An Alternative LSB Based Video Steganography

Kim Scicluna

Supervisor: Mr Alvin Cardona

June, 2018

Submitted to the Institute of Information and Communication Technology in partial fulfillment of the requirements for the degree of B.Sc. (Hons.) Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of Mr Alvin Cardona.

Kim Scicluna

October 30, 2018

Copyright Statement

In submitting this dissertation to the MCAST Institute of Information and Communication Technology I understand that I am giving permission for it to be made available for use in accordance with the regulations of MCAST and the College Library.

Kim Scicluna

Desiree‘,
Karmelitani Street,
Kercem KCM 1240

October 30, 2018

Dedication

This dissertation is dedicated to my family and friends for their endless support given throughout this journey.

Acknowledgements

I would like to thank my supervisor, Mr. Alvin Cardona who guided and inspired me with his knowledge and expertise in this area of study.
I also would like to thank my brother Mr. Wayne Scicluna for helping with technical issues encountered during this research.

Abstract

Encryption is a commonly used method of security that can help prevent access to the original contents of a message. Nonetheless, the unauthorised presence of communication between two or more parties can still prove to be a breach of privacy. Such a problem can be overcome with steganography, this being security by obscurity, which hides the presence of a communication channel. This might not prove very secure at first glance. However, as technology advances, new ways to implement steganography become possible. Steganography is nothing new, but digital steganography is an area of study which is still being explored. This research focuses on the feasibility of an LSB based video steganography technique which uses both the audio and frames as a cover for hiding any secret message.

This proposed technique takes advantage of the numerous number of frames and audio samples in a video, instead of utilising just the audio or the frames of the whole video. This gives the possibility of significantly increasing the capacity of secret data that can be embedded in the video. This was tackled by developing two sub-algorithms for audio and image steganography separately, which are based and built upon existing research. Both sub-algorithms were then adapted for a video steganography algorithm. The algorithms were developed with scalability in mind, meaning that a theoretical infinite amount of secret data can be embedded in a video, given that the video has enough capacity for the secret data. However, this is limited by the technology present today, for example, data type value ranges. Testing was done by a divide and conquer approach, first testing both sub-algorithms separately and then testing them together. The quality metric used in this study is the Peak Signal-To-Noise Ratio. The sub-algorithms were tested with different types of media and the best possible conditions to reduce detectability were explored, that is, hiding data in a grainy image. When possible, the sub-algorithms were also compared with algorithms from existing literature and existing tools.

Results gathered show ideal image and audio conditions for the sub-algorithms which were discussed further in detail in the chapter 5. Comparison between existing works was also done and the developed sub-algorithms and video steganography algorithm showed an overall improvement. To address the research question, there was an improvement of a video steganography approach utilising both image and audio streams over a video steganography approach which utilises only the audio or frames. An ideal secret data distribution ratio between the frames and the audio was also narrowed down. Ideal conditions of cover media were studied. In view of the findings and analysis presented in this study, further research may be carried out in other areas, including robustness against compression, more in-depth research of pre-processing the cover media, by for example including artificial noise, to make evidence of steganography less detectable, and testing against steganalysis attacks.

List of Acronyms

bpB - Bits Per Byte
dB - Decibels
DCT - Discrete Cosine Transform
DWT - Discrete Wavelet Transform
Hz - Hertz
kHz - Kilohertz (1000 Hz)
LSB - Least Significant Bit
MSE - Mean Square Error
NKS - No Key Steganography
PKS - Public Key Steganography
PRNG - Pseudorandom Number Generator
PSNR - Peak Signal-To-Noise Ratio
RGB - Red, Green, Blue
SD - Spatial Domain
SKS - Secret Key Steganography
TD - Transform Domain

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

No communication could be considered private on the internet. Privacy breaches happen every year, including that of governments exploiting wiretapping laws among other privacy invasion incidents exposed by whistleblowers namely Edward Snowden who said “I don’t want to live in a world where everything that I say, everything I do, everyone I talk to, every expression of creativity or love or friendship is recorded.” Security against these problems often exists in the form of cryptography: having the data on the internet encrypted, only to be decrypted by those who are authorised to view the data. However, cryptography does not hide the fact that a communication channel exists between two or more parties. This is where steganography comes in. With digital steganography, covert communication can be established between two or more parties by using innocent looking digital cover media, which secretly hold data embedded inside them.

This data can then be extracted only by those who are expecting data to be hidden inside the cover media. This process is known as 'security by obscurity'. Apart from real-world applications such as digital watermarking, steganography can also be seen as a theme in famous works of fiction such like Prison Break TV series and the Da Vinci Code.

1.2 Scope

At present, there is a lack of video steganography algorithms which take advantage of the characteristics of a video, including the numerous amounts of frames in a video and the audio stream of a video. Existing research on video steganography usually deals with either one of the two. The study's research hypothesis is: Is it possible to use all the frames and audio of a video for video steganography? Since a video is made up of images and audio, the areas of both image and audio steganography were explored.

1.3 Practical Relevance of Research

An improved video steganography algorithm offers a better and more efficient approach to the area of steganography. Although this research focuses on video steganography, extensive research on image and audio steganography has to be made individually. It is imperative that in-depth research on audio and image steganography will be carried out in video steganography. Developing optimal sub-algorithms for audio and image steganography is a goal for the research. This is achieved by improving on and adapting existing audio and image steganography algorithms to be used in a video. Using an LSB approach to the steganography helped keep the problem simple, while also being a contribution to the field of LSB based steganography.

1.4 Research Approach to be Taken

Before answering the research questions set, key literature was researched. The literature sources included valuable information regarding general useful steganography practices and existing implementations, theoretical or otherwise, by other researchers. The fields of audio and image steganography were researched to offer a solid foundation for the image and audio steganography sub-algorithms. Video steganography techniques were also researched. Existing steganography tools were obtained and utilised to compare and contrast results at the later stages.

The following chapter presents the literature review, followed by the research methodology. The developed sub-algorithms and video steganography algorithm were described in detail, with references to existing literature that aided in the implementation of the said algorithms. This stage is crucial as it sets down a plan that will guide this research to a sound conclusion. The gathering of results was done based on the approach chosen in the research methodology. Further discussion of the results was conducted to help analyse the gathered data to come to a conclusion for the hypothesis formulated.

Chapter 2

Literature Review

Steganography is the art of information hiding, a variant of cryptography, both of which can be used for beneficial or malicious intents. This research focused on embedding and distributing the secret data in both frames and audio of a video with the aim of creating a more robust steganography technique.

This raised the question: Could it be possible to create a robust video steganography algorithm, based on existing implementations that combine other techniques, i.e. audio and image steganography? Kour J. and Verma D. (2014) define steganography as being "the study of invisible communication". As Abdulzahra H. et al. (2014) state, steganography prevents any discovery of a communication channel between two or more parties. Steganography utilises a medium, which could be anything from a hand-written letter, a video, and any digital media. In this case the cover medium will be a video.

2.1 The Combined Use Of Cryptography And Steganography

The combined use of cryptography and steganography can result in a very secure communications means. By encrypting the data before the embedding process of the proposed video steganography technique, another layer of protection will be created against data extraction. As Siper A. et al. (2005) stated, Cryptography is the art of protecting information by transforming it into an unreadable format, called cypher text. This implies that encrypted information can be stolen. Aung and Naing (2014) proposed an image steganography technique that utilises both steganography and AES cryptography. Aung and Naing's (2014) implementation uses symmetric encryption. Symmetric cryptography uses a secret key to encrypt and decrypt the data, and this secret key is known by all the parties of the communication channel. Chhillar R. S. (2015) also explored the use of combining steganography and cryptography, where they state that the method is a good practice in steganography as it adds a layer of security against unauthorised access. However, they used asymmetric cryptography and LSB steganography. Asymmetric cryptography utilises public and private keys, where the public key can be shared with anyone, and the private key is kept secret by one user. Cryptography prevents unauthorised discovery of the contents of a communication between two or more parties. Cryptography can be implemented to work hand-in-hand with the proposed video steganography technique that will be developed. Since the techniques mentioned above cater only to image steganography, Abikoye

O. C. et al.'s (2012) method proves that such a method can also be applied to audio steganography; hence, such a technique can also be incorporated in the video steganography technique to be proposed in this research.

2.2 Image Steganography

Image steganography will make up half of the proposed technique, the other half being audio steganography. Singh A. and Singh S. J.'s (2014) overview of image steganography provides an insight into different image steganography techniques, all of which can be utilised for video steganography.

2.2.1 LSB Based Image Steganography

LSB insertion, which is a SD technique, is the most conventional and straightforward method for embedding data in a cover file. This method affects the SD which means that replaces the LSB of a pixel and replaces it with one bit of the secret data. As Deshmukh P. R and Rahangdale B. (2014) state, LSB steganography with videos offers large capacity due to the number of frames available, starting from about 24 frames per second. LSB steganography can still cause visual distortion with a big enough payload; however, some methods can decrease this distortion density caused. As shown in Figure 2.1 by Fraile R. (2015), and as described by Swain G. and Lenka S. K. (2014), LSB steganography in images is done by hiding the bits in the least significant bit of the RGB channels of an image.

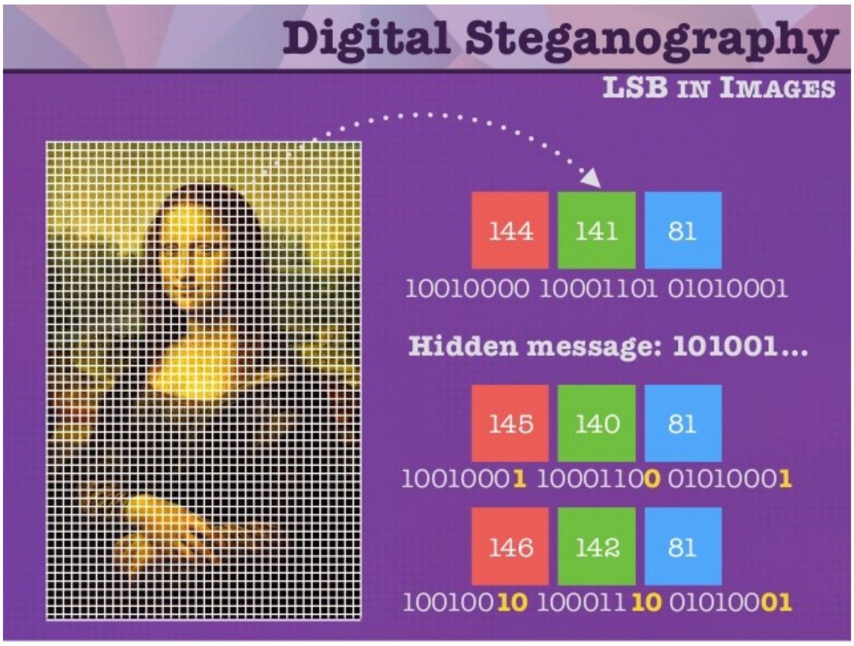

Figure 2.1: Visualized LSB Steganography in an Image (Fraile R., 2018)

2.2.2 Using a PRNG For LSB Based Image Steganography

An improved alteration to LSB steganography is described by Mohidul Islam S. M. et al. (2017). The authors explain that the blue colour is the least visible colour by the human eye; hence, it is said that the blue channel of an RGB image is best used for data embedding. To further protect against unauthorised extraction of the embedded data, Emam M. M. et al. (2016) propose an LSB method based on random pixel selection using a PRNG. This proposed technique works as a SKS technique, as opposed to a PKS technique where a public key is used for embedding, and a separate private key is used for extracting the secret data (Mishra M. et al., 2012). SKS is where the secret key, known only by those part of the covert communication channel, is used to generate a pseudorandom sequence that will be used to choose the pixels randomly for the embedding and extracting of the secret data. A PRNG provides the possibility for the distribution of data throughout the frames and audio of a video in the proposed technique. Sahu U. and Mitra S.'s (2011) video steganography method utilises LSB insertion with random frame selection and pixel position selection. Their methodology also supports Islam S. M. et al.'s (2017) suggestion as data is embedded mostly in the blue channel of the pixels, making it less perceivable by the human eye.

Manjula G. R. and Danti A. (2015) also propose a hash-based 2-3-3 LSB based algorithm, meaning that 8 bits are stored in a pixel, two being in red, 3 in green and 3 in blue. The hash function $m = k \ \% \ l$ was used, $m$being the bit position in an RGB pixel, k is the bit position of the secret eight bits to be embedded, and l being the number of LSB bits being utilised for embedding. Such a technique provided better PSNR results compared to the original algorithm. A hashing algorithm could prove useful for LSB selection and maybe even be adapted for a unique frame selection and pixel selection, instead of or in conjunction with a PRNG as proposed by Emam M. M. et al. (2016).

