National College of Ireland

Password storage protection using audio steganography with AES encryption

MSc Research Project
MSc in Cybersecurity

Nikhil Yadav
Student ID: x22187693

School of Computing
National College of Ireland

Supervisor: Prof. Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nikhil Adhikrao Yadav		
Student ID:	X22187693		
Programme:	MSc in Cybersecurity	**Year:**	2023-2024
Module:	MSc Research Practicum Part 2		
Supervisor:	Prof. Michael Pantridge		
Submission Due Date:	12/08/2024		
Project Title:	Password storage protection using audio steganography with AES encryption		
Word Count:	6562	**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.

Signature: Nikhil.A.Yadav

Date: 12/08/2024

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):	

PASSWORD STORAGE PROTECTION BY USING AUDIO STEGANOGRAPHY WITH AES ENCRYPTION

Nikhil Yadav
X22187693

Abstract

Securing end to end audio communication is very important but it is also having so many issues. This project presents a novel approach to secure password storage by combining audio steganography with AES encryption and SCRYPT hashing. The process starts with the user uploading an audio file and entering a password to be hidden. The original password is inserted into the audio file using Least Significant Bit (LSB) encoding by securing that the hidden data remains invisible. Following this the password experiences hashing using SCRYPT by generating a hash that serves as the key for AES encryption. The AES algorithm then encrypts the hashed password by producing ciphertext. The encryption and decryption processes have been measured for performance with results showing variations in encryption process time. The avalanche effect has been evaluated to test the sensitivity of the encryption algorithm to changes in the input password, revealing consistent high sensitivity with scores ranging from 48.23 to 51.62 which shows strong data encryption. For decryption the process includes retrieving the ciphertext from the steganographic audio which is going to extract and validate the hashed password and comparing it to the original hash. The decryption time is recorded and compared to the encryption time to evaluate performance. This approach is going to increase password security by combining steganography with advanced cryptographic techniques by having both secure storage and good data protection.

Keywords: Audio Steganography, AES Encryption, SCRYPT Hashing, Password Protection, LSB Encoding

1 Introduction

1.1 Background

Password storage protection is an important type of problem in digital security which requires good and robust methods to secure privacy and integrity (Yang et al., 2020). One of the most basic types of mechanisms for insecure networks is password authentication (Harba et al., 2018). There are some traditional approaches in encryption techniques like AES (Advanced Encryption Standard) to encode sensitive types of information. To increase security there are some type of techniques like audio steganography which can be combined. Audio steganography includes having secret or confidential type of data within audio files in a way that is invisible to human senses which is thereby giving an extra layer of hidden (Tan et al.,

2019). AES operates with keys of 128, 192, or 256 bits which gives different levels of security that depends on the key size chosen (Daoud et al., 2019).

Figure 1: AES Encryption Module

1.2 Aim of Study

The aim of this study is to increase password security with the combination of audio steganography with AES encryption and SCRYPT hashing techniques. By using audio files as a secret channel, the study aims to develop a strong system for hiding and securing sensitive type of password information. The methodology does includes first hashing the original password using SCRYPT to generate a hash key which is then used in the AES encryption process. This encrypted data is inserted within an audio file using Least Significant Bit (LSB) encoding by knowing that the original password remains hidden and protected from unauthorized types of access. The study aims to evaluate the performance of this system with metrics such as encryption and decryption process times and then evaluate its performance in maintaining data integrity and security. Also, the research explores the avalanche effect to know that even minor changes in the original password led to important changes in the ciphertext which is thereby having security. By providing a good approach to secure password storage and transmission, this study will look to give advancements in cryptographic practices and will give a good solution for protecting sensitive information in digital communications.

1.3 Research Objectives

There are some research objectives in this report are:

1. To evaluate the performance of combining AES encryption with SCRYPT hashing for securing passwords within audio files using audio steganography techniques.
2. To analyze the effect of different audio file sizes on encryption and decryption process times when passwords are inserted using audio steganography.
3. To evaluate the influence of Least Significant Bit (LSB) encoding on the audio file quality and the performance of hiding passwords in the audio.

1.4 Research Questions

There are some research questions in this report are:

1. How good is the combination of AES encryption and SCRYPT hashing in securing passwords when inserted within audio files using audio steganography techniques?
2. What are the affects of changing audio file sizes on the encryption and decryption process times for passwords hidden using audio steganography?
3. How does the Least Significant Bit (LSB) encoding method affect the quality of the audio file and the performance of password hiding?

2 Literature Review

2.1 AES Encryption in Steganography

In the literature review, the section on AES Encryption in Steganography examines the integration of Advanced Encryption Standard (AES) within various steganographic methods across five studies. It is going to explore how AES enhances data security by encrypting messages before embedding them into cover files with the help of some techniques such as Discrete Wavelet Transform (DWT), Least Significant Bit (LSB) steganography, and image/audio concealment. The review will show like how AES's role in reducing the weaknesses and ensuring robust encryption which will contribute to secure data transmission and storage in so many applications, from cloud computing to 3D image security. (Prasetyadi et al., 2018) proposes an append insertion steganography method to hide computer files of any type and size within selected cover files which do have aim to overcome format limitations which has been seen in popular steganography techniques. The method uses AES-256 (Rijndael algorithm) for encrypting messages with a secret thing. A specific kind of block of bytes ensures the message's integrity and all Implementation has been done with the help of C# programming language and .NET Framework in a Windows application, with each cover file containing exactly one hidden message file. Testing do include five random message files which has been verified for integrity using SHA-256 hashes before and after hiding which shows that all files do have their integrity post-recovery. There is another study which has been given by (Sari et al., 2019) who proposes a hybrid type of approach using AES for cryptography and DWT for steganography to secure and hide information. The process includes encrypting the message with the help of AES and then applying Huffman Coding for doing the compression before embedding it into the cover image via DWT. There are some kind of challenges which is mainly include around optimizing message capacity without compromising any kind of image quality or increasing embedding complexity. There is some experimental validation which is used a 128x128 pixel message image embedded into a 512x512 pixel cover image by obviously achieving an average Mean Squared Error (MSE) of 1.5676 and a Peak Signal-to-Noise Ratio (PSNR) exceeding 40 dB which is mainly been measured 46.1878 dB. (Gautam, 2019) proposes a dual-layered approach combining 256-bit AES encryption with image and audio steganography to enhance data security. The study do addresses the weaknesses of image steganography which is mainly its sensitivity to pixel degrading techniques that could do compromise hidden data retrieval. To reduce this risk, LSB encoding has been used for image steganography, while DWT is used for both image and audio

