National College of Ireland

Configuration Manual

MSc Research Project
MSc Cyber Security

Neha Singh
Student ID: X22132759

School of Computing
National College of Ireland

Supervisor: Evgeniia Jayasekera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Neha Singh ……

Student ID: X22132759……

Programme: MSc Cyber Security …… **Year:** 1……..

Module: Academic Internship ……

Supervisior: Evgeniia Jayasekera……

Submission Due Date: 14 December 2023 ……

Project Title: A Novel approach to Near Field Communication using B92 Quantum Key Distribution protocol

Word Count: 754…… **Page Count:** 5……

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: ……

Date: ……

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

Neha Singh
Student ID: X22132759

1 Introduction

This configuration manual offers comprehensive instructions on setting up and executing the proposed project. It bridges the gap between theoretical research and practical application, enabling readers to replicate and understand the practical aspects of the research. This research showcases using the B92 Quantum Key Distribution (QKD) protocol to generate a secure, shared key. It detects any intrusion or eavesdropping in the key generation phase and secures the interception of actual communication. PyCharm IDE is used to execute the Python code.

2 System Configuration

The required system configuration is highlighted in this section.

2.1. Hardware Configuration

Operating System: - Mac OS Sonoma 14.1.2
Processor: 1.4 GHz Quad-Core Intel Core i5
System: 64 bits
Hard drive: 256 GB
Memory (RAM): 8GB

2.2. Software Configuration

Table 1: Software Configuration

Tool	Version	Description
Python 3	3.9.6	Python3 was used to create the complete simulation because of its ease of use and efficiency in managing user interactions and random processes.
PyCharm	2023.3 (Community Edition)	PyCharm was used for the script's development and debugging. Its extensive feature set considerably aided the development process, which includes an integrated debugger, an intelligent code editor, and error highlighting.

3 Execution

This section describes the working, software installation and execution of the proposed system.

3.1. Software Installation

- Python 3.9.6 was used as it was a stable version with proper security updates. Link to download Python (Python Software Foundation, 2021). https://www.python.org/downloads/release/python-396/

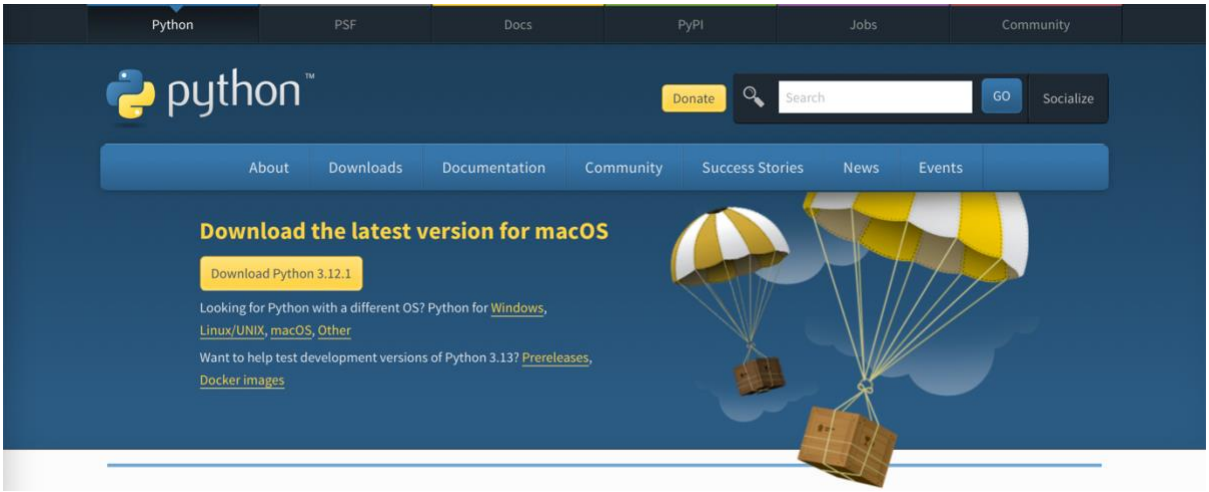

Figure 1: Download Python

- PyCharm IDE 2023.3 community version was used because it is free and has all the required functionalities to execute this project. Link to download the IDE (scroll to the bottom to download the community version (Jet Brains, n.d.). https://www.jetbrains.com/pycharm/download/?section=mac

