Configuration Manual

MSc Research Project
MSc Cybersecurity

Aryan Ingale
Student ID: x22178511

School of Computing
National College of Ireland

Supervisor: Rohit Verma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Aryan Sahebrao Ingale		
Student ID:	x22178511		
Programme:	MSc in Cybersecurity	**Year:**	2023-24
Module:	MSc Research Project		
Lecturer:	Rohit Verma		
Submission Due Date:	15th December 2023		
Project Title:	HexaCha: A Lightweight Hybrid Encryption Model for Password and Message Protection		
Word Count:	1328	**Page Count:**	10

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: Aryan Sahebrao Ingale

Date: 14th December 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):	

Table of Contents

Table of Figures

Configuration Manual

Aryan Ingale
x22178511

1 Introduction

1.1 Purpose of the Document

The research project namely "*HexaCha: A Lightweight Hybrid Encryption Model for Password and Message Protection*" is a hybrid encryption between Chacha20 and Honey Encryption algorithm, The purpose of this document is to provide an in-depth explanation to get the flask server and python application up and running on a fresh system, this document will go through the minimum requirements, steps to install and run the application, how to interpret application's output and finally a conclusion of all the key points discussed in every section.

1.2 Document Structure

The following structure will be followed in for this document to fulfill all the configuration and deployment requirements of HexaCha.

Title	**Description**
General Requirements and Information	This section will discuss the objectives of this research along with the general and minimum requirements needed by any user to setup and deploy the application on a new system.
Deployment and Interpretation	This section will discuss the procedure to execute the application, give inputs and interpret respective outputs given by the application.

2 General Requirements and Information

2.1 Objective of Research

The objective of the research was to create a lightweight, efficient, and economical hybrid encryption model comprising of Honey and ChaCha20 Encryption and compare it with an industry leading standard, in our case AES and Honey Encryption Hybrid, to validate the findings on which model would be best suited for low resource environments.

2.2 Requirements

This section will shed light on the system and software requirements needed by HexaCha, following table shows the minimum system requirements.

System Requirements	
Operating System	Windows 10 or greater
Minimum RAM	1gb or more
System Type	64-bit operating system, x64-based processor

2.2.1 Visual Studio Code

Initially we need to download and install the latest version of Visual Studio Code (VSCode) from the official Microsoft website [2]. Once the installation is completed we will need to get Python 3.0 [1] extension in it which can be done by the following steps:

- Step 1:

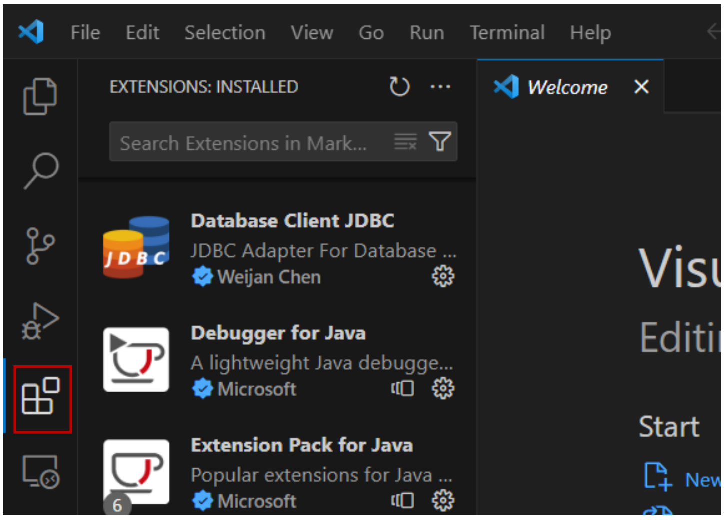

File Edit Selection View Go Run Terminal Help
EXTENSIONS: INSTALLED
Search Extensions in Mark...
Database Client JDBC
JDBC Adapter For Database ...
Weijan Chen
Debugger for Java
A lightweight Java debugge...
Microsoft
Extension Pack for Java
Popular extensions for Java ...
Microsoft
6
Welcome
Visu
Editi
Start
New

Figure 1: Extensions option

Once VSCode is running, located on the left side, is the extensions option as shown in Figure 1: Extensions option (highlighted on left side) and in the search box enter "*Python*" and install the extension as shown in Figure 2.

