Student Name: Holland Whitley

Advisor Name: Dr. West

Expected Date of Graduation: May 1, 2021

The Enigma Machine

Important Links:

GitHub Repository: https://github.com/HAWhitley/Enigma

Presentation video: https://youtu.be/rY5Tc8zXBIs

Description:

The Enigma Machine project is a program that recreates the encryption method of the Enigma Machine. It does this by having multiple different substitution ciphers that emulate the plugboard, rotors, and reflector of the real Enigma machine. It is also a study of whether or not frequency analysis is effective on the Enigma machine. This study goes into depth on why frequency analysis is effective or not.

Statement of Purpose:

Purpose:

The purpose of recreating the Enigma machine is to tie Computer Science and Cybersecurity together in one project. More specifically, it is to learn about one of the most famous encryption methods of the 20th century as well as to learn more about encryption in general. The purpose of studying the effectiveness of frequency analysis on the Enigma machine is to understand how frequency analysis works and what it works with.

Problem Statement:

It is one thing to study encryption and encryption breaking methods in a classroom. It is a completely different to dissect an encryption method, remake it by hand, and visually seeing why an encryption breaking method works or not. Encryption is a difficult subject to grasp. It is especially difficult for students to grasp when the subject is only spoken about theoretically. In class, symmetric and asymmetric encryption is brought up with only a brief explanation as well as some simple methods such as Caesar Cipher. As known from personal experience, a student usually does not fully understand it until they reach Network Penetration and Ethical Hacking or pass their Security+ exam.

The best way to aid their understanding in the subject is to give them hands-on experience with it. This does not mean that they should throw some plaintext into a pre-made encryption algorithm and try to decipher what it did. This means that the student should personally dissect the encryption method. Dissecting the encryption method will give the student ample opportunity to see each and every step of the process, thus furthering their understanding of it. By helping a student understand different methods as well as encryption overall, the student will be better equipped for the Security+ exam, be a better Cybersecurity student in the long run, and have more experience and understanding for the professional world.

To summarize, the best way to help a student learn more about encryption is to get hands-on experience with it.

Research and Background:

Initially, I knew little about the Enigma machine. I knew what it looked like. I knew that it was widely used in World War II by the Nazis. I knew how Alan Turing initially cracked the code of the Enigma machine. I also knew that it was a series of substitution ciphers. What I came to learn through research was what the different components of the Enigma machine were, how they worked individually, and how they worked together. I had to learn how the plugboard, the rotors, and the reflector worked. I learned that each component was a different substitution cipher that worked in a slightly different way.

The plugboard uses cables to swap two letters, which allows for a maximum of thirteen substitutions. It does not automatically change after a certain number of letters have been typed. It only moves when the person operating the machine has manually moved the cables.

The rotors have a number of settings that have different substitution ciphers. They also have a specific letter setting that offsets the substitution cipher by a certain number of letters. The right rotor will rotate one letter every time a letter is typed. The middle rotor will rotate one letter every 26 letters. The left rotor will rotate every 676 letters. Also, based on the setting of the rotor, when a rotor gets to a specific letter, the rotor to the left will rotate one letter.

The reflector also has a number of settings that have different substitution ciphers. However, the reflector remains stagnant unless the person operating the machine has manually changed the setting.

After learning about how the components worked, I had to learn about how the components worked together. When a letter is typed, it first goes through the plugboard. Then, it passes through each rotor from right to left. After that, it passes through the reflector. Once it goes through the reflector, it has to go back through the rotors. This time, it goes from left to right. It finally passes through the plugboard again. At this point, the letter is fully encrypted. If a rotor moves, it moves before the letter reaches it the first time. It will not move again until it reaches a specific letter setting or the required number of letters.

After figuring out how the Enigma machine worked and actually programming it, I had to learn about frequency analysis. I knew letters had different frequencies in the English language. However, I needed to learn what kinds of encryption it worked with to determine if it would work with the Enigma machine. After looking at the definition of frequency analysis, I knew that it would only work with substitution ciphers.

Project Language(s), Software, and Hardware:

Project Language: C++

Project Software: VMware, Ubuntu virtual machine, Vi text editor

Project Hardware: One laptop

Project Requirements:

The minimum viable product that is to be produced is a fully functional Enigma machine. The Enigma machine is to follow the Enigma I version. This includes a fully functional plugboard, three rotors, and a reflector.

Each rotor is to have letter settings as well as type settings. The type settings will be I, II, and III from the Enigma I version. The rotors will also have a double stepping mechanism. The middle and left rotor will step after every 26 rotations of the rotor to the right. The right rotor will step at every letter. Each letter will also step every time the rotor to the right passes a specific letter. This letter will be determined by the rotor type. The reflector will have the setting B from the Enigma I version.

As well as a fully functional Enigma machine, frequency analysis is to be performed on Enigma encryption. Once this has been done, a determination must be made of whether it works on the Enigma machine or not. Once this determination has been made, an examination of why it does or does not work must be made.

Project Implementation Description and Explanation:

The Enigma machine was developed with an assortment of arrays and a lot of ASCII manipulation.

For the plugboard, all settings are entered into an array according to user input. The location of the settings in the array will determine what substitutions will occur. For example, if ‘B’ is located at index 0, ‘A’ will be substituted with ‘B’ and vice versa.

For the rotors, the rotor type substitution settings are placed into an array. Whatever the input is from the rotors to the right or the plugboard will be substituted based on the settings of the rotor type. The rotor types are determined by user input. The rotor letter settings are also determined by user input.

The letter setting increases (rotates) according to the position of the rotor (right, middle, left). The right rotor increases (rotates) the letter setting at every letter. The middle rotor increases (rotates) the letter setting at every 26 rotations of the right rotor. The left rotor increases (rotates) the letter at every 26 rotations of the middle rotor.

The input letter is offset by the current letter setting and the initial letter setting. This positions the input letter correctly with the current settings of the rotor. For example, if the rotor type is I, the initial letter setting is ‘A’, the current letter setting is ‘B’, and the input letter is ‘A’, it will be substituted with ‘K’. If the current letter setting was ‘A’ instead of ‘B’, the input letter would be substituted with ‘E’.

