

**Audio Encryption and Authenticity using Digital Signature**



**MINI PROJECT REPORT**



**Submitted by**

**SHUBHAM GHOSH (715517104047)**

**THARAN S (715517104054)**

**A K SHARAN SHANKAR (715519104046)**

**in partial fulfilment for the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH,  
COIMBATORE 641 062**

**ANNA UNIVERSITY: CHENNAI - 600 025**

**JUNE 2022**

# **ANNA UNIVERSITY: CHENNAI - 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**Audio Encryption and using Digital Signature**” is the bonafide work of “**SHUBHAM GHOSH (715517104047), THARAN S (715517104054), A K SHARAN SHANKAR (715519104046)**” who carried out the project work under my supervision.

-----  
**SIGNATURE**

**Dr. R. Manimegalai**

**HEAD OF THE DEPARTMENT**

Professor and Head

Computer Science and Engineering

PSG Institute of Technology and

Applied Research,

Coimbatore – 641 062

-----  
**SIGNATURE**

**Dr D. Sivaganesan**

**SUPERVISOR**

Professor

Computer Science and Engineering

PSG Institute of Technology and

Applied Research,

Coimbatore – 641 062

**Submitted for the project viva-voce Examination held on \_\_\_\_\_**

-----  
**INTERNAL EXAMINER**

-----  
**EXTERNAL EXAMINER**

## **ACKNOWLEDGMENT**

First and foremost, we express our heartfelt gratitude to our honorable Managing Trustee, **SHRI. L GOPALAKRISHNAN** for his invaluable advice and moral support.

We would also like to express our deepest gratitude to our beloved Principal, **Dr. G. CHANDRAMOHAN, B.E.(Hons), M.Tech., Ph.D.**, for his overwhelming support and encouragement on this project.

We take this opportunity to extend my humble gratitude to the Secretary of our institution, **Dr. P. V. MOHANRAM, B.E. (Hons), M.Tech., Ph.D.**

We are greatly indebted to **Dr. R. MANIMEGALAI, M.E, Ph.D.**, Head of the Department, Computer Science and Engineering for her guidance and continuous support which was instrumental in the completion of this project.

We extend our thanks to our guide, **Dr. D. Sivaganesan, B.E., M.Tech., Ph.D.**, Professor, for his technical support and constant supervision without which we could not have completed this project study.

Finally, we would also like to whole heartedly thank our project coordinator **Mr. N. ARAVINDHRAJ, B.E, M.E**, Assistant Professor, for carrying out reviews smoothly and the valuable feedback provided at each step of the project development.

**SHARAN SHANKAR A K**

**SHUBHAM GHOSH**

**THARAN S**

## VeriGuide - Originality Report

### Individual Report

#### Background Information

File Name: ASEAV\_REPORT.docx  
Report Generated On: 06/06/2022, 03:51:23 AM

#### Similarity Statistics Overview

Similar Sentence(s) Found By VeriGuide: 28 out of 436 sentences = 6.42%

Similar Sentence(s) Filtered by User: 28 out of 436 sentences = 6.42%

Sentence(s) Selected By User To Export: 0

#### Similarity Statistics for Each Source

Entry	Source	From	Similarity
1	<a href="https://www.techtarget.com/searchsecurity/definition/digital-signature">https://www.techtarget.com/searchsecurity/definition/digital-signature</a>	Internet	14 / 436 = 3.21%
2	<a href="https://www.diva-portal.org/smash/get/diva2:695339/FULLTEXT01.pdf">https://www.diva-portal.org/smash/get/diva2:695339/FULLTEXT01.pdf</a>	Internet	6 / 436 = 1.38%
3	<a href="https://www.emptrust.com/blog/benefits-of-using-digital-signatures/">https://www.emptrust.com/blog/benefits-of-using-digital-signatures/</a>	Internet	4 / 436 = 0.92%
4	<a href="https://resources.infosecinstitute.com/topic/steganography-and-tools-to-perform-steganography/">https://resources.infosecinstitute.com/topic/steganography-and-tools-to-perform-steganography/</a>	Internet	3 / 436 = 0.69%
5	<a href="https://m.mu.edu.sa/sites/default/files/content/2018/12/Steganography%20Using%20Images.pdf">https://m.mu.edu.sa/sites/default/files/content/2018/12/Steganography%20Using%20Images.pdf</a>	Internet	2 / 436 = 0.46%
6	<a href="https://sectigo.com/resource-library/what-is-sha-encryption">https://sectigo.com/resource-library/what-is-sha-encryption</a>	Internet	2 / 436 = 0.46%
7	<a href="https://www.geeksforgeeks.org/what-is-digital-signature/">https://www.geeksforgeeks.org/what-is-digital-signature/</a>	Internet	2 / 436 = 0.46%
8	<a href="https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=48322&amp;section=3.2">https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=48322&amp;section=3.2</a>	Internet	2 / 436 = 0.46%
9	<a href="http://www.infosecwriters.com/text_resources/pdf/Steganography_AMangarao.pdf">http://www.infosecwriters.com/text_resources/pdf/Steganography_AMangarao.pdf</a>	Internet	1 / 436 = 0.23%
10	<a href="https://en.wikipedia.org/wiki/Cryptographic_hash_function">https://en.wikipedia.org/wiki/Cryptographic_hash_function</a>	Internet	1 / 436 = 0.23%
11	<a href="https://kaliboyz.com/wp-content/uploads/2020/11/Implementing-cryptography-using-python.pdf">https://kaliboyz.com/wp-content/uploads/2020/11/Implementing-cryptography-using-python.pdf</a>	Internet	1 / 436 = 0.23%

12	<a href="https://msrit-bucket.s3-us-west-2.amazonaws.com/Reports/Project+Abstracts+-+Pradarshana/Pradarshana+-+2018.pdf">https://msrit-bucket.s3-us-west-2.amazonaws.com/Reports/Project+Abstracts+-+Pradarshana/Pradarshana+-+2018.pdf</a>	Internet	1 / 436 = 0.23%
13	<a href="https://pure.ulster.ac.uk/en/publications/a-method-for-verifying-integrity-amp-authenticating-digital-media-2">https://pure.ulster.ac.uk/en/publications/a-method-for-verifying-integrity-amp-authenticating-digital-media-2</a>	Internet	1 / 436 = 0.23%
14	<a href="https://rdocs.rbi.org.in/rdocs/content/PDFs/GBS300411F.pdf">https://rdocs.rbi.org.in/rdocs/content/PDFs/GBS300411F.pdf</a>	Internet	1 / 436 = 0.23%
15	<a href="https://stackoverflow.com/questions/71497318/python-opencv-videocapture-returns-false-but-worked-initially">https://stackoverflow.com/questions/71497318/python-opencv-videocapture-returns-false-but-worked-initially</a>	Internet	1 / 436 = 0.23%
16	<a href="https://towardsdatascience.com/get-to-know-audio-feature-extraction-in-python-a499fdaefe42">https://towardsdatascience.com/get-to-know-audio-feature-extraction-in-python-a499fdaefe42</a>	Internet	1 / 436 = 0.23%
17	<a href="https://www.encryptionconsulting.com/education-center/what-is-sha/">https://www.encryptionconsulting.com/education-center/what-is-sha/</a>	Internet	1 / 436 = 0.23%

## ABSTRACT

The project aims to make the exchange of critical information more secure by making it discrete.