2.2.3 Multi-Cover Image Steganography

Thaneshawri M. and Arumugam N. (2012) propose a technique that utilises a variable amount of colour images as cover media for a single secret message. The embedding process, as described by the authors, requires all the amount of cover images for the data to be extracted. Such a technique provides a single high-quality cover image with the embedded data, high capacity and security. This is a form of SKS, as the N images are used as a key of a sort for the N+1th image. This is achieved using a number of Exclusive-OR operations. However, Thaneshawri M. and Arumugam N. (2012) show that such a technique decreases the PSNR of the cover image. Similarly, Baek J. et al. (2010) implement a similar technique to grayscale images. Steganography using a grayscale colour space has lower capacity. However, it offers more simplicity. These techniques make LSB insertion more robust against steganalysis attacks. Such a technique can be adapted to video steganography since a video is made up of several images. However, most video steganography techniques, like the proposed technique of Kakde Y. et al. (2015), do not integrate a multiple image steganography technique like the one mentioned by Thaneshawri M. and Arumugam N. (2012). By utilising aspects of both algorithms, there may be a possible increase in capacity and robustness.

2.3 Audio Steganography

Audio steganography also presents a good number of possible implementation to be utilised with video steganography. While some image steganography techniques can be applied to audio steganography, some are specific to the latter. The journal of Jayaram P. et al. (2011) goes through different audio steganography techniques. LSB Coding is done like image steganography, where the last bit of an audio sample is changed to represent a bit of the secret message. This is not very robust in audio steganography as it can produce distorted sound. The more bits in an audio sample, the smaller the distortion caused. However, with some enhancement, such a technique can become robust and efficient.

2.3.1 LSB Based Audio Steganography

Atoum M. S. et al. (2017) proposed advanced LSB technique that increased security. The authors explain that LSB is weak against intentional and unintentional attempts to extract the secret data. The proposed algorithm is an NKS, meaning no key is used for embedding and extracting the secret payload. Pixels are selected randomly by checking the MSB and LSB of a byte. If both have the same value, then the four secret bits are embedded in location two to five for each byte. Such a technique had higher PSNR values than a standard LSB technique. Bahl J. and Ramakishore R. (2015) propose a method using LSB and parity coding. Nosrati M. et al. (2012) write that parity coding separates the signal into regions of samples. The parity bit of the region is compared with the bit to be embedded. If they match, no change is done; otherwise the LSB is flipped. Such a method provides distortion as it is less obtrusive on the signal integrity. Figure 2.2 by Mishra S. et al. (2018) illustrates the steps taken in parity coding in three different regions.

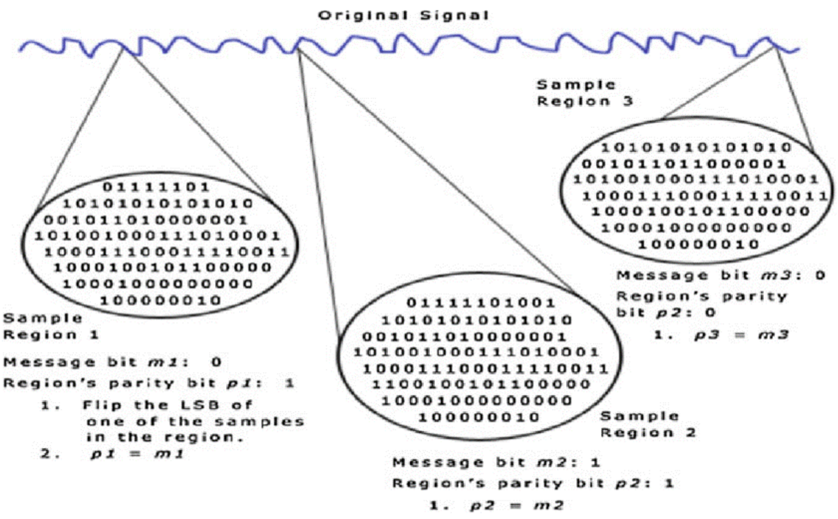

Figure 2.2: Parity Coding (Mishra S. et al., 2018)

The techniques mentioned above are all SD steganography techniques. TD steganography is where cover media, mainly being an image or audio, is transformed utilising a transform function such as DCT or DWT, and after that, the data is embedded. A prime example of this technique is the DWT steganography. This technique is more complicated but offers more robustness, compared with SD steganography. However, it is very secure and produces lossless stego-media (Kaur N. & Behal S., 2014).

2.4 Video Steganography

The areas mentioned, audio and image steganography, was used in this study. However, other existing video steganography techniques needed to be taken into consideration when it came to analysing the quality of the proposed video steganography technique with respect to the cover and payload sizes.

2.4.1 An Existing Video Steganography Tool

The problem with video steganography techniques is that they are most likely to stay stuck as a theoretical implementation, meaning that implementations of these are rarely found. As Hernandez-Castro J. and Sloan T. (2015) explain in their paper regarding forensic analysis of video steganography tools, this results in the tools available to the public being outdated and do not reflect the latest technological advances in this field. As Hernandez-Castro J. and Sloan T. (2015) mention a video steganography tool named MSU StegoVideo (Vatolin D., 2007), which was developed by a team of researchers. They managed to develop one of the few video steganography tools available that do decent video steganography, although, not being secure, and not a lot of information was given regarding this application by the authors. As this technique uses compression, the secret message embedded may not be retrieved with 100% integrity. Wu J. et al. (2010) explain that Vatolin D.'s (2007) implementation is an LSB based algorithm that also works with compression, at the cost of having partial message loss upon extraction.

2.5 Steganalysis

For one to develop a robust steganography technique, there needs to be a basic understanding of steganalysis. As Krenn R. (2004) argues in his paper, the need of better and evolving steganography techniques increases as steganalysis attacks become more efficient at detecting and even extracting the secret data from the cover media used. Steganalysis is the opposite of steganography. The need of steganalysis is crucial because steganography can be used for criminal acts. This underlines the need for forensic teams in law enforcement to develop new ways to detect hidden data embedded employing different steganography techniques.

2.5.1 Steganalysis Methods

Different steganalysis techniques exist for different types of steganography techniques, each catering for different algorithms and types of cover media used in the steganography process. Kaur M. and Kaur G. (2014) describe the different types of steganalysis techniques. Steganalysis techniques can be broadly classified into two separate categories, being signature steganalysis and statistical steganalysis. Statistical steganalysis utilises a file's statistical properties for signs of steganography. However, these two categories can be further divided into different methods. Specific statistical steganalysis methods target specific steganography techniques, such as Least Significant Bit Insertion, only by using, for example, Chi-square statistical attacks. These narrow down the detection rate because a lot of other steganography techniques exist. Universal statistical steganalysis target most of the known types of steganography techniques. It requires no previous knowledge of any embedding algorithms. This technique utilises neural networks and clustering algorithms to learn from existing steganography cover media. The two mentioned fall under the statistical steganalysis category. Signature steganalysis tackles steganography methods that manipulate the cover media in a way that it remains imperceptible, such as embedding data in palette tables.

2.6 Common Quality Metrics Used

The quality metrics used needed to be similar to existing research for ease of comparison. Dave M. and Somani H. (2016) wrote a survey on video steganography that also mentioned a list of quality features of a steganography technique. These measures are the following: Imperceptibility, Payload Size, and Robustness against statistical attacks, Security, Computational Cost and Perceptual Quality. Kharrazi M. et al. (2006) also give some parameters that are usually used to rate imperceptibility. First, the number of modifications done to the cover media show how detectable the stego-video file is. The fewer the changes, the less detectable. Another parameter is the MSE, which is an error metric calculated with the original cover media and the stego-video. Another relevant parameter to video steganography is the structural similarity measure. This provides a quantification of similarity between the cover and stego-media. The final and most commonly used metric that is derived from MSE is the PSNR.

2.7 Conclusion

The gap found in the research is the lack of use of both audio and image manipulation in video steganography. The discussed LSB based image steganography techniques, combined with LSB audio steganography techniques, would provide the basis of the methodology chosen. Furthermore, the discussion of quality metrics as seen throughout all the papers and the last two mentioned authors discussed provide a basis to have good quality metrics used to be analysed and compared with existing data in the discussion of the research.

Chapter 3

Research Methodology

3.1 Introduction

The proposed research's algorithm contains a combination of image steganography and audio steganography to develop a secret key, LSB based video steganography technique. The work was divided into two parts, namely, that of embedding and extracting the secret message. For added security, which is explained later on in this chapter, encryption and decryption was done on the secret message, prior and after embedding and extracting. The reason for the use of LSB embedding is because a basic LSB steganography algorithm is known to be more vulnerable than other techniques such as DCT and DWT mentioned by Kaur N. & Behal S., 2014. Hence, this research contributes further to LSB steganography in an attempt to develop a sound and robust LSB steganography algorithm, while maintaining its reputation as a relatively simple implementation as compared with other techniques in different domains.

3.2 Encryption And Decryption

The encryption of the secret message is a vital process to increase the security of the proposed video steganography algorithm. Chhillar R. S. (2015) states that encrypting the message before steganography is good practice as it is an extra security feature, which protects against the possibility of unauthorised data extraction from the cover video by steganalysis attacks. It serves as last resort, in the case where a steganalysis attack to extract the hidden data is successful. The type of cryptography used is symmetric cryptography, specifically, AES cryptography. Because AES is a symmetric algorithm, it is fast, and most importantly, as argued by Mohidul Islam S. M. et al. (2017), AES is a lossless encryption algorithm. This means that after decryption, the secret message is like it was before, unlike lossy encryption where certain information is permanently removed.

3.3 Pseudo-Random Number Generator

The proposed method functions using a secret key, where everyone in the covert communication channel has to know the shared secret key for where its value is used as a seed for a PRNG, and a secret key for the AES algorithm mentioned previously. This process is illustrated in Figure 3.1.

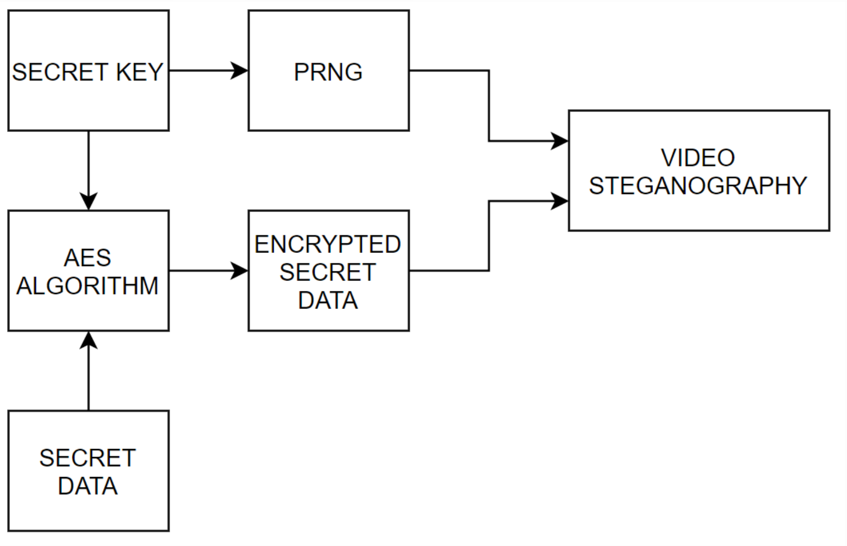

Figure 3.1: The use of the secret key

The PRNG used for the distribution of the secret data equally and randomly throughout the frames and audio samples by generating a sequence of pseudo-random numbers that indicate the pixels and samples which will contain the secret data, inspired by the research of Emam M. M. et al. (2016) and Sahu U. and Mitra S.'s (2011). The reason why SKS was used over NKS and PKS is that a SKS technique provides the needed key to be used for the encryption mentioned in the previous part and the PRNG sequence generated that effectively jumbles up and makes the data harder to reconstruct. If the wrong secret key is given, the algorithm extracts data that will not make any sense to the reader when reconstructed. The process of the PRNG is showcased in Figure 3.2 where eight pixels contain one bit of a byte of secret data each. It illustrated the different message extracted given a completely different PRNG sequence.

