Configuration Manual

Research Project
MSc Cloud Computing

Rajendra Chavan
Student ID: 21124213

National College of Ireland

Supervisor: Prof. Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rajendra Anil Chavan		
Student ID:	21124213		
Programme:	MSc in Cloud Computing	**Year:**	2022-2023
Module:	Research Project		
Lecturer:	Prof. Rashid Mijumbi		
Submission Due Date:	18th September 2023		
Project Title:	Cloud Data Security Improvement using cryptographic steganography by truly random and cryptographically secure random numbers		
Word Count:	607	**Page Count:**	3

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: Rajendra Anil Chavan

Date: 14th August 2023

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

Rajendra Chavan
Student ID: 21124213

1 Introduction

This Configuration Manual contains all the information needed to replicate the research and its findings on a local system. For establishing the coding environment, the system requirements for local run, file source, and Python packages are supplied.

2 System Specifications

Hardware Configuration for the local run:

- Processor: Intel 11th Gen Core i3 @2.4 GHz
- RAM: 16 GB DDR4 RAM 3200MHz
- Storage (SSD): 512GB
- Operating System: Windows 10, 64-bit

Software Packages for the local run:

- Python 3.9.17
- Visual Studio Code
- Python libraries

3 Cryptography Packages

PyCryptodome[1] is a Python library that provides cryptographic functions and utilities, allowing developers to implement a wide range of cryptographic operations in their applications. It is a fork of the older PyCrypto library and aims to provide a more up-to-date and actively maintained set of cryptographic tools. PyCryptodome supports various cryptographic algorithms, including symmetric and asymmetric encryption, digital signatures, hashing, key generation, and more (Hollman et al.; 2022).

[1] https://pypi.org/project/pycryptodome/

4 Installation of packages and a virtual environment

For the purpose of developing code on the PC, the following packages were installed. Before installing the packages needed for the creation and execution of the code, a virtual environment must first be constructed.

After installing visual studio code, open the visual studio and then open visual studio terminal as “bash” and then type the below commands to create environment,

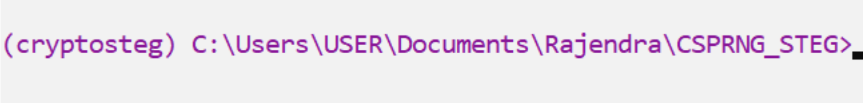
```
      python -m venv cryptosteg
```

This will create the virtual environment for the project with name, “`cryptosteg`”

This newly created environment must be activated before installing the packages. It is done using the following command, On the terminal,

```
      source env/bin/activate (only on Linux systems)
      source env/Scripts/activate (only on windows systems)
```

After this command, the prompt will change to the *cryptosteg* environment as seen below,



```
(cryptosteg) C:\Users\USER\Documents\Rajendra\CSPRNG_STEG>
```

The dependent packages can be installed within this cryptosteg environment:
We can install them in two different ways, either you can run the following commands, one by one on the terminal,

```
pip install pycryptodome  #cryptography package
pip install pillow        # add processing functionalities to the
                            python interpreter
```

or else by using the pre-existing file (requirements.txt) from the submitted code base, which will help in installing the required python packages in one go, for that you have to first unzip the code and copy the whole folder and paster it in the explorer tab in visual studio and then change directory to the location of project and then in the terminal window type,

```
      pip install – r requirements.txt
```

Now, the environment is configured and installed with the necessary packages to run the code.

In order to run the code you can select any images from the image folder as it currently only supports “.png” format.

```
      python csprng-steg.py hide images/cover_0002.png H3ll0RunT1m3
-f data_to_be_encrypted.txt
Where,
      csprng-steg.py: The python file that will do the processing
      hide: The Operation to be performed
      images/cover_0002.png: The image name to be used as cover for
      hiding the actual data.
      H3ll0RunT1m3: Password used for encryption.
      data_to_be_encrypted.txt: Message/data file where the actual
      data or message is stored
```

Once done, you can see a file in the folder generated as “Cover_Steg.png”, which consist of actual embedded data.

For decrypting the message from the image, just type,

```
    python csprng-steg.py show Cover_steg.png H3ll0RunT1m3 -o
decrypted_data.txt

Where,
    show: The operation to be performed
    Cover_Steg.png: The image which is been used to hide data.
    H3ll0RunT1m3: Password for decrypting the image
    decrypted_data.txt: Decrypted text message from the image.
```

References

Holman, M.A., Martínez, F. and Edwar, J.G., 2022. Implementation of Password Hashing on Embedded Systems with Cryptographic Acceleration Unit. *International Journal of Advanced Computer Science and Applications, 13*(2).

Secrets — Generate secure random numbers for managing secrets. Available at: https://docs.python.org/3/library/secrets.html (Accessed: 12 August 2023)