The message is hidden (embedded) in an audio file. This process is known as STEGANOGRAPHY, to be specific - AUDIO STEGANOGRAPHY.

**Steganography** is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data.

**Audio steganography** is about **hiding the secret message in the audio**. It is a technique used to secure the transmission of secret information or hide their existence. It provides confidentiality (makes it discrete) to the said message is encrypted.

- The steganography algorithm used here introduces a negligible amount of noise to the audio which is typically undetectable by humans.
- This modified audio file is the encrypted and sent to the user via suitable medium/media.

**Keywords:** Encryption, Decryption, Cryptography, Steganography, Cipher, Hash, LSB.

## LIST OF CONTENTS

CHAPTER NO	TITLE	Page No.
	ACKNOWLEDGEMENT	
	ABSTRACT	
	LIST OF CONTENTS	
	LIST OF FIGURES	
	LIST OF TABLES	
1	<b>INTRODUCTION</b>  1.1 About Project  1.2 Audio Steganography  1.3 Digital Signature  1.4 Problem Statement  1.5 Proposed Solution	1
2	<b>LITERATURE SURVEY</b>	11

	2.1 Audio Encryption and Authenticity using Digital Signature	
<b>3</b>	<b>SYSTEM DESCRIPTION</b>  3.1 Problem statement  3.2 Project description  3.3 Project Goals	<b>14</b>
<b>4</b>	<b>SYSTEM DESIGN AND IMPLEMENTATION</b>  4.1 Requirement Gathering  4.2 System flow  4.3 System Implementation	<b>16</b>
<b>5</b>	<b>RESULT AND ANALYSIS</b>	<b>21</b>
<b>6</b>	<b>CONCLUSION AND FUTURE ENHANCEMENTS</b>	<b>22</b>
	<b>APPENDIX</b>	<b>24</b>
	<b>REFERENCES</b>	<b>27</b>



## LIST OF FIGURES

Figure Number	Title	Page number
1.1	Pre Hashing	1
1.2	Post Hashing	2
1.3	Receiver Side	2
1.4	Steganography Overview	3
1.5	Steganography and de-steganography	4
1.6	LSB Coding	4
1.7	Overview of Sender	10
1.8	Overview of Receiver	10
A1	Sender Side	34
A2	Receiver Side before tampering	35
A3	Tampering the audio	35
A4	Receiver Side after Tampering	36

## LIST OF ABBREVIATIONS

Abbreviation	Expansion
WAV	Waveform Audio File Format
MSG	Message
Stego	Steganography/Steganographic
RSA	Rivest-Shamir-Adleman
AES	Advanced Encryption Standard
SHA	Secure Hash Algorithm
PIN	Personal Identification Number
CA	Certificate Authority
TSP	Trust Service Provider
CRC	Cyclic Redundancy Code
PKI	Public Key Infrastructure
LSB	Least Significant Bit
UI	User Interface
TUI	Textual User Interface
GUI	Graphical User Interface
AUD	Audio
API	Application Programming Interface

# CHAPTER 1

## INTRODUCTION

### 1.1 ABOUT PROJECT

The project aims to make the exchange of critical information more secure by making it discrete.

The message is hidden (embedded) in an audio file. This process is known as STEGANOGRAPHY, to be specific - AUDIO STEGANOGRAPHY.

Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data.

Audio steganography is about hiding the secret message into the audio. It is a technique used to secure the transmission of secret information or hide their existence. It also may provide confidentiality to secret message if the message is encrypted.

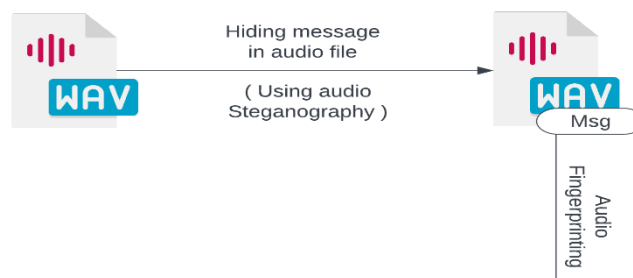


Figure 1.1: Pre Hashing

The process used introduces a negligible amount of noise to the audio file which is undetectable by humans.

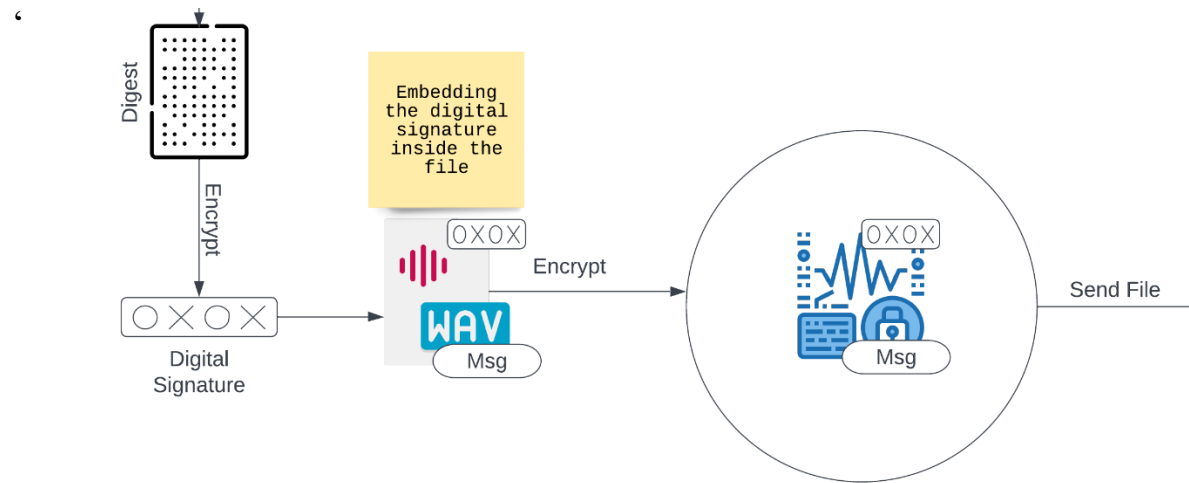


Figure 1.2: Post Hashing

This modified audio file is the encrypted and sent to the receiver via suitable media.

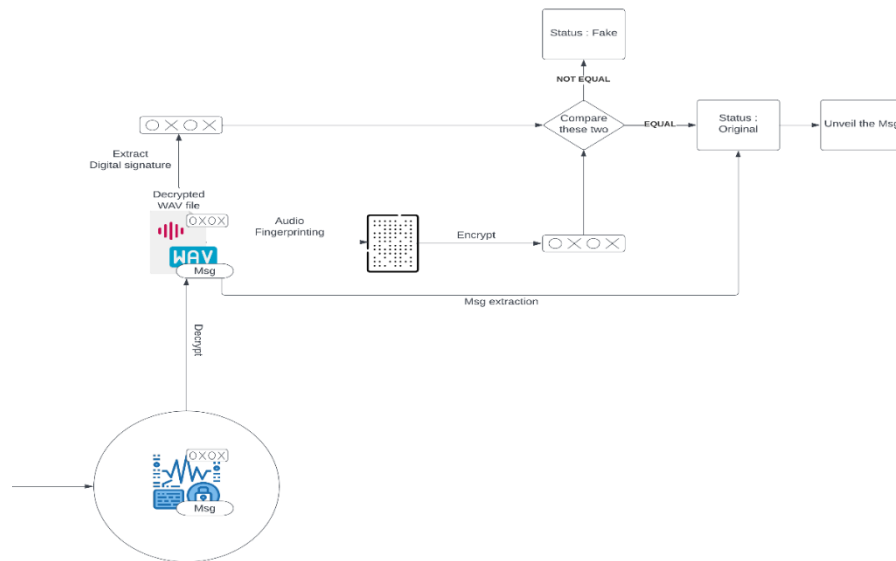


Figure 1.3: Receiver Side

## 1.2 AUDIO STEGANOGRAPHY

In steganography does not alter the structure of the secret message, but hides it inside a cover image so that it cannot be seen. . A cipher text, for instance, can easily grab the attention of attackers. On the other hand, an “invisible” message doesn’t grab the attention in the first place.