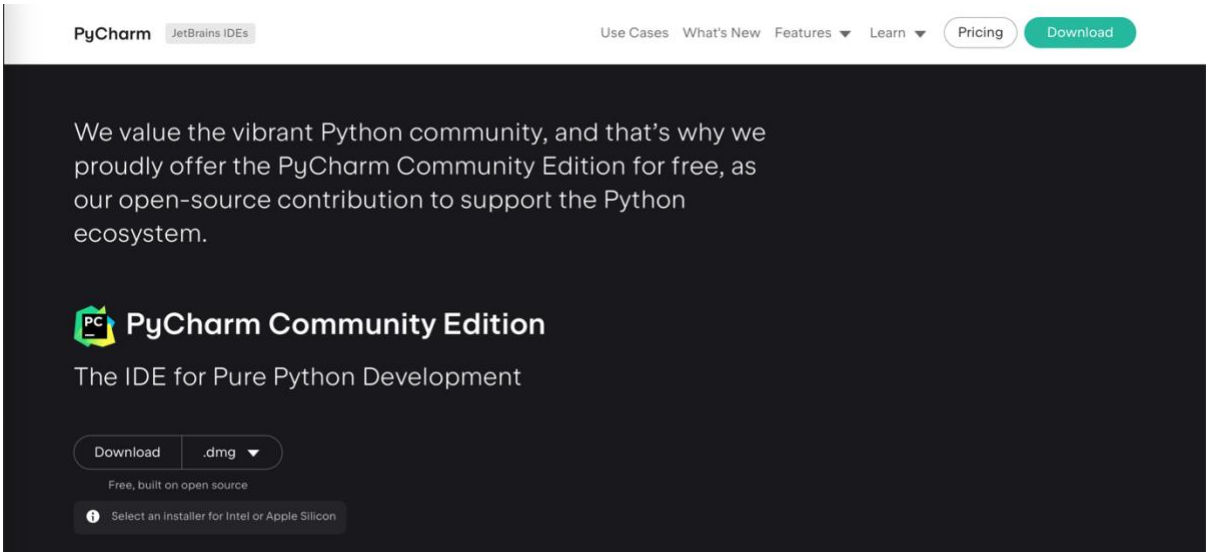

Figure 2: Download PyCharm

3.2. Program Execution

After successfully installing Python and PyCharm, Open the Project in PyCharm.

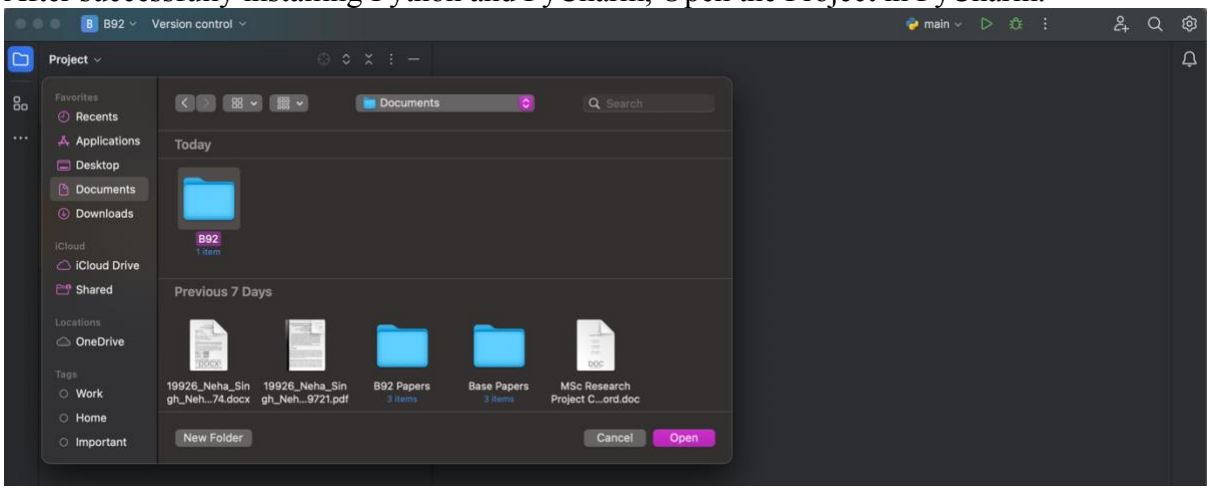

Figure 3: Open the Project in PyCharm

Once the project is open in PyCharm, open the main.py file.

