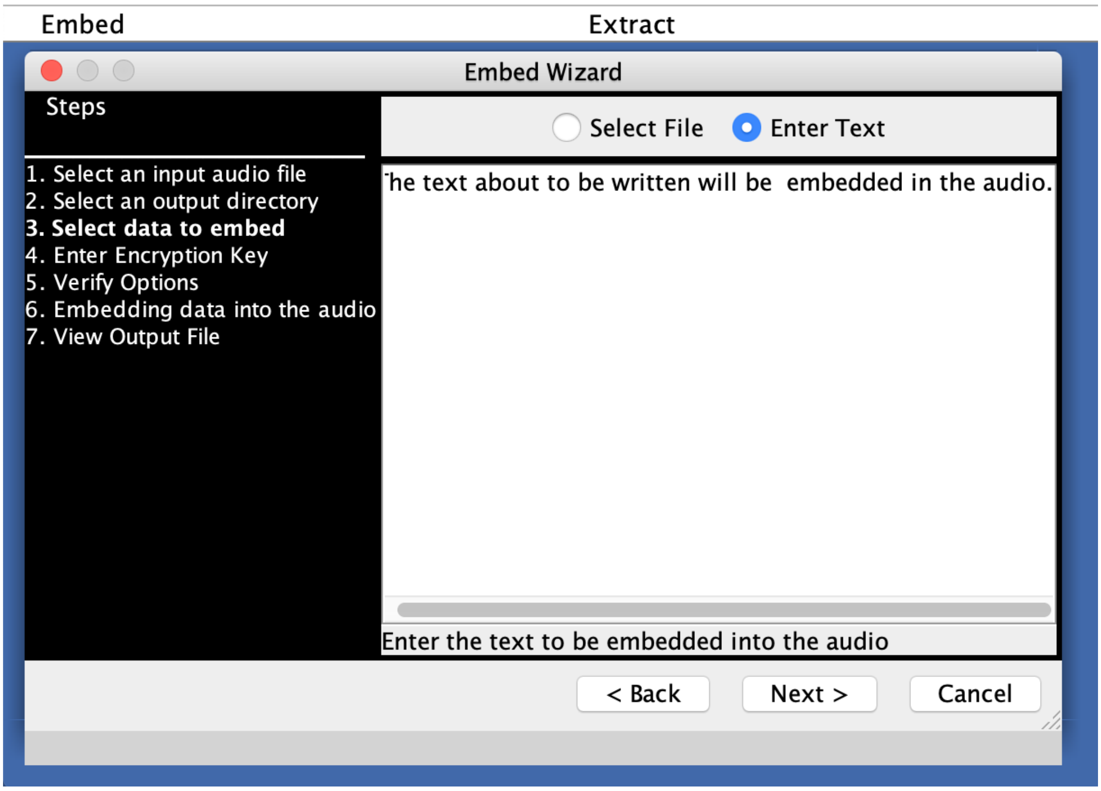


1. Select an input audio file
2. Select an output directory
3. Select data to embed
4. Enter Encryption Key
5. Verify Options
6. Embedding data into the audio
7. View Output File

ıents/JavaApplication1/./speechwav.wav

Fig 14 - Step 1- The audio file is selected

Then the message which will be encrypted and then embedded is either written or a file is attached.



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

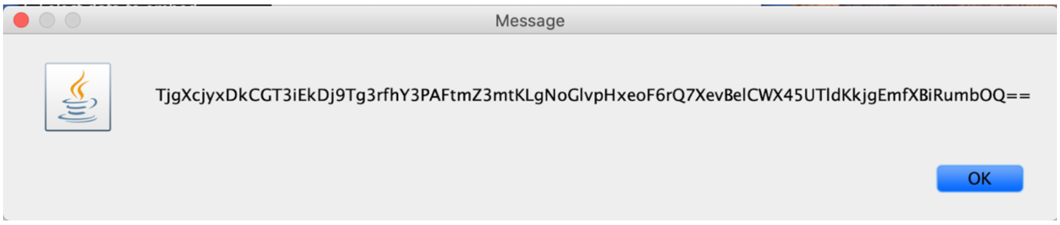
○ Select File ● Enter Text

'he text about to be written will be embedded in the audio.