In other word, steganography prevents an unintended recipient from suspecting that the data exists. In addition, the security of classical steganography system relies on secrecy of the data encoding system. Once the encoding system is known, the steganography system is defeated.

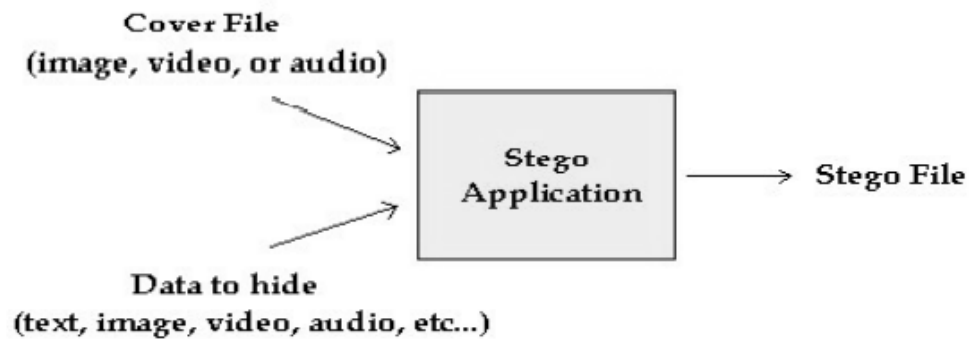


Figure 1.4: Steganography Overview

The steganography application hides different types of data within a cover file. The resulting stego also contains hidden information, although it is virtually identical to the cover file. What Steganography essentially does is exploit human perception; human senses are not trained to look for files that have information hidden inside of them, although there are programs available that can do what is called Steganalysis.

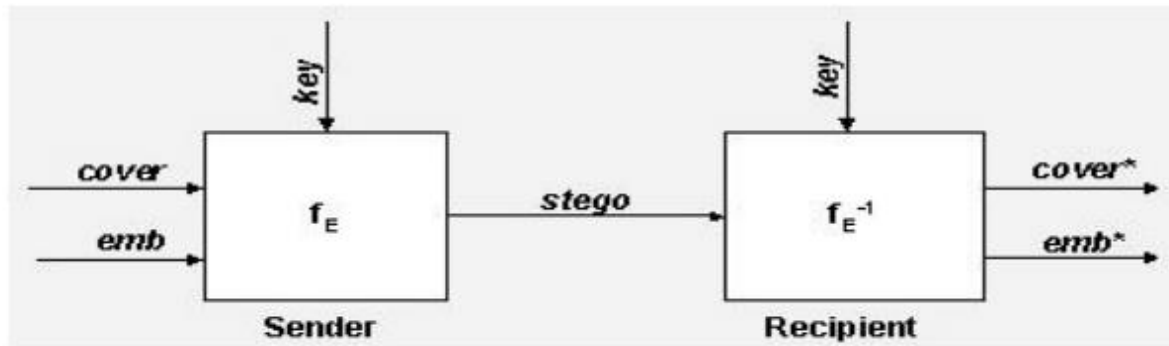


Figure 1.5: Steganography and de-steganography

## LSB CODING:

Sampled Audio Stream (16-bit)	'HEY' in binary	Audio stream w/ message encoded
1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0	0	1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0
0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1	1	0 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1	0	1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0	0	0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1
0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0	1	0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0
0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1	0	0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 0	0	0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 0
0 0 0 0 0 1 0 1 0 1 1 1 0 1 0 1	0	0 0 0 0 0 1 0 1 0 1 1 1 0 1 0 1
1 1 1 1 0 1 0 1 1 0 1 0 1 0 1 1	0	1 1 1 1 0 1 0 1 1 0 1 0 1 0 1 1
0 1 1 1 0 0 1 1 0 0 1 0 1 0 1 0	1	0 1 1 1 0 0 1 1 1 0 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1	0	1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1
0 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0	0	0 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0	1	0 1 0 1 0 0 0 1 1 0 1 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0	0	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0	1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0	0	0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0	1	0 1 0 1 0 1 0 1 1 0 0 0 1 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1	0	1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1	1	0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1	1	0 1 0 1 0 1 0 1 1 0 0 0 1 0 1 0
0 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1	0	0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0
0 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0	0	0 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0
0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 0	1	0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 1

↑  
LSB column

Figure 1.6: LSB Coding

### **1.3 DIGITAL SIGNATURE**

The entire concept of Digital Signature revolves around Public Key Infrastructure (PKI). An algorithm such as RSA can be used for this purpose. In this particular infrastructure, a pair of keys is generated. One is known as a public key and the other one is known as private key. Both the keys are related.

The entity responsible for the creation of the digital signature usually uses the private key in question to make the (encrypt) digital signature. The signature along with the private key is then sent to the receiver.

In case the receiver is not able to decrypt the signature using the sender's public key, it denotes that there is something wrong with either the key or the signature itself. This could be taken as an indication of forgery/tampering.

The entity responsible for the creation of the digital signature is to keep the private key confidential. If another entity manages to get their hands on the private key, then fake or fraudulent signature can be created.

#### **Benefits of digital signatures**

The primary use or benefit of digital signatures is the fact that it allows the parties involved to know whether the contents have been tampered with or not. Features which are used in digital signatures are as follows:

1. Something that is used to verify and authenticate things like email, username etc. is Personal Identification Number or PIN. It is widely used these days, and is one of the simpler methods out there.
2. An encryption method that uses two keys instead of one is known as asymmetric encryption.

3. Checksum is long alphanumeric string that is basically the sum of the correct digits in a piece of digital data. Comparisons are made against this to check for any errors that might have crept in. It basically acts as a data fingerprint.
4. An error-detecting code usually used in networks and storages to detect any changes that might have crept in intentionally/unintentionally in the data is CRC (Cyclic Redundancy Code)
5. Validation via Certificate Authority (CA). CAs are regarded as the trusted non first-party entities that issue digital signatures. They do so by authenticating, accepting and maintaining DCs (Digital Certificate). Their usage stops the creation of counterfeit certificates.
6. Someone or something that does validations on behalf of a company and provides signature validation reports is known as a TSP (Trust Service Provider)

**Other benefits to using digital signatures include the following:**

1. One of the most important benefits is timestamping. It is especially useful when timing is of utmost importance. For example in stock trading, lottery tickets etc.
2. Digital signatures are accepted worldwide and adheres to government laws and regulations. As the time passes, Digital Signatures are becoming an integral part of the world.
3. Processes such as signing, storage and exchange of documents physically is exhausting and time consuming. This is resolved by digital Signatures as it is much faster than the traditional ways, saving a lot of time.
4. Physical resources like papers, labors etc. involve a high amount of cost. This is eliminated by Digital Signatures, as it doesn't require any of it.



