National College of Ireland

# Configuration Manual

MSc Research Project
MSc Cybersecurity Top-up

Gary Crowe
Student ID: 23182652

School of Computing
National College of Ireland

Supervisor: Mark Monaghan

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Gary Crowe | | |
| **Student ID:** | 23182652 | | |
| **Programme:** | MSc. Cybersecurity Top-up | **Year:** | 1st Year |
| **Module:** | MSc (Research Practicum Part 2) | | |
| **Lecturer:** | Mark Monaghan | | |
| **Submission Due Date:** | 11/08/2025 | | |
| **Project Title:** | Configuration Manual | | |
| **Word Count:** | 979 | **Page Count:** | 7 |

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:** | Gary Crowe |
| **Date:** | 06/08/2025 |

**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

Gary Crowe
Student ID: 23182652

## 1 Introduction

The following is to show a potential reader the environment they must set up to run all the code used in the creation of the RSAnalyzer and the associated tests.

This document will go over what needs to be installed to successfully run the code and how to use RSAnalyzer.

## 2 Environment Set-up

This section details the tools used to create RSAnalyzer. Note this is not a guide how to install these tools as that information can be easily found elsewhere. Table 2.1 shows all tools used in this project.

| Tool Name | Version | Purpose |
|---|---|---|
| Python 3 | 3.11.7 | The version of Python that most of the code for this project was written in. |
| Anaconda | 24.11.3 | Needed for extra libraries |
| Jupyter-notebook | 7.0.8 | Required to read/run the files the models were trained in. |
| TensorFlow | 2.10.1 | Some of the regression models were trained using tensorflow. This requires Python 3.9.21 to run. |
| VSCode | N/A | Any version of any IDE will do here as long as you can run the code. VSCode is just what I used for this project and it is what some of the example screenshots are taken from. |
| pickle | 4.0 | For saving and loading the machine learning models |
| Scikit-learn | 1.6.1 | For saving and loading the machine learning models |

# 3 Using RSAnalyzer

RSAnalyzer is the main artefact of this project, so it will require some explanation of how to use it.

## 3.1 Running RSAnalyzer

Simply run *RSAnalyzer.py* in the IDE of your choice.


**Figure 3.1.1: RSAnalyzer - Main Menu**

From Figure 3.1.1 you have the following options:

- **Analyze** – Have the program cryptanalyze and break your ciphertext – Input "a".
- **Encrypt** – Input a word with 7 letters of less and get a ciphertext and a public to use in the Analyze feature – Input "e".
- **Quit** – Exit the program – Input "q".
- **Help –** Gives a description of what RSAnalyzer is – Input "h".

## 3.2 Running RSAnalyzer – Encrypt

To have something to analyse, from the welcome screen input "e".

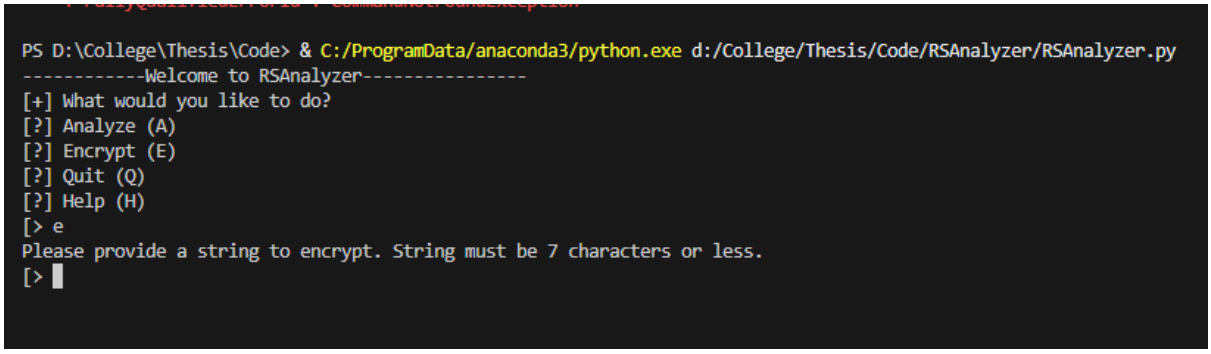


**Figure 3.2.1 - Encrypt Feature**

You are then prompted to enter a word with 7 characters or less (Figure 3.2.1), Sticking with English words is recommended but if the user sticks to ASCII characters, you are free to input whatever you like. For the purposes of this document the word "candle" will be used. Input your chosen word as depicted in Figure 3.2.2

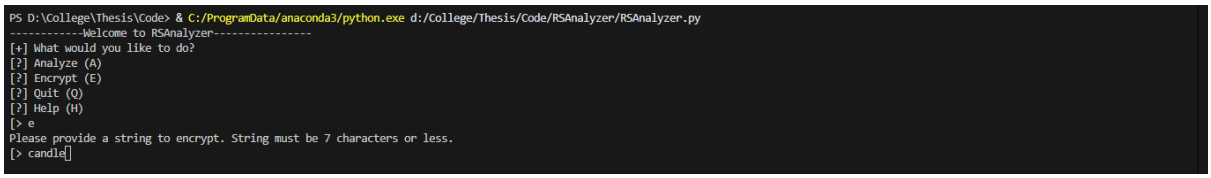


**Figure 3.2.2 - Encrypt Feature - Input**

The word will then be encrypted with a random weakness and returned to the screen along with the corresponding public key as depicted in Figure 3.2.3. Note that this may take a few seconds but if the process has not been completed in 3-5 minutes at most, simply stop the program and run again. Or you could wait if you want to but there is no telling how long this will take as the numbers used in the encryption are randomized as well.