Enter the text to be embedded into the audio

Fig 14 - Step 3- Enter the data to embed

After selecting the encryption key, the message is then encrypted into a cypher text shown below

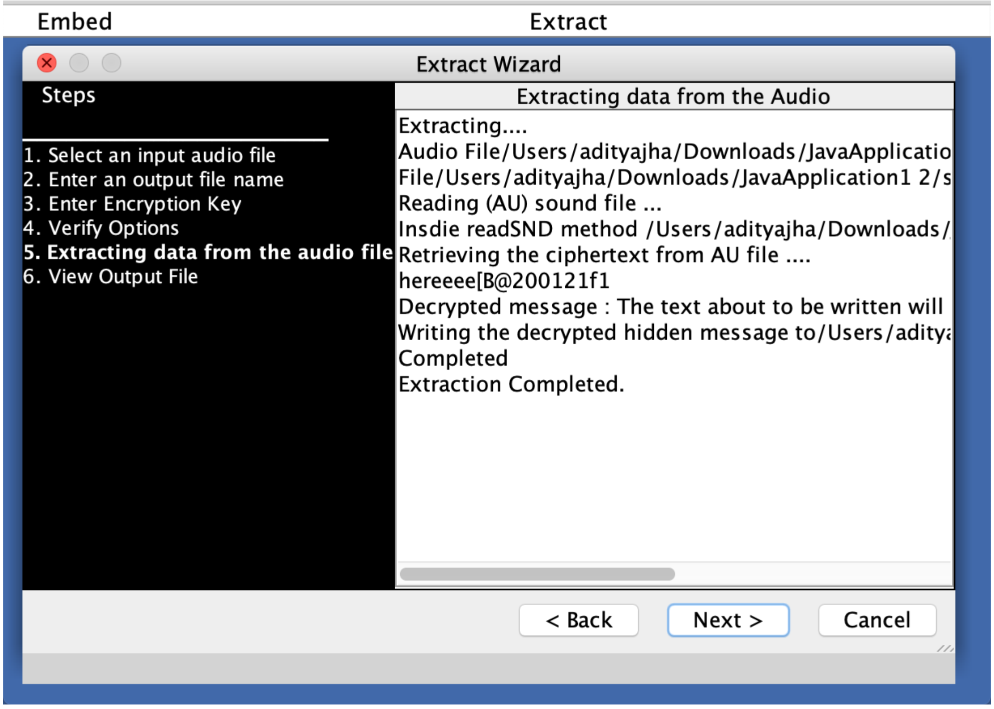


```
TjgXcjyxDkCGT3iEkDj9Tg3rfhY3PAFtmZ3mtKLgNoGlvpHxeoF6rQ7XevBelCWX45UTldKkjgEmfXBiRumbOQ==
```

Fig 15. Cypher text

The cypher key is then embedded into the audio file using LSB steganography.

A stego-audio of name speechwav1.wav of similar size 117 kb will be created and saved in the selected directory.
To extract the message, the extract option is selected and after following the steps and inserting the AES key, the output will be shown at the same time a secret.txt of 60kb file will be created with the message.



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

```
Extracting....
Audio File/Users/adityajha/Downloads/JavaApplicatio
File/Users/adityajha/Downloads/JavaApplication1 2/s
Reading (AU) sound file ...
Insdie readSND method /Users/adityajha/Downloads/
Retrieving the ciphertext from AU file ....
hereeee[B@200121f1
Decrypted message : The text about to be written will
Writing the decrypted hidden message to/Users/adity
Completed
Extraction Completed.
```

Fig 16 - Step 5- Output of the extraction process

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

Evaluation and Testing

Due to some time restrictions and error in program development, we had to make two Java projects. One in which above stated audio steganography is implemented and in another one, the testing of the audio is done. The evaluation and testing are done in this project. In this project, the Signal to Noise Ratio(SNR value) and Mean Square Error(MSE) is calculated.