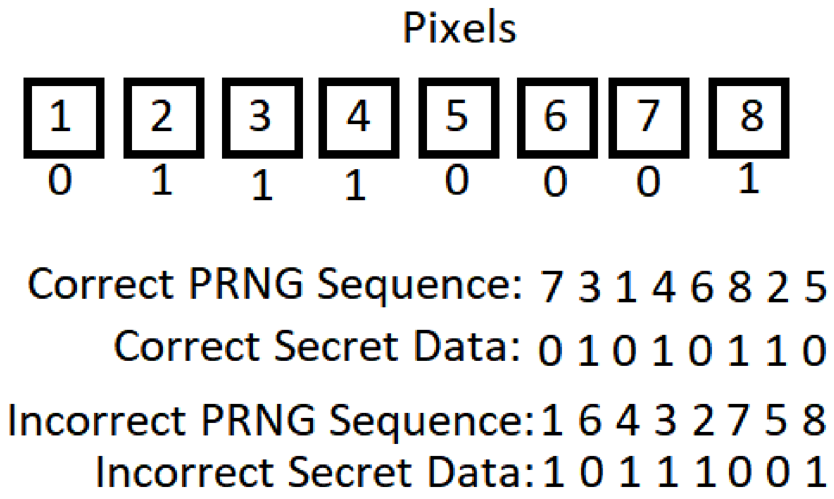
Pixels
1 2 3 4 5 6 7 8
0 1 1 1 0 0 0 1
Correct PRNG Sequence: 7 3 1 4 6 8 2 5
Correct Secret Data: 0 1 0 1 0 1 1 0
Incorrect PRNG Sequence: 1 6 4 3 2 7 5 8
Incorrect Secret Data: 1 0 1 1 1 0 0 1

Figure 3.2: An example of the use of a PRNG sequence

3.4 The Image Steganography Sub-Algorithm

The sub-algorithm used on the frames of the video was based on random pixel selection and random frame selection, similar to that proposed by Emam M. M. et al. (2016) and Sahu U. and Mitra S.'s (2011) respectively. The central concept behind this decision is the difficulty presented to reconstruct the secret message without knowledge or access to the secret key, because of the use of a PRNG which generates a sequence of numbers that will be used to randomly distribute the secret bits throughout the pixels of each frame. This is an improvement over the implementation of Deshmukh P. R. and Rahangdale B. (2014), where they only used one frame from the whole video to hide the data.

3.4.1 The Steps Of The Sub-Algorithm

Embedding Data

The first step of the embedding process is to store the length of the secret message to be embedded in the frames, with this value being 16-bits and will be stored in the first frame. With this value, the bytes per frame can be calculated, and it is used to generate an x amount of non-repeating, random points for embedding. The embedding process took Manjula G. R. and Danti A.'s (2015) research in consideration of using a 2-3-3 LSB algorithm for each frame. The embedding rate of the proposed algorithm was one byte per pixel, embedding two bits in the red channel, three bits in the green and three bits in the blue. This process is illustrated in Figure 3.3 where the values of the hash function are: m=4,2 for Red, m=1,4,2 for Green and m = 1,4,2 for Blue.

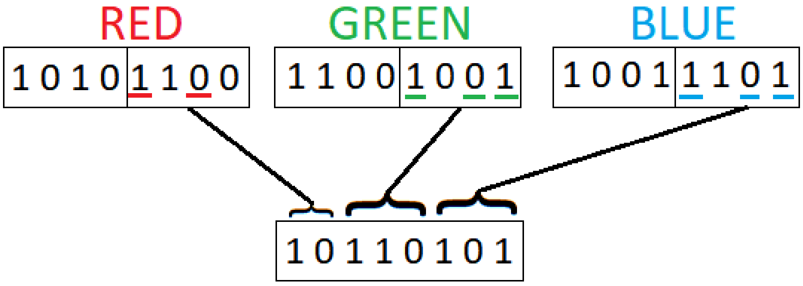

Figure 3.3: 2-3-3 LSB embedding

The use of the hash function indicates the position where each bit will be embedded. Manjula G. R. and Danti A. (2015) prove that such a technique provided better PSNR values, making it less detectable. This process is shown in Figure 3.4.

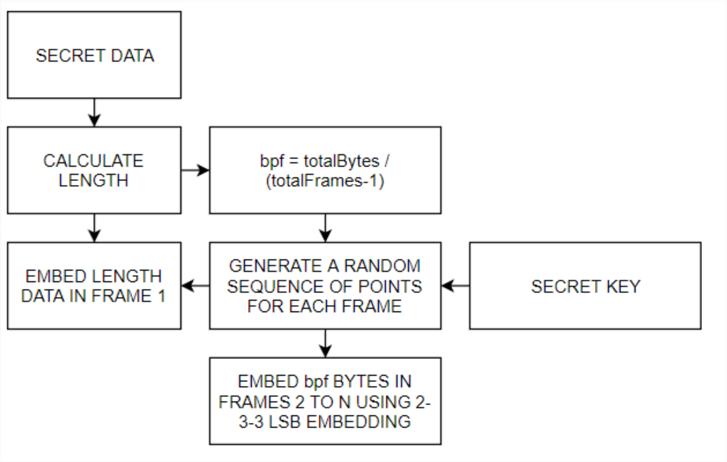
SECRET DATA
CALCULATE LENGTH
bpf = totalBytes / (totalFrames-1)
EMBED LENGTH DATA IN FRAME 1
GENERATE A RANDOM SEQUENCE OF POINTS FOR EACH FRAME
SECRET KEY
EMBED bpf BYTES IN FRAMES 2 TO N USING 2-3-3 LSB EMBEDDING

Figure 3.4: The embedding process

Extracting Data

The extraction process requires the same secret key used in the embedding and the cover video. The first step would be to extract the length of the secret data from the first frame. Since the sub-algorithm embeds 16 bits of data, the PRNG, using the same seed will generate the same four points where the data was embedded. The extracted value will indicate how many bytes are embedded in each remaining frame. Hence an x amount of points will be randomly generated for each frame to help rebuild the secret data. This whole process is illustrated in Figure 3.5.

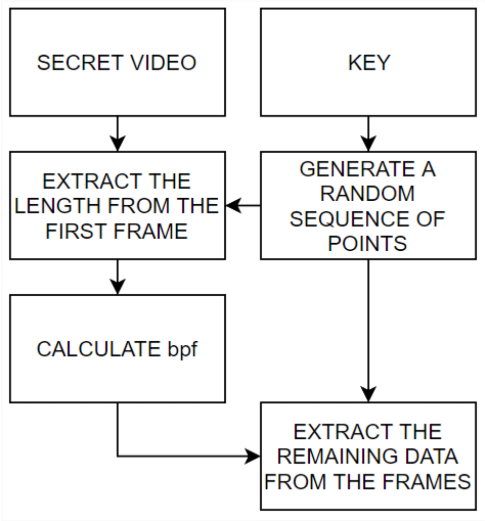
SECRET VIDEO
KEY
EXTRACT THE LENGTH FROM THE FIRST FRAME
GENERATE A RANDOM SEQUENCE OF POINTS
CALCULATE bpf
EXTRACT THE REMAINING DATA FROM THE FRAMES

Figure 3.5: The extracting process

3.5 The Audio Steganography Sub-Algorithm

The audio steganography technique used on the audio stream was an LSB based approach that utilises sample regions for data hiding. This process utilises a PRNG as a method to randomly distribute the data throughout the regions. This was done in support of Atoum M. S. et al. (2017)'s research that explains LSB and its weakness against intentional and unintentional attempts at extracting the secret data. The method chosen will protect against the mentioned possibility of exploitation by making it much harder to reassemble the secret message without the random sequence generated with the given key.

3.5.1 The Steps of the Sub-Algorithm

Aspects from Bahl J. and Ramakishore R.'s (2015) implementation were used. The idea of taking sample regions was adopted. However, the regions taken would have one byte of data, instead of one bit, embedded in them. A region will usually include thousands of samples, more than enough for the data to fit. The eight bits will be inserted pseudo-randomly in LSB of that specific region. The whole process of distributing the data throughout regions in a random order was implemented to find a middle ground between Parity Coding and LSB embedding to increase the robustness of simple LSB embedding while increasing the capacity of a Parity Coding technique while trying to maintain the quality of the audio stream. The mentioned steps are illustrated in Figure 3.6.

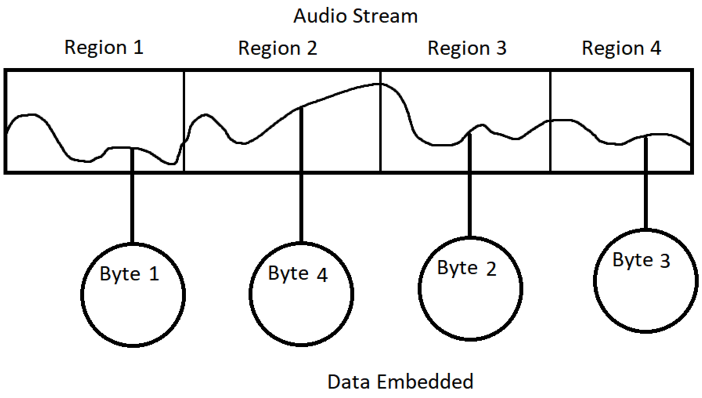

Figure 3.6: Bytes randomly embedded in regions of the audio stream

The data was then simply extracted by generating the pseudo-random number sequence and the bits are retrieved in the same order as they were embedded. This is similar to the image steganography's sub-algorithm, where first the length of the message is extracted.

3.6 The Video Steganography Algorithm

The proposed video steganography technique will adopt the mentioned techniques for the audio and frames. The implementation will improve on the implementations of Sahu U. and Mitra S.'s (2011) and Kakde Y. et al. (2015), mainly by taking advantage of a large number of audio samples and frames present in a video as opposed to focusing the secret data to be embedded in one hidden frame or audio only. The research will also aim to try to maximise the capacity of secret data that can be embedded while maintaining the fidelity of the cover media, which is a goal of steganography in general. The idea of security is also dealt with by implementing a failsafe encryption technique, using the stego-key presented by the user to the secret data prior embedding, as mentioned in Section 3.1.

3.6.1 The Steps of the Algorithm

Embedding Data

The encryption of the secret message is followed by splitting the data by percentage between the audio and frames. After this, the video is split into frames and audio. The aforementioned steganography techniques will be applied, using the same stego key for the frames and audio using the divided, encrypted, and secret message. After the image and audio steganography parts are done, the video is recompiled with the updates frames and audio streams. For testing purposes, the secret message is hashed for future use. This process is clearly illustrated in Figure 3.7.

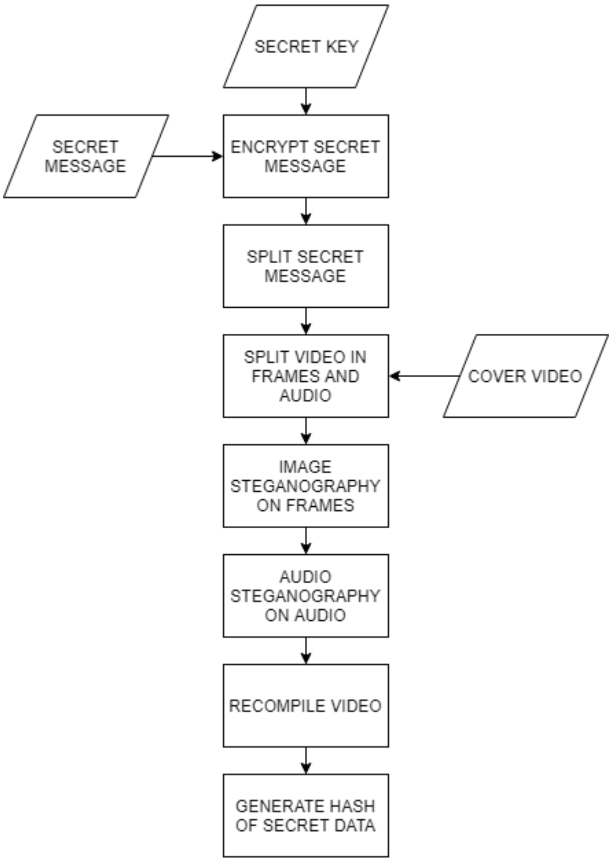
SECRET KEY
SECRET MESSAGE
ENCRYPT SECRET MESSAGE
SPLIT SECRET MESSAGE
SPLIT VIDEO IN FRAMES AND AUDIO
COVER VIDEO
IMAGE STEGANOGRAPHY ON FRAMES
AUDIO STEGANOGRAPHY ON AUDIO
RECOMPILE VIDEO
GENERATE HASH OF SECRET DATA

Figure 3.7: The Embedding Process

Extracting Data

The extraction process functions by first splitting the video into frames and audio, and then, by using the given secret key, it extracts the data from the audio and the frames. Finally, the extracted data is reassembled into one. After this process, the secret data is decrypted and displayed. For testing purposes, the secret message's hash is compared with the one generated at the end of the embedding process to verify data integrity. The extracting process is illustrated in Figure 3.8.

