**B. TECH CYBER SECURITY**

**OBJECT ORIENTED PROGRAMMING WITH PYTHON BCB 101**

**GROUP 2**

**BACKGROUND**

Technology today is not what it was a few decades ago, today we all have become accustomed to the Technology around us, we depend heavily on it, because It serves it purpose, which is to make our lives easier, and one of the most impactful thing that Technology has brought about is the Internet, which makes all of us connected in a way which seemed impossible a few decades ago, with this most growing part of connectivity is communication and the most efficient way of doing it so through a text message which can be done in many ways as of today, some including

* Email
* SMS (Short Message Service) provided by ISP’s
* MMS (Multimedia Message Services)
* Third Party social media, etc.

Most of these services already provide encryption but to the connection and not the data that is passed

hence it makes it difficult to get into the connection but that does mean it doesn’t happen or can't. Hence my team has taken on the task to bring a solution to this issue

**PROBLEM STATEMENT**

Because of the rapidly growing cyber world today due to the mass use of technologies and the internet,

people have started doing malicious things with whatever information they can get, which is not good because we embed our identity to our technologies and also over the internet, hence we share sensitive information when communicating over the internet, which is “a big problem” when someone malicious intents get hold of the information we share, they can “pretend to be you” on the internet and nobody would know, because verification on the internet is what information you have. Since Text messages are transported in readable text, over a secured connection all one would have to do is to bypass the

security of the end-to-end connection made and they would be able to gain access to said information

**OBJECTIVE**

As cybersecurity students my team have tasked ourselves to come out with an application that improves upon the existing messaging system. The approach we took was to apply encryption to everything at the individual level. This will be done such that the message to be sent will be encrypted with an algorithm in such a way that it wouldn't be human readable before passing it over the encrypted network, and can only be decrypted with a secret key Hence the encryption will happen on both levels.

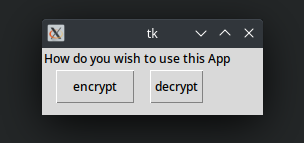
**DESIGN AND IMPLEMENTATION**

For this task we could have used if not any general-purpose language but we decided to use Python because it's much easier to maintain an develop with. The GUI Library used is “Tkinter” as it comes with python out of the box (usually but not always) and doesn’t need addition installation and requirements to start developing with.

Now to get to the main point of this application which is the encryption, my Team decided to go with it this way

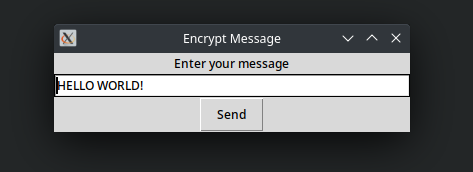
* the message to be sent will be encrypted before sending
* the encryption algorithm will swap **each** letter with the letter beside it. An example of this would be the word “HELLO WORLD!”, we will first swap the first two letters: “’EH’LLO WORLD!” then the next ones, which does not include the already swapped on:”’EHLL’O WORLD!” and so on until It reaches the end: “EHLL OOWLR!D”
* the decryption will just be the reverse of the encryption

As for the design, the first thing that shows up is a main window with a label of “How do you wish to use this app” which is asking you if you want to encrypt or decrypt a message with two buttons with each having encrypt and decrypt respectively e.g. (Fig 1).



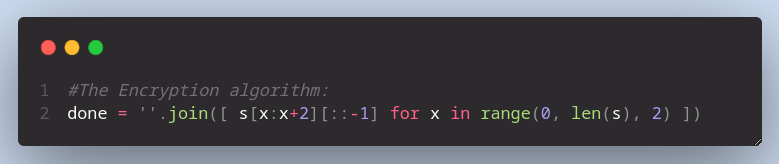
[Fig 1]

If you pick either, another window will pop up with the label “Enter your message” with a text box below, e.g. (Fig 2) in which you can Enter the message you want to encrypt or decrypt and below it Is button with “send” on it and when you click the button, it will take the words from the text box and pass it through the algorithm and depending on what you picked, it will decrypt or encrypt it then display the encrypted message or decrypted message



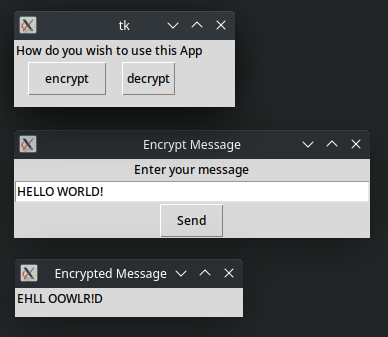
[Fig 2]

As for the source code, there are many parts, but the algorithm used is a one liner code which is due to how robust python is: (Fig 3)

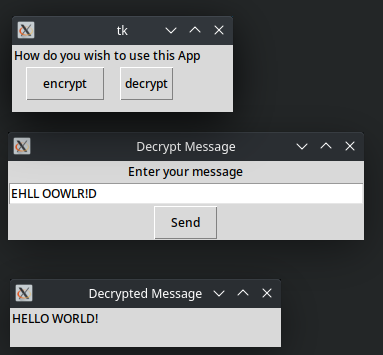
 [Fig 3]

With just this code, it will swap each pair of letters in the string(message) which is how our encryption Is done

Below is an example of an encryption (fig 4) and decryption (fig 5)



[ Fig 4]



[ Fig 5]