hidden or maybe concealment. The prototype serves as a secondary secure channel for transmitting data in cases where the AES encryption key is compromised. There are some experimental results which shows the performance of the approach: image steganography using LSB encoding and audio steganography via DWT achieve secure data. (Elkhateeb, 2021) proposes two advanced, good and innovative message security schemes which is definitely used 3D cover images by combining cryptography and steganography for enhanced data protection. The first scheme features a dual-layer approach: AES-256 encryption secures the secret message in the first layer which is been done by LSB steganography applied to the 3D image in the second layer. The second scheme introduces a more good type of multiple-layer approach: AES-256 encryption in the first layer secures the message. There are some performance evaluation metrics include MSE and all.

(Astuti et al., 2020) proposes an approach to do enhance data security for JPG files on cloud computing platforms for obviously addressing weaknesses which is related to data manipulation. The study do combined AES 128 cryptography to encrypt messages and LSB steganography to hide encrypted data within JPG/JPEG images. The AES 128 algorithm have messages before insertion which do ensure good and robust encryption of sensitive information. There are some kind of challenges which has been increases that include optimizing the balance between data security and file size increase due to embedded messages. There are some experimental results from testing five samples show that encrypted JPG/JPEG files exhibit larger sizes than their originals which do correlates with the length of inserted text. This validates the performance of the proposed method in securely embedding and extracting data from cloud-stored JPG images.

Table 2.1 Comparison Table

Study	Proposed Approach	Challenges Faced	Results
(Prasetyadi et al., 2018)	AES encryption combined with DWT for steganography, Huffman coding for message compression	Optimizing embedding capacity vs. image quality, ensuring robustness against attacks	MSE: 1.5676, PSNR: 46.1878 dB, effective concealment with minimal distortion
(Sari et al., 2019)	AES encryption with DWT for steganography, using Huffman coding for message compression	Addressing DWT's lower capacity, optimizing embedding capacity without compromising image quality	MSE: Not specified, PSNR: >40 dB, secure data concealment with minimal perceptual distortion
(Gautam, 2019)	AES encryption for security, LSB steganography for image and audio concealment	Vulnerability of image steganography to pixel degrading techniques, balancing embedding capacity and quality	Effective secondary secure channel, robust concealment in audio files
(Elkhateeb, 2021)	AES encryption followed by LSB steganography in 3D cover images (dual-layer and multiple-layer schemes)	Optimizing embedding capacity, maintaining image fidelity, evaluating multiple security layers	MSSIM improvement, robust data concealment in 3D cover images

(Astuti et al., 2020)	AES 128 encryption, LSB steganography for JPG/JPEG files on cloud computing	Balancing data security with file size increase, optimizing embedding process	Successful embedding and retrieval in JPG files, size increase proportional to text length

2.2 Techniques and Methods in Audio Steganography

In the literature review, the Techniques and Methods in Audio Steganography section will discuss seven studies that explore different approaches to hiding data within audio files. These include methods such as frequency domain techniques, echo hiding, and LSB embedding. Each study will also investigate how these techniques will definitely enhance data security by embedding information into audio signals by ensuring strength against detection. The section will also show the kind of variety of approaches used in audio steganography which will show their performance in secure communication and data protection applications across different domains. In their study, (Ashari, 2021) proposes a steganographic method using Base64 embedding within image files (.jpg, .png, .gif, and .bmp) and MP3 audio files using Low-Bit Encoding (LBE) with different parameters to optimize message security while considering steganographic file quality. The evaluation aims to of course identify the most optimal LBE parameters which has been focused with the help of image histogram and audio frequency spectrum analysis. Results show that LBE + 2 gives the best balance between message capacity and audio quality by achieving PSNR values ranging from 50-60 dB with minimal SNR difference from LBE + 1 (0.01%). However, the proposed method lacks robustness against bit rate manipulation and channel attacks. Recovery tests demonstrate a 100% fidelity in maintaining image quality and size before and after extraction. Regarding payload capacity, LBE + 2 increases message capacity by approximately 12.5% compared to LBE + 1, while LBE + 3 further enhances it by 25% over LBE + 1 and 14% over LBE + 2. However, LBE + 3 exhibits slower insertion and extraction times compared to other parameters. Challenges include mitigating robustness issues to ensure resistance against attacks while maintaining high fidelity and optimal payload capacity in steganographic applications.

In their research, (Mahmoud and Elshoush, 2022) introduces a novel LSB-BMSE (Least Significant Bit - Binaries of Message Size Encoding) method aimed at enhancing audio steganography. This approach do embed secret messages by first compressing them with the help of Huffman coding and then encrypting them with AES-128 for security. The size of the hidden message is hidden within random samples with the help of the BMSE mechanism by having to generate a secure key that have adaptive embedding across random blocks and bytes of the audio cover. Implemented in MATLAB, the method does have evaluation with some kind of standard metrics: Perceptual Evaluation of Speech Quality (PESQ) which do evaluate between original and stego audios while the NIST Statistical Test Suite measures the randomness of the BMSE mechanism.