For the reflector, settings are placed into an array. Whatever the input is from the left rotor will be substituted by the settings of the reflector. For example, if the input from the left rotor is ‘A’, it will be substituted with ‘Y’.

Test Plan:

With each prototype in the Enigma Machine, there are five tests. For the double stepping prototype, there are size tests. Each test has specific settings. Throughout each prototype, the test’s settings remain the same. In some prototypes, settings are added to the test, such as adding the rotor settings. In the process of testing, I will enter some clear text to encrypt. Then, the program should encrypt the clear text and output it onto the screen. All non-alphabetic text will remain the same. To ensure that the encryption process is running correctly, I will calculate all encryption by hand and develop the expected output. I will also run the plaintext through a simulator to check my work.

Test 1 will be the control settings: nothing changed on the plugboard, all letter settings to ‘a’ on the rotors, and all rotor settings on type 1. Test 2 will only change the plugboard settings. Test 3 will only change the rotor letter settings. Test 4 will only change the rotor type settings. Test 5 will only change the right rotor letter setting to set off the double stepping mechanism on the right rotor. Test 6 will only change all of the rotor letter settings to set off the double stepping mechanism on all of the rotors.

Cleartext Phrase: “Hello, my name is Holland.”

1. Plugboard

   Test 1:

Plugboard settings: a-a, b-b, c-c, d-d, e-e, f-f, g-g, h-h, i-i, j-j, k-k, l-l, m-m, n-n, o-o, p-p, q-q, r-r, s-s, t-t, u-u, v-v, w-w, x-x, y-y, z-z

Test2:

Plugboard settings: a-z, b-y, c-x, d-w, e-v, f-u, g-t, h-s, i-r, j-q, k-k, l-l, m-m, n-n, o-o, p-p

Test 3:

Plugboard settings: a-a, b-b, c-c, d-d, e-e, f-f, g-g, h-h, i-i, j-j, k-k, l-l, m-m, n-n, o-o, p-p, q-q, r-r, s-s, t-t, u-u, v-v, w-w, x-x, y-y, z-z

Test 4:

Plugboard settings: a-a, b-b, c-c, d-d, e-e, f-f, g-g, h-h, i-i, j-j, k-k, l-l, m-m, n-n, o-o, p-p, q-q, r-r, s-s, t-t, u-u, v-v, w-w, x-x, y-y, z-z

2. Plugboard + Right Rotor (Single Stepping)

Test 1:

Right Rotor Letter: a

Right Rotor Type: 1

Test 2:

Right Rotor Letter: a

Right Rotor Type: 1

Test 3:

Right Rotor Letter: z

Right Rotor Type: 1

Test 4:

Right Rotor Letter: a

Right Rotor Type: 1

3. Plugboard + Right Rotor + Middle Rotor (Single Stepping)

Test 1:

Middle Rotor Letter: a

Middle Rotor Type: 1

Test 2:

Middle Rotor Letter: a

Middle Rotor Type: 1

Test 3:

Middle Rotor Letter: y

Middle Rotor Type: 1

Test 4:

Middle Rotor Letter: a

Middle Rotor Type: 2

4. Plugboard + Right Rotor + Middle Rotor + Left Rotor (Single Stepping)

Test 1:

Left Rotor Letter: a

Left Rotor Type: 1

Test 2:

Left Rotor Letter: a

Left Rotor Type: 1

Test 3:

Left Rotor Letter: x

Left Rotor Type: 1

Test 4:

Left Rotor Letter: a

Left Rotor Type: 3

5. Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector (Single Stepping)
6. Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector + Left Rotor + Middle Rotor + Right Rotor + Plugboard (Single Stepping)
7. Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector + Left Rotor + Middle Rotor + Right Rotor + Plugboard (Double Stepping)

Test 5:

Plugboard settings: a-a, b-b, c-c, d-d, e-e, f-f, g-g, h-h, i-i, j-j, k-k, l-l, m-m, n-n, o-o, p-p, q-q, r-r, s-s, t-t, u-u, v-v, w-w, x-x, y-y, z-z

Right Rotor Letter: q

Right Rotor Type: 1

Middle Rotor Letter: a

Middle Rotor Type: 1

Left Rotor Letter: a

Left Rotor Type: 1

Test 6:

Plugboard settings: a-a, b-b, c-c, d-d, e-e, f-f, g-g, h-h, i-i, j-j, k-k, l-l, m-m, n-n, o-o, p-p, q-q, r-r, s-s, t-t, u-u, v-v, w-w, x-x, y-y, z-z

Right Rotor Letter: q

Right Rotor Type: 1

Middle Rotor Letter: q

Middle Rotor Type: 1

Left Rotor Letter: q

Left Rotor Type: 1

For the frequency analysis prototype, there will also be three different tests. For each test, a large amount of text will be run through the Enigma Machine. Then I will run the encrypted text through frequency analysis and cross check it with the original text. Then I will determine the frequency analysis's accuracy.

Test 1: "The Cask of Amontillado" by Edgar Allen Poe

Test 2: "Rumpelstiltskin" by Brothers Grimm

Test 3: "Chapter 1" from *Systems Analysis and Design* by Rosenblatt

Test Results:

Enigma Machine –

- Plugboard:

Test 1: Passed

Expected Result: HELLO, MY NAME IS HOLLAND

Actual Result: HELLO, MY NAME IS HOLLAND

Test 2: Passed

Expected Result: SVLLO, MB NZMV RH SOLLZNW

Actual Result: SVLLO, MB NZMV RH SOLLZNW

Test 3: Passed

Expected Result: HELLO, MY NAME IS HOLLAND

Actual Result: HELLO, MY NAME IS HOLLAND

Test 4: Passed

Expected Result: HELLO, MY NAME IS HOLLAND

Actual Result: HELLO, MY NAME IS HOLLAND

- Plugboard + Right Rotor (Single Stepping):