5. Reducing paper usage reduces the amount of trees cut down, which in turn has a great positive impact on the environment.
6. Chances of mistakes and misplacements are very high in the traditional mode. However, this is very low in case of Digital Signatures as everything is done digitally.

### **Creation of a Digital Signature**

Firstly, a software or an algorithm provides the one way hash for the data under consideration, i.e., the data to be signed.

On a very basic level, a hash is a sequence of alphanumeric character that is irreversible and is unique for a particular piece of information. The creator's private key is used to encrypt the hash. For this Asymmetric Algorithms like RSA can be used. The output hence received, along with the various other information is collectively known as the digital signature.

There are a few reasons for encrypting the hash and not the message itself. Firstly, hash is irreversible whereas encryption is not. We don't want the intruder to be able to get the data back from the signature itself. Another reason is the fact that hashes are of fixed length and are very short when compared to large chunks of texts. Hence, encrypting a hash is comparatively easier and less time consuming than encrypting the message itself.

The hash for a particular piece of data is unique to it. Change in even a single character will result in the hash being different. Now it is clear why hashes are chosen as the first priority when it comes to data integrity.

Say the decrypted hash matches with the newly computed hash of the same data. Then it implies that the data has not been compromised. If it doesn't match however, it means that either the integrity of the data was compromised or the public key and the private key are not related.

## **1.4 PROBLEM STATEMENT**

Information is one of the most valuable assets that any individual or any organization holds these days. Naturally, with great importance comes great threats. As a result, many malicious organizations/individuals are specifically inclined towards acquiring information of the said individuals/organizations. In these cases, it becomes extremely important to safeguard the information.

Most of the theft or manipulation in information takes place when it is being transferred from one point to another. Even though organizations or individuals might use security mechanisms like encryption, it still draws the attention of a few malicious individuals towards it. Seldom the information is not stolen completely, rather it is modified (can be intentional or unintentional) so that its original purpose is lost.

Hence, the ultimate goal is to make the information secure and discreet (so that the malicious individuals do not even realize that a critical information is present in the first place) all the while ensuring that the recipient gets to know whether they have received the intended message without any modification/tampering or not.

In order to make all the aforementioned points come true, this project is being developed.

## **1.5 PROPOSED SOLUTION**

### Process 1

Audio Steganography is used to hide the message or information in the audio file.

### Process 2

This process is involved for the creation of a digital signature for a file. In this, we first hash the original audio file, which gives a sequence of pseudo-random characters known as a digest. Then the digest is encrypted using a cipher (usually AES ciphers), which produces another seemingly meaningless sequence of characters. This is known as the digital signature, and is guaranteed to be unique for a given file.

This process doesn't affect the original audio file.

### Process 3

The file with the hidden message is now encrypted. The digital signature is also added to the encrypted file.

### Process 4

The resultant of the previous step is now sent to the receiver end.

### Process 5

This is the receiver side. The digital signature is now extracted. The encrypted contents are decrypted. The embedded digital signature is then checked with a newly created digital signature of the decrypted file.

If any modification has taken place, the match returns negative, else positive.

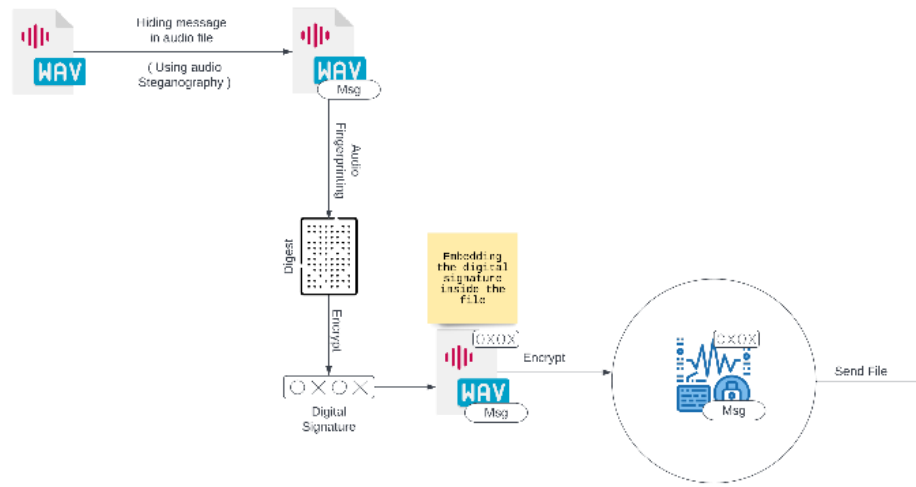


Figure 1.7: Overview of Sender

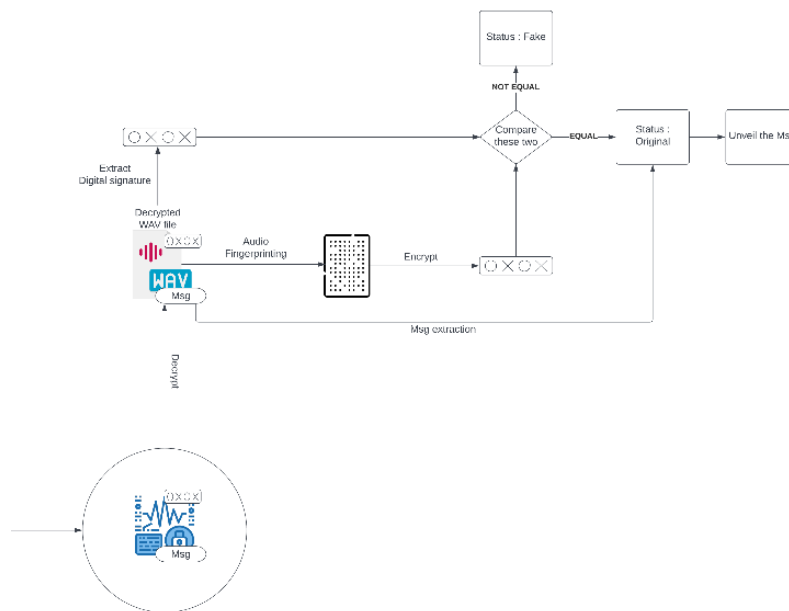


Figure 1.8: Overview of Receiver

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 Audio Encryption and Authenticity using Digital Signature**

##### **Audio Steganography Techniques: A Survey;**

By - Shilpi Mishra, Virendra Kumar Yadav, Munesh Chandra Trivedi and Tarun Shrimali

**LSB coding:** LSB hiding is an easy and quick procedure for inserting data in a sound sign. In LSB approach LSB of binary series of every illustration of digital audio data is interchanged with binary equivalent of secret data. The limit is stand out bit every sample of the spread sound which could be less for some applications. In LSB code, the perfect information transmission rate is 1 kbps in line with 1 kHz. In a few executions of LSB code, on the other hand, the two least significant bits of a series are supplanted with two data bits. This will increase the quantity of information that may be encoded however conjointly will increase the quantity of ensuing noise within the audio file additionally. A more modern methodology is to utilize a pseudorandom amount generator to extend the information over the sound document in an arbitrary way. One prevalent methodology is to utilize the random temporary strategy, in which a secret key controlled by the sender is utilized as a concept as a part of a pseudorandom amount generator to make an irregular succession of test lists. This approach is two deprivation connected with the utilization of systems like LSB coding. The human ear is exceptionally delicate and can frequently distinguish even the scarcest bit of commotion brought into a sound document, second deprivation on the other hand is that this is not strong. In the event that a sound record installed with a mystery message utilizing either LSB coding was resample, the implanted data would be lost.