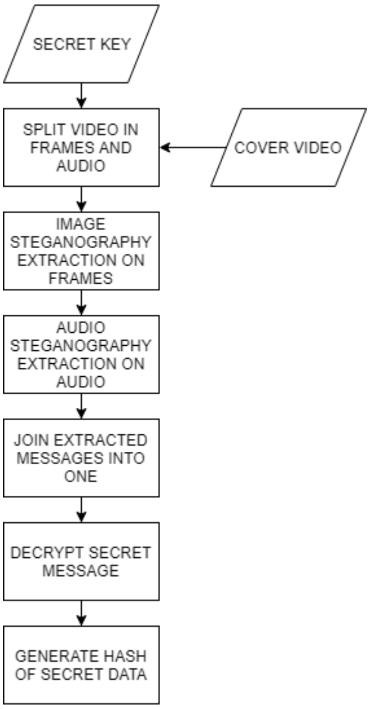
SECRET KEY
SPLIT VIDEO IN FRAMES AND AUDIO
COVER VIDEO
IMAGE STEGANOGRAPHY EXTRACTION ON FRAMES
AUDIO STEGANOGRAPHY EXTRACTION ON AUDIO
JOIN EXTRACTED MESSAGES INTO ONE
DECRYPT SECRET MESSAGE
GENERATE HASH OF SECRET DATA

Figure 3.8: The Extracting Process

3.7 Assumptions Taken

The following is a list of assumptions which were used in this research methodology, mainly regarding the key-agreement protocol to be used between the users in the covert communication channel that will utilise the proposed technique.

1. In this research, it was assumed that the users who are communicating have prior knowledge of the single secret key phrase that will be used to embed and extract data. Such issue is usually not tackled in the reviewed literature, and it is a whole separate area of study.

2. Another assumption (and limitation) in this research was that AVI files will be used with the algorithm. This is because of the AVIFile

Windows library that was used in the development of the prototype, which will be used to gather the results needed.

3. If the produced stego-media is compressed with a lossy compression technique, it will lose its embedded data. This is also limitation of the research since SD techniques are being used for steganography.

3.8 Cover Media To Be Used

For the testing of the image steganography sub-algorithm, common test images such as Lena, Baboon and Peppers were used. Apart from these, images with noise, among others were used. For the testing of the audio steganography sub-algorithm, frequency tones, a silent audio clip and a human speech audio clip were used for testing, among others. For the video steganography algorithm, ten second video clips were used for testing. This is because the cover videos used were uncompressed AVI files, which is a limitation of the proposed research, since dealing with compression can lead to far more complicated obstacles that need to be traversed. Uncompressed AVI files lead to very large sizes if they are long enough. Another reason for the use of AVI files is because of the use of Windows' AVIFile library which only works with AVI files.

3.9 Data To Be Embedded

The type of data to be embedded and extracted does not make any difference because the algorithm works on a bit level. However, text will be primarily used to test the algorithm and to gather the results as it is the easiest way to generate the necessary amount of realistic data since each character is one byte. Lorem Ipsum text is considered a good industry standard dummy text. A simple online Lorem Ipsum generator is an excellent tool that was used to generate as much data as it will be needed to test the stego-video quality under different payload sizes. Online Lorem Ipsum generators gave the ability to generate a given amount of bytes of dummy text. This proved helpful and made it easier to carry out the testing process. As previously mentioned, two hash codes were generated after the embedding and the extracting; this was be done for testing purposes to check if the integrity of the secret message after extracting is kept intact.

3.10 Data Gathering And Analysis

The proposed research aimed to develop a proper video steganography technique that obtains satisfactory quality metrics under different conditions such as the secret data embedded and length of the video. A number of stego videos were created with different secret message sizes and different cover media sizes. Mainly, the relationship between these two variables was be analysed under these different conditions. Some of the metrics mentioned by Dave M. and Somani H.'s (2016) survey, mainly, Imperceptibility and Payload Size were used.

Imperceptibility is one of the most critical metrics. Building on Kharrazi M. et al.'s (2006) research, PSNR values will be used to calculate imperceptibility. These values are also mainly used as quality metrics in this area of study, and it would be easier to compare the proposed work with existing work. Furthermore, the payload size will be calculated by the unit of bits per bytes of data for both the image steganography and audio steganography sub-algorithms and the video algorithm as a whole. Hence, the maximum secret data size that can be embedded, with respect to the cover media size, will be deduced.

The analysis of the proposed algorithm was carried out on the whole video to analyse the overall performance of the algorithm but also on the frames and audio separately. This was meant to analyse the feasibility of using both the audio and the frames for data embedding for the amount of effort it takes. The imperceptibility metrics mentioned by Kharrazi M. et al. (2006) were tabulated. Graphs were used to compare and contrast the different findings gathered in this research for further discussion.

Chapter 4

Findings

4.1 Quality Metrics Used

PSNR is the primary metric used to measure how perceivable the resulting noise caused by the algorithm in the gathered results was, as mentioned by Kharrazi M. et al.'s (2006). PSNR is an error metric, measured in dB. PSNR is a full-reference quality metric, meaning that for the PSNR to be calculated, there is a need for the original file and the changed file. PSNR is defined as being the ratio between the maximum power of a signal and the power of the unwanted noise affecting the fidelity of its representation. PSNR is derived from the MSE, which measures the average squares of errors/deviations.

MSE is defined by the following formula, where the lower the resulting value, the less error there is:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{4.1}$$

PSNR is defined by the following formula, where the higher the resulting value, the less noise there is:

$$PSNR = 10 \cdot \log \left(\frac{MAX^2{}_I}{MSE} \right) \tag{4.2}$$

This shows that PSNR is an inverse to MSE, hence the higher the value, the better the quality. To cater for coincidences in the results, each test was carried out for three or two times, and an average was calculated and used as the final result for each test.

4.2 Payload Capacity and Embedding Rate

The payload capacity of a cover file is the total amount of data, measured in bytes, that can be stored. The embedding rate is the amount of bits stored in a one byte, measured by bpB.

4.2.1 Image Capacity

The image steganography sub-algorithm stores 1 byte in every RGB pixel. This means that if a cover image of resolution 128 by 128 pixels is used, the total number of bytes that can be stored in that image is 16,384 bytes, without any data loss, hence the formula for the capacity is:

$$C_{\text{Image}} = \mathit{Width} \times \mathit{Height} \tag{4.3}$$

In a 24 bit depth image, this makes the embedding rate equal to 3 bpB. If more data than the total capacity is stored, data loss and corruption will occur.

4.2.2 Audio Capacity

The payload capacity of the implemented audio steganography sub-algorithm was found to be:

$$C_{\text{Audio}} = \lfloor \frac{\mathit{Subchunk2Size}}{8} \rfloor \tag{4.4}$$

Subchunk2Size is extracted from the header information of the Wave file. Just like the payload capacity of the image steganography sub-algorithm, if this limit is exceeded, the embedded secret data will be corrupted, resulting in unreadable data. The reason for the subtraction of 300 is because the first 300 bytes were chosen to hold the message length. The maximum embedding rate of the sub-algorithm scales depending on the secret message size. When the maximum amount of data is embedded, the embedding rate is 8 bpB, while the value can go down to any value above 0 if the secret message size is very small.

4.2.3 Video Capacity

The video steganography algorithm adds up the two mentioned payload capacity formulae. The payload capacity of a video is determined by the formula:

$$C_{\text{Video}} = ((N - 1) \times C_{\text{Image}}) + C_{\text{Audio}} \tag{4.5}$$

4.3 The Data Gathering Process

The testing of the video steganography algorithm was split into three stages, being the testing of each individual sub-algorithm and the video steganography algorithm combining both. The image steganography used for the frames of a cover video was tested separately from the audio steganography sub-algorithm used for the audio stream of the video. This was done to analyse separately the viability of each part of the video steganography algorithm that combines both of the sub-algorithms mentioned. Both the image and audio steganography were tested with different scenarios and compared together. Graphs were extracted from the data for visualisation purposes. When possible, the data were compared with data from the reviewed literature. However, this was not always possible due to lack of information about the tests carried out by other researchers. To correctly compare the results with existing literature, the same cover media and secret message size had to be used. The testing of the video steganography algorithm was tested under different types of ten-second clips. The reason for the ten-second clips was because they were large enough to store sufficient amount of data while having a reasonably small size since uncompressed AVI files take up a decent amount of storage. The video steganography algorithm was tested against existing steganography software.

4.4 Results Obtained from the Image Steganography Sub-Algorithm

The Image Steganography sub-algorithm was tested against a similar image steganography technique from Manjula G.R. and Danti A. (2015). Furthermore, testing of the algorithm was done on cover images of different sizes and types.

4.4.1 Comparison with a Similar LSB Based Image Steganography Algorithm

To correctly compare the results with existing literature of Manjula G.R. and Danti A. (2015), the same cover media and secret message size were used. The reason behind choosing this specific literature to compare and contrast the results with is because this algorithm uses a similar embedding technique to the proposed sub-algorithm. This is shown in Figure 4.1.

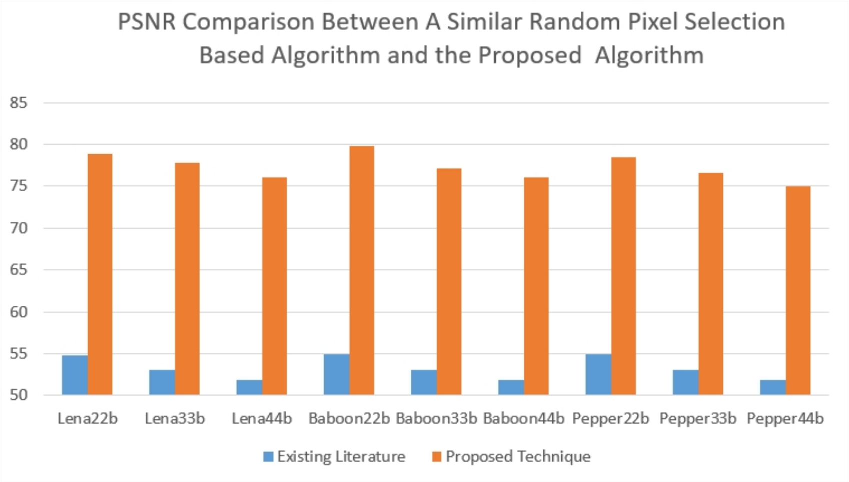

Figure 4.1: PSNR comparison between a similar random pixel selection based algorithm and the proposed algorithm

4.4.2 Testing with Solid Coloured and Noisy Images

The following tests could not be compared with existing literature due to the lack of similar tests carried under the same conditions by other research papers. Three other specific tests were carried out on two solid coloured images, being an entirely white image, a completely black image and an image with salt and pepper noise, each having the resolution of 128 pixels by 128 pixels. This test was carried out to test if it is more beneficial to embed data in existing noise rather than a solid coloured/cartoon image. As shown in the table, the test was done with four different message sizes. The biggest message was chosen to be 16,384 bytes, meaning the maximum amount of data that can fit in the image of the mentioned dimensions. Furthermore, the values of 75%, 50% and 25% of 16,384 were taken to test with different message sizes. The results are illustrated in Figure 4.2

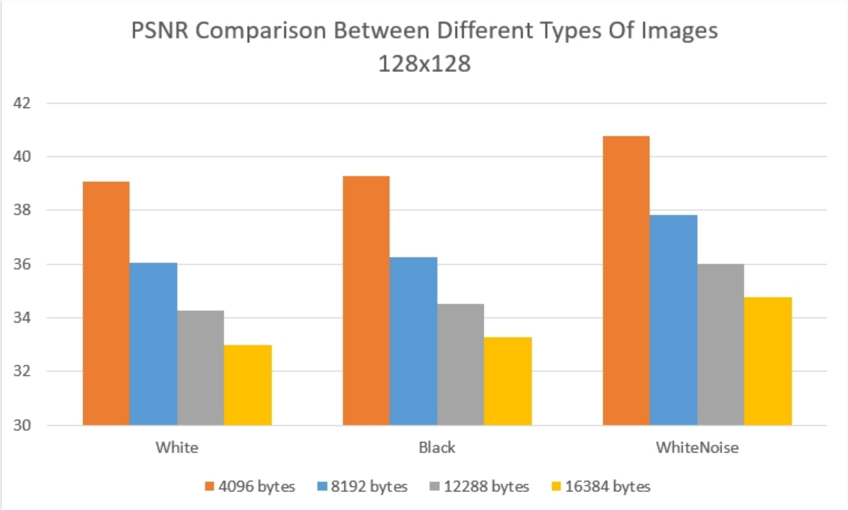

Figure 4.2: PSNR comparison between different types of images 128x128

The results obtained show that it is more beneficial to use an image with as much noise as possible. This can be seen as the WhiteNoise image has about 2 dB more than the black and white images. This is not a big, noticeable difference. However, the present noise helps to obscure the generated noise by the sub-algorithm. The results gathered in this test also highlight the relationship between the PSNR and the message size mentioned previously.