Test 1: Passed

Expected Result: VDYHP, SG IZBH AG IFKMSDR

Actual Result: VDYHP, SG IZBH AG IFKMSDR

Test 2: Passed

Expected Result: PRYHP, SV IVBD FA DFKMUDX

Actual Result: PRYHP, SV IVBD FA DFKMUDX

Test 3: Passed

Expected Result: QGWYS, UL AVIY PL AMEKUGB

Actual Result: QGWYS, UL AVIY PL AMEKUGB

Test 4: Passed

Expected Result: VDYHP, SG IZBH AG IFKMSDR

Actual Result: VDYHP, SG IZBH AG IFKMSDR

- Plugboard + Right Rotor + Middle Rotor (Single Stepping):

Test 1: Passed

Expected Result: AKIFN, OJ EXUB YP AXAIENR

Actual Result: AKIFN, OJ EXUB YP AXAIENR

Test 2: Passed

Expected Result: YHIFN, OY EOUS PW HXAIMNF

Actual Result: YHIFN, OY EOUS PW HXAIMNF

Test 3: Passed

Expected Result: YFSPO, WF UTRO MR TBWSKOD

Actual Result: YFSPO, WF UTRO MR TBWSKOD

Test 4: Passed

Expected Result: PJYKL, WE AQGF MN PQPYALV

Actual Result: PJYKL, WE AQGF MN PQPYALV

- Plugboard + Right Rotor + Middle Rotor + Left Rotor (Single Stepping):

Test 1: Passed

Expected Result: ENVGW, YZ LRAK CH EREVLWU

Actual Result: ENVGW, YZ LRAK CH EREVLWU

Test 2: Passed

Expected Result: CQVGW, YC LYAS HB QREVOWG

Actual Result: CQVGW, YC LYAS HB QREVOWG

Test 3: Passed

Expected Result: RLUYW, IL PSXW TX SEIUZWM

Actual Result: RLUYW, IL PSXW TX SEIUZWM

Test 4: Passed

Expected Result: ETQXV, UJ BICL ZN EIEQBVM

Actual Result: ETQXV, UJ BICL ZN EIEQBVM

- Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector (Single Stepping)

  Test 1: Passed

  Expected Result: QKWLV, AT GBYN UD QBQWGVC

  Actual Result: QKWLV, AT GBYN UD QBQWGVC

  Test 2: Passed

  Expected Result: UEWLV, AU GAYF DR EBQWMVL

  Actual Result: UEWLV, AU GAYF DR EBQWMVL

  Test 3: Passed

  Expected Result: CMJRT, GM FWYT VY WDGJUTI

  Actual Result: CMJRT, GM FWYT VY WDGJUTI

  Test 4: Passed

  Expected Result: QZEJW, CX RPUG TK QPQERWO

  Actual Result: QZEJW, CX RPUG TK QPQERWO

- Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector + Left Rotor + Middle Rotor + Right Rotor + Plugboard (Single Stepping)

  Test 1: Passed

  Expected Result: GMHWP, KX WEDR NF SJNFQTB

  Actual Result: GMHWP, KX WEDR NF SJNFQTB

  Test 2: Passed

  Expected Result: NLSDP, KG DLWM LG SQNUPGC

  Actual Result: NLSDP, KG DLWM LG SQNUPGC

Test 3: Passed

Expected Result: GPXQS, IS LUSF DY KAQAYBP

Actual Result: GPXQS, IS LUSF DY KAQAYBP

Test 4: Passed

Expected Result: MFNCZ, KU CIDI YE LFONYQI

Actual Result: MFNCZ, KU CIDI YE LFONYQI

- Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector + Left Rotor + Middle Rotor + Right Rotor + Plugboard (Double Stepping)

Test 1: Passed

Expected Result: GMHWP, KX WEDR NF SJNNTZW

Actual Result: GMHWP, KX WEDR NF SJNNTZW

Test 2: Passed

Expected Result: NLSDP, KG DLWM LG SQNNEAW

Actual Result: NLSDP, KG DLWM LG SQNNEAW

Test 3: Passed

Expected Result: GPXQS, IS LUSF DY KAQAOFM

Actual Result: GPXQS, IS LUSF DY KAQAOFM

Test 4: Passed

Expected Result: MFNCZ, KU CIDI YE LFODCBN

Actual Result: MFNCZ, KU CIDI YE LFODCBN

Test 5: Passed

Expected Result: CYBHV, JL UPFA SZ TVWWIYK

Actual Result: CYBHV, JL UPFA SZ TVWWIYK

Test 6: Passed

Expected Result: GLJEV, LT CGTI WT OXCIPPI

Actual Result: GLJEV, LT CGTI WT OXCIPPI

Frequency Analysis –

- Test 1: Failed

  Expected Result: “The Cask of Amontillado” by Edgar Allen Poe

  Actual Result: Gibberish

- Test 2: Failed

  Expected Result: “Rumpelstiltskin” by Brothers Grimm

  Actual Result: Gibberish

- Test 3: Failed

  Expected Result: “Chapter 1” from *Systems Analysis and Design* by Rosenblatt

  Actual Result: Gibberish

Why?

Frequency analysis only works when the substitution cipher doesn’t change throughout the ciphertext. The Enigma machine changes the substitution cipher at every letter. One letter of plaintext can have different ciphertext letters within the same phrase (i.e., ‘A’ can be ‘C’ and ‘M’ in the same phrase).

Challenges Overcome:

The main challenge that I faced was a lack of good resources about the Enigma machine. Most sources about it were oversimplified for my purposes. They were good for giving a general overview about how it worked, but they did not go into great detail about how the individual components worked. I was lucky enough to find one website that gave a very good explanation about the different components. This gave me a great starting point.

After I overcame that challenge, I had no way of knowing if my implementation was correct. I knew that it worked based on my understanding of the Enigma machine, but I did not know whether or not it was accurate to the actual Enigma machine. To try and make sure that my implementation was accurate, I needed to search for a pre-made simulator of the Enigma machine. This proved to be very difficult. Many simulators had different versions of the machine, and I could not see their process of encryption. I knew from looking at simulators that my implementation was completely different from everything else. Eventually, I did find a simulator that showed exactly what it was doing in each step. This helped with correcting the mistakes that I made. Most of my implementation was correct, but there were tiny mistakes that completely threw the final encrypted message off.

