Research and Analysis on Enigma Cipher Machine

A PROJECT REPORT

Submitted by

AKSHAT RAWAT (20BCY10036)

KUNAL DHINGRA (20BCY10018)

SAKSHAM GUPTA (20BCY10088)

SHASHWAT CHANDRA (20BCY10023)

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

WITH SPECIALIZATION IN

CYBERSECURITY AND DIGITAL FORENSICS

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

APRIL 2022

BONAFIDE CERTIFICATE

Certified that this project report titled **"Research and Analysis on Enigma Cipher Machine"** is the bonafide work of "**AKSHAT RAWAT (20BCY10036), KUNAL DHINGRA (20BCY10018), SAKSHAM GUPTA(20BCY10088), SHASHWAT CHANDRA(20BCY10023)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR	**PROJECT GUIDE**
Dr. R. Rakesh Senior Assistant professor School of Computing Science and Engineering VIT BHOPAL UNIVERSITY	Dr. S. Rajasoundaran Assistant professor School of Computing Science and Engineering VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on _______________

ACKNOWLEDGEMENT

First, we would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

We wish to express our heartfelt gratitude to Dr R.Rakesh, Program Chair, Division of Cyber Security and School of Computing Science for much of his valuable support and encouragement in carrying out this work.

We would like to thank our project guide Dr .S Rajasoundaran for continually guiding and actively participating in our project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

ABSTRACT

The Enigma machine is a cipher device. Used in the early- to mid-20th century .It was employed extensively by Nazi Germany during World War II, in all branches of the German military. The Enigma machine was considered so secure that it was used to encipher even the most top-secret messages. In spite of a relatively large number of publications about breaking Enigma by the Allies before and during World War II, this subject remains relatively unknown not only to the general public, but also to people professionally involved in cryptologic research.

TABLE OF CONTENTS

CHAPTER 1:

1.1 INTRODUCTION

The military Enigma machine was a portable electro-mechanical rotor encrypting machine used during the Second World War mainly by the German military and government services.

1.2 MOTIVATION FOR THE WORK

Having a very keen interest in the subject of Cyber Security and Digital Forensics, our main objective is to research **Enigma Cipher-Decipher Machine** and implement the knowledge gained into a modern day working model that will use modern day programming languages like Python and Java to run the same algorithm on a computer

Ultimately, we would be analyzing the time required to break the cipher text of an Enigma Machine on a present day computer

1.3 About Introduction to the project including techniques

In the project, certain papers from different authors have been used to research more on the topic.

1.4 PROBLEM STATEMENT

- Research papers available
`

- Commercial, diplomatic, and military communication of Nazi were exposed.

1.5 OBJECTIVE OF THE WORK

To research the Enigma Machine. To find out how the Nazi’s used this machine to encipher even the highest miltary messages and codes. To also implement the knowledge gained by this research in the form of a virtual machine made using python language.

1.6 ORGANIZATION OF THE PROJECT

The research was divided into several other researches already conducted by different authors. After sufficient study the best ones were selected for analysis and comparison.

1.7 SUMMARY

This work aims to provide in-depth research on the enigma cipher machine which was famously used by Nazi German forces during WWII.

CHAPTER 2:

Inside the Enigma Machine

By Shannon Riffe, Carnegie Mellon University

Methodology and Conclusion

- The underlying principle of an Enigma machine cipher is that of **letter substitution**, meaning that each letter of our plaintext (undeciphered message) is substituted by another letter.

- The machine's advantage include the fact that the cyphers it employs are based on random substitution, making them unpredictable and formula-free.
- On the other hand, messages were often found with errors due to the inaccuracy of operators which proved to be its main disadvantage.

The Mathematics of the Enigma Machine

By Manchester Metropolitan University

Overview and Conclusion:-

The Bombe (Group Theory) was a machine that could be used to find the key to an Enigma cipher.
It was modeled on the Polish machine, Bomba, which could also do the same before Enigma was enhanced.

Group theory worked by finding a fingerprint (pattern) given by the settings. Using the crib the operator would make a list of what the letters in the plaintext encrypted to in the ciphertext to , this could then be used to find the fingerprint.

- **Advantages:**

The machine's advantage over other machines is that its ciphers are random substitution ciphers, meaning they are formula-free and difficult to crack.

- **Disadvantage:**

The primary issue of the machine was that it never encrypted a letter to itself, causing cribs to be used.

Alan Turing, Enigma, and the Breaking of German Machine Ciphers in WWII

By Lee A.Gladwin

Alan Mathison Turing, a mathematician, first proposed the idea of building a universal machine capable of mimicking any other machine in his seminal article "On Computable Numbers" (1936). He insisted that "whatever a human computer [i.e., a person] does

Those who dealt with numbers could be replaced by a machine "!!

Radio communications were used to coordinate and direct Germany's rapidly advancing army, but these Morse code signals could not be relayed in clear, unencrypted text. These intimate conversations between generals and armies, or admirals and fleets, could be overheard by the adversary. It was necessary to use code.

The Enigma code was based on the
Encoding letters using the rotor principle. It
consisted of three rotors, each roughly four inches in diameter.
and a half inches in diameter, with twenty-six letters positioned randomly around its circumference. There were, in turn, twenty-six letters arranged randomly around its circle.
merely the appropriate electrical connections
beneath the letters There were three rotors installed.