**Parity coding:** The signal is separated into gathering of specific pattern or test known as pattern location and this sample region every bit encoded from the Audio Steganography Techniques: A Survey 583 concealed message in a pattern location parity bit. In the event that the equality bit of a choose area does not coordinate, the mystery bit to be encoded procedure flips the LSB of one of the samples inside the location. So the sender has to a greater extent a decision in encoding the secret bit.

**Advantage:** The sender has even more a decision in encoding the mystery bit and the sign can be modified in a more subtle way.

**Limitation:** This technique like LSB code is not strong in quality. The limit remains the same as that of LSB strategy.

**Echo Hiding:** This technique acquaints a shorten echo with the host signal and afterward inserts information in it. Three operation of echo signal are controlled for concealing information: Initial Amplitude, the offset (delay), the decay rate. Echo hiding strategy inserts information inside mask audio sign by presenting an echo.

1. The confidential message sent will pass through the two layers and then be embedded in the cover message on the third layer. The stego message is sent to the receiver over some network which is assumed to be secured, and the message is retrieved by doing the aforementioned operations in reverse order.
2. To achieve this, dual layer randomization approach is used. First layer of randomization is achieved by randomly selecting the byte number or samples. An additional layer of security is provided by randomly selecting the bit position at which embedding is done in the selected samples. Using this proposed algorithm the transparency and robustness of the steganographic technique is increased.

3. The main reason why LSB method is not preferred usually lies in its lack of robustness. Rather than using one bit for storing the information, two bits (2nd and 3rd LSB's) can be used for the hiding purpose. The mentioned method has an added benefit of extra storage capacity. The changes occurred in the stego file is to be minimal. For this purpose a filter is designed. The filtered file and the stego file together are used to form the unique key.
4. Robustness can be increased by embedding message bits into multiple and higher layer values. This can be achieved by using genetic algorithms. The robustness definitely increases against intentional attacks, but it also has an added benefit of protecting against unintentional factors such as noise.

## **CHAPTER 3**

### **SYSTEM DESCRIPTION**

#### **3.1 PROBLEM STATEMENT**

The exchange of critical information is a vulnerable process in today's world. In order to exchange highly confidential messages discreetly and to ascertain the authenticity of the message at the receiver end, this project is being developed.

#### **3.2 PROJECT DESCRIPTION**

The project aims to make the exchange of critical information more secure by making it discrete.

The message is hidden (embedded) in an audio file. This process is known as STEGANOGRAPHY, to be specific - AUDIO STEGANOGRAPHY.

Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data.

Audio steganography is the process of embedding a confidential message in the some audio, effectively hiding it. It is a technique used to secure the transmission of secret information or hide their existence. It also may provide confidentiality to secret message if the message is encrypted

In steganography does not alter the structure of the secret message, but hides it inside a cover image so that it cannot be seen. A cipher text, for instance, can easily grab the attention of attackers. On the other hand, an "invisible" message doesn't grab the attention in the first place.

In other words, steganography prevents an unintended recipient from suspecting that the data exists. In addition, the security of classical steganography



system relies on secrecy of the data encoding system. Once the encoding system is known, the steganography system is defeated.

### **3.3 PROJECT GOALS**

1. The project aims to make the exchange of critical information more secure by making it discrete.
2. To hide a given piece of information inside the audio file using a method known as steganography.
3. To introduce negligible noise to the audio file, which would be undetectable by the human ear.
4. To encrypt the message-containing file, making it suitable for transmission and preventing it from falling into the wrong hands.

## **CHAPTER 4**

### **SYSTEM DESIGN AND IMPLEMENTATION**

#### **4.1 REQUIREMENTS GATHERING**

VS Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

The python packages used are wave, hashlib, cryptography. All of them can be installed on the system beforehand, primarily using the pip command.

#### **4.2 SYSTEM FLOW**

##### Process 1

Audio Steganography is used to hide the MESSAGE in the file.

##### Process 2

To acquire the Digital Signature we “HASH” the ORIGINAL AUDIO using hashing algorithm, where DIGEST is generated and the digest is encrypted to form the DIGITAL SIGNATURE.

This process (2) doesn't affect the original audio file.

##### Process 3

Original audio is ENCRYPTED.

##### Process 4

Encrypt the audio file. Add the Digital Signature along with the encrypted file.

#### Process 5

Send the SIGNED ENCRYPTED FILE to the receiver side.

#### Process 6

Now extract Digital signature from the file and DECRYPT the encrypted file. Compare Sign and Decrypted file. If it's mismatched, TAMPERING has occurred else VERIFIED.

### **4.3 SYSTEM IMPLEMENTATION**

The system is developed using python with few algorithms implementation such as Fernet for encryption, sha256 for hashing (Digest creation).

#### **sender.py**

Function: add\_msg

Parameters: audio\_file, string, output.

The audio file with name audio\_file will be opened in binary mode. Every frame of the said file is read. The bytes are stored as bytearray (mutable) in an object named frame\_bytes.

String is the message to be hidden. The length of the audio file and the string is compared, the remaining space in the string is filled with a placeholder value, which is '@' in this case. After this, every bit of the string is converted to a byte long (8 bits) by using lstrip and rjust. It is then stored in a list object named bits.

Then iteration is done for bits and frame\_bytes simultaneously using enumeration object. That is, for ith iteration, we select bits[i] and frame\_bytes[i]. A mask is created for frame\_bytes[i] by AND-ing it with 245 (11111110). Hence, it leaves the last bit empty for modification, which we do by OR-ing with bits[i].

Hence, only the LSB of the  $i$ th frame is changed for every  $i$ . Every modified bit is then stored in a bytearray object named `modified_bytes`.

The contents of `modified_bytes` is then recombined to form an output audio file which is named as `output`.

#### Function: `hash_audio`

Parameters: `input`

An audio file named `input` is read and its contents are hashed using the sha256 algorithm. The digest so obtained is returned. This is an intermediate function, and the user doesn't have access to it.

#### Function: `generate_sign`

Parameters: `digest`

This function encrypts the digest using fernet algorithm to form the sign and returns the key thus generated and the sign. This is also an intermediate function, and the user has no access to it.

#### Function: `implant_key`

Parameters: `sign`, `key`

This function produces a hybrid signature inside which the key is embedded, so that the process of decoding can be automated at the receiver side. It calls `generate_index()`, which returns an index based on which the key is implanted to the sign. This too is an intermediate function.

#### Function: `audio_encrypt`

Parameters: `audio_file`, `key`, `output`, `implant_key`

This function encrypts the message-bearing audio file (audio\_file) using the given key and adds to it the hybrid key generated earlier (implant\_key) and stores the contents in an audio file named output.

