Configuration Manual

MSc Internship
Cyber Security

Manmeet Kumar
Student ID: x21113602

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Manmeet Kumar		
Student ID:	X21113602		
Programme:	MSc Cyber Security	**Year:**	2022-2023
Module:	MSc Internship		
Supervisor:	Arghir Nicolae Moldovan		
Submission Due Date:	15/12/2022		
Project Title:	Secure Transmission of Data using Audio Steganography and Cryptography		
Word Count:	715	**Page Count:**	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Manmeet
Date: 15/12/2022

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

Manmeet Kumar
X21113602

1. Introduction

This manual contains information on the proposed model's setup and requirements, including particular libraries and essential applications/softwares. This setup handbook also includes instructions on how to perform the experiment.

2. Hardware

- Processor: 9th Gen Intel® Core ™ i7-9750 CPU @ 2.6 GHz, 2592 Mhz, 6 Core(s)
- Installed RAM: 16.0 GB, DDR4
- System Type: 64-bot OS, x64-based processor
- SSD: 237 GB
- Graphics card: NVIDIA GeForce GTX 1650

3. Software and Tools Used

- Operating System: Kali Linux 5.18.0
- Programming Language: Python 3.10.8
- Code Editor: Visual Studio Code
- Steganography Tool: Steghide 0.5.1
- Other Tools: Microsoft Excel, NumPy 1.23.4

4. Configuration Steps

1. Python is pre-installed on Kali Linux. To check its version, use below command:

   To check its version:
   # python3 –V

   To update python version:
   # sudo apt-get install python3

Fig. 1: Python Installation

2. Download and follow steps mentioned in below link to install Visual Studio Code in Kali Linux.

   Download link: https://code.visualstudio.com/download [1]
   Installation Guide: https://code.visualstudio.com/docs/setup/linux [2]

3. To install Steghide use below command[3]:

   # sudo apt install steghide

Fig. 2: Steghide Installation

5. Implementation

The implementation starts by importing required libraries. These imported libraries are used for different purposes like running OS commands, performing mathematical operations, usage of audio files in the project etc.

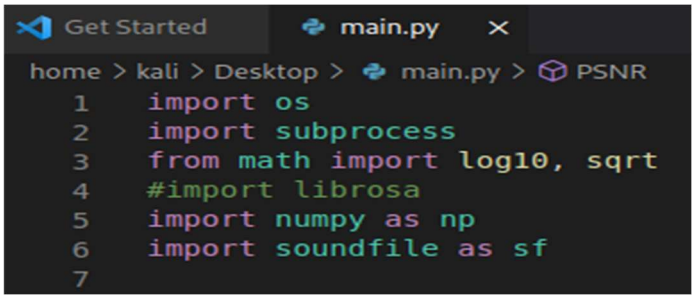

Fig. 3: Python Libraries

After importing all the required libraries, we iterated through folders containing cover audio files and secret data files and select these files to perform encryption and embedding of secret data files into cover audio files.

```
8   coverFiles = []
9   secretFiles = []
10  for i in os.listdir('/home/kali/Desktop/Coverfiles/'):
11      coverFiles.append(i)
12  # print(coverFiles)
13
14  for i in os.listdir('/home/kali/Desktop/Secret-message/'):
15      secretFiles.append(i)
16  # print(secretFiles)
17
```

Fig. 4: Iteration through Cover audio and secret data files

Next, we used a kali Linux tool named Steghide to perform different types of encryptions on all the files present in “Secret-message” folder then embed these encrypted files into cover audio files present in “Coverfiles” folder and the resultant Stego-audio files are saved in a folder named “Embedfiles”

```
cmd = 'steghide embed -cf /home/kali/Desktop/Coverfiles/'+cvFile+' -ef /home/kali/Desktop/Secret-message/'+sctFile+' -p Pass@123@ -e '+ency+' -sf /home/kali/Desktop/Embedfiles/'+outputF
```

Fig. 5: Steghide Operation

After getting all the stego-audio files we calculated MSE and PSNR for all the files to do evaluation of the experiment.[4]

```
25  def PSNR(original, compressed):
26      signal1, sr1 = sf.read(original,dtype='float32')
27      signal2, sr2 = sf.read(compressed,dtype='float32')
28      err = np.mean((signal1 - signal2) ** 2)
29      mse = sqrt(err)
30      if(mse == 0):  # MSE is zero means no noise is present in the signal .
31                     # Therefore PSNR have no importance.
32          return 100
33      max_pixel = 65536
34      psnr = 20 * log10(max_pixel / mse)
35      result=[psnr,mse]
36      return result
```

Fig. 6: MSE & PSNR calculation

After calculating PSNR and MSE for all the Stego-audio files, we put all the required fields like “CoverFile name”, “size”, “SecretFile name”, “size”, “Encryption used”, “Output file name”, “PSNR” and “MSR” in CSV for evaluation part.[5]

```
csvData = []
for coverFile in range(len(coverFiles)):
    for secretFile in range(len(secretFiles)):
        for encryption in range(len(encryptions)):
            data ={}
            cvFile=coverFiles[coverFile]
            cvFileSize=os.path.getsize('/home/kali/Desktop/Coverfiles/'+cvFile)/1024/1024
            sctFile=secretFiles[secretFile]
            sctFileSize=os.path.getsize('/home/kali/Desktop/Secret-message/'+sctFile)/1024/1024
            ency = encryptions[encryption]

            outputFname = cvFile.split('.')[0]+'_'+sctFile.split('.')[0]+'_'+ency+'.wav'
            cmd = 'steghide embed -cf /home/kali/Desktop/Coverfiles/'+cvFile+' -ef /home/kali/Desktop/Secret-message/'+sctFile+' -p Pass@123@ -e

            print(subprocess.getoutput(cmd))
            # outFileSize=os.path.getsize('/home/kali/Desktop/Embedfiles/'+outputFname)/1024/1024

            data['CoverFile']=cvFile
            data['CoverFile_size']=cvFileSize
            data['SecretFile'] = sctFile
            data['SecretFile_size'] =sctFileSize
            data['Encryption'] = ency
            data['outputFile'] = outputFname
            # data['outputFile_size'] = outFileSize
            result= PSNR('/home/kali/Desktop/Coverfiles/'+cvFile,'/home/kali/Desktop/Embedfiles/'+outputFname)
            data['PSNR'] = result[0]
            data['MSE'] = result[1]
            csvData.append(data)

```

```
import csv
with open('thesisOutput.csv', 'w') as csvfile:
    fieldnames = ['CoverFile', 'CoverFile_size','SecretFile','SecretFile_size','Encryption','outputFile','PSNR','MSE']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    for i in csvData:
        writer.writerow(i)
```

Fig. 7: CSV code

References

[1] “Download Visual Studio Code - Mac, Linux, Windows.” https://code.visualstudio.com/Download (accessed Dec. 15, 2022).

[2] “Running Visual Studio Code on Linux.” https://code.visualstudio.com/docs/setup/linux (accessed Dec. 15, 2022).

[3] “steghide | Kali Linux Tools,” *Kali Linux*. https://www.kali.org/tools/steghide/ (accessed Dec. 15, 2022).

[4] “Python | Peak Signal-to-Noise Ratio (PSNR) - GeeksforGeeks.” https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/ (accessed Dec. 15, 2022).

[5] “csv — CSV File Reading and Writing,” *Python documentation*. https://docs.python.org/3/library/csv.html (accessed Dec. 15, 2022).