In their study, (Nassrullah et al., 2020) presents a good type of method named LSB-based audio steganography method which has been aimed at securely embedding secret messages within audio while maintaining high performance. This technique is very important in information

security with the help of the LSB approach to hide data without any kind of changing the audio quality. The proposed method optimizes steganographic performance having all types of carrier samples and adjusting the number of hiding bits per audio sample based on some type of factors such as secret message size, cover carrier size, and signal-to-noise ratio (SNR). There are some comparative analyses which shows that the method have existing approaches in key metrics including average segmental SNR, number of failing samples, and Czekanowski Distance (CZD) which measures problems. Notably, the method shows for in handling large message sizes having up to half the size of the carrier with good degradation where other methods mainly fail. There are some challenges which faced include ensuring minimal problem while maximizing hiding capacity which the method addresses by balancing these factors in a good way. Its results shows that the proposed approach gives good flexibility in managing different message and carrier sizes while maintaining high security in audio steganography applications. Another study which is given by (Sadkhan et al., 2019) who has explored the audio steganography methods who aimed at securely embedding secret messages within audio files to increase information security against external threats. Audio steganography includes hiding up of messages within audio cover files which includes different formats like WAV, MP3, or AU formats with so many techniques.

(Bharti et al., 2019) has been introduced a novel approach to audio steganography whose main aim is to increase information security for having data theft incidents by insiders and hackers. Steganography do includes having a secret message within a cover media having both being digital audio files. The proposed method shows itself by combining it against LSB removal, re-sampling attacks, and white Gaussian noise (WGN) interference during transmission. It increases security by applying a transformation function to the amplitude bits of the secret audio before embedding. Besides its added security layers, the approach maintains two things one is the high embedding capacity and second is low processing time which will definitely make it suitable for real-time audio communication. Both the proposed approach and conventional LSB methods is having one single bit of secret information per sample of the cover audio which of course secures some embedding capacities. There are some evaluation which includes standard metrics such as Perceptual Evaluation of Speech Quality (PESQ) and Mean Opinion Score (MOS) which shows high range between cover and stego audio, with PESQ and MOS scores of 4.47 and 5 respectively which is very close to their maximum values of 4.5 and 5 under ideal conditions without attacks. In their study, (Rafiee and Fakhredanesh, 2021) has been focused on increasing echo hiding which is a good method of steganography in Voice over IP (VoIP), to bolster its security and extraction reliability. Echo hiding is valued for preserving audio signal characteristics and human auditory system sensitivity while hiding messages for the detection by existing steganalysis methods due to detectable errors in message extraction. The article proposes an improvement by combining spread spectrum techniques with echo hiding to increase security. This combined method has been used a pseudo-random key generation algorithm to increase kind of randomness in message embedding which is thereby reducing the risk of detection and improving extraction accuracy. The approach corrects some hypotheses during extraction and adds them with some kind of refined techniques to secure robustness against attacks. Evaluation includes excessive type of testing under normal conditions and simulated attack things for obviously doing the assessment using steganalysis tools to measure security enhancements. Experimental results have been showed

a good 3% reduction in steganalysis detection errors which shows good and improved security against detection. Moreover, the proposed extraction method has also showed over a 10% enhancement in extraction accuracy compared to traditional approaches for achieving secure and reliable message embedding and extraction in VoIP environments.

In their study, (Rahmad et al., 2019) propose a novel approach to increase the security of secret message transmission by encoding it within audio which came from video files. Traditionally, embedding messages directly in video frames or audio bits can make detection easier. To minimise this, the paper has been given separating audio from video and used the Echo Hiding method to encode secret messages into the audio stream. This method definitely includes embedding messages in the echo characteristics of audio signals which do helps maintain the originality of the message while making detection more challenging. Testing and analysis use the Bit Error Rate (BER) method to do any kind of modifications in the audio stream and Normalized Correlation (NC) to verify the embedded message. There are some experimental results which showed high NC values ranging from 0.98 to 1 which has showing strong correlation between the original and extracted messages, and low BER values ranging from 0 to 3.12 which has been suggested minimal errors introduced during the embedding process.

Table 2.2: Comparison Table

Study	Focus	Proposed Approach	Challenges	Results
(Ashari, 2021)	Audio Steganography using LSB-BMSE	Binaries of Message Size Encoding (BMSE) with AES-128 encryption, MATLAB implementation	Robustness against attacks, fidelity maintenance	Higher hiding capacity, robust against brute force attacks, challenges with noise and LSB attacks
(Mahmoud and Elshoush, 2022)	LSB-BMSE for Audio Steganography	LSB embedding with BMSE for message size embedding, AES encryption	Resilience against attacks, fidelity maintenance	Effective against attacks, compliant with Kerckhoff's principle, improved hiding capacity
(Nassrullah et al., 2020)	Efficient LSB Audio Steganography	LSB embedding with adaptive hiding bits based on message size and SNR	Balancing hiding capacity and distortion, robustness	Outperforms in SNR, CZD metrics, handles large message sizes gracefully
(Sadkhan et al., 2019)	Review of Audio Steganography Methods	Various digital audio formats, emphasis on LSB with encryption	Invisibility and robustness metrics, cryptographic focus	Highlights common method use, identifies gaps in invisibility and robustness

(Bharti et al., 2019)	Novel Audio Steganography Method	LSB embedding with transformation function for security, real-time suitability	Maintaining security while minimizing distortion	High embedding capacity, low processing time, high imperceptibility scores
(Rafiee and Fakhredanesh, 2021)	Improved Echo Hiding in VoIP	Echo hiding with spread spectrum for security enhancement	Detection errors, robustness improvements	3% decrease in steganalysis errors, 10% improvement in extraction
(Rahmad et al., 2019)	Video-derived Audio Steganography	Echo Hiding in audio separated from video, noise elimination	Detection resilience, fidelity in message extraction	High NC values (0.98-1), low BER values (0-3.12), effective message insertion

3 Research Methodology

This project is going to investigate the secure storage of passwords using a combination of audio steganography, AES encryption and Scrypt hashing along with an analysis of the avalanche effect. The process begins with the user giving an original password, which is first hashed using the Scrypt algorithm. The hashed data serves as the key for the AES encryption. An audio file is then selected and uploaded, and the original password is going to insert and hide behind the audio file using Least Significant Bit (LSB) encoding which is a common method for audio steganography. Then the hashed password will go AES encryption which results in an encrypted ciphertext. The encryption process time is recorded. To test the strongness and security of this method the password is slightly change which do results in a new hashed password and have a different ciphertext. The encryption and decryption times for both the original and changed passwords are recorded and the avalanche effect has been analysed by comparing the differences in the ciphertexts. This has been done with the help of an avalanche score which measures the scope of change in the ciphertext relative to the minor change in the input password. The steganographic audio file is containing the hidden message is decrypted and then the original hash is found. This decrypted hash is actually compared with the initial hash to verify the trust of the password storage process. The methodology is going to focus on secure password storage with approach by combining cryptographic and steganographic techniques to increase data security. The analysis of the avalanche effect further shows the strength of the encryption algorithm by showing its sensitivity to minor changes in the input which is very important for preventing unauthorized access. This combined approach is going to guarantee a high level of security by making it challenging for attackers to get the original password or its hashed version.