The mentioned test was also carried out to visualise the effect of steganography on these 128x128 images because of the solid background. The noise caused can be faintly seen in Figure 4.3 and Figure 4.4, increasing in intensity as the message size increases.

Figure 4.3: Visualised noise after embedding in a solid black coloured image.

To further highlight the changes done by the sub-algorithm, the brightness and contrast values of the images were adjusted accordingly with an image editing tool. The resulting images are shown in Figure 4.4; the leftmost image being less distorted and the rightmost image being the most distorted, as every pixel was changed.

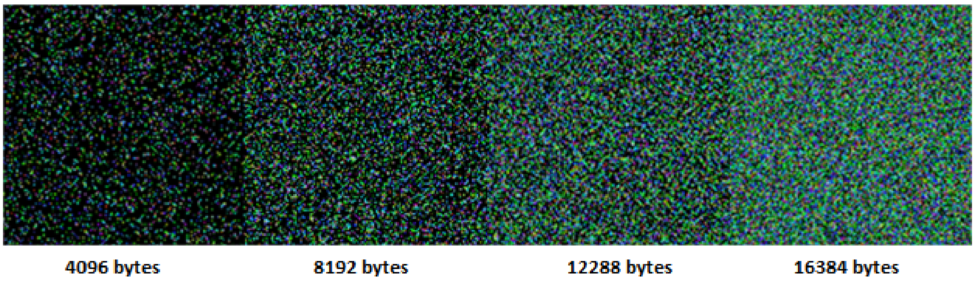

Figure 4.4: Enhanced the noise of Figure 4.3 for better visualisation

4.4.3 Testing with Different Resolutions

Another test was carried out to analyse how the sub-algorithm works with a similar image with different resolution. From the results gathered, it was evident that the higher the resolution the less noise caused, as the number of pixels increase. This is highlighted in the Figure 4.5. This test was carried out with four different secret message sizes and all the tests demonstrated similar results.

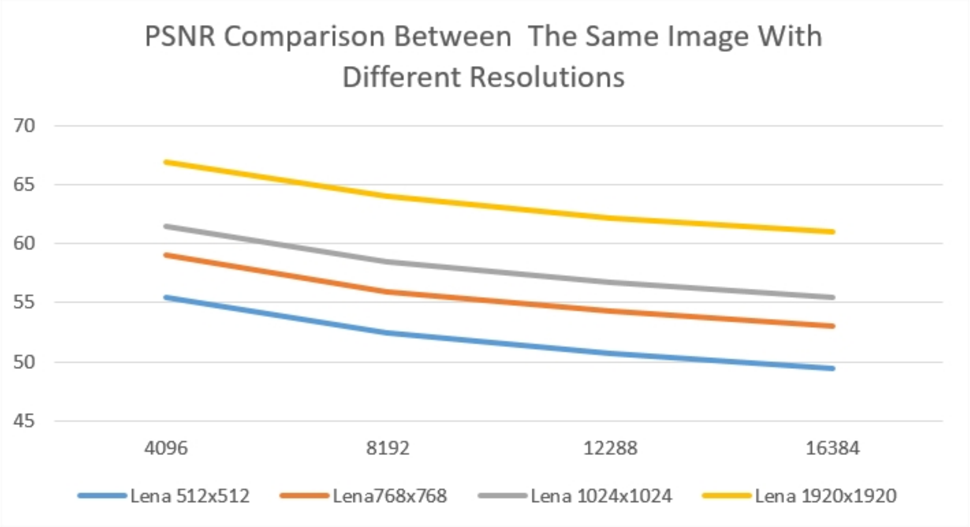

Figure 4.5: PSNR comparison between the same image with different resolutions

4.4.4 Testing with Images of Increasing Colour Depth

The next test carried out for the image steganography sub-algorithm included three images of the same resolution, but each increase in colour depth. This test was done to see if there is any effect on the PSNR if there are more colours present in a picture. The pictures used are found in Figure 4.6.

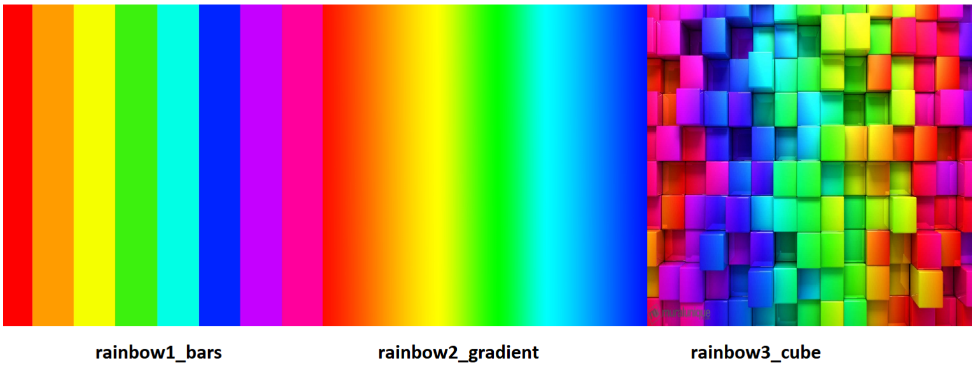

Figure 4.6: The cover images used for the colour depth test

The results obtained, as illustrated in Figure 4.7, showed no significant difference in results when the same secret messages were used throughout, with rainbow3_cube having a slightly higher PSNR in the 8192-byte test and having a slightly lower PSNR in the 12,288 byte and 16,384 byte tests.

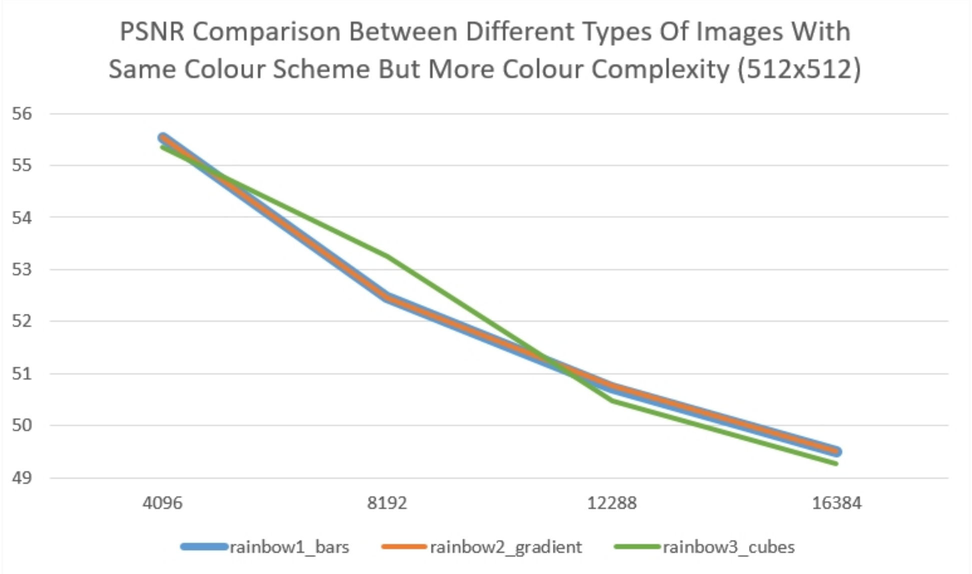

Figure 4.7: PSNR comparison between different types of images with the same colour scheme but more colour complexity (512x512)

4.4.5 Effect of JPEG Compression On PSNR

The final test of the sub-algorithm was carried out to see the effect of JPEG compression on a stego-image. As expected, the JPEG compression drastically decreases the PSNR of the image due to changes in the pixels. This change is highlighted in Figure 4.8 where a clear difference could be seen between the original lossless PNG and the lossy JPEG images.

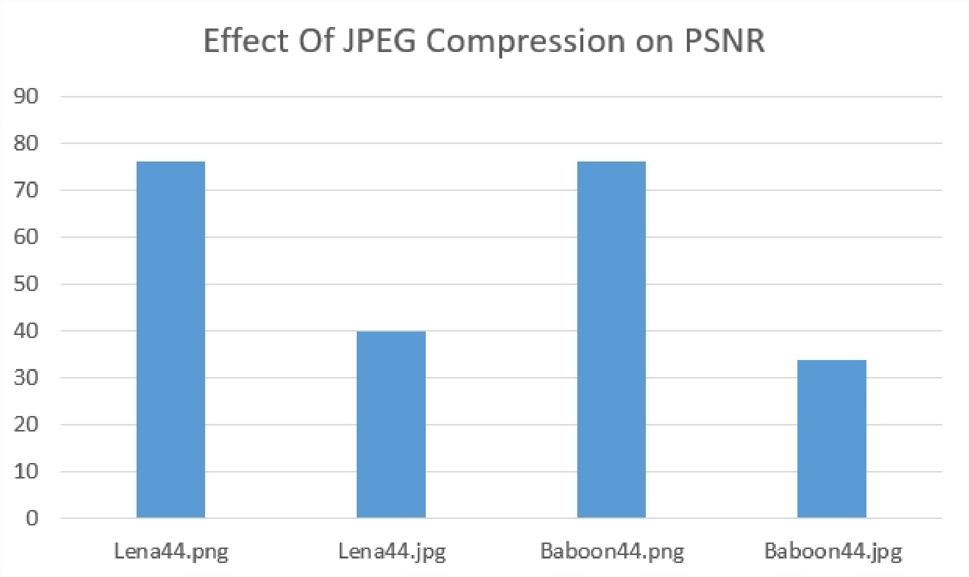

Figure 4.8: Effect of JPEG compression on PSNR

4.5 Results Obtained from the Audio Steganography Sub-Algorithm

The audio steganography sub-algorithm was tested slightly differently due to the lack of good literature results to compare the obtained results. Having this problem, the algorithm was instead tested under different types of wave files, utilising frequency tones, Gaussian noise, a tone sweep and speech with background noise.

4.5.1 Tone, Silence and Noise Testing

The first test was done on six different frequency tones, complete silence and white Gaussian noise, each being 10 seconds long. Three secret message sizes were used in this test, these being: 22 bytes, 33 bytes and 44 bytes. These results can be seen in Figure 4.9.

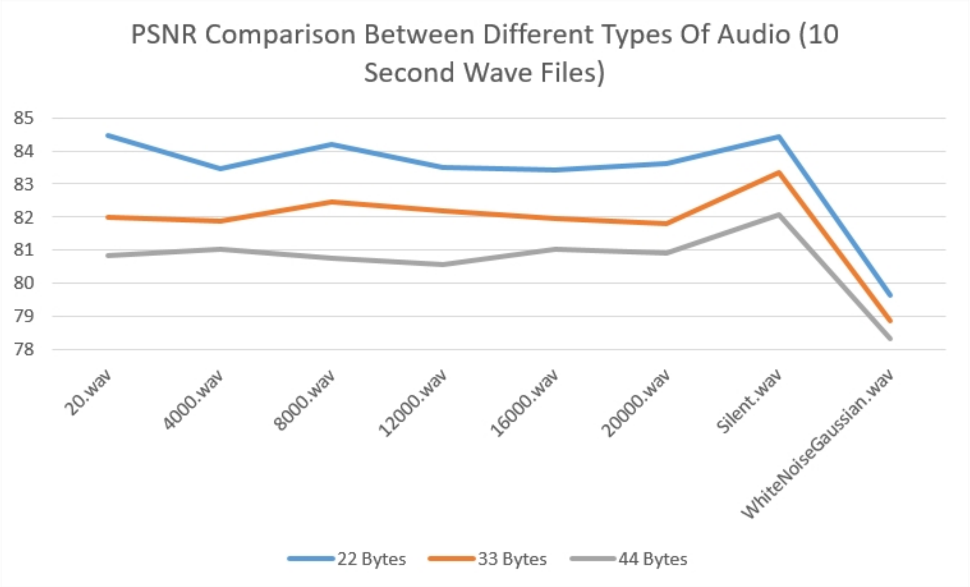

Figure 4.9: PSNR comparison between different types of audio

The findings show an unusual pattern. The tones do not affect the PSNR a lot, with the values varying by a maximum of 1 dB. However, when the tests were carried out on the Silent.wav file, there was an increase in all tests by about 1 dB. This, however, makes the noise generated even more audible. The opposite can be said to the WhiteNoiseGaussian.wav result, where the PSNR plummeted by an average of 4 dB with all message sizes.