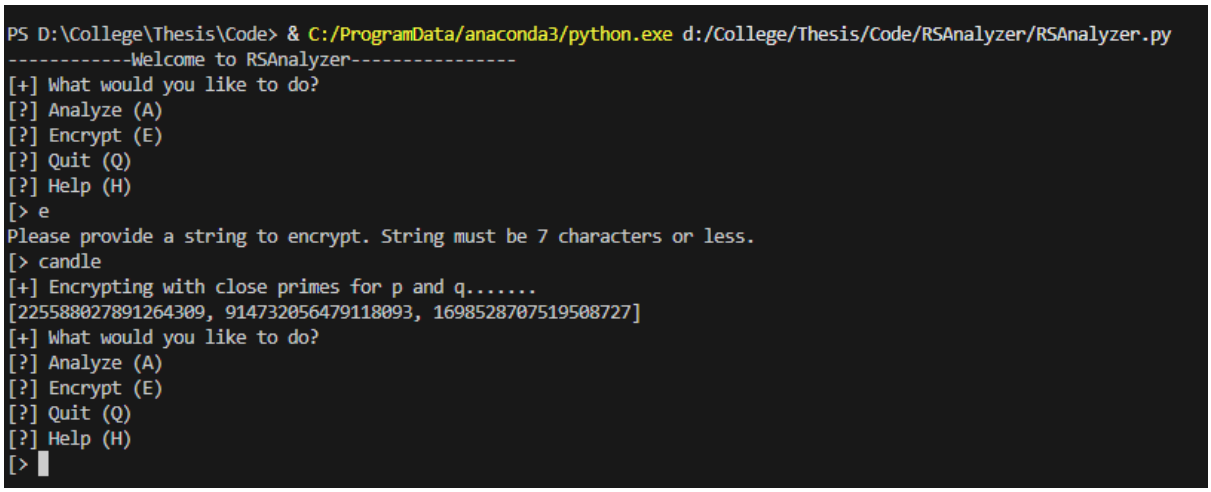


**Figure 3.2.3 - Encrypt Feature – Output**

As can be seen in Figure 3.2.3, the output comes in 3 parts:

1. The ciphertext
2. The exponent of the public key
3. The modulus of the public key

The user is also told what weakness the ciphertext has. This is to identify if the RSAnalyzer classifies the correct weakness or not.

Now the user has an input to analyze!

## 3.3 Running RSAnalyzer - Analyze

As can be seen in Figure 3.2.3, after the encryption process is complete the user is brought back to the main screen. From here the user can input “A”/”a” to analyze a ciphertext.

From here the user can input a ciphertext, exponent and modulus (in that order) of their choice. Then the user can choose between 4 models to analyze the ciphertext with:

1. Decision Tree – input “dt”.
2. Random Forest – input "rf”.
3. Support Vector Machine – input “svm”.
4. Neural Network – input “nn”.

RSAnalyzer then attempts to break the ciphertext as depicted in Figure 3.3.1 and Figure 3.3.2

```
PS D:\College\Thesis\Code> & C:/ProgramData/anaconda3/python.exe d:/College/Thesis/Code/RSAnalyzer/RSAnalyzer.py
-----------Welcome to RSAnalyzer---------------
[+] What would you like to do?
[?] Analyze (A)
[?] Encrypt (E)
[?] Quit (Q)
[?] Help (H)
[> a
[+] Please provide input in the following order Ciphertext e n:
[> 225588027891264309 914732056479118093 1698528707519508727
[+] Select a machine learning model to analyse with
[?] Decision Tree (DT)
[?] Random Forest (RF)
[?] Support Vector Machine (SVM)
[?] Neural Network (NN)
[> dt
```

**Figure 3.3.1 - Analysis Feature - Input**

(If copy and pasting remember to remove the commas.)

```
PS D:\College\Thesis\Code> & C:/ProgramData/anaconda3/python.exe d:/College/Thesis/Code/RSAnalyzer/RSAnalyzer.py
-----------Welcome to RSAnalyzer---------------
[+] What would you like to do?
[?] Analyze (A)
[?] Encrypt (E)
[?] Quit (Q)
[?] Help (H)
[> a
[+] Please provide input in the following order Ciphertext e n:
[> 225588027891264309 914732056479118093 1698528707519508727
[+] Select a machine learning model to analyse with
[?] Decision Tree (DT)
[?] Random Forest (RF)
[?] Support Vector Machine (SVM)
[?] Neural Network (NN)
[> dt
[+] Loading model
[+] Model Loaded!
[+] 225588027891264309 914732056479118093 1698528707519508727
[+] Performing cryptanalysis now.....
[+] N was constructed with close prime numbers for p and q.
[+] Using Fermat's factorisation on 1698528707519508727.
[+] Factors of 1698528707519508727 found!
[+] 1303276141 and 1303276147
[+] Calculating the Euler Toitent Function of n
[+] Phi of N is: 1698528704912956440
[+] Calculating d.....
[+] Decrypting Cipertext...
[+] 225588027891264309^38929823682006757 mod 1698528707519508727
[+] Decrypted message is: 109270115052645
[+] The Original Message is: candle
[?] Would you like to continue using RSAnalyzer? (Y/N)
[>
```

**Figure 3.3.2 - Analysis Feature – Output**

RSAnalyzer classifies the ciphertext and attempts to obtain the original plaintext exploit the weakness it thinks the ciphertext has. RSAnalyzer then gives a detailed breakdown of how it arrived at the plaintext.

The user can then decide to continue to use RSAnalyzer, in which case they will be brought back to the main screen or to exit the program.