On a steel rod, inside the Enigma. In the event that a
The first rotor was turned when the typewriter key was pushed.
As a fresh contact was made, the circuit was changed, and a letter on the lampboard or screen was illuminated.
Assume the rotor was set at "A"before typing in text and that the text consisted solely of the letter "A" typed repeatedly. On the first occasion, the "A"key might light up the "H" on the lampboard; on a second occasion, "Y" and so on.

The Cryptanalysis Of The Enigma Cipher

Bialystok University of Technology, Poland

The military Enigma machine was a portable electro-mechanical rotor encrypting machine used during the Second World War mainly by the German military and government services. Since then the Polish cryptologists systematically worked on: decoding ciphers, constantly modified manners of generating secret messages and modernized constructions of Enigma machines

The Germans used different kinds of Enigma machines (also commercial), modified their constructions and changed the manner of generating secret messages. For the following research the M3 Enigma machine has been selected

The M3 Enigma machine is a combination of electrical and mechanical subsystems. This machine consists of an alphabetical 26-letter keyboard, a lampboard (set of 26 lights), a set of three rotating, encrypting disks (called drums) placed on a shared shaft, two fixed wheels: an entry wheel and a reflector, a plugboard, a battery, and a turning mechanism (used to turn one, two or three drums after pressing any key).

The Enigma Cipher Machine

~By Gary M. Bateman

For the sake of security, a number of modifications were made in subsequent generations. The basic machine featured a typewriter-like keyboard, was roughly the size of an office typewriter, and was driven by an electric motor that comprised of a number of rotating drums or rotors.

Enigma was the most sophisticated encoding mechanism ever devised by the human mind. The Allied-Axis fight for mastery over one another resulted in the fast technical and scientific development that has occurred in the globe since World War II, progress that we all take for granted and regard as commonplace in the 1980s.

The need for security in high level communications and intelligence work has never diminished, rather, it has magnified. Today many codes and ciphers that are created by different governments of the world are so complex in nature that only a high speed computer would ever have a chance of breaking them.

A review on mathematical strength and analysis of Enigma

By Kalika Prasad and Munesh Kumari

In 1926 the German Army started taking interest in Enigma machines due to its specific and high security strength. In 1927 they built their first prototype, and used it exclusively to test its capacity. The final version for use in the German army was Enigma-I and was ready for deployment by 1932. Initially the Enigma cipher machines were free for use by any organization. But after 1932, the German army changed the rule and thereafter use of any Enigma machine had to be approved by the German Army.
This paper is the extension of same work on ’the cryptographic mathematics of enigma’ .
The total theoretical number of possible Enigma configurations is simply the product of all five variable components which is approximately 3 ×

10114 .
As it is estimated that there are only about 10^{80} atoms in the entire observable universe. No wonder, the German Cryptographer had confidence in their machine!

Sockets selection:
Since we have to choose cables from those 26-cables, so total possible combinations of different sockets are ($^{26}C2_p$) or ($^{26}{}_{2p)}$.

First, insert the first end of the first cable into any of those 2p sockets and then the second end of the first cable, which has total (2p − 1) free sockets available to choose. Next, insert the first end of the second cable and thus the second end of the second cable has (2p − 3) free sockets available to choose. This pattern continuous down up to last one cable, whose second end has only one free socket left to connect

Thus, the total possible number of ways in which those p-cables could have been connected with those 2p sockets are (2p − 1)(2p − 3)(2p − 5)...(3)(1) = (2p − 1)!!
Therefore, the number of different connection which could have been made by an Enigma cipher machine is given by
$^{26}C_{2p} \times (2p-1)!! = 26! /\{(26-2p)! \times p! \times 2\ p\}$

Values of p	Combinations	Values of p	Combinations
0	1	7	1,305,093,289,500
1	325	8	10,767,019,638,375
2	44850	9	53,835,098,191,875
3	3,453,450	10	150,738,274,937,250
4	164,038,875	11	205,552,193,096,250
5	5,019,589,575	12	102,776,096,548,125
6	100,391,791,500	13	7,905,853,580,625

Fig : Possible plugboard Combination

The history of cracking the ENIGMA machine

By - Marie-José Durand-Richard

Overview and Conclusion:-

Breaking the Enigma code was a major challenge during World War II. Its success was supported by the efficiency of the work resulting from the involvement of mathematicians and engineers in the practices of military cryptanalysts, strongly endorsed by the political context. Some mechanical processes had already been introduced in cryptology after World War I. These transformations were then pursued at Bletchley Park, and on an industrial scale, which from then on would definitely become one of the characteristic features of cryptology. Cryptanalytic methods had to draw on mathematics and logic to overcome the considerable increase in the number of encryption combinations to consider after the mechanization of Enigma. The success of Bletchley Park continued the first efforts by the Poles on the military Enigma machines, and of Knox's team on the commercial machines. The theory of permutations mobilized by Rejewski, as well as the probabilities and logic mobilized by Turing, could not have been efficient without their respective Bombes.