### **receiver.py**

#### Function: extract\_components

Parameters: audio\_file

This function separates the encrypted contents from the hybrid signature and returns them separately. It returns the Meta data of the audio file, the encrypted contents and the hybrid signature.

#### Function: decrypt\_audio

Parameters: encrypted\_contents, key, params

This function decrypts the encrypted frames (encrypted\_contents) returned by extract\_components() using the key. It then writes the decrypted contents and the params back to an audio file. This new file so formed contains the hidden message.

#### Function: ex\_msg

Parameters: audio\_file

This function extracts the hidden message from the given audio\_file and displays it. This is done by reading every frame of the audio\_file and AND-ing it with 1 (00000001), which yields the last bit where our string is stored. The last bits of every frame is then recombined to get the hidden message.

#### Function: verify\_signature

Parameters: audio\_file, sign, key

This function decrypts the sign (the sign extracted from the audio file) using the key to get a digest. Then it hashes the file currently at hand to get another digest. Then it compares both the digests to check for their similarity. If they match exactly,

then the signature is verified and the content is not tampered. Else, the content is concluded to be tampered.

### **tamper.py**

This file simulates the tampering of the file. When run, this file selects a bunch of audio frames and mutates them to some other value. Its objective is to showcase the verification capabilities of the software.

## **CHAPTER 5**

### **RESULT AND ANALYSIS**

Hence, an application software has been developed that achieves the said requirements, which are steganography and encryption. The application hence produced is extremely lightweight and flexible. The implementation is easy to understand, thanks to it being a product of python.

The application secures the messages by embedding it in an audio file and encrypts the audio file as well. There's no overhead of sending keys by the sender as everything is embedded securely in the file itself, making the extraction and decryption process near automatic. The encryption algorithm used here is also extremely powerful i.e., secure.

An abstracted version of the software is planned to be released as a publicly accessible commodity, making it useful for real-life applications. An API (Application Program Interface) can also be developed by doing slight modifications to the actual software, which will take its utility a notch higher. It can be then used as a “medium” for encryption and hashing for day-to-day communication or message exchange.

The application software also provides a learning opportunity for individuals, mostly students, to enter the world of cryptography and explore the same. The functions are documented and are well defined.

Even though the software has countless advantages, there are a few limitations, which are documented in the following chapter and can be rectified easily.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE ENHANCEMENTS**

#### **6.1 Conclusion**

In today's world where the theft of the most valuable asset, i.e. information, is common, a method has to be developed to eliminate or at least minimize theft of information. This software helps in achieving the same to a large extent. The software aims at not only making the transmission of information secure, but also to make it discrete. This is because an asset that does not draw the attention of entities in the first place would not be stolen. Making the transmission discrete is an underrated approach to information security, as it reduces the need to invest in resource intensive methods such as encryption, though these methods can and should be kept as a backup in case theft occurs.

Many software often forget the aspect of tampering during transmission. As a result if the information is tampered during transmission, there's no way for the receiver to know that it was tampered and the receiver believes that the tampered message is the one that was originally sent by the sender. This, in most of the cases, leads to inconsistency. However, this software captures this aspect as well and offers checking at the receiver side so that appropriate measures can be taken by the involved parties and the inconsistency is hence minimized to a great extent.

#### **6.2 Future Enhancements**

The software covers a few of the major aspects of information security. However, it does lack in some minor aspects which can be rectified in future enhancements easy, thanks to the software being extremely flexible. Some of these enhancements are explained as follows.

The software is entirely offline, which means the entire processing takes place on a local machine, which is actually a good thing when security is concerned, as it has no influence from the outside world via the internet. However, it does pose a slight issue as the actual transmission takes place via a third-party medium. This can



be rectified in future by providing a secure transmission media, allocated specifically for the users of this software. Though challenging, but it is certainly plausible.

Another one of the aspects that the software can improve upon is its User Interface (UI). The software provides a Textual User Interface (TUI) which is frowned upon by a major portion of the public and certainly presents itself as challenging to a lot of individuals. However, this feature is pseudo-intentional, as a Graphical User Interface (GUI) comes with its own set of loopholes and disadvantages, which might pose a threat to security, which in-turn nullifies the entire motive of the software.

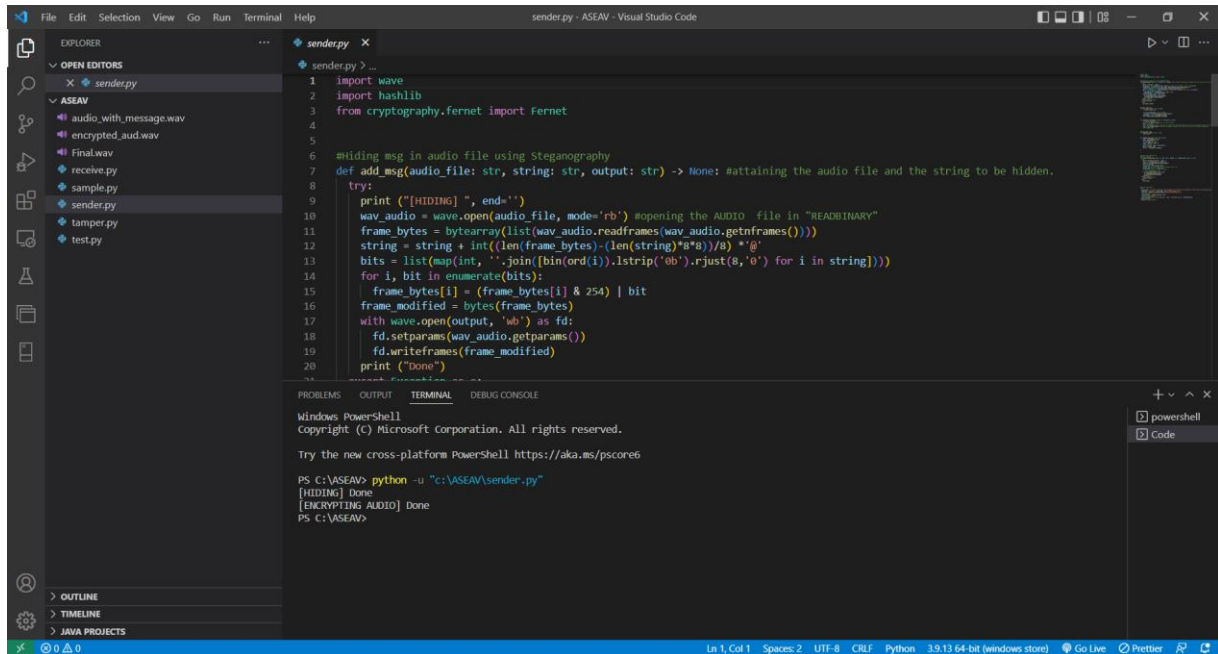
The software caters to a niche category of people; hence it is not understandable by a major portion of the public. This can be fixed by adding a User Interface (as mentioned above) and using user-friendly and easy to understand terms.

The encryption and hashing algorithms used in the software are verified and without a doubt, strong. In fact, cracking one of the messages is expected to take thousands of years quite literally. However, it is still not up to the level used in military applications. Hence, even stronger algorithms can be used, which will allow this software to be used even for military purposes. However, this “limitation” too was kept keeping the power of an average computer in mind. The newer and stronger algorithms require a lot of computation power.