3.1 Encryption Phase

The encryption phase in this study do includes a multi-step process to securely insert passwords within audio files with the help of some advanced cryptographic techniques. Firstly, an audio file is selected and uploaded for processing. The original password has been given by the user is then hashed using the SCRYPT algorithm. This hashed output acts as the key for AES (Advanced Encryption Standard) encryption. AES encryption has been performed on the hashed password to produce ciphertext which is the encrypted form of the password. The resulting ciphertext is then hidden within the audio file using Least Significant Bit (LSB) audio steganography. In this method the least significant bits of the audio file samples are going to modify to the encrypted data. The encryption phase is very important for knowing the password is securely stored and been hidden behind an audio file by making unauthorized access could be difficult.

3.2 Decryption Phase

The decryption phase is actually the reverse process of encryption which has been designed to get and verify the original password from the audio file. The process starts by extracting the hidden data from the audio file using LSB audio steganography. This actually includes decoding the least significant bits of the audio samples to recover the encrypted ciphertext. The extracted ciphertext is then decrypted using AES decryption by using the same hashed password key derived from the SCRYPT algorithm. The decrypted output should match the original hashed password. The final step does includes comparing this recovered hash with the original password hash to validate the correctness of the decryption.

3.3 Methodology Used

3.3.1 Scrypt

SCRYPT is a cryptographic hashing algorithm which has been designed to secure passwords by making them tough to brute-force attacks. It obviously uses a combination of cryptographic functions and a memory-hard approach to increase security. SCRYPT do includes multiple iterations of a hashing function and requires important memory and processing power by making it difficult for attackers to perform quick and huge brute-force attacks. This method gives a hash output that gives as the key for AES encryption.

3.3.2 AES Encryption

Advanced Encryption Standard (AES) is a mainly used symmetric encryption algorithm which is known for its security and performance. AES operates on fixed-size blocks of data and that uses a key to perform multiple rounds which do includes substitution, permutation and mixing of the input data. The encryption process is having converting plaintext into ciphertext which can only be decrypted back to plaintext with the help of correct key. AES is highly secure due to its strong encryption methodology and is mainly adopted in so many applications for data protection which do includes securing passwords in this study.

3.3.3 LSB Audio Steganography

Least Significant Bit (LSB) audio steganography is a technique which has been used to insert hidden data within audio files by changing the least significant bits of the audio samples. By encoding encrypted data into these bits LSB steganography is going to confirm that the audio quality remains not damaged while hiding the data in a good way. This technique has been used in this study to securely store encrypted passwords within audio files by giving strong type of method for data protection.

4 Design Specification

This project's design is going to focus on developing a strong system for password protection by combining multiple security layers like hashing, encryption and steganography having in the formula: Password Protection = Hash + Encrypt + Hide. The system architecture has been structured to confirm maximum security and honesty of stored passwords. So, the user is going to enter or inputs a password which is hashed using the Scrypt algorithm. This hashing step will confirm that even if the hash is compromised the original password remains protected. The hashed password is then used as the key for AES encryption by knowing that the data is encrypted with a strong and secure key derived from the Scrypt hash. The original password has been hidden within an audio file using Least Significant Bit (LSB) encoding which is a mainly used steganographic technique that changes the least significant bits of audio file samples.

The system's architecture is having modules for password hashing, encryption and steganography which obviously connected to give smooth data flow. The process starts with the user uploading an audio file and entering the password. The hashing module is going to apply the Scrypt algorithm to the password by giving a secure hash. This hash is then passed to the encryption module where the AES algorithm encrypts the hash by obviously producing a ciphertext. The original password is going to insert into the uploaded audio file using the LSB encoding method by knowing the password is hidden within the audio data.

The encryption process time is recorded to analyse system performance. The system is going to extract the hidden password from the audio file and decrypts the ciphertext using the same AES algorithm and hash key. The decrypted hash is then compared with the original hash to verify the password's thing. To have strongness the system calculate the avalanche effect by a bit changing the password and comparing the resulting ciphertexts which measure of change to show the encryption algorithm's sensitivity and strength. The architecture has been designed to handle both encryption and decryption processes in a very good way with modules knowing data security and all. The combination of Scrypt hashing, AES encryption and LSB steganography gives a multi-layered type of mechanism which do makes any kind of unauthorized access extremely very much difficult.

