Configuration Manual

MSc Academic Internship
Cyber Security

Anurag Chandrabali Gautam
Student ID: x18145060

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Anurag Chandrabali Gautam

Student ID: X18145060

Programme: MSc in Cyber Security **Year:** 2019-2020

Module: Academic Internship

Lecturer: Mr. Imran Khan

Submission Due Date: 12/12/2019

Project Title: Secure End to End transmission using Audio Steganography and AES encryption

Word Count: **Page Count:**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ……

Date: 12th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the	□

project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	

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

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

Configuration Manual

Anurag Chandrabali Gautam
Student ID: x18145060

1 Introduction

This article will give us an insight on how the proposed prototype is executed and can be utilized. This paper provides the prototype which gives us the secondary secure channel when the AES encryption key is compromised, or the attacker has used pixel degrading method on the stegoimage which will allow attacker to recover the embedded secret message. This prototype uses 256-bit AES encryption for robust ciphertext (Chernev, 2019) and for embedding ciphertext in the image we have used LSB encoding and for embedding stegoimage in the audio file we have used Discrete Wavelength Transform. It is difficult to do steganalysis attack on the audio file. (Garg & Kaur, 2017)

2 Configuration of System

2.1 Hardware Configuration

- Operating System: Windows 7 or later
- Processor: 2 or more CPU cores
- System: 32-bits or 64-bits
- Hard Disk: 256 Gb or more
- RAM: 2 GB or more

2.2 Software Configuration

This part of the article illustrates the information about the tools and skills used while developing the prototype.

Tool	Version	Illustration
Python (Windows 64 bit)	3.8	Python programming language is used to run the encryption and decryption script.
MATLAB	R2018a	It is used to embed and extract the secret message in image and image in audio file.

Table 1: Applications used in this prototype.[1][2]

[1] https://www.python.org/downloads/release/python-380/

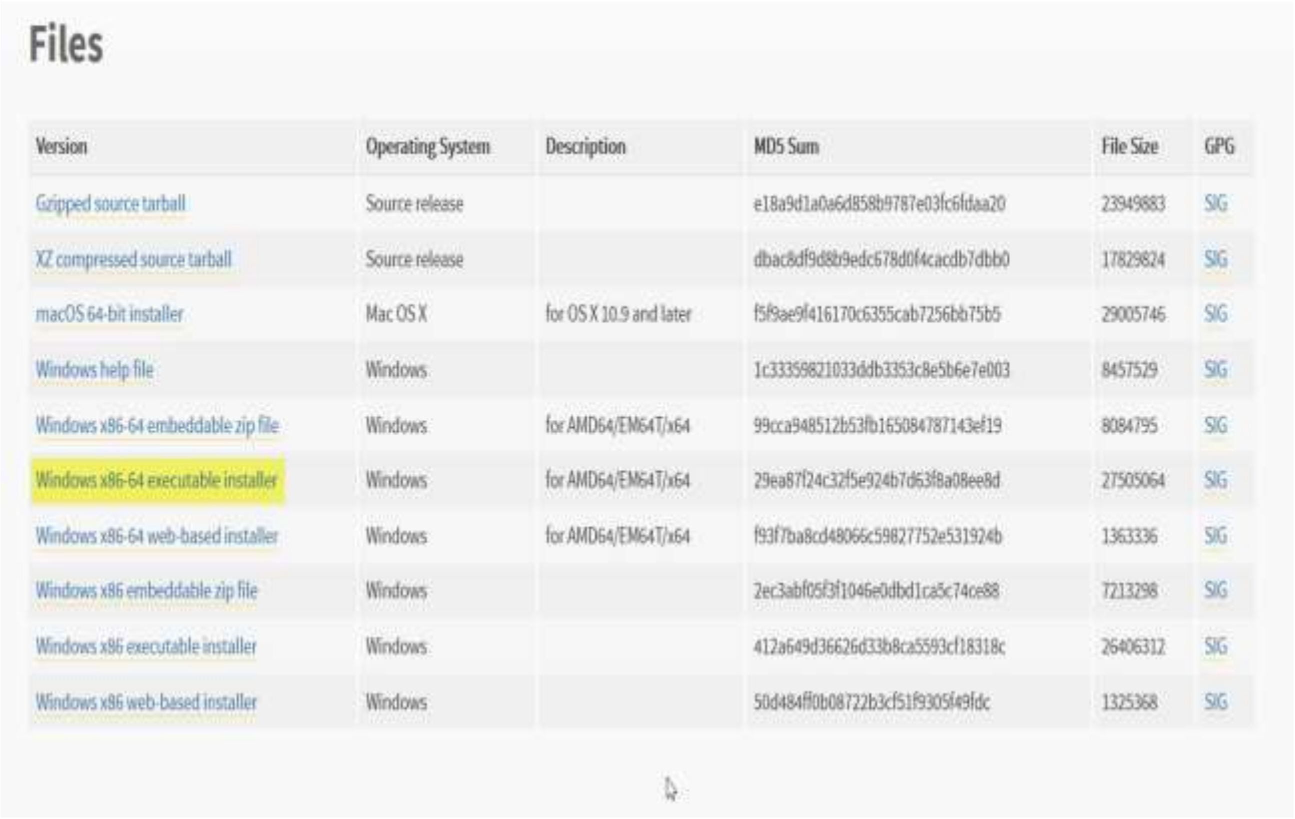
[2] https://uk.mathworks.com/products/new_products/release2018a.html