Another challenge that I faced was figuring out whether frequency analysis worked on the Enigma machine or not. My initial process of testing this out was putting a large chunk of plaintext into my Enigma machine and throwing it into a pre-made decoder that used frequency analysis. I figured that I would try and see if it worked before I made my own decoder. It did not take long to figure out that it did not work. I had no idea why because I learned that frequency analysis works with substitution ciphers. The Enigma machine is a bunch of substitution ciphers strung together, so it theoretically should have worked. Then, I realized that the substitution cipher changes at every letter. Frequency analysis only works when the substitution cipher doesn’t change. Therefore, I switched from making a decoder using frequency analysis for the

Enigma machine to doing a study of why frequency analysis does not work on the Enigma machine.

Future Enhancements:

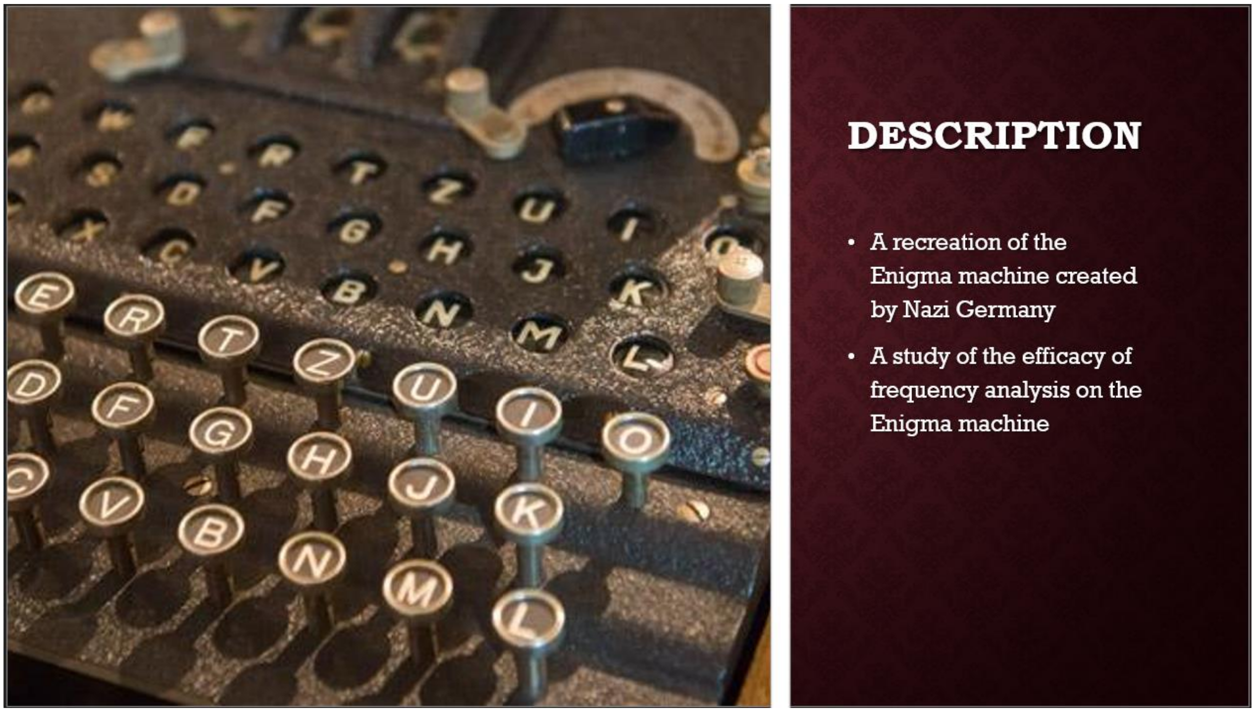
Future enhancements would involve any parts of the Enigma machine that were not included in this implementation. The parts that were not included were extra rotor settings (IV, V, VI, VII, VIII, Beta, Gamma), a fourth rotor, extra reflector settings (Reflector C, Reflector B Thin, Reflector C Thin), and ring settings.

They would also involve more studies on the effectiveness of different types of code breaking methods on the Enigma machine. An example of this could include an expansion of the known method of breaking it (i.e., using a known word or phrase in the text to break it). This could possibly be done by using common three letter words such as “the”, “she”, “his”, and “her”. A study could also be done about what effect of no white space or punctuation could have on the code breaking process. There are many possibilities.

Defense Presentation Slides:

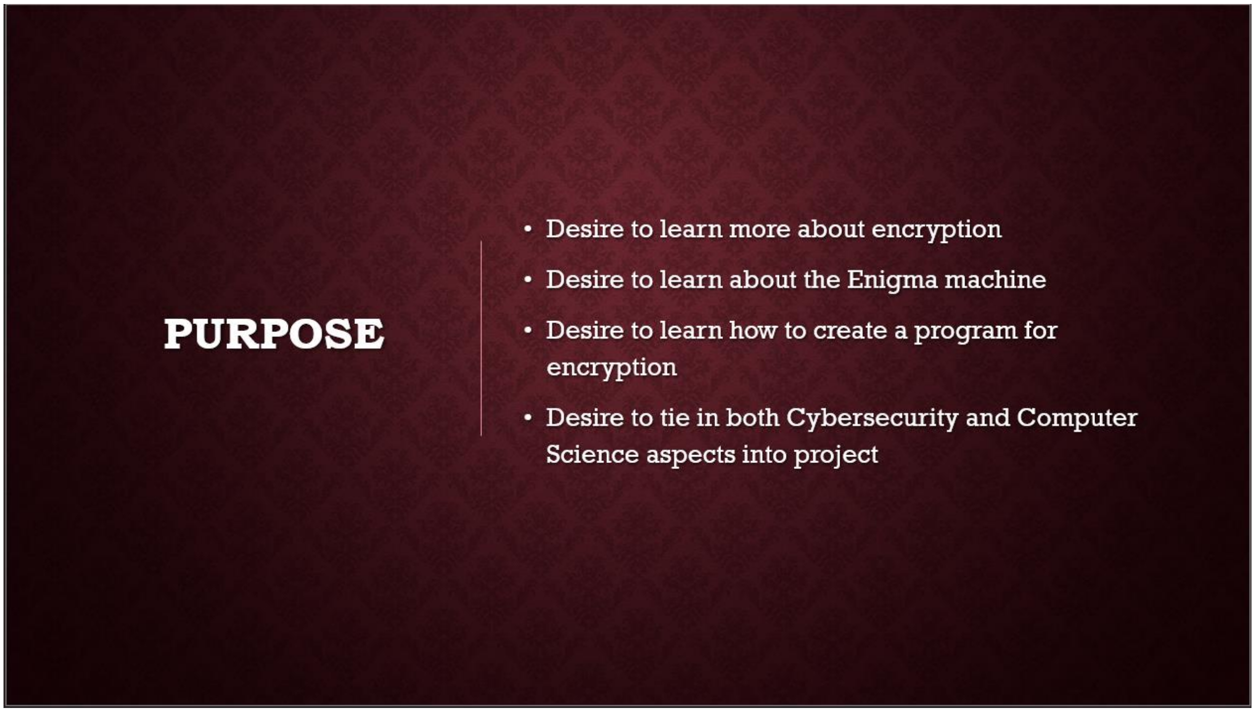
THE ENIGMA MACHINE

Holland Whitley



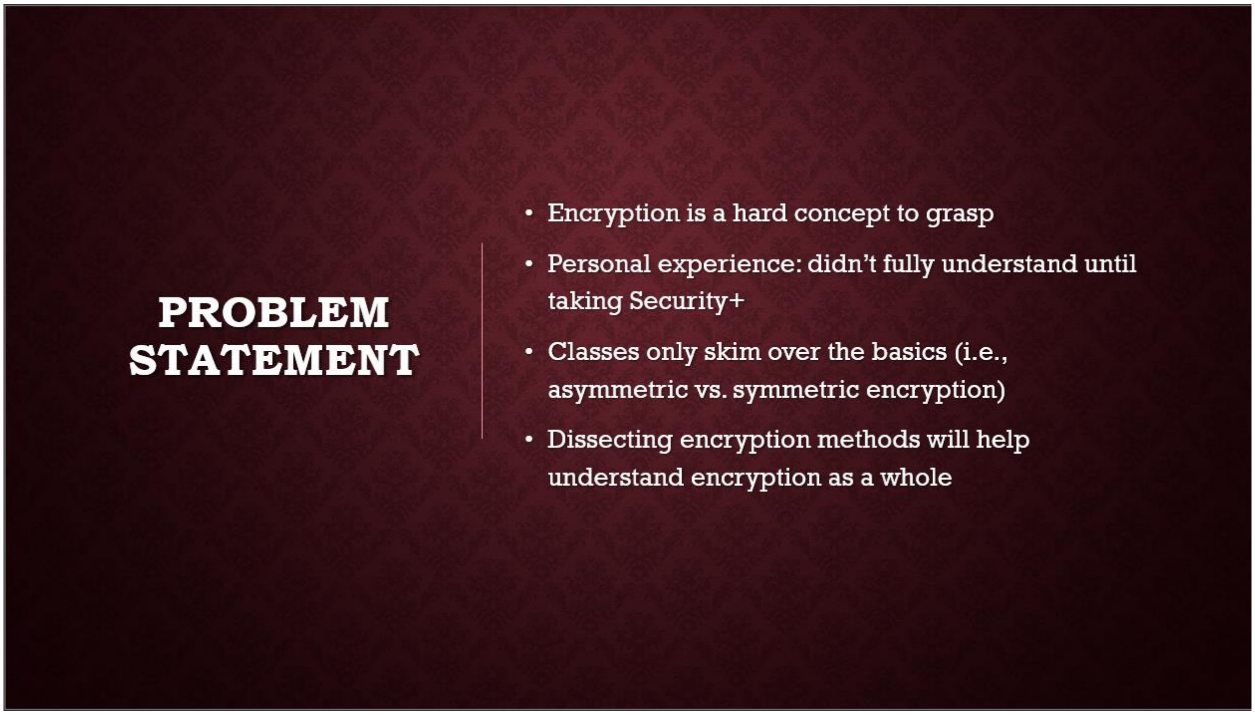
DESCRIPTION

- A recreation of the Enigma machine created by Nazi Germany
- A study of the efficacy of frequency analysis on the Enigma machine



PURPOSE

- Desire to learn more about encryption
- Desire to learn about the Enigma machine
- Desire to learn how to create a program for encryption
- Desire to tie in both Cybersecurity and Computer Science aspects into project

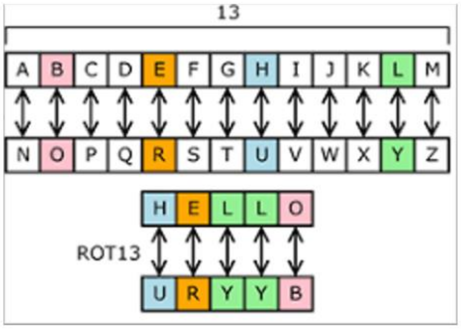


PROBLEM STATEMENT

- Encryption is a hard concept to grasp
- Personal experience: didn't fully understand until taking Security+
- Classes only skim over the basics (i.e., asymmetric vs. symmetric encryption)
- Dissecting encryption methods will help understand encryption as a whole