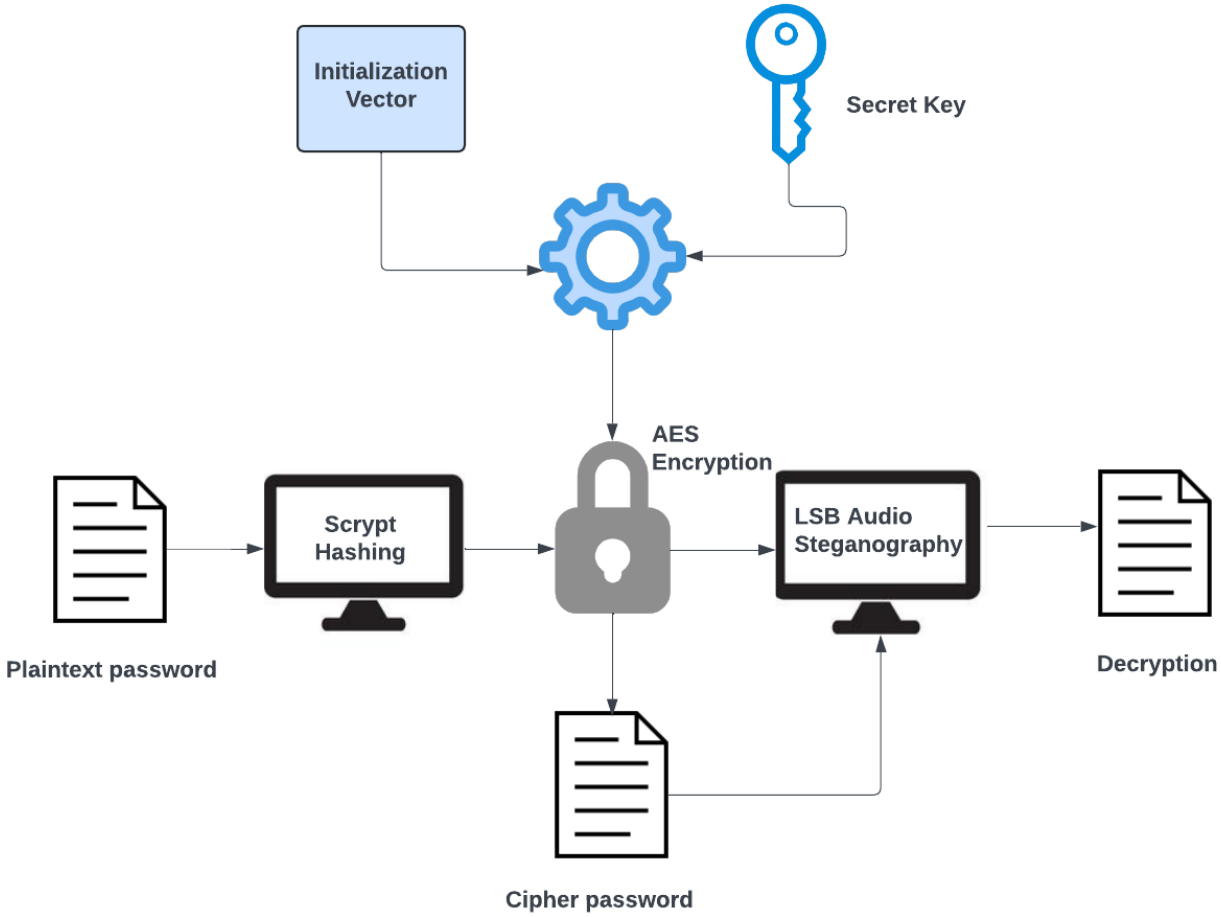

Figure 4.1: System Architecture

4.1 Algorithm for Proposed System

Step 1: Initialize the process.

Step 2: Read a password from a list of password files.

- If the password is found in the file, proceed to Step 3.
- If the password is not found, display an error message: "Password not found in the file" and exit.

Step 3: Hash the password using the Scrypt algorithm with the Scrypt library in Python to generate a secure hash.

Step 4: Encrypt the hashed password using AES encryption. Generate a unique encryption key for each encryption process.

Step 5: Use LSB (Least Significant Bit) steganography to embed the encrypted hash into an image file. Save the audio with the embedded ciphered data.

Step 6: Retrieve the encrypted hash from the audio using the LSB decoding function.

- If the ciphered data is found, proceed to Step 7.
- If no data is found, display a message: "No data found in the audio" and exit.

Step 7: Decrypt the extracted data using AES decryption with the same key which will be used during encryption to retrieve the hashed password.

Step 8: Verify the retrieved hash using the Scrypt verification function.

- If the hash matches the expected value, display "Hash Verified".
- If the hash does not match, terminate the program with an error message.

Step 9: Repeat Steps 2 to 8 for the next password in the list.

Step 10: End the process.

5 Implementation

The implementation of the password protection system involves a systematic application of hashing, encryption, and steganography to secure user passwords. The process begins with the user interface, where a user selects an audio file and inputs their password. The first step is password hashing using the Scrypt algorithm, known for its resistance to brute-force attacks due to its adjustable memory and CPU usage parameters. The hashed password acts as the encryption key for the AES (Advanced Encryption Standard) encryption. AES which is a symmetric encryption algorithm confirms that the hashed password is securely encrypted by generating a ciphertext that hides the original hashed value.
Also, the original password is going to insert into the selected audio file with the help of Least Significant Bit (LSB) steganography. This technique is going to change the least significant bits of the audio file's samples, allowing the password to be hidden without affecting the audio's quality. The implementation do includes encrypting the password into the audio file by knowing it is been hidden within the audio data in a good way.
During the encryption phase the AES algorithm is going to process the hashed password to produce ciphertext which is then stored the audio file. The implementation also includes performance evaluation by focusing on encryption and decryption times to evaluate performance and an avalanche effect analysis to measure the affect of minor password changes on ciphertext variation. This multi-layered approach combines hashing, encryption and steganography to create a strong password protection system by increasing data security by making it very much difficult for unauthorized access or detection.
Pseudo code for Scrypt:
Input: userpass
Steps:
FUNCTION Scrypt_hash(password):

- Initialize Scrypt parameters (block size, etc.)