CHAPTER 3: REQUIREMENT ARTIFACTS

3.1 INTRODUCTION

The user does not require any high-end specification in his device to run our project. The project was made using python programming language.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- 1.6 GHz or faster processor
- 1 Gb of RAM

SOFTWARE REQUIREMENTS:

- Visual Studio Code

3.3 SUMMARY

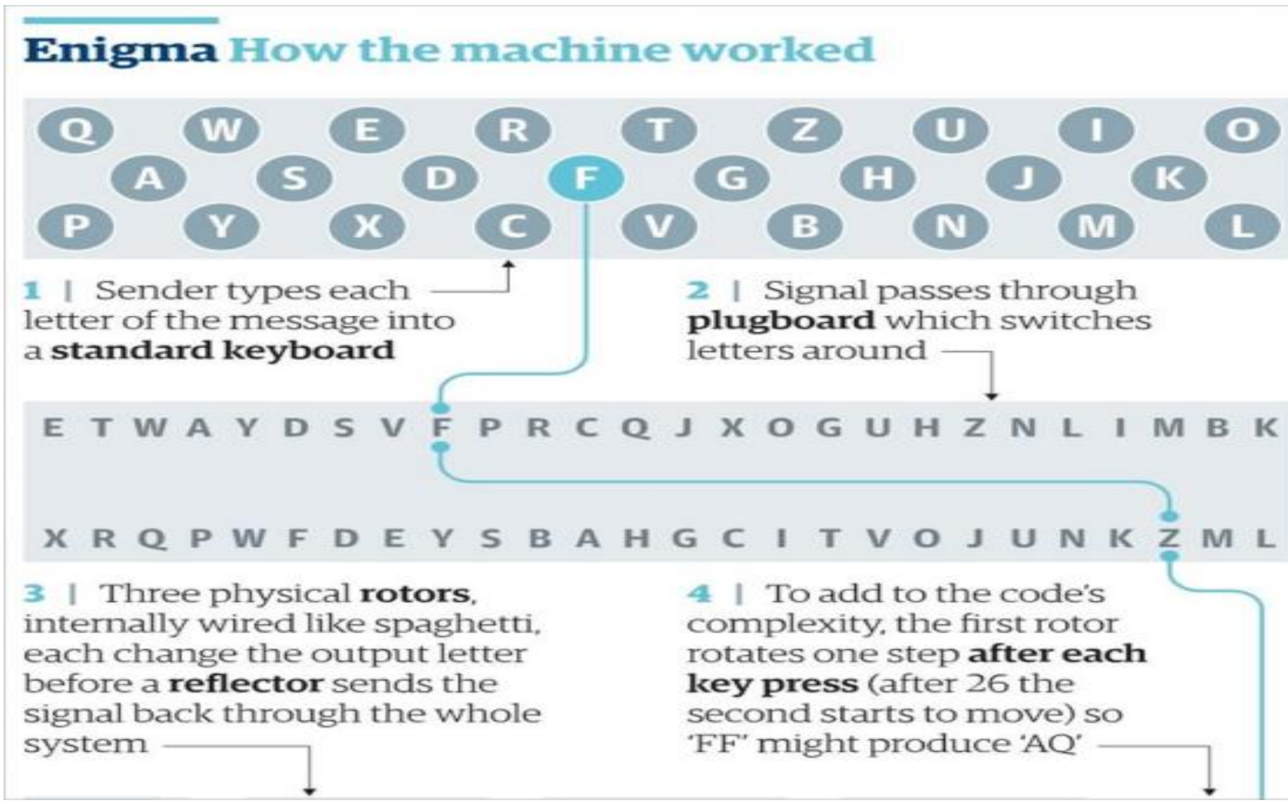
The project runs on basic requirements like a stable internet connection, a 4 Gb RAM, and requires Visual Studio Code. The model is not a complex project and is accessible to users without any difficulty.