Multiple steganography algorithms can be introduced, the user can select one based on their requirements, and what they are willing to trade-off.

A light-weight version of the software can be launched as a website, which might encourage the public towards encryption, hashing, steganography and cryptography in general, which will accelerate the growth of information security methodologies and will be beneficial to the mankind as a whole.

## APPENDIX



The screenshot displays the Visual Studio Code interface with a Python script named `sender.py` open in the editor. The script implements a steganography function to hide a message in an audio file. The Explorer sidebar on the left shows the project structure, including files like `audio_with_message.wav` and `encrypted_aud.wav`. The terminal at the bottom shows the command `python -u "C:\ASEAV\sender.py"` being executed, with output indicating the message was successfully hidden in the audio file.

```
1 import wave
2 import hashlib
3 from cryptography.fernet import Fernet
4
5
6 #hiding msg in audio file using Steganography
7 def add_msg(audio_file: str, string: str, output: str) -> None: #attaining the audio file and the string to be hidden.
8     try:
9         print("[HIDING] ", end='')
10        wav_audio = wave.open(audio_file, mode='rb') #opening the AUDIO file in "READONLY"
11        frame_bytes = bytearray(list(wav_audio.readframes(wav_audio.getnframes())))
12        string = string + int((len(frame_bytes)-(len(string)*8))/8) * '@'
13        bits = list(map(int, ''.join([bin(ord(i)).lstrip('0b').rjust(8,'0') for i in string]))
14        for i, bit in enumerate(bits):
15            frame_bytes[i] = (frame_bytes[i] & 254) | bit
16        frame_modified = bytes(frame_bytes)
17        with wave.open(output, 'wb') as fd:
18            fd.setparams(wav_audio.getparams())
19            fd.writeframes(frame_modified)
20        print("Done")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\ASEAV> python -u "C:\ASEAV\sender.py"
[HIDING] Done
[ENCRYPTING AUDIO] Done
PS C:\ASEAV>
```

Figure A1: Sender side

Note how in the above, two more files named “audio\_with\_message.wav” and “encrypted\_aud.wav” are created. The user sends the latter to the receiver. The receiver is then supposed to extract the contents from it. The receiver side has been shown in the next page.

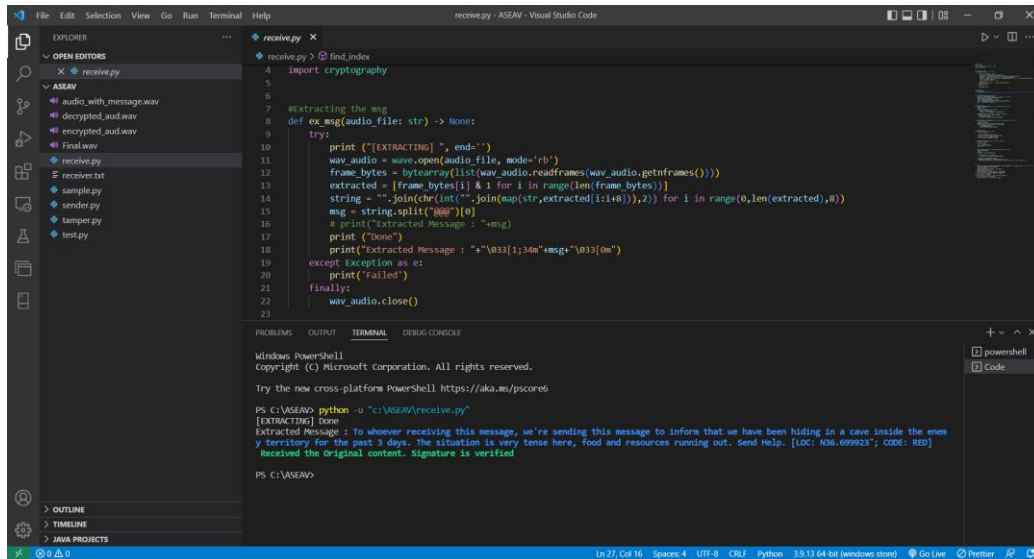


Figure A2: Receiver side before tampering

Note how a file named “decrypted\_aud.wav” is created. It is the audio containing the message.

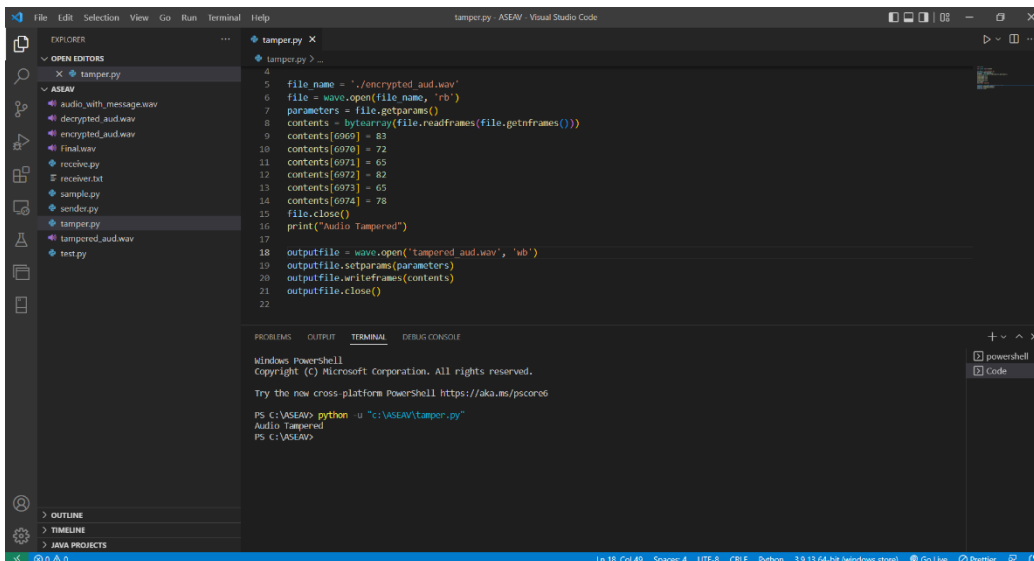


Figure A3: Tampering the audio

A new file named “tampered\_aud.wav” is generated, which is a representation of a tampered audio file.