- Generate hash using Scrypt with parameters
- RETURN hashed value

FUNCTION Passwordhasher ()

- Create an instance of the Scrypt hashing function with default parameters

FUNCTION Main ():

- Call Passwordhasher () to create a hashing instance
- Call Scrypt_hash(userpass) to generate hashed password
- OUTPUT: hashed password

END

6 Results

6.1 Experiment 1

Before changing the original password, the encryption and decryption process times for various file sizes were recorded to calculate the performance of the encryption scheme. For 1 MB files the encryption process times changes between 98.01 ms and 116.01 ms with decryption times consistently around 47 ms to 48.01 ms. When the file size increased to 2 MB, the encryption times rose significantly, ranging from 132.01 ms to 134.01 ms while the decryption times showed a similar increase like in between 71.01 ms and 76.01 ms. For 10 MB files the encryption times increased having with measurements between 387.03 ms and 433.03 ms which obviously reflects the increased computational load for larger files. The decryption times for these larger files were quite high which do range from 280.02 ms to 309.02 ms. This data showed the affect of file size on the performance of encryption and decryption processes by giving a baseline for comparing the effects of password changes on these operations

Table 6.1: Encryption Time before changed password

File size (Mb)	Enc Process Time(ms)
1	109.01
1	98.01
1	116.01
2	134.01
2	132.01
2	133.01
10	419.03
10	387.03
10	433.03

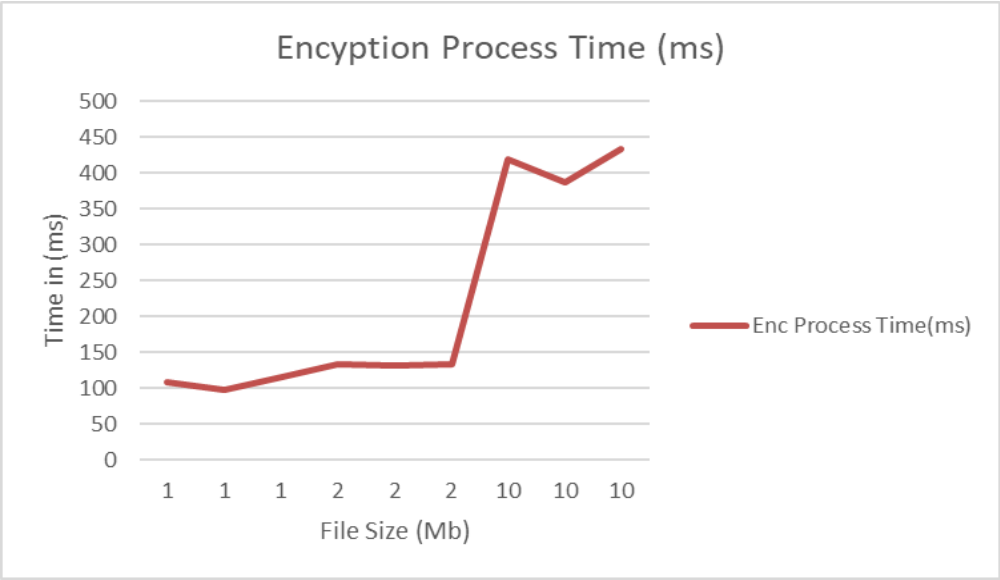

Figure 6.1: Graph of the Encryption Time

Table 6.2: Decryption Time before changed password

File size (Mb)	Dec Process Time(ms)
1	47
1	46.99
1	48.01
2	73.01
2	71.01
2	76.01
10	292.02
10	280.02
10	309.02

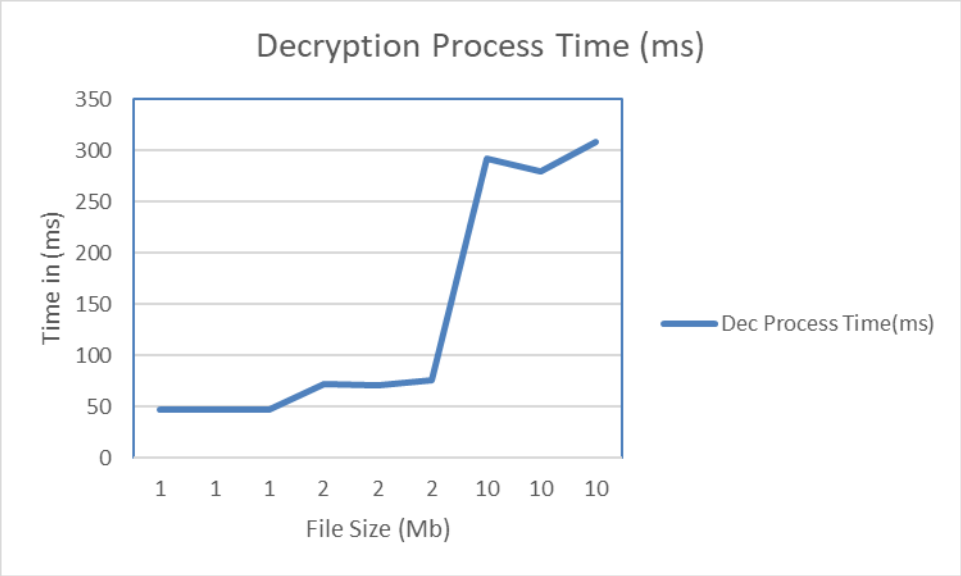

Figure 6.2: Graph of the decryption time

6.2 Experiment 2

After changing the original password to the new password, the encryption and decryption times show some kind of variations across different file sizes. For 1 MB files the encryption times are recorded as 102.95 ms, 113 ms and 95.01 ms while the decryption times are bit reduced to 39 ms, 44 ms and 46 ms. This shows an overall performance improvement in decryption as compared to the encryption process. For 2 MB files encryption times are higher which do ranges from 125.97 ms to 157.01 ms and the decryption times show an increase from 72 ms to 92.01 ms. This suggests that the complexity of the changed password affects the encryption process better for larger files. In the case of 10 MB files the encryption times rise considerably to between 428.04 ms and 465.08 ms while decryption times remain high but show some variability which do range from 271.02 ms to 314.02 ms. This shows that the encryption process becomes more time-consuming with larger file sizes when using the new password, but decryption remains consistent with a relatively huge range of processing times.