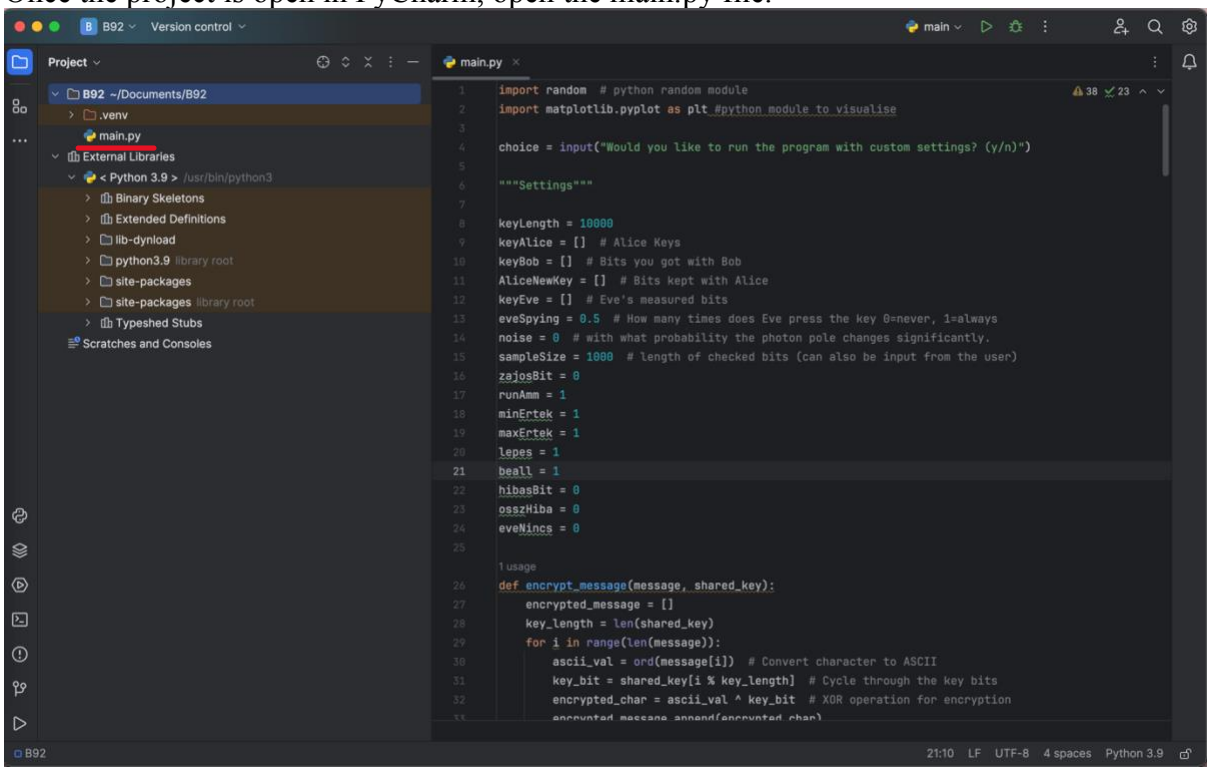

Figure 4: Open the main.py file

Install any required packages.

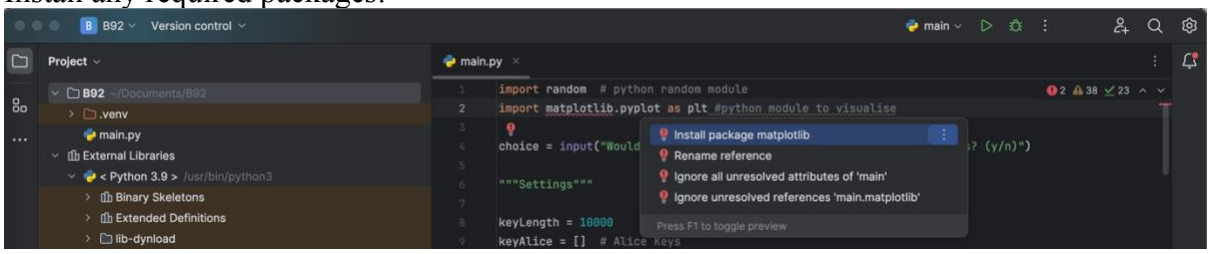

Figure 5: Install required packages

After the required packages are installed, the program is ready to execute. Once we run the code, we get asked if we want to provide custom settings or initialise with the default settings.

```
/usr/bin/python3 /Users/nehstruck/Documents/B92/main.py
Would you like to run the program with custom settings? (y/n)
```