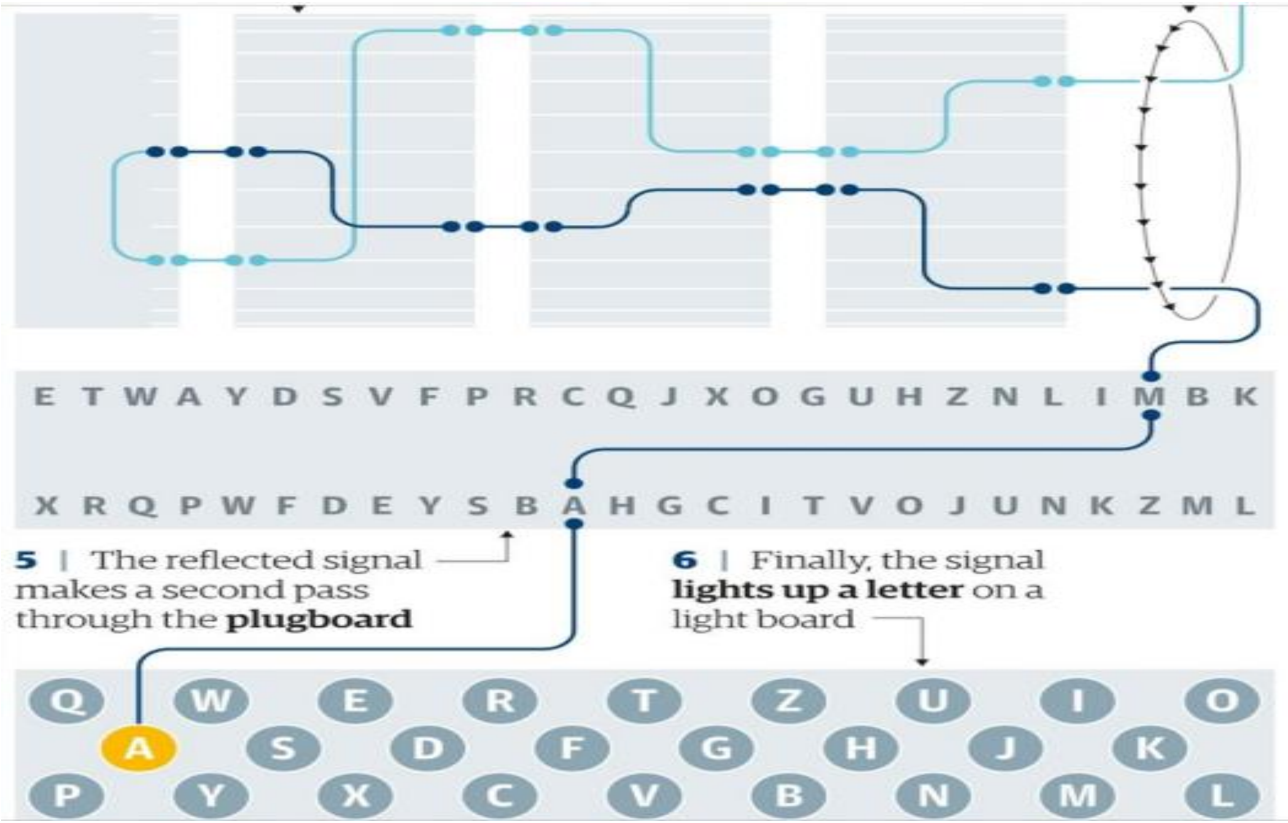
.

CHAPTER 4:

4.1 FUNCTIONAL MODULES DESIGN AND ANALYSIS

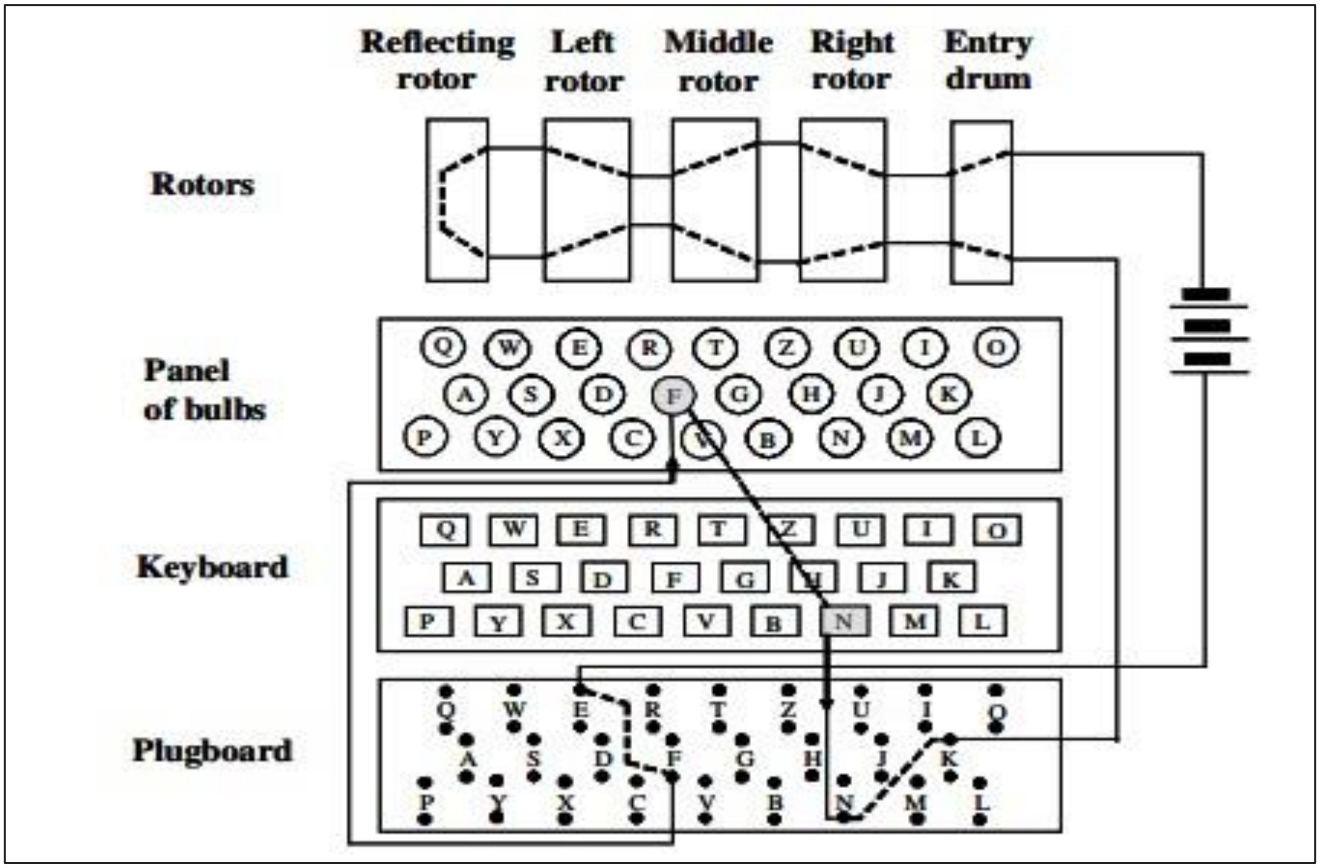
Modules Used:





4.2 ARCHITECTURAL DESIGNS

Figure : Functional Diagram of the military Enigma



4.3 USER INTERFACE DESIGN

Library Tkinter is used for user interface design.