3 Functioning

This part to the manual shows the step by step method used for installing and configuring applications for proposed prototype and its working as well.

Installation of Applications

Python 3.8.0 is the newest release of the python language. We can download it from the following link.
https://www.python.org/downloads/release/python-380/



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		e18a9d1a0a6d858b9787e03fc6fdaa20	23949883	SIG
XZ compressed source tarball	Source release		dbac8df9d8b9edc678d0f4cacdb7dbb0	17829824	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	f5f9ae9f416170c6355cab7256bb75b5	29005746	SIG
Windows help file	Windows		1c33359821033ddb3353c8e5b6e7e003	8457529	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	99cca948512b53fb165084787143ef19	8084795	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	29ea87f24c32f5e924b7d63f8a08ee8d	27505064	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	f93f7ba8cd48066c59827752e531924b	1363336	SIG
Windows x86 embeddable zip file	Windows		2ec3abf05f3f1046e0dbd1ca5c74ce88	7213298	SIG
Windows x86 executable installer	Windows		412a649d36626d33b8ca5593cf18318c	26406312	SIG
Windows x86 web-based installer	Windows		50d484ff0b08722b3cf51f9305f49fdc	1325368	SIG

Figure 1: Python 3.8.0 executable file

MATLAB R2018a can be download using following link. It is a paid software, but student can opt for 30 days free trial for students by registering with their student credentials.

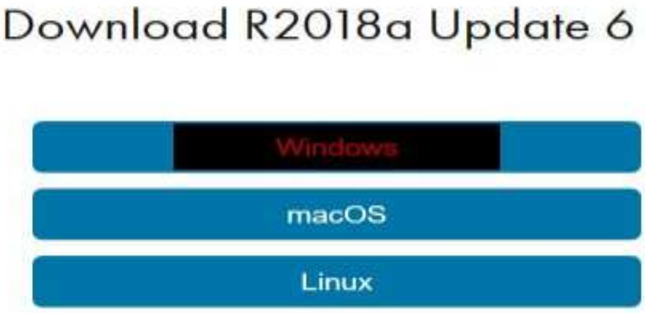
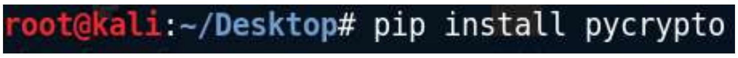


Figure 2: MATLAB download file

3.1 Working

As few libraries of the python were unable to install in the windows 10 due to some reason, that’s why I have used python 2 in Kali Linux OS as an alternative for this.