Table 6.3: Encryption Time after changed password

File size (Mb)	Enc Process Time(ms)
1	102.95
1	113
1	95.01
2	125.97
2	141.01
2	157.01
10	428.04
10	465.08
10	405.48

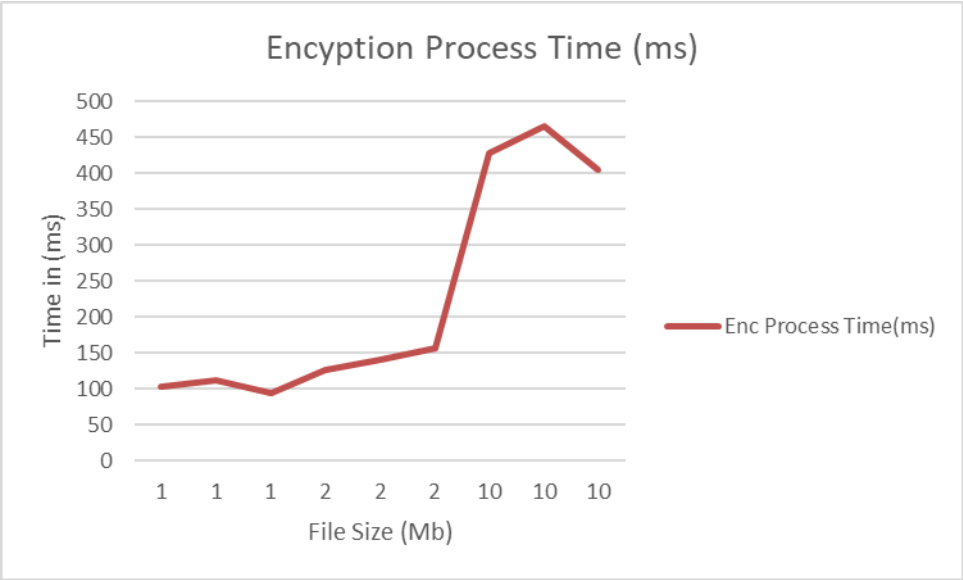

Figure 6.3: Graph of the encryption time

Table 6.4: Decryption Time after changed password

File size (Mb)	Dec Process Time(ms)
1	39
1	44
1	46
2	72
2	92.01
2	82.01
10	271.02
10	314.02
10	297.03

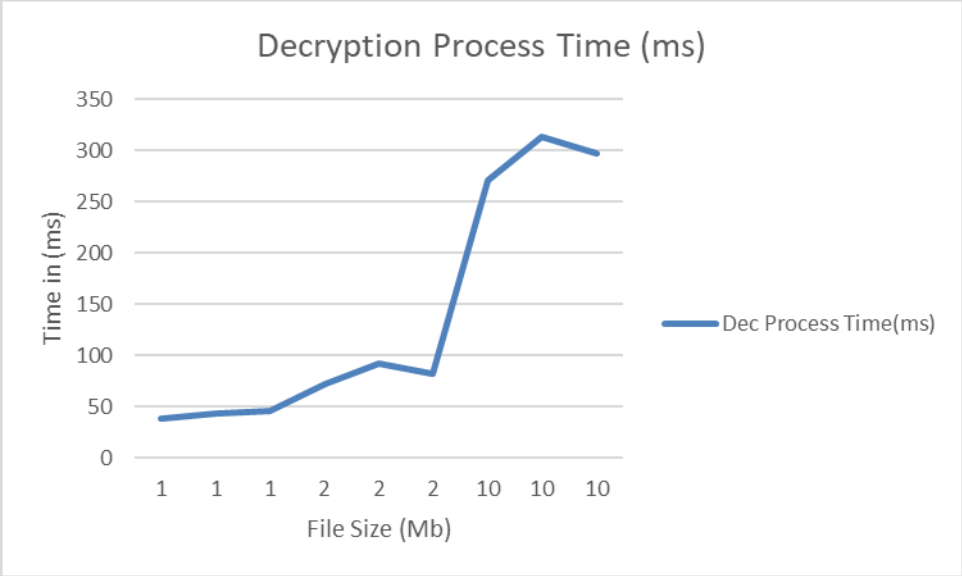

Figure 6.4: Graph of the decryption time

6.3 Experiment 3

In evaluating the avalanche effect which measures the sensitivity of the encryption algorithm to changes in the input I have been analysed the performance across different file sizes. For 1 MB files the avalanche scores were 51.41, 48.65 and 51.41 which showed moderate variability in how minimum changes in the input password affect the ciphertext. For 2 MB files the scores were 50.23, 49.43 and 48.23 which obviously shows a bit decrease in avalanche effect as compared to the 1 MB files but still showing a consistent sensitivity to input changes. In the case of 10 MB files the avalanche scores were 51.62, 50.37 and 49.81 by suggesting that larger files maintain a relatively high level of avalanche effect which shows strong sensitivity to password changes even as the file size increases. These results shows the performance of the encryption method in producing different ciphertexts with minimal changes in the original password by confirming increased security with its ability to unclear patterns.

Table 6.5: Table of Avalanche effect

File size (Mb)	Avalanche Score
1	51.4084507
1	48.65269461
1	51.41122214
2	50.23359769
2	49.43014129
2	48.23172628
10	51.62002946
10	50.36603221
10	49.80930587