BACKGROUND/RESEARCH

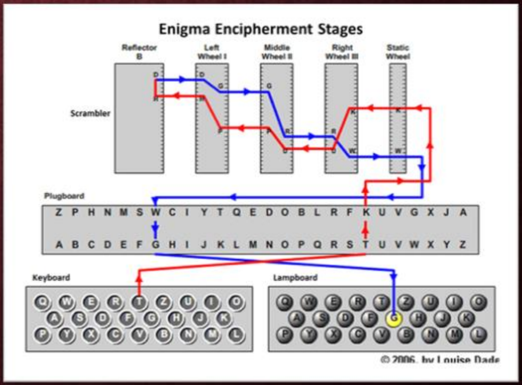
THE BASICS



- The Enigma machine is a series of four substitution ciphers that change at every letter.
- Substitution cipher – a cipher in which the letters of plaintext are systematically replaced by substitute letters

BASIC COMPONENTS

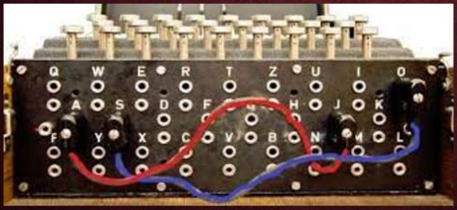
- Plugboard – substitution cipher made with physical plugs
- Three Rotors – substitution ciphers based on setting
- Reflector – substitution cipher
- Keyboard
- Lampboard



© 2006, by Louise Dade

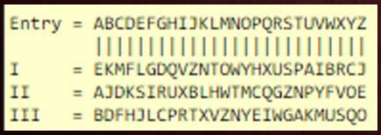
PLUGBOARD, KEYBOARD, & LAMPBOARD

- Keyboard – Input mechanism
- Lampboard –Output mechanism
- Plugboard – substitution between two letters (A=Z, Z=A), which is made by physical plugs

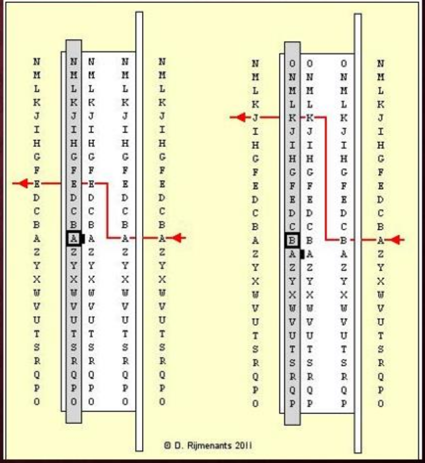


ROTOR SETTINGS

- Rotor – a substitution cipher that throws the ciphertext into the next rotor, reflector, or plugboard
- Letter setting – shifts initial letter over by the offset of the letter setting
- Three Rotor Settings (I, II, III) – each rotor can be on either setting



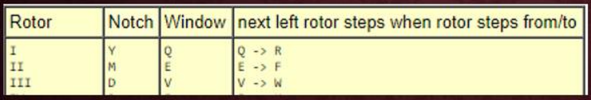
```
Entry = ABCDEFGHIJKLMNOPQRSTUVWXYZ
        ||||||||||||||||||||||||||
I     = EKMFLGDQVZNTOWYHXUSPAIBRCJ
II    = AJDKSIRUXBLHWTMCQGZNPYFVOE
III   = BDFHJLCPRTXVZNYEIWGAKMUSQO
```



© D. Rijmenants 2011

ROTOR MOVEMENT

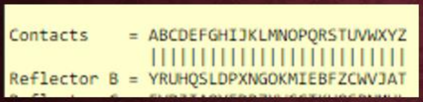
- Movement by Rotor Placement –
  - Right rotor – increases letter setting at every letter
  - Middle rotor – increases letter setting at every 26 rotations of the right rotor
  - Right rotor – increases letter setting at every 26 rotations of the middle rotor
- Movement by Rotor Setting –



Rotor	Notch	Window	next left rotor steps when rotor steps from/to
I	Y	Q	Q -> R
II	M	E	E -> F
III	D	V	V -> W

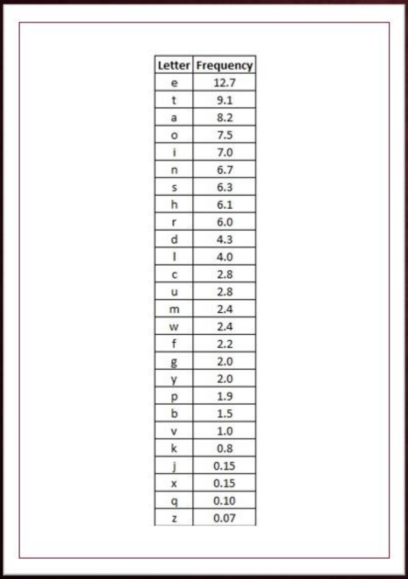
REFLECTOR SETTINGS

- Reflector - A substitution cipher that throws the ciphertext back through the rotors



```
Contacts    = ABCDEFGHIJKLMNOPQRSTUVWXYZ
              ||||||||||||||||||||||||||
Reflector B = YRUHQSLDPXNGOKMIEBFZCWVJAT
```