4.4 SUMMARY

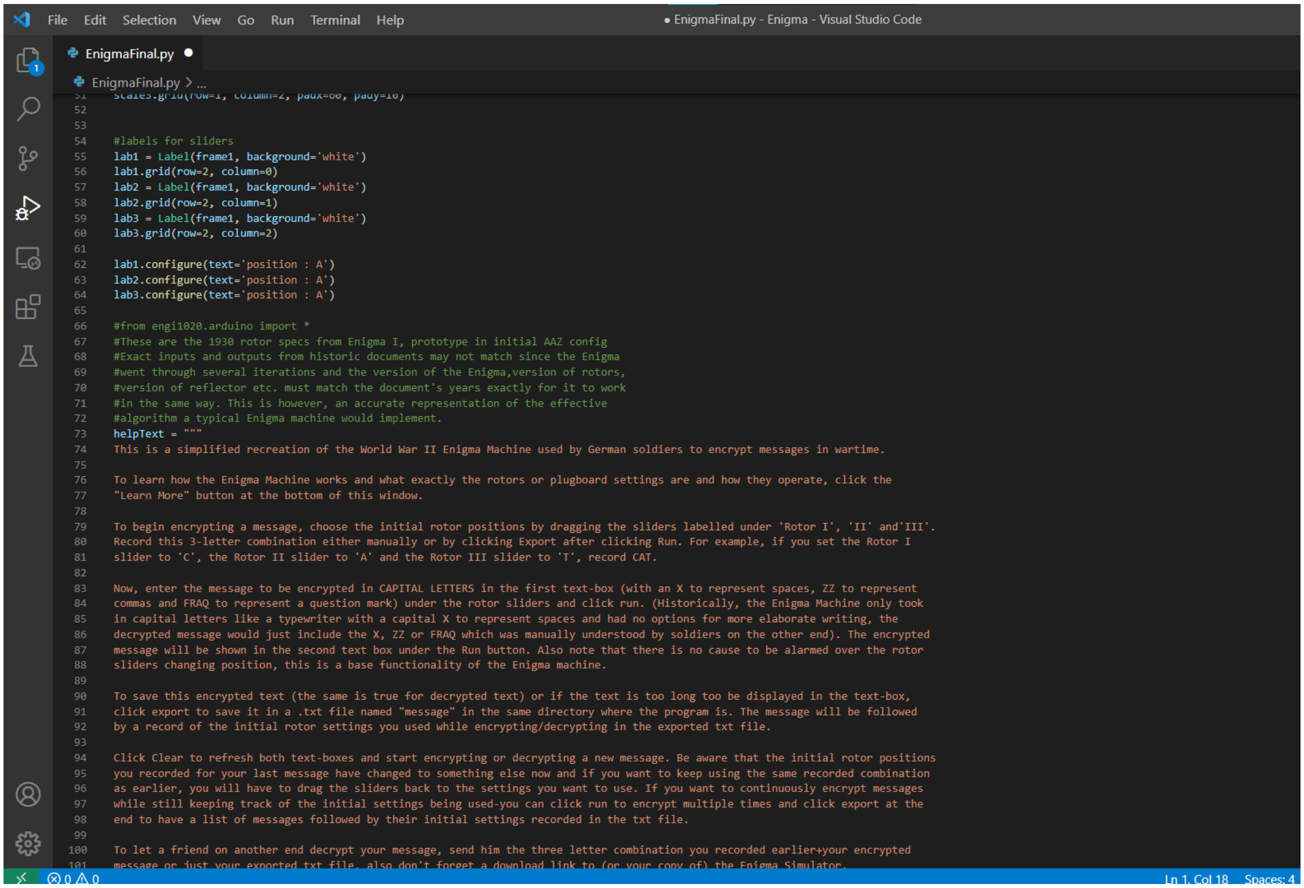
The entire project is based on the simple command of Python which makes it open-source and user friendly

CHAPTER-5:

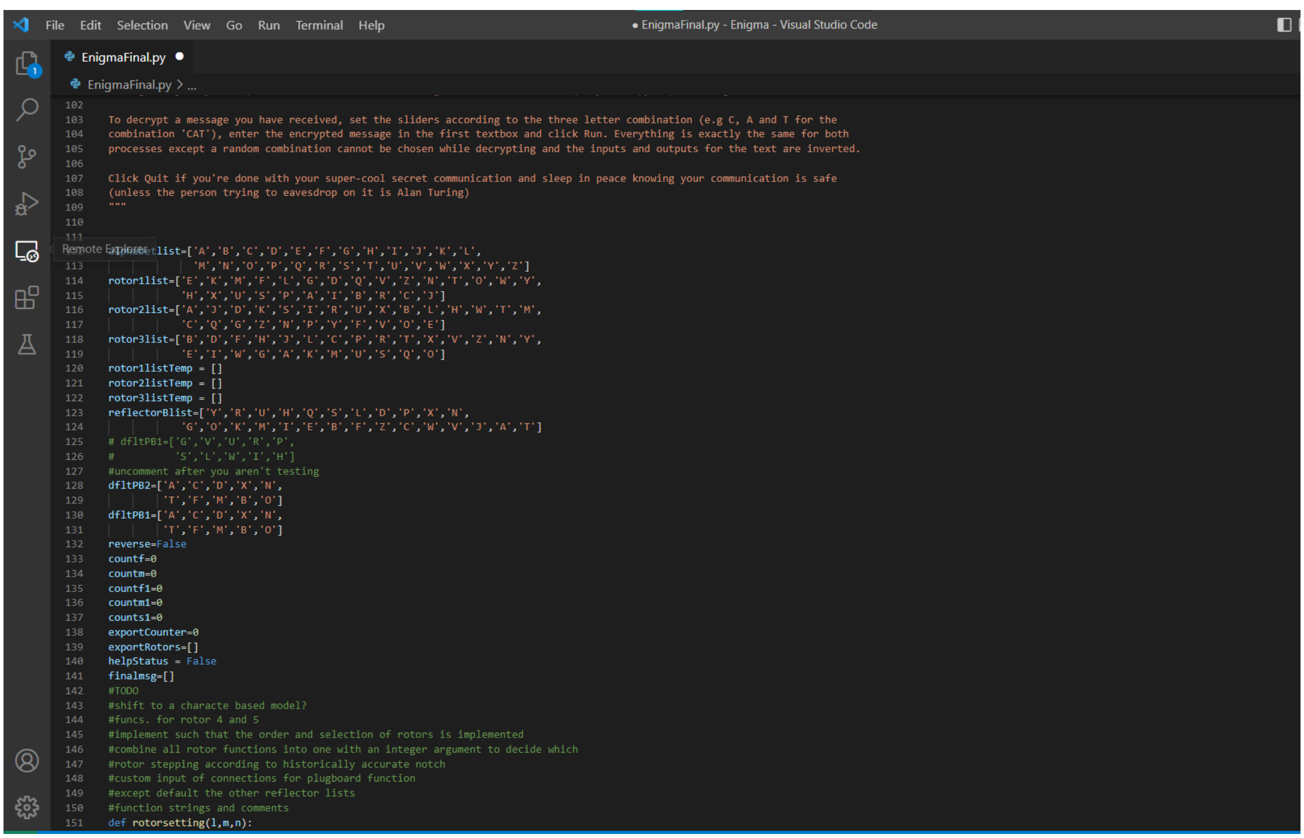
TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Technical coding and code solutions

```python
#import_libraries
from tkinter import *
from tkinter.messagebox import *
import webbrowser
#window
window = Tk()
window.title("MUN Enigma Machine")
window.configure(background='gray')

#logo and title
image = PhotoImage(file="ENIGMA MACHINE.png")
Label(window, image=image, background='white').pack(padx=5, pady=5, side=TOP)


#frame
frame1 = Frame(window, borderwidth=0, relief=FLAT, background='gray')
frame1.pack(side=TOP, padx=10, pady=10)

#slider updates
def update1(x):
    x = int(x)
    alphabetList = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
    lab1.configure(text='position : {}'.format(alphabetList[x]))
def update2(x):
    x = int(x)
    alphabetList = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
    lab2.configure(text='position : {}'.format(alphabetList[x]))
def update3(x):
    x = int(x)
    alphabetList = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
    lab3.configure(text='position : {}'.format(alphabetList[x]))

#rotor labels
rotor1lab = Label(frame1, text='Rotor I', padx=10, pady=5, borderwidth=0, background='white')
rotor1lab.grid(row=0, column=0)
rotor2lab = Label(frame1, text='Rotor II', padx=10, pady=5, borderwidth=0, background='white')
rotor2lab.grid(row=0, column=1)
rotor3lab = Label(frame1, text='Rotor III', padx=10, pady=5, borderwidth=0, background='white')
rotor3lab.grid(row=0, column=2)


#sliders
rotor1_var = DoubleVar()
scale1 = Scale(frame1, from_=0, to=25, variable = rotor1_var, showvalue=0, command=update1, length= 150, background='white')
scale1.grid(row=1, column=0, padx=60, pady=10)
rotor2_var = DoubleVar()
scale2 = Scale(frame1, from_=0, to=25, variable = rotor2_var, showvalue=0, command=update2, length= 150, background='white')
scale2.grid(row=1, column=1, padx=60, pady=10)
rotor3_var = DoubleVar()
scale3 = Scale(frame1, from_=0, to=25, variable = rotor3_var, showvalue=0, command=update3, length= 150, background='white')
```



```
#labels for sliders
lab1 = Label(frame1, background='white')
lab1.grid(row=2, column=0)
lab2 = Label(frame1, background='white')
lab2.grid(row=2, column=1)
lab3 = Label(frame1, background='white')
lab3.grid(row=2, column=2)

lab1.configure(text='position : A')
lab2.configure(text='position : A')
lab3.configure(text='position : A')

#from engi1020.arduino import *
#These are the 1930 rotor specs from Enigma I, prototype in initial AAZ config
#Exact inputs and outputs from historic documents may not match since the Enigma
#went through several iterations and the version of the Enigma,version of rotors,
#version of reflector etc. must match the document's years exactly for it to work
#in the same way. This is however, an accurate representation of the effective
#algorithm a typical Enigma machine would implement.
helpText = """
This is a simplified recreation of the World War II Enigma Machine used by German soldiers to encrypt messages in wartime.

To learn how the Enigma Machine works and what exactly the rotors or plugboard settings are and how they operate, click the
"Learn More" button at the bottom of this window.

To begin encrypting a message, choose the initial rotor positions by dragging the sliders labelled under 'Rotor I', 'II' and'III'.
Record this 3-letter combination either manually or by clicking Export after clicking Run. For example, if you set the Rotor I
slider to 'C', the Rotor II slider to 'A' and the Rotor III slider to 'T', record CAT.

Now, enter the message to be encrypted in CAPITAL LETTERS in the first text-box (with an X to represent spaces, ZZ to represent
commas and FRAQ to represent a question mark) under the rotor sliders and click run. (Historically, the Enigma Machine only took