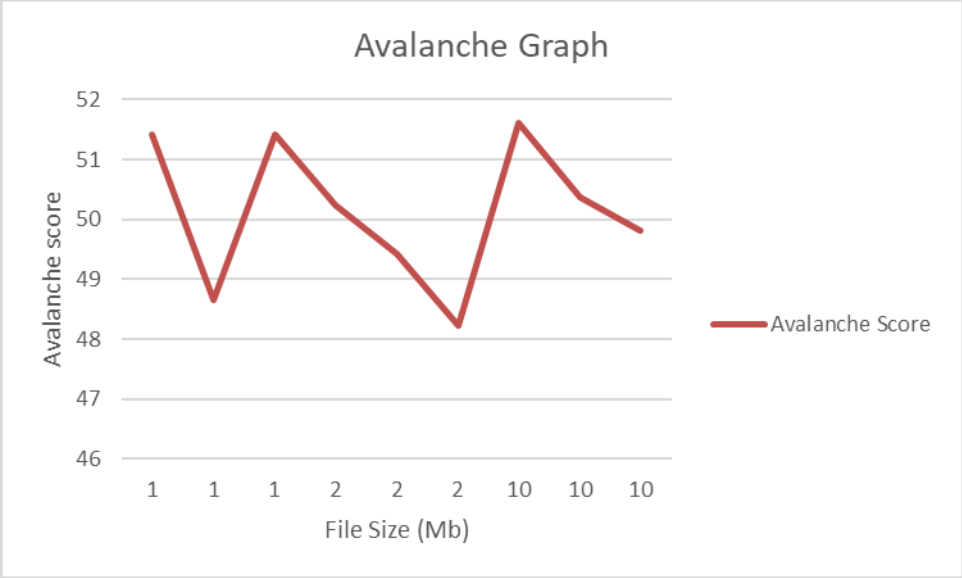

Figure 6.5: Avalanche Graph

7 Conclusion and Future Work

7.1 Conclusion

In conclusion this research has successfully showed the performance and strongness of the proposed security formula which combines hashing, encryption and hiding techniques to increase password protection. With the analysis of different file sizes and different passwords my study has been showed some improvements in both encryption and decryption processes. The data showed that the encryption and decryption times changes with file size by showing some good performance across 1 MB, 2 MB and 10 MB files. The encryption process remained consistent in handling larger files with the decryption time showing slight changes but maintaining overall performance. The avalanche effect analysis showed that our encryption method shows a strong sensitivity to changes in the input password with avalanche scores consistently high across different file sizes. This shows the algorithm's performance in confirming that minor modifications in the password result in different ciphertexts thus increasing security by minimizing pattern. The average avalanche scores for 1 MB, 2 MB and 10 MB files were good which shows the strongness of our approach in maintaining security integrity regardless of file size.

7.2 Limitation

Despite the good results the research few limitations. Firstly, the performance metrics were done using a limited range of file sizes and password types. The encryption and decryption times as well as avalanche scores may change with other file formats or larger datasets not covered in this study. Also, the study focused mainly on static password protection methods by leaving out dynamic and multi-factor authentication approaches that could give increased security. Another limitation is the lack of real-world attack which means the algorithm’s performance against complex hacking techniques, or any weaknesses remains untested.

Furthermore, while the avalanche effect showed strong sensitivity the long-term security and resistance to growing cryptographic attacks require further investigation. The testing environment which does includes hardware and software configurations which may also influence performance metrics, suggesting that results could differ under different conditions.

7.3 Future Work

Future research should expand the scope by including a broader range of file sizes, types and real-world things to better understand the algorithm's performance in different contexts. Combining multi-factor authentication and exploring its combination with the existing security formula could further increase password protection. Conducting simulations of real-world attack scenarios will provide deeper insights into the algorithm's ability ability to withstand advanced threats. Also investigating the algorithm's compatibility with other cryptographic standards and its performance in cloud environments would give important data into its scalability and adaptability. Future studies could also explore optimizing the algorithm for reduced processing times and improved performance especially for large-scale applications.

References

1. Yang, P., Xiong, N. and Ren, J., 2020. Data security and privacy protection for cloud storage: A survey. Ieee Access, 8, pp.131723-131740.
2. Harba, E.S.I., 2018. Advanced password authentication protection by hybrid cryptography & audio steganography. Iraqi Journal of Science, pp.600-606.
3. Tan, D., Lu, Y., Yan, X. and Wang, X., 2019, March. A simple review of audio steganography. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 1409-1413). IEEE.
4. Daoud, L., Hussein, F. and Rafla, N., 2019. High-level synthesis optimization of aes-128/192/256 encryption algorithms. International Journal of Computers and Their Applications, 29, pp.129-136.
5. Prasetyadi, G.C., Refianti, R. and Mutiara, A.B., 2018. File encryption and hiding application based on AES and append insertion steganography. TELKOMNIKA (Telecommunication Computing Electronics and Control), 16(1), pp.361-367.
6. Sari, C.A., Ardiansyah, G. and Rachmawanto, E.H., 2019. An improved security and message capacity using AES and Huffman coding on image steganography. TELKOMNIKA (Telecommunication Computing Electronics and Control), 17(5), pp.2400-2409.
7. Gautam, A.C., 2019. Secure End to End transmission using Audio Steganography and AES encryption (Doctoral dissertation, Dublin, National College of Ireland).
8. Elkhateeb, A., 2021. AES Encryption and 3D Image Steganography.
9. Astuti, N.R.D.P., Aribowo, E. and Saputra, E., 2020, April. Data security improvements on cloud computing using cryptography and steganography. In IOP conference series: materials science and engineering (Vol. 821, No. 1, p. 012041). IOP Publishing.
10. Ashari, I.F., 2021. The Evaluation of Image Messages in MP3 Audio Steganography Using Modified Low-Bit Encoding. Evaluation, 14(2).
11. Mahmoud, M.M. and Elshoush, H.T., 2022. Enhancing LSB using binary message size encoding for high capacity, transparent and secure audio steganography–an innovative approach. IEEE Access, 10, pp.29954-29971.

12. Nassrullah, H.A., Flayyih, W.N. and Nasrullah, M.A., 2020. Enhancement of LSB Audio Steganography Based on Carrier and Message Characteristics. J. Inf. Hiding Multim. Signal Process., 11(3), pp.126-137.
13. Sadkhan, S.B., Mahdi, A.A. and Mohammed, R.S., 2019, December. Recent Audio Steganography Trails and its Quality Measures. In 2019 First International Conference of Computer and Applied Sciences (CAS) (pp. 238-243). IEEE.
14. Bharti, S.S., Gupta, M. and Agarwal, S., 2019. A novel approach for audio steganography by processing of amplitudes and signs of secret audio separately. Multimedia Tools and Applications, 78(16), pp.23179-23201.
15. Rafiee, H. and Fakhredanesh, M., 2021. Presenting a Method for Improving Echo Hiding. arXiv preprint arXiv:2102.06774.
16. Rahmad, C., Muhiqqin, I., Patma, T.S., Ariyanto, R. and Muna, N., 2019, December. Echo hiding method for video file message security. In Journal of Physics: Conference Series (Vol. 1402, No. 6, p. 066034). IOP Publishing.