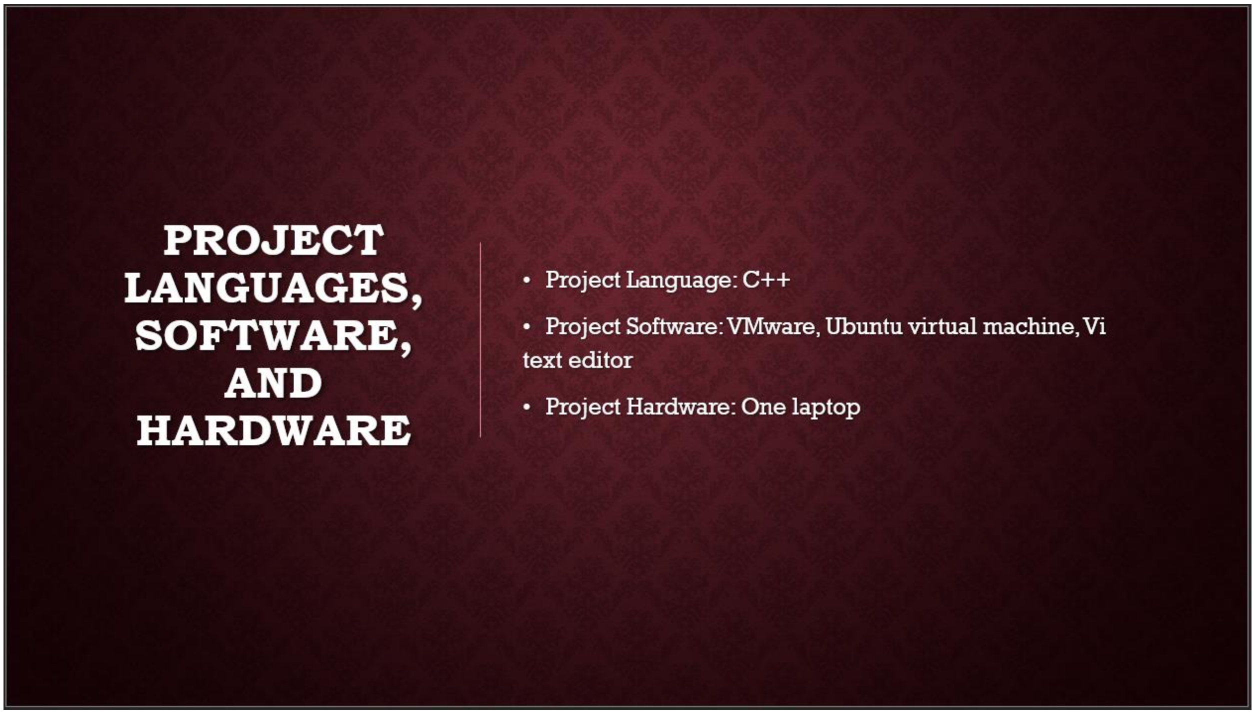
- Does not move, always constant
- Does not have any letter settings



Letter	Frequency
e	12.7
t	9.1
a	8.2
o	7.5
i	7.0
n	6.7
s	6.3
h	6.1
r	6.0
d	4.3
l	4.0
c	2.8
u	2.8
m	2.4
w	2.4
f	2.2
g	2.0
y	2.0
p	1.9
b	1.5
v	1.0
k	0.8
j	0.15
x	0.15
q	0.10
z	0.07

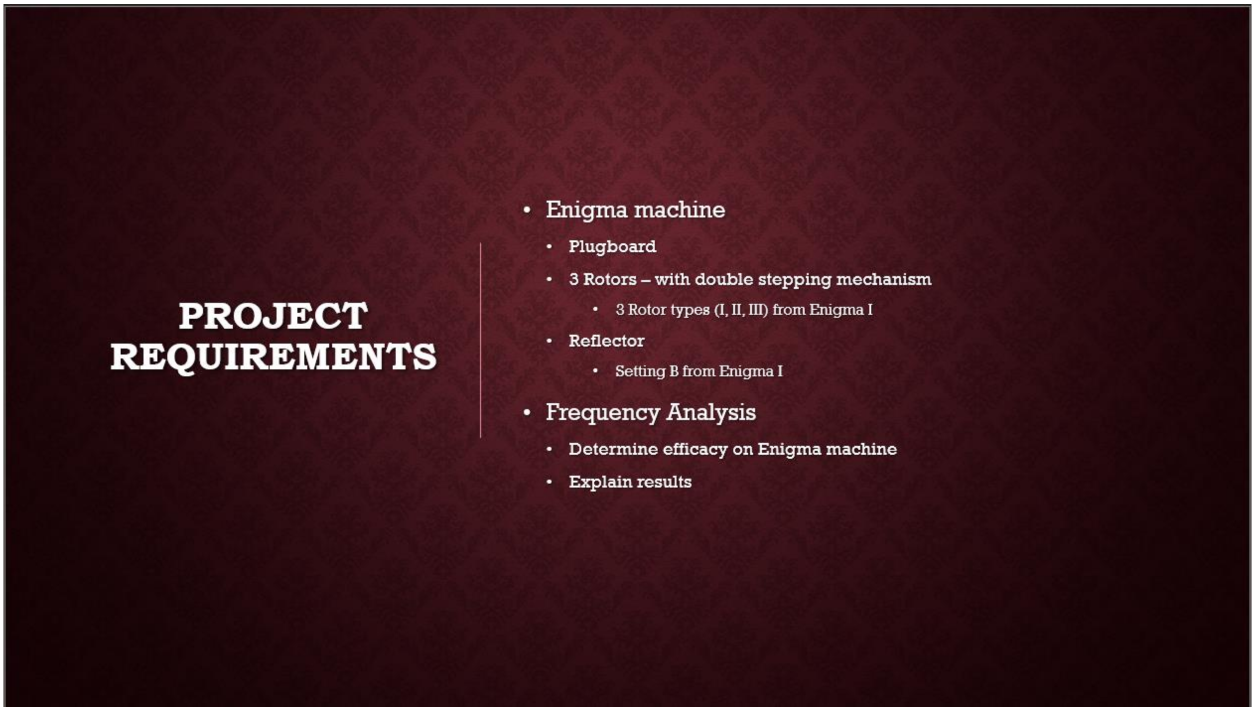
FREQUENCY ANALYSIS

- The frequency of letters or groups of letters in a ciphertext can be translated to the frequency of letters or groups of letters in plaintext