This means that more noise is caused; however, it was indistinguishable from the noise of the original file, hence it is less perceivable by the human ear.

4.5.2 Tone Sweep Versus Speech with Cheering Test

Another test was carried out with larger secret data sizes and longer audio clips. The comparison was made between a clean tone sweep, ranging from 20 Hz to 20 kHz. This was compared to audio consisting of human speech and background cheering. The results gathered are displayed in Figure 4.10.

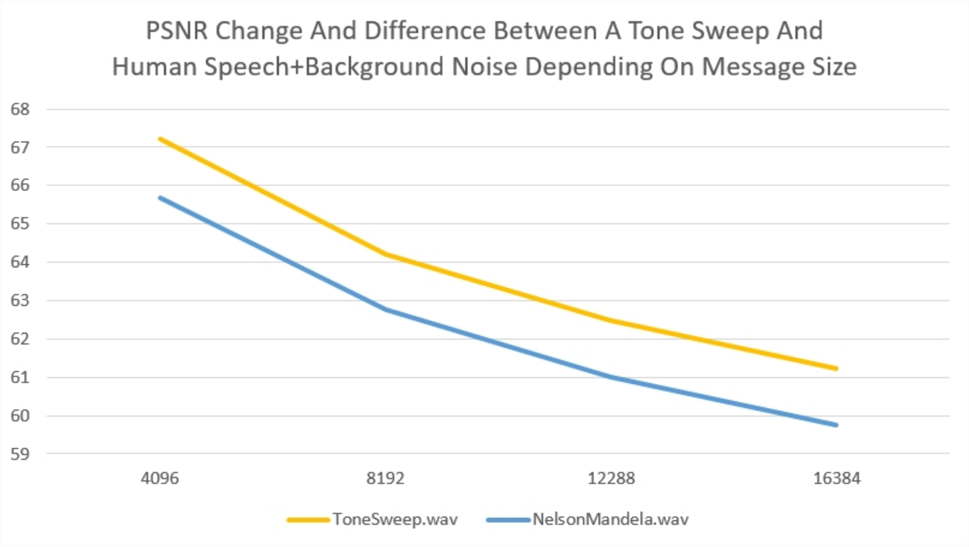

Figure 4.10: PSNR change and difference between tone sweep and human speech with cheering

This test resulted in a very perceptive stego-audio file, as static can easily be heard on both files. The ToneSweep wave file had a slightly higher PSNR. However, the difference is not significant as it is that of a 1 dB increase.

4.6 Results Obtained from the Video Steganography Algorithm

The video algorithm, which combines the previously mentioned algorithms in this section was also tested under different conditions. The total noise generated by both the frame and audio sub-algorithms were then added up, and a final PNSR value in decibels was extracted. It was also tested against existing video steganography software for comparison.

4.6.1 Distribution Ratio Test

The first test carried out for the video steganography algorithm tested the different amounts of secret message distribution between the frames and audio streams. The test uses 2048 bytes of secret data. Five different test cases were simulated, each splitting the secret data differently between the frames and audio. For example, in a 25% distribution test, 25% of the secret data was embedded in the frames while the remaining 75% is embedded in the audio. Similarly, a 75% distribution test embeds 75% of the data in the frames while the remaining 25% are embedded in the audio. The test was also carried out on different resolutions of the same video, indicated by the last three digits of the file name of the X-Axis. The results were gathered by calculating the average PSNR of the frames and the PSNR of the audio stream. Both values were added together, and the total noise caused was calculated. All of this is shown in Figure 4.11.

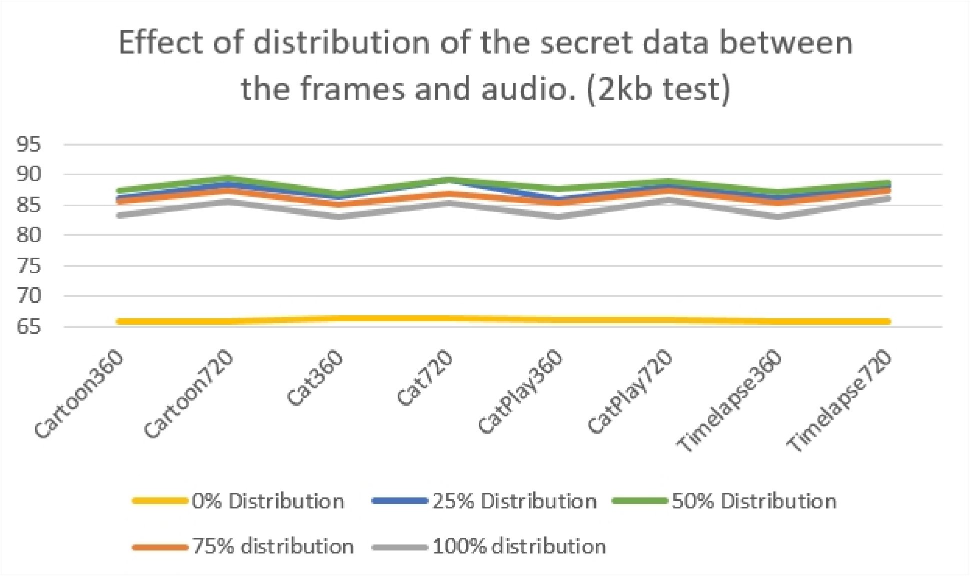

Figure 4.11: Effect of distribution of the secret data between the frames and audio. (2kb test)

This test indicates that there is an advantage to splitting the data between the audio and frames, rather than using either on their own. The 25%, 50 and 75% distribution results were very close to each other, fluctuating by 1 dB here and there. The 0% test, had very poor results, having a PSNR of 20 dB lower than the rest of the results. The best distribution seems to be between 25% and 75%, with results being the best at the 50% distribution ratio. This can support the fact that a video steganography technique utilising both the audio and frames at the same time, improves the overall PSNR. As also concluded from the image steganography results gathered in the previous section, the increased resolution increases the PSNR by a couple of decibels.

4.6.2 Comparision with MSUStegoVideo

The following test compares the algorithm with an algorithm from the video steganography tool developed by Vatolin D. (2007). This algorithm was chosen because it is considered as one of the only free and good video steganography tools out there as Wu J. et al. (2010) and Hernandez-Castro J. and Sloan T. (2015) explain. The data embedded was 2,048 bytes and the same four and ten- second videos were used. The proposed algorithm was used with a 50% embed rate as it proved to offer the best results in the previous test. The results gathered are shown in Table 4.1.

Video	**Message Size (Bytes)**	**MSUStegoVideo**	**Proposed Method**
Cartoon	2048	16.3	85.50766
Cat	2048	15.6	85.1124
CatPlay	2048	17.1	85.30766
Timelapse	2048	16.7	85.20305

Table 4.1: Comparison between MSUStegoVideo and the proposed algorithm

As the values indicate, the average PSNR of the algorithm from the literature is far lower than those obtained from the proposed algorithm, meaning that the proposed algorithm is better concerning the PSNR. This shows improvement over this algorithm by Vatolin D. (2007) when it comes to noise caused by the steganography process. Even when taking the peak PSNR value of the algorithm from the literature, which is an average of 47 for all four videos, which is still lower than the average of the proposed algorithm.

Chapter 5

Discussion Of Results

5.1 Introduction

In this study, the researcher has investigated the possibility of creating a video steganography algorithm based on the basic LSB technique. This incorporated image and audio steganography techniques inspired by existing literature. Since a digital video file is nothing more than a container for a sequence of images and an audio stream, the main hypotheses regarding the possibility utilising both the video frames and audio streams were first tested in a divide and conquer manner. This was achieved by developing two sub-algorithms involving audio and images separately. Both algorithms developed are based on the most influential aspects of different techniques. On the other hand, the divide and conquer approach allowed for easier testing and performances comparison, by first focusing on both parts separately and then combining them as a whole. This structure is shown in Figure 5.1.

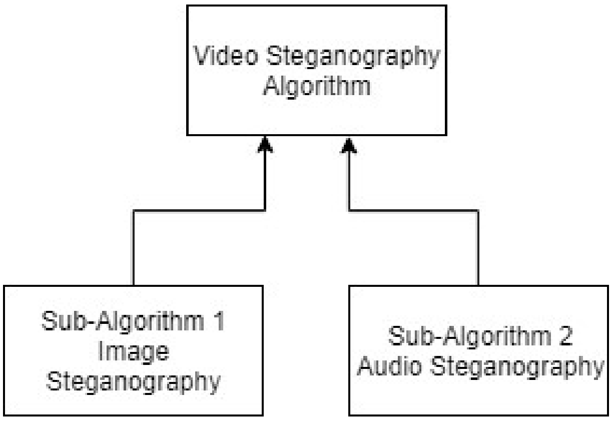
Video Steganography Algorithm
Sub-Algorithm 1 Image Steganography
Sub-Algorithm 2 Audio Steganography

Figure 5.1: The structure of the video steganography algorithm

As mentioned in Chapter 3, an LSB based algorithm was chosen because of its simple nature. This LSB based video steganography algorithm is a contribution to steganography in the SD, reviewed by Singh A. and Singh S. J. (2014), which is considered more vulnerable when compared to other techniques such as those in the TD, mentioned by Kaur N. and Behal S. (2014). The developed algorithm and sub-algorithms were planned with robustness in mind, while staying as scalable as possible, meaning that the algorithms, from the logic aspect, have no limits in terms of message size or carrier size. However, the implemented prototype did have a limit because of the data types used, mostly limited by 2,147,483,647, which is the integer's upper limit. This limits the image steganography sub-algorithm by limiting the maximum data that can be embedded in one image by 2,147,483,647 bits. The same limit is present in the audio steganography sub-algorithm with the maximum number of bits in a single sample region being 2,147,483,647 bits. This issue can be solved by investing more time to make it even more efficient by modifying arrays to use datatypes with more extensive ranges.

Furthermore, all the tests carried out had a 100% message integrity on the extraction process, meaning no loss of embedded data occurred at any single time during the steganography process for the image, audio and video steganography. Any tests that did not have a 100% message integrity due to bugs, which were later fixed, were discarded.

5.2 The Image Steganography Sub-Algorithm

The image steganography sub-algorithm also delivered excellent results, compared with one of the best performing image steganography algorithm mentioned in the Literature Review.

5.2.1 Discussion of the Results of the Sub-Algorithm

The first test was compared with literature from Manjula G.R. and Danti A. (2015). This was done because of a similar hash-based LSB embedding used in both their research and this study. The proposed sub-algorithm utilised Emam M. M. et al.'s (2016) PRNG approach for random pixel selection from the image. This was a significant factor that showed vast improvement on security and robustness by randomly scattering the secret data rather than embedding the data in sequential pixels. This approach not only helped to increase the PSNR, but it is a way to increase further robustness of steganalysis attacks that attempt to extract the data, with the weakest link of the algorithm being the key used to generate the pseudo-random numbers. The improved results show that a less intrusive image steganography sub-algorithm was developed, meaning it caused less noise on the cover image.

The next tests tested the algorithm under different conditions of cover media, starting from solid black and white images and later compared to salt and pepper noise. These tests were carried out to research ideal types of cover media, for example, if it is more beneficial to embed data in existing noise rather than an image with few colours. Interestingly, images with solid colours had identical PSNR values. However, the image with the salt and pepper noise had a 4.1% increase in PSNR. Although this might seem like a small value, the fact that the existing noise makes it even harder for a human to see any changes done, it makes for an exciting finding that can be adapted to any steganography. Existing noise in the cover media will make it more difficult for a human to detect any noise caused in the embedding process. This will be discussed further in subsection 5.4.5.

The relationship between the carrier size and PSNR when using the sub-algorithm was also studied. The results indicate that like most techniques mentioned in Chapter 2, the higher the size of the cover media, the higher the PSNR. This is because a smaller percentage of pixels are altered in a 1920x1920 image when compared with a 512 x 512 image. It could be said that the noise cause is less dense in a higher resolution image when contrasted with a lower resolution image. In simple terms, this could be an analogy to a bucket of water with salt, where the more water present, the more the salt is diluted.