in capital letters like a typewriter with a capital X to represent spaces and had no options for more elaborate writing, the
decrypted message would just include the X, ZZ or FRAQ which was manually understood by soldiers on the other end). The encrypted
message will be shown in the second text box under the Run button. Also note that there is no cause to be alarmed over the rotor
sliders changing position, this is a base functionality of the Enigma machine.

To save this encrypted text (the same is true for decrypted text) or if the text is too long too be displayed in the text-box,
click export to save it in a .txt file named "message" in the same directory where the program is. The message will be followed
by a record of the initial rotor settings you used while encrypting/decrypting in the exported txt file.

Click Clear to refresh both text-boxes and start encrypting or decrypting a new message. Be aware that the initial rotor positions
you recorded for your last message have changed to something else now and if you want to keep using the same recorded combination
as earlier, you will have to drag the sliders back to the settings you want to use. If you want to continuously encrypt messages
while still keeping track of the initial settings being used-you can click run to encrypt multiple times and click export at the
end to have a list of messages followed by their initial settings recorded in the txt file.

To let a friend on another end decrypt your message, send him the three letter combination you recorded earlier+your encrypted
message or just your exported txt file. also don't forget a download link to (or your copy of) the Enigma Simulator.
```



```
To decrypt a message you have received, set the sliders according to the three letter combination (e.g C, A and T for the
combination 'CAT'), enter the encrypted message in the first textbox and click Run. Everything is exactly the same for both
processes except a random combination cannot be chosen while decrypting and the inputs and outputs for the text are inverted.

Click Quit if you're done with your super-cool secret communication and sleep in peace knowing your communication is safe
(unless the person trying to eavesdrop on it is Alan Turing)
"""


alphabetlist=['A','B','C','D','E','F','G','H','I','J','K','L',
              'M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
rotor1list=['E','K','M','F','L','G','D','Q','V','Z','N','T','O','W','Y',
            'H','X','U','S','P','A','I','B','R','C','J']
rotor2list=['A','J','D','K','S','I','R','U','X','B','L','H','W','T','M',
            'C','Q','G','Z','N','P','Y','F','V','O','E']
rotor3list=['B','D','F','H','J','L','C','P','R','T','X','V','Z','N','Y',
            'E','I','W','G','A','K','M','U','S','Q','O']
rotor1listTemp = []
rotor2listTemp = []
rotor3listTemp = []
reflectorBlist=['Y','R','U','H','Q','S','L','D','P','X','N',
            'G','O','K','M','I','E','B','F','Z','C','W','V','J','A','T']
# dfltPB1=['G','V','U','R','P',
#          'S','L','W','I','H']
#uncomment after you aren't testing
dfltPB2=['A','C','D','X','N',
         'T','F','M','B','O']
dfltPB1=['A','C','D','X','N',
         'T','F','M','B','O']
reverse=False
countf=0
countm=0
countf1=0
countm1=0
counts1=0
exportCounter=0
exportRotors=[]
helpStatus = False
finalmsg=[]
#TODO
#shift to a characte based model?
#funcs. for rotor 4 and 5
#implement such that the order and selection of rotors is implemented
#combine all rotor functions into one with an integer argument to decide which
#rotor stepping according to historically accurate notch
#custom input of connections for plugboard function
#except default the other reflector lists
#function strings and comments
def rotorsetting(l,m,n):
```