SNR is a measurement which calculates the comparison between the desired audio signals and the distorted audio signals. MSE calculates the square of the errors between the original and the encrypted audio.

For testing, we have taken 5 sample audio files in which message “This is steganography” is embedded and then SNR and MSE values are calculated.

Audio Extension	Audio Type	Audio Size(in kb)	SNR value(in dB)	MSE	Bit Error
Speechwav.wav	Original	117	5.636014274227477	0.15479867783745405	0
	Encrypted	117	5.747632917700155	0.39344463122204887	
Clapred.wav	Original	256	7.5766343874891575	0.15898864469793175	0
	Encrypted	256	7.612811887505895	0.3987338018000628	
Ensoniq-ZR-76-01-Dope-77.mp3	Original	549	11.19705257143037	7.354646117922261E-5	1
	Encrypted	548	11.196995444906776	0.008575923342662444	
file_example_1.au	Original	1100	10.107580458739328	0.23544766008853912	482
	Encrypted	618	10.407824829594965	0.4852294921875	
file_example_2.au	Original	2100	10.175634697657541	78.21623235940933	1482
	Encrypted	618	10.326870997665527	8.843994140625	

From the evaluation, it can be concluded there is no data loss in wav audio file formats, in mp3, very little is lost but in .au files a large amount of data is getting lost.

Limitations

The limitation of this project is, it is not much effective for .au extension file. Also, two projects are implemented, one for audio steganography and one for evaluating the SNR and MSE values.

Conclusion and Future Work

The proposed method enhances the security of audio steganography by applying the AES encryption, using the concepts of both steganography and cryptography. Before embedding of data, the data is first converted into the encrypted text maintaining the authenticity of the data during transmission.

So, in future work, the project should be enhanced in such a way that audio steganography could be implemented for .au file extensions, and both the projects could be merge. Also, another encryption algorithm can be tried accordingly.

References

[1] Kessler, G., 2020. *An Overview Of Cryptography*. [online] Garykessler.net. Available at: <https://www.garykessler.net/library/crypto.html#intro> [Accessed 3 August 2020].

[2] P. P. Balgurgi and S. K. Jagtap, "Intelligent processing: An approach of audio steganography," *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, 2012, pp. 1-6, DOI: 10.1109/ICCICT.2012.6398182.

[3] R. Subhashini and K. B. Bagan, "Robust audio watermarking for monitoring and information embedding," *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, Chennai, 2017, pp. 1-4, DOI: 10.1109/ICSCN.2017.8085650.

[4] S. P. Rajput, K. P. Adhiya and G. K. Patnaik, "An Efficient Audio Steganography Technique to Hide Text in Audio," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, 2017, pp. 1-6, DOI: 10.1109/ICCUBEA.2017.8463948.

[5] Shweta Vinayakarao Jadhav and A.M. Rawate, "A New Audio Steganography with Enhanced Security based on Location SelectionScheme", *International Journal of Scientific Engineering and Research*, 2016.

[6] S. P. Rajput, K. P. Adhiya and G. K. Patnaik, "An Efficient Audio Steganography Technique to Hide Text in Audio," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, 2017, pp. 1-6, doi: 10.1109/ICCUBEA.2017.8463948.

[7] C. C. Sobin and V. M. Manikandan, "A Secure Audio Steganography Scheme using Genetic Algorithm," 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 2019, pp. 403-407, doi: 10.1109/ICIIP47207.2019.8985689.

[8]Shujun Li, Xuanqin Mou, Yuanlong Cai, Improving security of a chaotic encryption approach,Physics Letters A,Volume 290, Issues 3–4,2001,Pages 127-133,ISSN 0375-9601,https://doi.org/10.1016/S0375-9601(01)00612-0.(http://www.sciencedirect.com/science/article/pii/S0375960101006120).

[9]M. Anwar, M. Sarosa and E. Rohadi, "Audio Steganography Using Lifting Wavelet Transform and Dynamic Key," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 133-137, doi: 10.1109/ICAIIT.2019.8834579.