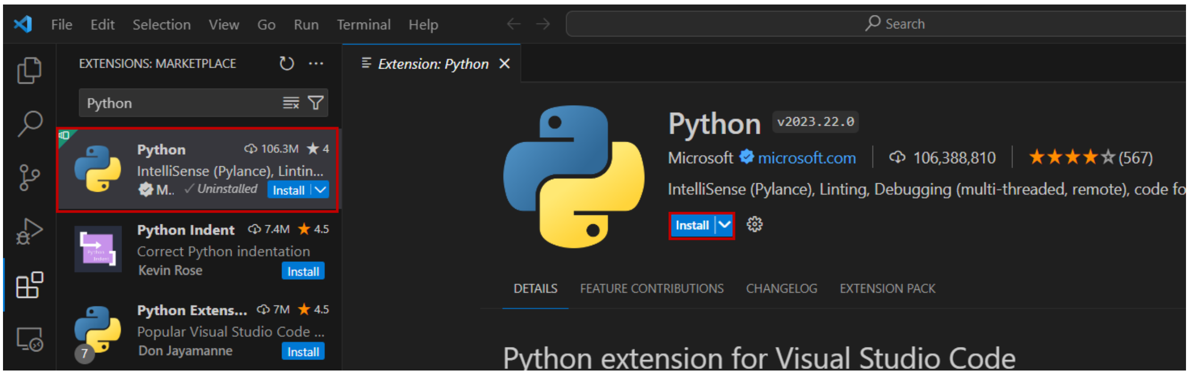

File Edit Selection View Go Run Terminal Help
Search
EXTENSIONS: MARKETPLACE
Python
Python 106.3M ★ 4
IntelliSense (Pylance), Lintin...
M.. ✓ Uninstalled Install
Python Indent 7.4M ★ 4.5
Correct Python indentation
Kevin Rose Install
Python Extens... 7M ★ 4.5
Popular Visual Studio Code ...
Don Jayamanne Install
Extension: Python
Python v2023.22.0
Microsoft microsoft.com 106,388,810 ★★★★☆ (567)
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), code fo
Install
DETAILS FEATURE CONTRIBUTIONS CHANGELOG EXTENSION PACK
Python extension for Visual Studio Code

Figure 2 : Python Extension

- <u>Step 2:</u>

Now import the project file in VSCode for which we will extract the zip folder and follow the steps '*File -> Open Folder -> Select the **Application Code** folder*' and the folder will be visible in '*Explorer*' window.

- <u>Step 3:</u>

Once the folder is imported in VSCode, we need to install all the essential libraries which are imported in the primary python script, this can be done by browsing into the flaskapp folder as shown in Figure 3.

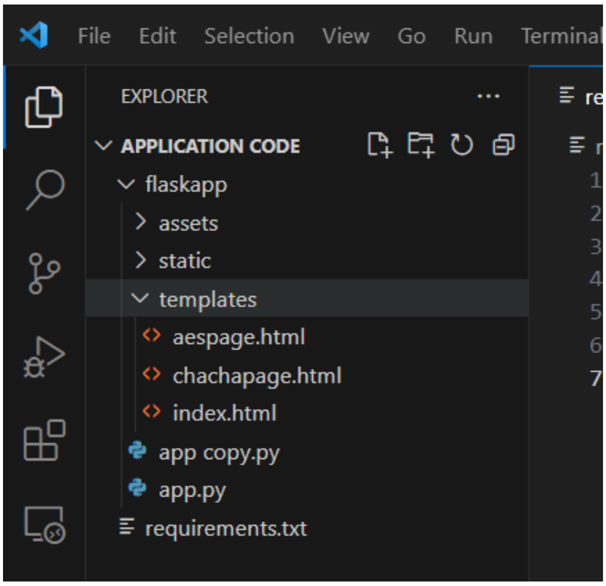
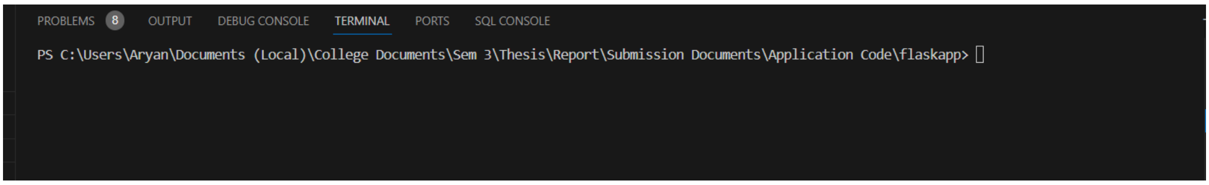