The test carried out between images with the same colour scheme with added colour complexity was chosen to test the algorithm's performance between simple images and complex images regarding the number of colours. The results indicate no considerable difference between the three images. However, the third most complex image had a slightly higher PSNR value in one of the tests. The effect of the number of colours present on the PSNR of the stego-image can be an area that can be expanded upon with further research.

5.3 The Audio Steganography Sub-Algorithm

The testing of the audio steganography sub-algorithm provided a bit of a challenge. This was due to the fact that existing literature did not provide enough information about their testing. This made it difficult to compare the algorithm with existing literature. To cater for this, the algorithm was tested with different types of audio clips as cover media, as mentioned in section 4.5. When the produced stego-audio clips were analysed by ear alone, it was evident that the audio is most susceptible to noise, as it produced audible cracking noise even by embedding small amounts of data. The more data embedded, the more frequent the cracking noise can be heard. As discussed in the previous section, a way to obscure this noise is to hide it in existing noise. If, for example, Gaussian noise is used as a cover audio clip, the noise generated is next to indistinguishable. In simpler words, the lower the sound quality where more noise is present than usual, the better it is to hide the noise caused by the embedded data.

5.3.1 Discussion of the Results of the Sub-Algorithm

The testing done on different frequency tones showed no real difference in PSNR, as the values fluctuate by a maximum of 1 dB. However, when it came to the silent wave file, PSNR values increased but the noise could still be easily heard. The test with the Gaussian noise had its PSNR values plummet by about 4 dB; however, as mentioned before the noise caused by the sub-algorithm could not be distinguished.

The final test done on the audio steganography sub-algorithm compared a tone sweep from 20 Hz to 20 kHz with human speech the change in PSNR was about 1 dB higher for the tone sweep. However, the noise generated could be heard less in the NelsonMandela.wav file due to the video not being perfect quality. The lower quality the audio, the less perceivable the noise is, as mentioned in the first paragraph of this subsection.

This test also highlighted the relationship between the message size and PSNR. Just like the image steganography algorithm's results, the two are inversely proportional to each other, meaning the larger the secret message the, the more noise is generated in the stego-audio file generated by the algorithm.

5.4 The Video Steganography Algorithm

The video steganography algorithm proved to be challenging to implement due to the required codecs and the different types of stream types, especially for the audio. It was a bit difficult to find a video that works until the compatible codecs that work the video processing library used in the development of the prototype. The codecs used for the implementation of the video steganography algorithm were the RV24 video codec and the ARAW audio codec. That being said, the logic of the algorithm can still be used for any codec, assuming the video processing library used is compatible with them.

5.4.1 Addressing the Research Question

Is it viable to create a video steganography algorithm that utilises both image and audio steganography? By dividing and analysing both audio and image steganography sub-algorithms separately, the efficiency of a video algorithm that utilises image and audio steganography was assessed to help answer the research question. As the previously discussed results indicate, the developed sub-algorithms show that they are efficient enough to be used for a video steganography algorithm. The secret message"s integrity was kept after the extraction process. Ideal video conditions to help to hide the noise caused by the steganography were also researched.

5.4.2 The Effect of the Distribution of Data Between Audio and Frames

As the results in Figure 4.11 show, when data was embedded in a video's frames and audio stream, there were improved results. The 0% distribution test, meaning 0% of the secret message was embedded in the frames, and 100% was embedded in the audio, shows how poorly an LSB based audio steganography algorithm works when compared to the proposed video algorithm utilising both the frames and audio. The same can be said when a 100% distribution value was used, meaning all the data was embedded in the frames. The resulting PSNR value was not as low as the audio. However, it was still about 3 dB lower than the proposed method of distributing data between the frames and audio streams. The 25% to 75% distribution values had the best results, coming close to each other, varying by 1 dB to 2 dB. There still was an improvement, albeit a small one, when the secret data was split; hence the effect of the distribution ratio on the PSNR values was studied with these results.

5.4.3 Discussing MSU StegoVideo and the Proposed Algorithm

The comparison with one of the most used, free video steganography tool called MSU StegoVideo by Vatolin D. (2007) indicates a definite improvement. The MSU StegoVideo tool uses compression and LSB steganography. The proposed video-algorithm does not offer compression options, unlike MSU StegoVideo, resulting in a relatively large stego-video which can make it cumbersome to transfer via the internet. The compression that MSU StegoVideo does is the reason for the low PSNR results as explained by Hernandez-Castro J. and Sloan T. (2015) and Wu J. et al. (2010). A similar result from the proposed image steganography sub-algorithm can be seen in the results of Figure 4.8 were compressed images are compared to their uncompressed counterparts. The compressed images had a lower PSNR value by an average of 39.34 dB when compared with their uncompressed.

5.4.4 The Payload Capacity

The proposed video steganography algorithm also has a comparatively larger capacity, when compared with existing works such as those from Deshmukh P. R. and Rahangdale B. (2014). Sahu U. and Mitra S.'s (2011) implementation utilises a selection of frames similar to the proposed algorithm. By utilising all the frames, the average PSNR was kept higher because the overall noise density in an image is lower, while drastically increasing the capacity. The addition of the audio channel further increases the payload capacity. If a larger secret message needs to be passed to another person in the communication channel, a longer video can be used, creating a scaling video steganography algorithm with no limits of storage, with the only trade-off being noise caused in the process.

5.4.5 The Perceivability Of The Algorithm

The most perceivable part of the resulting videos of the proposed algorithm was the audio. Even small amounts of data caused audible noise in the audio stream. On the other hand, it is challenging for humans to perceive pixel changes in the frames, especially if the video is of low quality. A way to counter this is to use a cover video with noisy audio to mask the audible pops and cracks created by the audio steganography sub-algorithm. While an increase of 3 dB may not seem much between a frames- only video steganography and a video-audio steganography technique, it is a still an improvement and could be beneficial when used with the correct cover media, that is, noisy cover media. This can be achieved with preprocessing before any steganography is done on the video, which could be a beneficial improvement upon the proposed algorithm.

Chapter 6

Conclusion

The findings obtained proved to be very helpful in answering the hypothesis. An improved video steganography algorithm was developed and showed promising results namely that if both the audio and frames are used, there would be higher PSNR values, causing a lower impact on video quality. Along with this, an increase in message capacity of a video was achieved. The ideal cover media conditions to help camouflage the noise caused by the steganography were also explored. In this regard, audio proved to be the most sensitive to data embedding. It was found that the noisier the original audio was, before embedding, the less noticeable the noise caused by the embedded data is. These findings could also be applied to the other algorithms.

The contributions to this area of study were made by developing two sub-algorithms tackling audio and image steganography, along with a video steganography algorithm utilising both of them. The audio and image steganography algorithms were developed to also be able to work independently, hence being a contribution to their respective area of study. This was possible by building on other particular works such as those by Emam M. M. et al. (2016) and Manjula G. R. and Danti A. (2015), which helped in constructing the algorithms used in this work. Hence, a better version of the algorithm was developed. This research helped in trying to populate the gap in this research area.

6.1 Limitations

The limitations identified in this study are the following:

1. Processing took a considerable amount of time to process large videos. Such limitation could be overcome with the implementation of multithreading that can speed up the process by dividing the processing among multiple threads. Solving this issue, the algorithm could be tested with larger secret messages and cover videos.

2. It has been assumed that the single secret key phrase is known by both communicating parties and used to embed and extract data. Such issue is usually not tackled in the reviewed literature, as it is a whole separate area of study, referred as a key-agreement protocol.

3. AVI files were used as cover videos. This was limited to only AVI files due to the library used in the development of the prototype. This limitation did not allow testing with the same videos of different formats.

4. If the produced stego-media is compressed with a lossy compression technique, it will lose its embedded data.

5. Building on limitation 4 and as explained in Chapter 5, compression on the video with the embedded data destroys the integrity of the secret message upon extraction. This limitation allowed for large AVI files to be compiled after embedding, which might prove cumbersome to transfer over the internet with low bandwidth. This is common to any steganography algorithms which do not support compression.

6. The limits set by the data types used to implement the algorithm were a limitation. An example of this would be the 2,147,483,647 bit limit that can be embedded inside a single frame or sample region, because of the integer's maximum value.

6.2 Further Research

Further research is mostly based on the limitations mentioned in the previous section as a way to overcome them. Possible further research topics include:

1. Implementing a similar video steganography algorithm that can hold message integrity after compression (such as MPEG-4 or MOV) of a video containing embedded data. At the same time, the processed video will have a smaller size, hence being more lightweight. This can solve the problems presented by limitations 3, 4 and 5.

2. Using PKS instead of SKS is a possible research topic that can further increase the security of the algorithm. This problem is presented by limitation 2.

3. Researching about the possibility of a relationship between video characteristics (amount of frames, number of samples etc.) and distribution ratio of the secret data to be embedded to obtain optimal PSNR values.

4. This proposed algorithm can also be tested against existing steganalysis methods mentioned in Chapter 2, among others. This will test the algorithm's robustness against known steganalysis attacks.

5. Possible implementation of pre-processing techniques, such as adding noise, to be done prior to the steganography that can help to camouflage the existence of embedded data in the cover media, as discussed in Chapter 5.

Bibliography

Abdulzahra, H., Ahmad, R.O.B.I.A.H. and Noor, N.M., 2014. Combining cryptography and steganography for data hiding in images. ACACOS, Applied Computational Science, pp.978-960.

Abikoye, O.C., Adewole, K.S. and Oladipupo, A.J., 2012. Efficient data hiding system using cryptography and steganography.

Atoum, M.S., Alnabhan, M.M. and Habboush, A., 2017. ADVANCED LSB TECHNIQUE FOR AUDIO STENOGRAPHY.

Aung, P.P. and Naing, T.M., 2014. A novel secure combination technique of steganography and cryptography. International Journal of Information Technology, Modeling and Computing (IJITMC), 2(1), pp.55-62.

Baek, J., Kim, C., Fisher, P.S. and Chao, H., 2010, June. (N, 1) secret sharing approach based on steganography with gray digital images. In Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on (pp. 325-329). IEEE.

Bahl, M. and Girdhar, A., 2012. Steganography using the Technique of Orderly Changing of Pixel Components. International Journal of Computer Applications, 58(6).

Chhillar, R.S., 2015. Data Hiding using Advanced LSB with RSA Algorithm. International Journal of Computer Applications, 122(4).

Deshmukh, P.R. and Rahangdale, B., 2014. Hash Based Least Significant Bit Technique For Video Steganography. Int. Journal of Engineering Research and Applications, 4(1), pp.44-49.

Dmitriy Vatolin OP. 2007. Msu stegovideo. Available at http://goo.gl/XqiWi1.

Emam, M.M., Aly, A.A. and Omara, F.A., 2016. An Improved Image Steganography Method Based on LSB Technique with Random Pixel Selection. (IJACSA) International Journal of Advanced Computer Science and Applications, 7(3).

Fraile, R. (2018). Steganography: Hiding your secrets with PHP. [online] Slideshare.net. Available at: https://www.slideshare.net/raulfraile/steganography-hiding-your-secrets-with-php [Accessed 6 Mar. 2018].

Jayaram, P., Ranganatha, H.R. and Anupama, H.S., 2011. Information hiding using audio steganographya survey. The International Journal of

Multimedia & Its Applications (IJMA) Vol, 3, pp.86-96.

Kakde, Y., Gonnade, P. and Dahiwale, P., 2015, March. Audio-video steganography. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on (pp. 1-6). IEEE.

Kaur, M. and Kaur, G., 2014. Review of various steganalysis techniques. International Journal of Computer Science and Information Technologies, 5(2), pp.1744-1747.

Kaur, N. and Behal, S., 2014. A Survey on various types of Steganography and Analysis of Hiding Techniques. International Journal of Engineering Trends and Technology, 11(8), pp.387-91.

Kharrazi, M., Sencar, H.T. and Memon, N., 2006, October. Cover selection for steganographic embedding. In Image Processing, 2006 IEEE International Conference on (pp. 117-120). IEEE.

Kour, J. and Verma, D., 2014. Steganography TechniquesA Review Paper. International Journal of Emerging Research in Management &Technology ISSN, pp.2278-9359.

Krenn, R., 2004. Steganography and steganalysis.

Manjula, G.R. and Danti, A., 2015. A novel hash based least significant bit (2-3-3) image steganography in spatial domain. arXiv preprint arXiv:1503.03674.

Mansi Dave, and Ms. Hinal Somani. (2016) 'A Survey On Digital Video Steganography Techniques Used For Secure Transmission Of Data', Internation Journal Of Advance Research And Innovative Ideas In Education, 2(6), pp. 479-485IJARIIE.

Mishra, M., Mishra, P. and Adhikary, M.C., 2012. Digital image data hiding techniques: A comparative study. arXiv preprint arXiv:1408.3564.