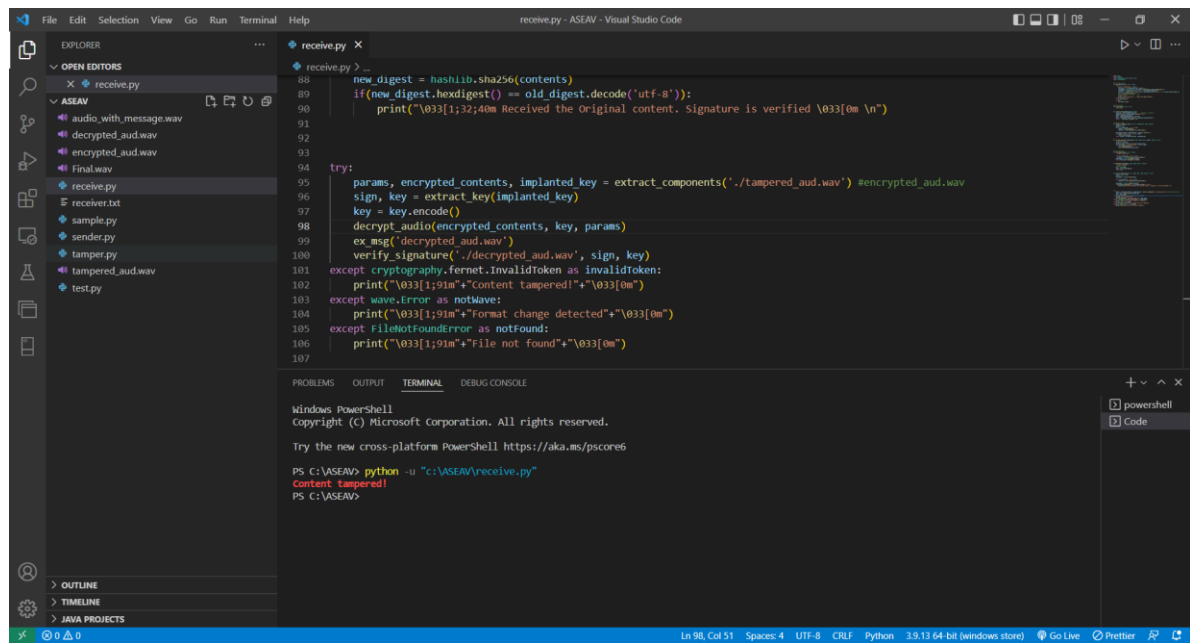


Figure A4: Receiver side after tampering

The extraction process fails as the audio file was tampered.

## REFERENCES

- [1] A method for verifying integrity & authenticating digital media: Applied Computing and Informatics, ISSN: 2210-8327, Vol: 14, Issue: 2, Page: 145-158.
- [2] Alam, Shahzad & Jamil, Mohammad & Saldhi, Ankur & Ahmad, Musheer. (2015). Digital image authentication and encryption using digital signature. 332-336. 10.1109/ICACEA.2015.7164725.
- [3] Renza, Diego & Ballesteros, Dora & Lemus, Camilo. (2017). Authenticity verification of audio signals based on fragile watermarking for audio forensics. Expert Systems with Applications. 91.
- [4] <https://www.geeksforgeeks.org/what-is-digital-signature/>
- [5] <https://www.diva-portal.org/smash/get/diva2:695339/FULLTEXT01.pdf>
- [6] G. W. Romney and D. W. Parry, "A Digital Signature Signing Engine to the Integrity of Digital Assets," 2006 7th International Conference on Information technology Based Higher Education and Training, 2006, pp. 800-805, doi: 10.1109/ITHET.2006.339702.
- [7] K.Geetha And P.VanithaMuthu," Implementation of ETAS (Embedding Text in Audio Signal) Model to Ensure Secrecy", International Journal on Computer and Engineering, Vol. 02, No. 04, 20 I O.
- [8] Cvejic N. and Seppanen T. "Increasing the capacity of LSB based audio", Proc. 5th IEEE International Workshop on Multimedia Signal Processing, St. Thomas, V I, December 2002, pp.336- 338.
- [9] Walter Bender, Daniel Gruhl, Norishige Morimoto, Anthony Lu,"Techniques for Data Hiding", IBM Systems Journal, vol. 35, no. 3 and 4, pp. 313-336, 1996.
- [10] Request for Information on Audio Fingerprinting Technologies.  
[http://www.riaa.org/pdf/RIAA IFPI Fingerprinting RFI.pdf](http://www.riaa.org/pdf/RIAA_IFPI_Fingerprinting_RFI.pdf). 2001.

- [11] L. Boney, A. Tewfik, and K. Hamdy, "Digital Watermarks for Audio Signals," IEEE Proceedings Multimedia, 473-480, 1996.
- [12] <https://blog.jscrambler.com/hashing-algorithms>
- [13] [https://en.wikipedia.org/wiki/Hash\\_function](https://en.wikipedia.org/wiki/Hash_function)
- [14] [https://en.wikipedia.org/wiki/Fingerprint\\_\(computing\)](https://en.wikipedia.org/wiki/Fingerprint_(computing))
- [15] [https://en.wikipedia.org/wiki/Acoustic\\_fingerprint](https://en.wikipedia.org/wiki/Acoustic_fingerprint)
- [16] <https://www.encryptionconsulting.com/education-center/what-is-sha/>
- [17] <https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm>
- [18] [https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)
- [19] <https://sectigo.com/resource-library/what-is-sha-encryption>
- [20] <https://www.geeksforgeeks.org/fernet-symmetric-encryption-using-cryptography-module-in-python/>
- [21] <https://cryptography.io/en/latest/fernet/>
- [22] <https://en.wikipedia.org/wiki/WAV>
- [23] <https://docs.python.org/3/library/wave.html>
- [24] <https://sound.stackexchange.com/questions/41567/difference-between-frame-and-sample-in-waveform>



# COIMBATORE INSTITUTE OF TECHNOLOGY

(A Government Aided Autonomous Institution Affiliated to Anna University)  
Avinashi Rd, Civil Aerodrome Post, Coimbatore, Tamil Nadu 641014



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING  
A NATIONAL LEVEL TECHNICAL SYMPOSIUM

ELECTERA  
2K22

## CERTIFICATE OF PARTICIPATION

THIS IS TO CERTIFY THAT Mr./Ms.

**SHARAN SHANKAR A K**

participated in the **TEK-THESIS** event organized by the Department of EEE, **COIMBATORE INSTITUTE OF TECHNOLOGY**, as a part of **Electera 2K22**, held on 27th MAY, 2022.

HEAD OF THE DEPARTMENT,  
EEE







# COIMBATORE INSTITUTE OF TECHNOLOGY

(A Government Aided Autonomous Institution Affiliated to Anna University)  
Avinashi Rd, Civil Aerodrome Post, Coimbatore, Tamil Nadu 641014



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING  
A NATIONAL LEVEL TECHNICAL SYMPOSIUM

ELECTERA  
2K22

## CERTIFICATE OF PARTICIPATION

THIS IS TO CERTIFY THAT Mr./Ms.

**SHUBHAM GHOSH**

participated in the **TEK-THESIS** event organized by the Department of EEE, **COIMBATORE INSTITUTE OF TECHNOLOGY**, as a part of **Electera 2K22**, held on 27th MAY, 2022.

HEAD OF THE DEPARTMENT,  
EEE







# COIMBATORE INSTITUTE OF TECHNOLOGY

(A Government Aided Autonomous Institution Affiliated to Anna University)  
Avinashi Rd, Civil Aerodrome Post, Coimbatore, Tamil Nadu 641014



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING  
A NATIONAL LEVEL TECHNICAL SYMPOSIUM

ELECTERA  
2K22

## CERTIFICATE OF PARTICIPATION

THIS IS TO CERTIFY THAT Mr./Ms.

**THARAN S**

participated in the **TEK-THESIS** event organized by the Department of EEE, **COIMBATORE INSTITUTE OF TECHNOLOGY**, as a part of **Electera 2K22**, held on 27th MAY, 2022.

HEAD OF THE DEPARTMENT,  
EEE