Figure 3 : Flaskapp Folder

Once in the folder right click on '*flaskapp*' option of Explorer and select '*Open in integrated terminal*' which opens the folder in a terminal as shown in Figure 4.



```
PS C:\Users\Aryan\Documents (Local)\College Documents\Sem 3\Thesis\Report\Submission Documents\Application Code\flaskapp>
```

Figure 4 : Folder in Integrated Terminal

In this terminal run the following command to install all the required libraries:

```
pip install -r ./requirements.txt
```

Once this is all done the system is ready to run the flask server and HexaCha.

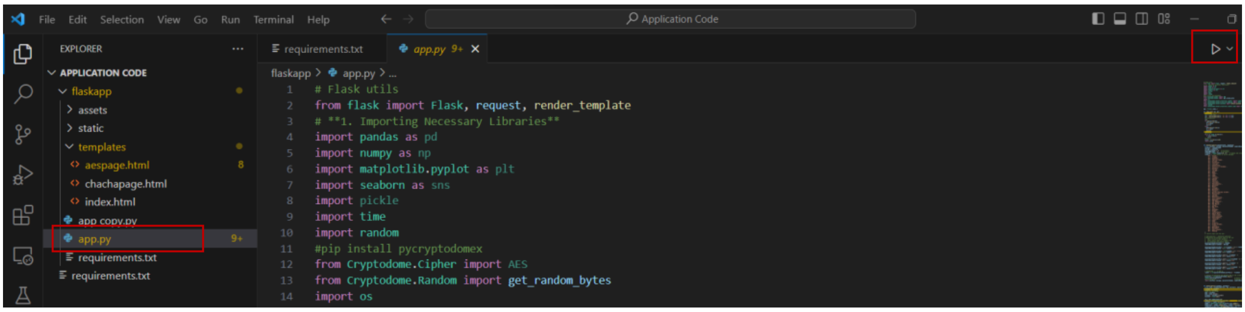
3 Deployment and Interpretation

This section will explain how to get the server running and interpretation of application output.

3.1 Deployment

The application files and libraries were successfully imported in the system now to run and get outputs from HexaCha we will need to follow the below steps:

In Explorer, select the ‘*app.py*’ file and either select the play button as highlighted in Figure 5 or use “*Ctrl + F5*” shortcut while the python script is selected in Explorer.

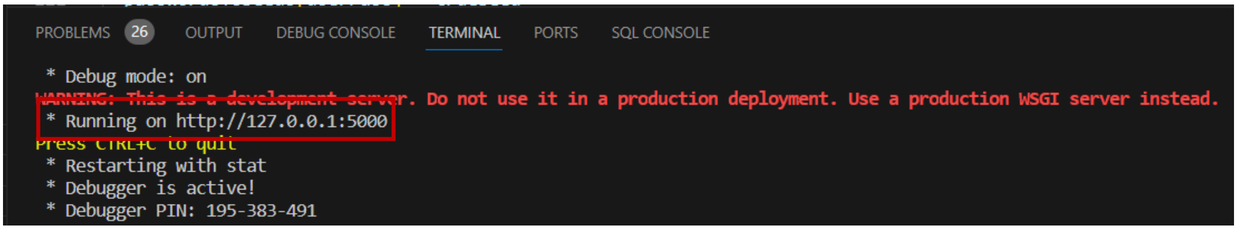


```python
# Flask utils
from flask import Flask, request, render_template
# **1. Importing Necessary Libraries**
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import time
import random
#pip install pycryptodomex
from Cryptodome.Cipher import AES
from Cryptodome.Random import get_random_bytes
import os
```

Figure 5 : Code Execution

The script will start the server on localhost port 5000, of which the output is displayed in the terminal (Figure 6).