Mishra, S., Yadav, V.K., Trivedi, M.C. and Shrimali, T., 2018. Audio Steganography Techniques: A Survey. In Advances in Computer and Computational Sciences (pp. 581-589). Springer, Singapore.

Nosrati, M., Karimi, R. and Hariri, M., 2012. Audio steganography: A survey on recent approaches. world applied programming, 2(3), pp.202-205.

S.M. Mohidul Islam, MD. Altab Hossin, R.K Shah , P.K Bipin, 2017, Bit Adjusting Image Steganography in Blue Channel using AES and Secured Hash Function, IJCSMC, Vol. 6, Issue. 11, November 2017, pp. 25-30.

Sahu, M.U. and Mitra, M.S., 2015. A Secure Data Hiding Technique Using Video Steganography. International Journal of Computer Science & Communication Networks, 5(5), pp.348-357.

Singh, A. and Singh, S.J., 2014. An Overview of Image Steganography Techniques. International Journal of Engineering and Computer Science, 3(07).

Siper, A., Farley, R. and Lombardo, C., 2005. The rise of steganography. Proceedings of Student/Faculty Research Day, CSIS, Pace University.

Sloan, T. and Hernandez-Castro, J., 2015. Forensic analysis of video steganography tools. PeerJ Computer Science, 1, p.e7.

Swain, G. and Lenka, S.K., 2014. Classification of image steganography techniques in spatial domain: a study. International Journal of Computer Science & Engineering Technology, 5(03), pp.219-232.

Thaneshwari, M. and Arumugam, N., 2012, April. Implementation of Steganography Secret Sharing approach (N, 1) for Color Digital Images. In IJCA Proceedings on International Conference in Recent trends in Computational Methods, Communication and Controls (ICON3C 2012) (No. 2). Foundation of Computer Science (FCS).

Wu, J., Zhang, R., Chen, M. and Niu, X., 2010, April. Steganalysis of msu stego video based on discontinuous coefficient. In Computer Engineering and Technology (ICCET), 2010 2nd International Conference on (Vol. 2, pp. V2-96). IEEE.

Appendix A

Appendices

A.1 Raw Data Gathered

Video	**DataEmbedded**	**Avg Frame PSNR**	**Audio PSNR**	**Added PSNR**
Video	DataEmbedded	Avg Frame PSNR	Audio PSNR	Added PSNR
Cartoon360	2048	inf	65.9	65.9
Cartoon720	2048	inf	65.9	65.9
Cat360	2048	inf	66.3	66.3
Cat720	2048	inf	66.3	66.3
CatPlay360	2048	inf	66.1	66.1
CatPlay720	2048	inf	66.1	66.1
Timelapse360	2048	inf	65.8	65.8
Timelapse720	2048	inf	65.8	65.8

Table A.1: 0% Distribution Tests For Video Steganography

Video	**DataEmbedded**	**Avg Frame PSNR**	**Audio PSNR**	**Added PSNR**
Video	DataEmbedded	Avg Frame PSNR	Audio PSNR	Added PSNR
Cartoon360	2048	85.9	67.1	85.95687709
Cartoon720	2048	88.3	67.1	88.33282028
Cat360	2048	86.3	66.9	86.34957959
Cat720	2048	89.1	66.9	89.1260903
CatPlay360	2048	85.8	67.2	85.85953933
CatPlay720	2048	88.1	67.2	88.13515809
Timelapse360	2048	86.1	66.92	86.15214044
Timelapse720	2048	88.1	66.92	88.1329712

Table A.2: 25% Distribution Tests For Video Steganography

Video	**Data Embedded**	**Avg Frame PSNR**	**Audio PSNR**	**Added PSNR**
Cartoon360	2048	87.2	69	87.26524064
Cartoon720	2048	89.3	69	89.34034276
Cat360	2048	86.8	68.8	86.86829128
Cat720	2048	89.1	68.8	89.14034276
CatPlay360	2048	87.5	69.1	87.56232529
CatPlay720	2048	88.9	69.1	88.94523977
Timelapse360	2048	87	68.9	87.06674867
Timelapse720	2048	88.7	68.9	88.74523977

Table A.3: 50% Distribution Tests For Video Steganography

Video	**DataEmbedded**	**Avg Frame PSNR**	**Audio PSNR**	**Added PSNR**
Video	DataEmbedded	Avg Frame PSNR	Audio PSNR	Added PSNR
Cartoon360	2048	85.3	72.2	85.50766331
Cartoon720	2048	87.1	72.2	87.23830902
Cat360	2048	84.9	71.9	85.11238402
Cat720	2048	86.8	71.9	86.93830902
CatPlay360	2048	85.1	72	85.30766331
CatPlay720	2048	87.2	72	87.32921342
Timelapse360	2048	85	71.8	85.2030451
Timelapse720	2048	87.1	71.8	87.22631452

Table A.4: 75% Distribution Tests For Video Steganography

Video	**DataEmbedded**	**Avg Frame PSNR**	**Audio PSNR**	**Added PSNR**
Video	DataEmbedded	Avg Frame PSNR	Audio PSNR	Added PSNR
Cartoon360	2048	83.2	inf	83.2
Cartoon720	2048	85.6	inf	85.6
Cat360	2048	82.9	inf	82.9
Cat720	2048	85.4	inf	85.4
CatPlay360	2048	83.1	inf	83.1
CatPlay720	2048	85.9	inf	85.9
Timelapse360	2048	83	inf	83
Timelapse720	2048	86.1	inf	86.1

Table A.5: 100% Distribution Tests For Video Steganography

Cover Image (512x512)	**Message Size (bytes)**	**PSNR1**	**PSNR2**	**PSNR3**	**Average PSNR**
Lena	22	78.61	78.2	79.78	78.86333333
Lena	33	76.56	78.63	78.24	77.81
Lena	44	76.05	75.76	76.3	76.03666667
Baboon	22	77.22	80.79	81.56	79.85666667
Baboon	33	77.42	77.32	76.54	77.09333333
Baboon	44	76.3	75.14	76.8	76.08
Pepper	22	77.04	79.55	78.84	78.47666667
Pepper	33	76.99	75.75	77.07	76.60333333
Pepper	44	75.1	75.32	74.62	75.01333333
Fruits	22	78.35	78.47	78.6	78.47333333
Fruits	33	76.99	76.9	77.02	76.97
Fruits	44	74.43	75.08	75.16	74.89

Table A.6: 22 byte, 33 byte, 44 byte Image Steganography Results

Cover Image	**Message Size (bytes)**	**PSNR 1**	**PSNR 2**	**PSNR 3**	**Avg. PSNR**
White4	16384	33	33.01	33	33.00333333
White3	12288	34.23	34.26	34.28	34.25666667
White2	8192	36.05	36.02	36.01	36.02666667
White1	4096	39.07	39.06	39.05	39.06
Black4	16384	33.26	33.29	33.25	33.26666667
Black3	12288	34.53	34.5	34.57	34.53333333
Black2	8192	36.28	36.31	36.21	36.26666667
Black1	4096	39.19	39.31	39.28	39.26
WhiteNoise4	16384	34.75	34.77	34.77	34.76333333
WhiteNoise3	12288	36.03	36.01	36.03	36.02333333
WhiteNoise2	8192	37.86	37.76	37.87	37.83
WhiteNoise1	4096	40.85	40.72	40.73	40.76666667
Baboon4	16384	49.49	49.57	49.45	49.50333333
Baboon3	12288	50.74	50.76	50.75	50.75
Baboon2	8192	52.4	52.47	52.57	52.48
Baboon1	4096	55.53	55.57	55.38	55.49333333
Peppers4	16384	49.22	49.29	49.3	49.27
Peppers3	12288	50.57	50.56	50.61	50.58
Peppers2	8192	52.28	52.37	52.22	52.29
Peppers1	4096	55.33	55.24	55.3	55.29
Lena4	16384	49.36	49.46	49.39	49.40333333
Lena3	12288	50.67	50.76	50.75	50.72666667
Lena2	8192	52.43	52.51	52.45	52.46333333
Lena1	4096	55.58	55.49	55.4	55.49
Lena7684	16384	52.97	52.98	52.97	52.97333333
Lena7683	12288	54.25	54.2	54.28	54.24333333
Lena7682	8192	55.97	56	55.9	55.95666667
Lena7681	4096	59.18	58.85	59.06	59.03
Lena10244	16384	55.46	55.44	55.52	55.47333333
Lena10243	12288	56.56	56.72	56.7	56.66
Lena10242	8192	58.37	58.49	58.51	58.45666667
Lena10241	4096	61.52	61.44	61.55	61.50333333
Lena19204	16384	60.91	60.97	60.94	60.94
Lena19203	12288	62.19	62.1	62.21	62.16666667
Lena19202	8192	63.99	63.98	63.9	63.95666667
Lena19201	4096	66.78	66.95	67.08	66.93666667
rainbow1bars4	16384	49.49	49.51	49.47	49.49
rainbow1bars3	12288	50.72	50.76	50.73	50.73666667
rainbow1bars2	8192	52.41	52.43	52.55	52.46333333
rainbow1bars1	4096	55.53	55.43	55.63	55.53
rainbow2gradient4	16384	49.56	49.45	49.52	49.51
rainbow2gradient3	12288	50.8	50.75	50.75	50.76666667
rainbow2gradient2	8192	52.38	52.48	52.53	52.46333333
rainbow2gradient1	4096	55.5	55.67	55.47	55.54666667
rainbow3cubes4	16384	49.31	49.22	49.3	49.27666667
rainbow3cubes3	12288	50.45	50.49	50.49	50.47666667
rainbow3cubes2	8192	52.3	52.16	55.29	53.25
rainbow3cubes1	4096	55.46	55.3	55.29	55.35

Table A.7: Larger Capacity Image Steganography Results

AudioFile	**Message Size (bytes)**	**PSNR 1**	**PSNR2**	**PSNR3**	**Avg. PSNR**
20.wav	22	84.786	84.114	84.52	84.47333333
20.wav	33	82.303	81.775	81.866	81.98133333
20.wav	44	80.663	80.878	81.028	80.85633333
4000.wav	22	83.672	83.401	83.336	83.46966667
4000.wav	33	81.864	82.101	81.639	81.868
4000.wav	44	80.805	81.262	81.027	81.03133333
8000.wav	22	84.276	84.276	84.041	84.19766667
8000.wav	33	82.514	82.46	82.46	82.478
8000.wav	44	80.771	80.771	80.7	80.74733333
12000.wav	22	83.208	83.887	83.401	83.49866667
12000.wav	33	82.007	82.055	82.459	82.17366667
12000.wav	44	80.359	80.916	80.492	80.589
16000.wav	22	83.401	83.534	83.336	83.42366667
16000.wav	33	81.774	82.103	82.006	81.961
16000.wav	44	80.878	81.427	80.805	81.03666667
20000.wav	22	83.672	83.814	83.335	83.607
20000.wav	33	81.64	81.958	81.865	81.821
20000.wav	44	81.221	80.989	80.558	80.92266667
NelsonMandela.wav	22	87.73	87.815	87.989	87.84466667
NelsonMandela.wav	33	86.897	86.14	86.695	86.57733333
NelsonMandela.wav	44	85.206	85.349	85.114	85.223
Silent.wav	22	84.523	84.523	84.275	84.44033333
Silent.wav	33	83.025	83.537	83.469	83.34366667
Silent.wav	44	82.056	82.154	81.961	82.057
ToneSweep.wav	22	89.047	88.968	88.447	88.82066667
ToneSweep.wav	33	87.506	87.128	87.858	87.49733333
ToneSweep.wav	44	86.828	86.733	86.414	86.65833333
WhiteNoiseGaussian.wav	22	79.448	79.893	79.555	79.632
WhiteNoiseGaussian.wav	33	78.996	78.996	78.675	78.889
WhiteNoiseGaussian.wav	44	78.252	78.252	78.458	78.32066667
NelsonMandela.wav	16384	59.73	59.745	59.757	59.744
NelsonMandela.wav	12288	60.99	61.011	61.006	61.00233333
NelsonMandela.wav	8192	62.753	62.746	62.75	62.74966667
NelsonMandela.wav	4096	65.676	65.664	65.717	65.68566667
ToneSweep.wav	16384	61.239	61.227	61.235	61.23366667
ToneSweep.wav	12288	62.501	62.463	62.466	62.47666667
ToneSweep.wav	8192	64.22	64.192	64.224	64.212
ToneSweep.wav	4096	67.166	67.298	67.225	67.22966667

Table A.8: Audio Steganography Results