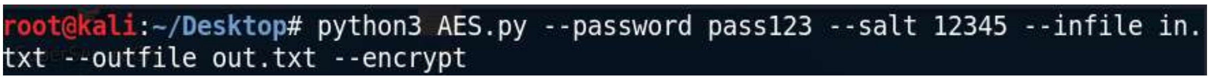
Following figure 3 shows that before starting the encryption, we have to install some python libraries. Following command is used to install library “pycrypto”.



```
root@kali:~/Desktop# pip install pycrypto
```

Figure 3: Installing pycrypto library

Figure 4 shows the command to run the AES python code to encrypt the secret message.
--password: It indicates the encryption password.
--salt: salting the password.
--infile: It is the file in which the secret message is saved and imported
-- outfile: It is the file in which the ciphertext of the secret message will be stored
--encrypt: for encryption

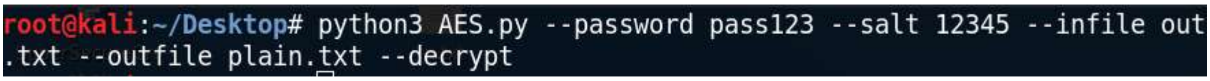


```
root@kali:~/Desktop# python3 AES.py --password pass123 --salt 12345 --infile in.
txt --outfile out.txt --encrypt
```

Figure 4: Encryption command for secret message

Figure 5 shows the command to run the AES python code to decrypt the ciphertext which is to be extracted from the stegoimage.

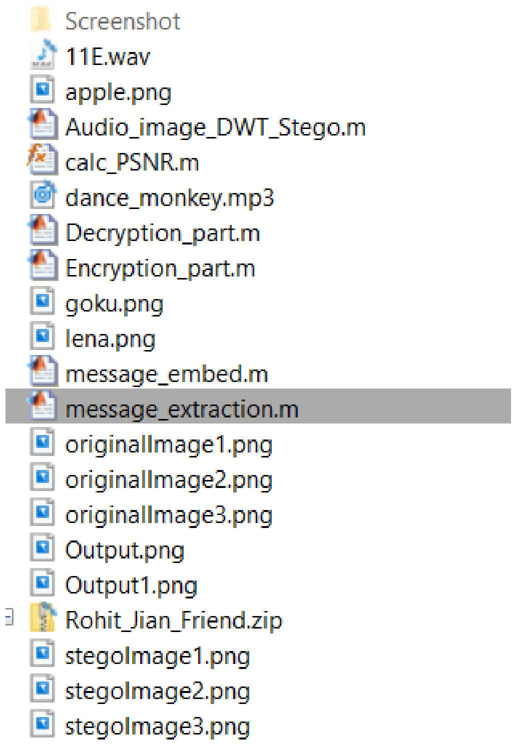
--password: It indicates the decryption password.
--salt: salting the password.
--infile: It is the file in which the ciphertext is saved and imported
-- outfile: It is the file in which the secret message will be stored
--decrypt: for decryption



```
root@kali:~/Desktop# python3 AES.py --password pass123 --salt 12345 --infile out
.txt --outfile plain.txt --decrypt
```