Figure 6: Choice for custom settings

If we select ‘n’, the code initialises the default values; on selecting ‘y’, we need to input custom values. Here, we chose ‘y’ and are providing custom values. Here, we provided the key length, Eavesdropping percentage, noise level and key length. We are taking the best-case scenario with no eavesdropping and noise.

```
Would you like to run the program with custom settings? (y/n)y
Key length:100
How many times does Eve listen to the message as a percentage (0-1):0
Noise level (0-1):0
Number of bits to check (recommended: <key length/10):10
```

Figure 7: Input custom values

According to the B92 QKD protocol, the code first generates Alice’s Key and creates a subset of bits from Alice's original key that are kept after a process of key sifting called the Alice retained bits. It is held separately from the public shared bits to generate the secret shared key.

```
Alice's generated key:
[0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1,
Alice's retained bits:
[0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0]
Length of Alice's key: 31
```

Figure 8: Alice's keys

Alice sends the keys to Bob, who measures them.

```
Bob's measured bits
[0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0]
length of Bob's key: 31
```

Figure 9: Bob's measurements

Since we did not initiate eavesdropping, Eve does not measure the bits.

```
Eve's measured bits:
[]
length of Eve's key: 0
```

Figure 10: Eve's measurements

Finally, Alice and Bob generate the same secret shared key used for Encryption and Decryption.

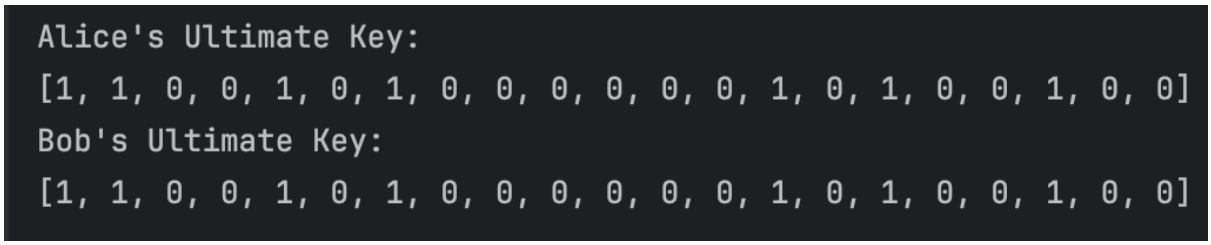

Figure 11: Alice and Bob shared key

The output provides the errors. Since we did not initiate any noise or eavesdropping, the expected errors are 0, and we can conclude that no eavesdropping was attempted.

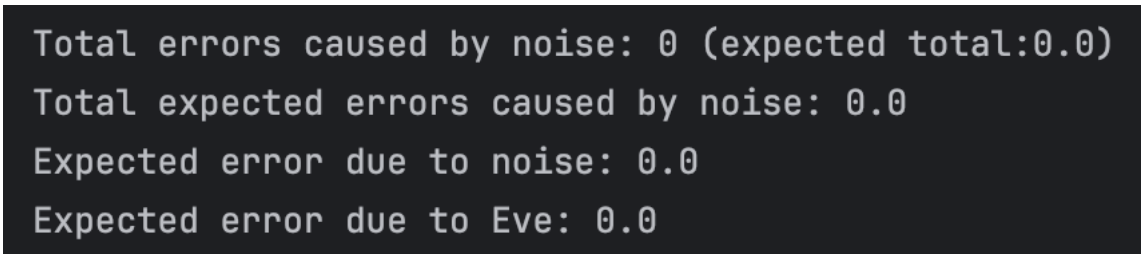

Figure 12: Noise and Eavesdropping Errors

Once the secret shared key is generated, it can encrypt and decrypt the data. We also get the Quantum Bit Error Rate (QBER).

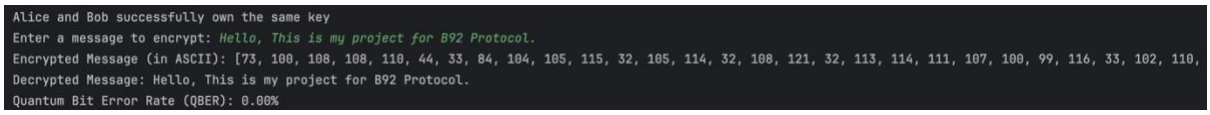

Figure 13: Encryption and Decryption

4 References

Jet Brains, n.d. *Getting started.* [Online]
Available at: https://www.jetbrains.com/help/pycharm/getting-started.html
[Accessed 14 Dec 2023].

Python Software Foundation, 2021. *Python 3.9.6 documentation.* [Online]
Available at: https://docs.python.org/release/3.9.6/
[Accessed 14 Dec 2023].