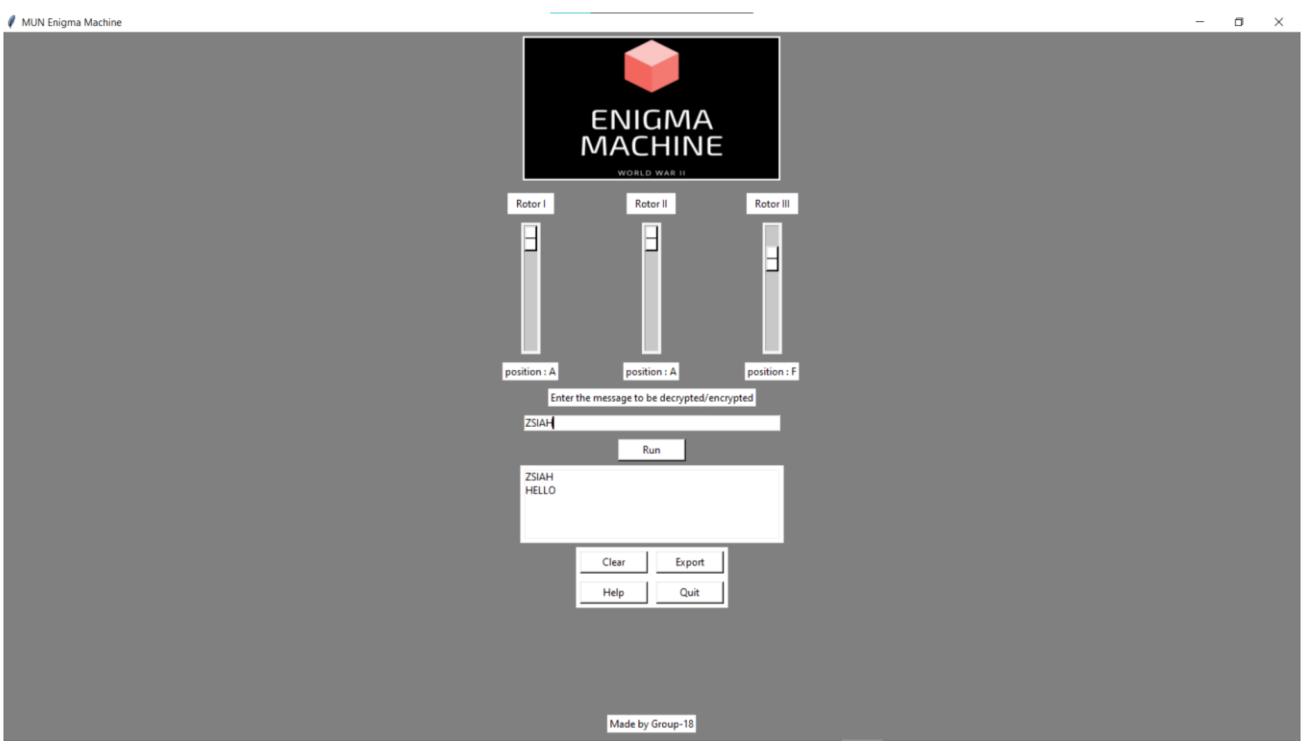
5.2 Summary

This chapter consists of segments of codes of the prototype

CHAPTER-6:

PROJECT OUTCOME AND APPLICABILITY

6.1 Significant project outcomes



6.2 Project applicability on Real-world applications

This model could be used in several universities for research purposes to an indepth idea of the working and applicability of ‘The Enigma machine’ which was one of the greatest achievements in the history of cyberworld.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1 Outline

Discovery Phase:

- Primary Research
  - The Enigma machine is a cipher device developed and used in the early- to mid-20th century to protect commercial, diplomatic, and military communication.
- Existing Content and Design Analysis
  - The four main components of the Enigma: Four rotors, a lampboard, a keyboard and a plugboard.

- Gather Data
    - The Enigma machine is a complicated apparatus consisting of a keyboard, a set of rotors, an alphabet ring, and plug connections, all configurable by the operator. For the message to be both encrypted and decrypted, both operators had to know two sets of codes.

Strategy Phase

- Determine Requirements
    - Software Requirements: Visual Studio Code
      Python IDLE 3.9
    - Stackoverflow
    - Stable Internet Connectivity

Design and Development Phase

- The Enigma machine implemented a substitution cipher, which encrypts a message by substituting one character for another/

Limitation/Constraints of the Enigma Machine.

- Inside the Enigma Machine - A major flaw with the Enigma code was that a letter could never be encoded as itself. In other words, an “M” would never be encoded as an “M.” This was a huge flaw in the Enigma code because it gave codebreakers a piece of information they could use to decrypt messages.

7.4 Inference

The Enigma machine produced encoded messages. Electrical signals from a typewriter-like keyboard were routed through a series of rotating wheels as well as a plugboard that scrambled the output but did so in a way that was decipherable with the right settings. Once a coded message was produced by one machine, it was sent via Morse code to an Enigma operator based elsewhere. That operator would use the same daily key code to set up their own machine in the same way, and they could then simply input the cyphered text on the keyboard.

REFERENCES

1. https://www.cmu.edu/news/stories/archives/2019/october/inside-the-engima-machine.html
2. https://www.mmu.ac.uk/media/mmuacuk/content/documents/school-of-computing-mathematics-and-digital-technology/mathematics-student-posters/Enigma.pdf
3. https://arxiv.org/pdf/2004.09982.pdf
4. https://www.archives.gov/files/publications/prologue/1997/fall/turing.pdf
5. https://link.springer.com/content/pdf/10.1007%2F3-540-39200-9_7.pdf
6. https://uwspace.uwaterloo.ca/bitstream/handle/10012/11023/FAINT_STEPHANIE.pdf?sequence=1
7. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.953.1316&rep=rep1&type=pdf
8. https://www.sciencedirect.com/topics/computer-science/enigma-machine