Figure 6 : Terminal Output



```
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 195-383-491
```

3.2 I/O Operations and Interpretation

Following a successful deployment this section will continue with the I/O operations and discuss how to understand the algorithm results.

- Step 1:

Opening any local browser and entering the localhost URL ‘*http://127.0.0.1:5000*’ will open the application interface (Figure 7). It consists of two options one for HexaCha and one for AES both of which have their independent hybrid algorithms.

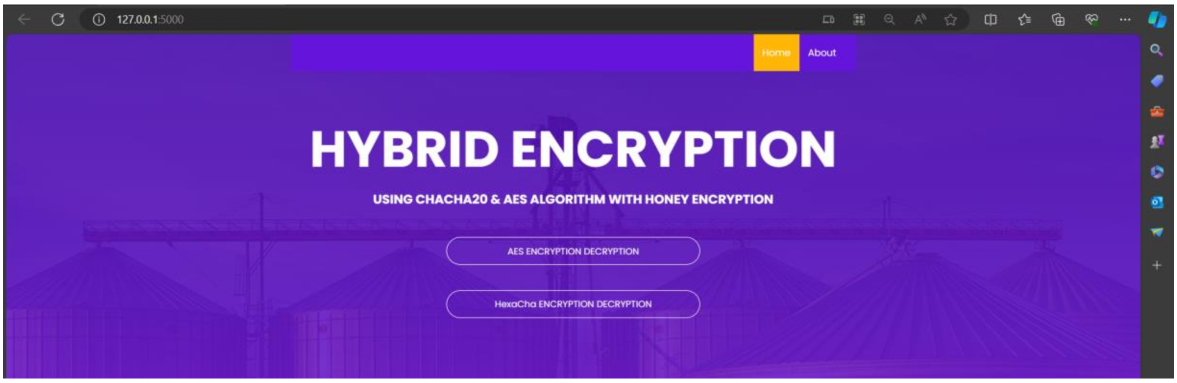
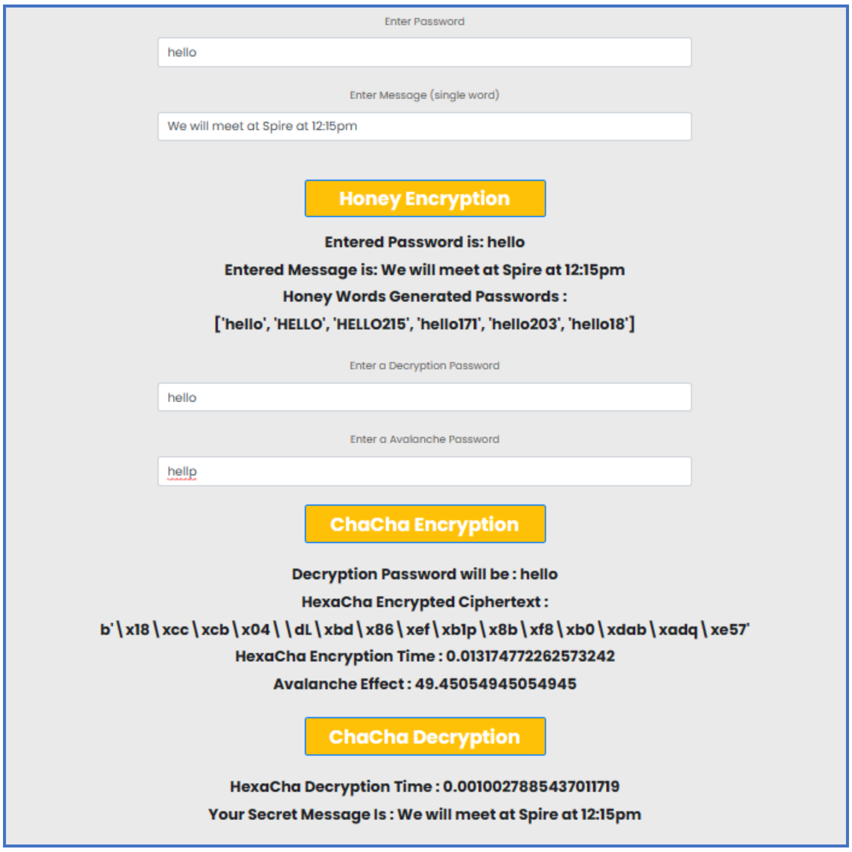