[10] J. Mohajon, Z. Ahammed and K. Hasan Talukder, "An Improved Approach in Audio Steganography Using Genetic Algorithm with K-Bit Symmetric Security Key," 2018 21st International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2018, pp. 1-6, doi: 10.1109/ICCITECHN.2018.8631918.

[11] S. Hemalatha, U. D. Acharya and A. Renuka, "Audio data hiding technique using integer wavelet transform", *International Journal of Electronic Security and Digital Forensics*, vol. 8, no. 2, pp. 131-147, 2016

[12] A. Al-Hooti, M. Hatem, S. Djanali and T. Ahmad, "Audio data hiding based on sample value modification using modulus function", *Journal of information processing systems*, vol. 12, no. 3, 2016.

[13] K.-C. Choi, C.-M. Pun and C. P. Chen, "Application of a generalized difference expansion based reversible audio data hiding algorithm", *Multimedia Tools and Applications*, vol. 74, no. 6, pp. 1961-1982, 2015.

[14] A. H. Ali, L. E. George, A. Zaidan and M. R. Mokhtar, "High capacity transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain", *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 31 487-31 516, 2018.

[15] P. Zhang, Y. Li, X. Ma, Y. Fan and X. Chen, "Efficient audio data hiding via parallel combinatory spread spectrum", *2015 8th International Congress on Image and Signal Processing (CISP)*, pp. 814-818, 2015.

[16] A. H. Ali, M. R. Mokhtar and L. E. George, "Enhancing the hiding capacity of audio steganography based on block mapping", *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 7, pp. 1441-1448, 201711

[17] Della Baby, Jitha Thomas, Gisny Augustine, Elsa George, Neenu Rosia Michael,A Novel DWT Based Image Securing Method Using Steganography,Procedia Computer Science,Volume 46,2015,Pages 612-618,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2015.02.105. (http://www.sciencedirect.com/science/article/pii/S1877050915001696)

[18] B. Geethavani, E. V. Prasad and R. Roopa, "A new approach for secure data transfer in audio signals using DWT," 2013 15th International Conference on Advanced Computing Technologies (ICACT), Rajampet, 2013, pp. 1-6, doi: 10.1109/ICACT.2013.6710492.

[19]R. Das, D. Mukherjee, R. S. Singh, S. Godara and S. Kumar, "DWTAS: A robust discrete wavelet transform approach towards audio steganography," 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Bangkok, 2017, pp. 198-204, doi: 10.1109/IEMECON.2017.8079588.

[20] Djebbar, F., Ayad, B., Meraim, K.A. *et al.* Comparative study of digital audio steganography techniques. *J AUDIO SPEECH MUSIC PROC.* 2012, 25 (2012). https://doi.org/10.1186/1687-4722-2012-25

[21] Gruhl D, Bender W: Echo hiding, Proceeding of the 1st Inforomation Hiding Workshop, Lecture Notes in Computer Science,. (Isaac Newton Institute, England, 1996), pp. 295–315

[22] Sun Micro. Java Security Overview, White paper, April 2005.

[23] SSL Client Socket Example Retrieved. DOI=http://www. exampledepot. com/egs/javax. net. ssl/client. Html.

[24] MyKong. JCE Encryption-Data Encryption Standard (DES) Tutorial. DOI=http://www. mkyong. com/java/jce-encryption-data-encryption-standard-des- tutorial.

[25]"How to Write Insecure Code | OWASP", *Owasp.org*, 2020. [Online]. Available: https://owasp.org/www-community/How_to_write_insecure_code. [Accessed: 09- Aug- 2020].

[26]"What Is AES - The World's Most Popular Encryption Method", *TechJury*, 2020. [Online]. Available: https://techjury.net/blog/what-is-aes/. [Accessed: 08- Aug- 2020].

[27]Lindawati and R. Siburian, "Steganography implementation on android smartphone using the LSB (least significant bit) to MP3 and WAV audio," 2017 3rd International Conference on Wireless and Telematics (ICWT), Palembang, 2017, pp. 170-174, doi: 10.1109/ICWT.2017.8284161.