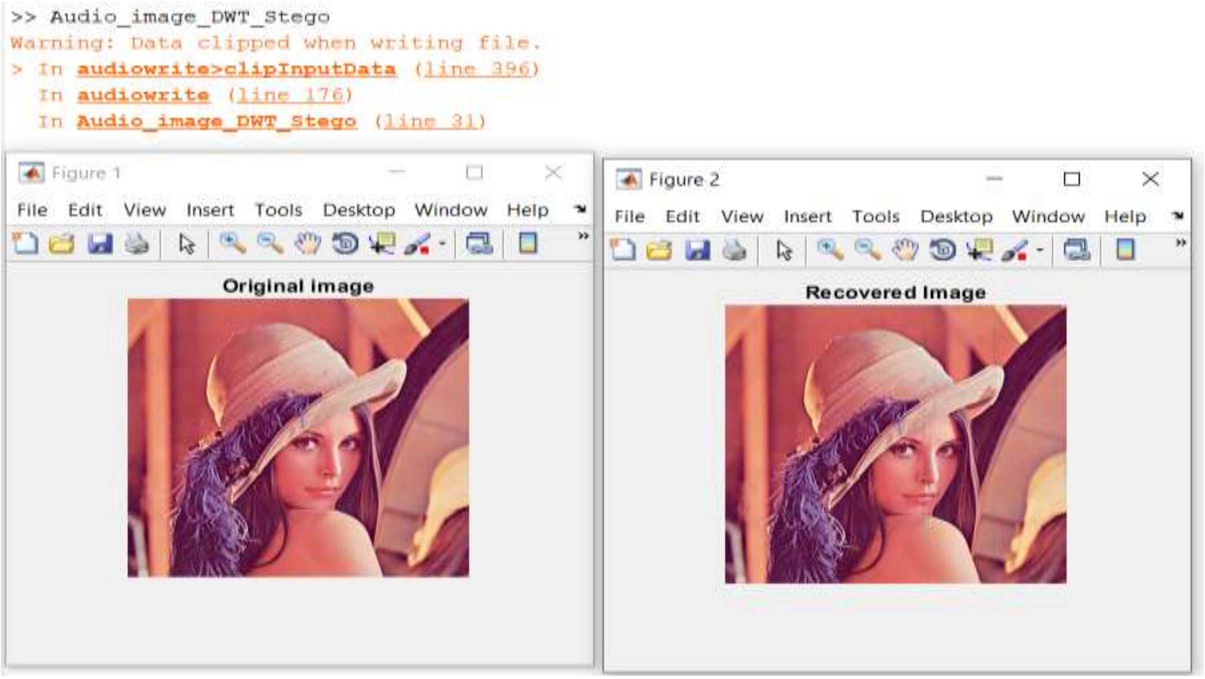
Figure 5: Decryption command for ciphertext

Following figure 6 incidates the file and images path of the image steganography. This is the path in which the stegoiamge is to be stored. After running the message_embed.m file we will create two iamge file would be named as originalimage1.png and stegoiamge1.png which contains the ciphertext of the secret message as we can see in the below figure 6.



```
Screenshot
11E.wav
apple.png
Audio_image_DWT_Stego.m
calc_PSNR.m
dance_monkey.mp3
Decryption_part.m
Encryption_part.m
goku.png
lena.png
message_embed.m
message_extraction.m
originalImage1.png
originalImage2.png
originalImage3.png
Output.png
Output1.png
Rohit_Jian_Friend.zip
stegoImage1.png
stegoImage2.png
stegoImage3.png
```

Figure 6: File path where the stegoimage will be stored



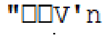
```
>> Audio_image_DWT_Stego
Warning: Data clipped when writing file.
> In audiowrite>clipInputData (line 396)
  In audiowrite (line 176)
  In Audio_image_DWT_Stego (line 31)
```

Figure 1
File Edit View Insert Tools Desktop Window Help
Original image
Figure 2
File Edit View Insert Tools Desktop Window Help
Recovered Image

Figure 7: Embedding and Extraction of stegoimage1 in audio file

From the figure 6, we have the file path which contains cover audio file and stegoimage which is to be embed in the audio file. Encryption _part.m is run in the MATLAB to embed the stegoimage1.png in the cover audio file which is dance_monkey.mp3 . Decryption_part.m is run in the MATLAB to extract the embedded stegoimage1.png. From figure 7, we can see the recovered image and steganographic audio file is saved in the filepath and it also runs in the background after running the code.

From figure 8, we can see that there is number of bits loss while extracting the ciphertext from the stegoimage. Due to loss the bits, the embedded ciphertext has been lost. We couldn’t able to recover the ciphertext fully from the stegoimage.



```
"□□V'n
```

Figure 8: Ciphertext extracted from the stegoimge.

4 References

[1] Y. Garg and A. Kaur, "A Case study on Steganography and its Attacks," *International Journal of Engineering Trends and Technology(IJETT),* p. 5, 2017.

[2] B. Chernev, "What Is AES and Why You Already Love It," 12 03 2019. [Online]. Available: https://techjury.net/what-is-aes/#gref.