127.0.0.1:5000
Home
About
HYBRID ENCRYPTION
USING CHACHA20 & AES ALGORITHM WITH HONEY ENCRYPTION
AES ENCRYPTION DECRYPTION
HexaCha ENCRYPTION DECRYPTION

Figure 7 : Home Page of the Application

- Step 2:



Enter Password

hello

Enter Message (single word)

We will meet at Spire at 12:15pm

Entered Password is: hello

Entered Message is: We will meet at Spire at 12:15pm

Honey Words Generated Passwords :

['hello', 'HELLO', 'HELLO215', 'hello171', 'hello203', 'hello18']

Enter a Decryption Password

hello

Enter a Avalanche Password

hellp

Decryption Password will be : hello

HexaCha Encrypted Ciphertext :

```
b'\x18\xcc\xcb\x04\ \dL\xbd\x86\xef\xb1p\x8b\xf8\xb0\xdab\xadq\xe57'
```

HexaCha Encryption Time : 0.013174772262573242

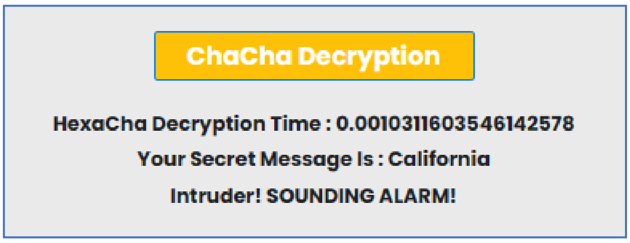
Avalanche Effect : 49.45054945054945

HexaCha Decryption Time : 0.0010027885437011719

Your Secret Message Is : We will meet at Spire at 12:15pm

Figure 8 : Input and Output of HexaCha

The above figure shows the working of HexaCha Algorithm with different user inputs, as depicted the snapshot, initially the user will input a password which will be used by ChaCha20 for encryption and a message which will be mapped to the original password and encrypted with the password using ChaCha20, then user clicks on ‘*Honey Encryption*’ button which generates a dictionary of honey words and maps them with their fake messages and stores them in the system. If the user wants to decrypt and get the message, they will enter a decryption password along with an avalanche password which is used to calculate avalanche effect. If the decryption password matches the encryption password, then the original message is displayed which in our case is “*We will meet at Spire at 12:15pm*”. Similarly, if the decryption password matches to any of the honeywords then a fake message is generated by the system (Figure 9).



HexaCha Decryption Time : 0.0010311603546142578
Your Secret Message Is : California
Intruder! SOUNDING ALARM!

Figure 9 : Fake message displayed.

Similar mechanisms have been implemented for both AES and HexaCha, also the systems give out additional details such as encryption time, decryption time and avalanche effect calculations for all the relevant operations for comparative evaluation.

4 References

[1] Microsoft, "Python - Visual Studio Marketplace," [Online]. Available: https://marketplace.visualstudio.com/items?itemName=ms-python.python. [Accessed 01 December 2023].

[2] Microsoft, “Visual Studio Code - Code Editing. Redefined,” [Online]. Available: https://code.visualstudio.com/. [Accessed 01 December 2023].