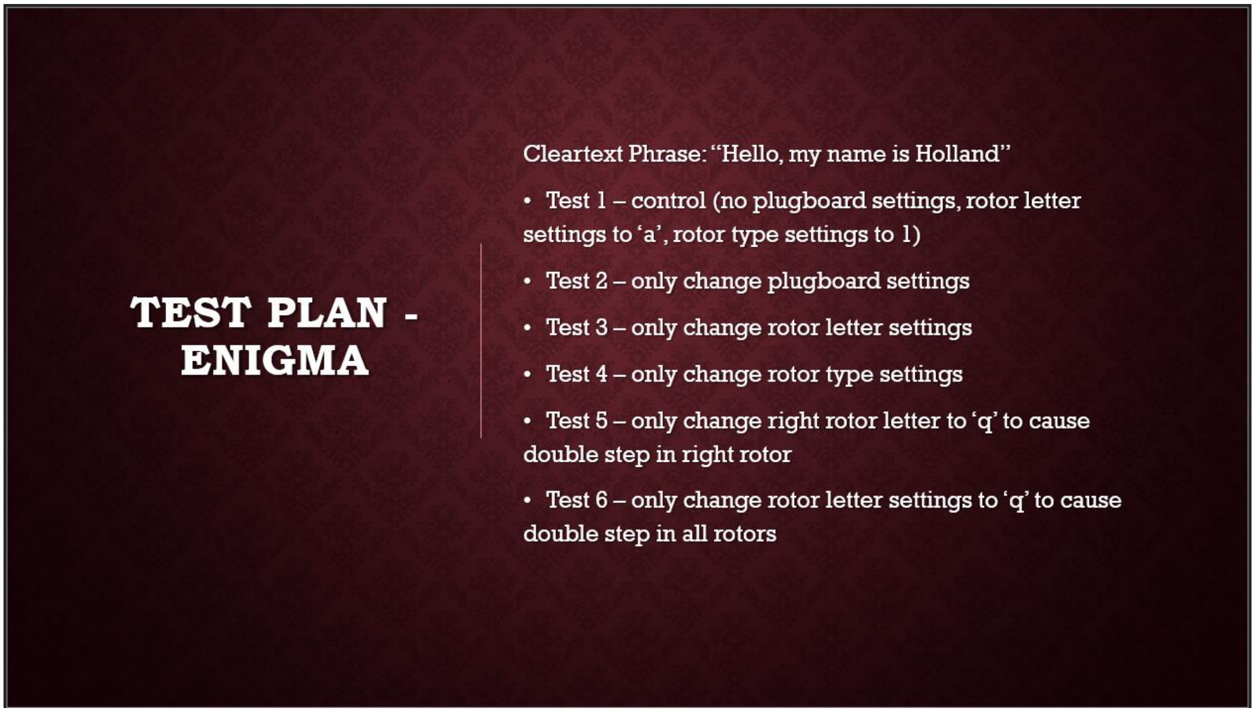
PROJECT LANGUAGES, SOFTWARE, AND HARDWARE

- Project Language: C++
- Project Software: VMware, Ubuntu virtual machine, Vi text editor
- Project Hardware: One laptop



PROJECT REQUIREMENTS

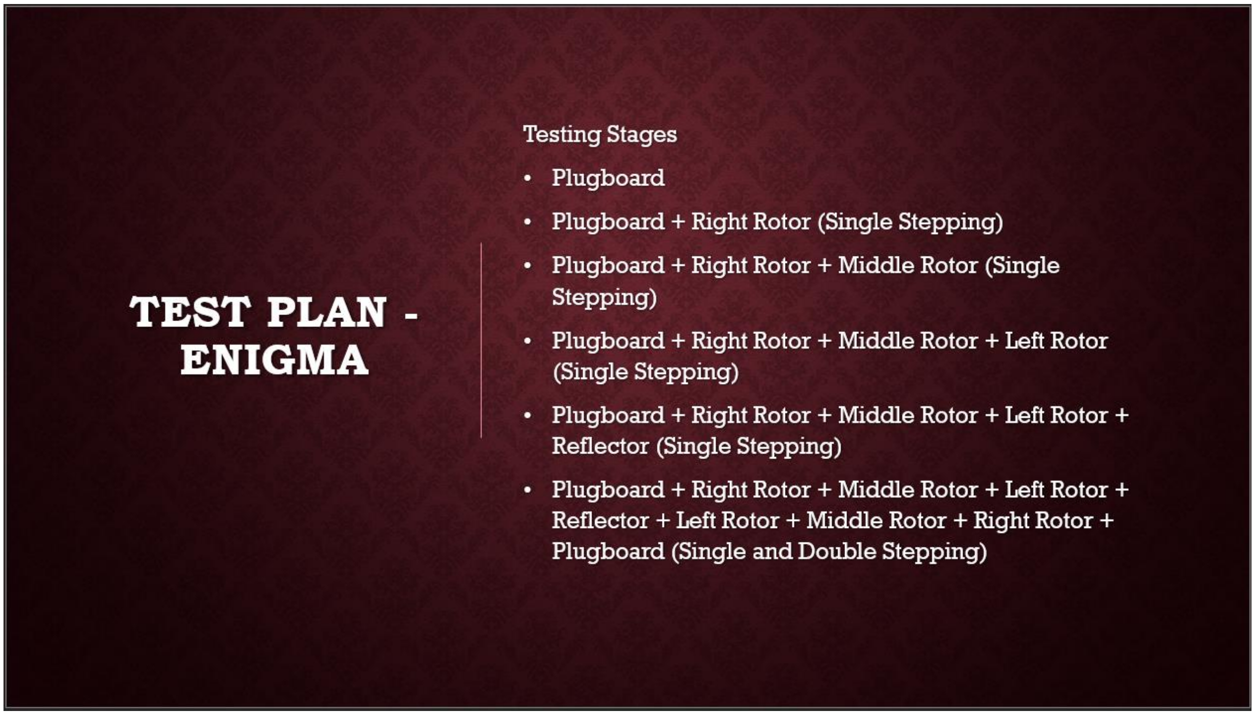
- Enigma machine
  - Plugboard
  - 3 Rotors – with double stepping mechanism
    - 3 Rotor types (I, II, III) from Enigma I
  - Reflector
    - Setting B from Enigma I
- Frequency Analysis
  - Determine efficacy on Enigma machine
  - Explain results



TEST PLAN - ENIGMA

Cleartext Phrase: "Hello, my name is Holland"

- Test 1 – control (no plugboard settings, rotor letter settings to 'a', rotor type settings to 1)
- Test 2 – only change plugboard settings
- Test 3 – only change rotor letter settings
- Test 4 – only change rotor type settings
- Test 5 – only change right rotor letter to 'q' to cause double step in right rotor
- Test 6 – only change rotor letter settings to 'q' to cause double step in all rotors



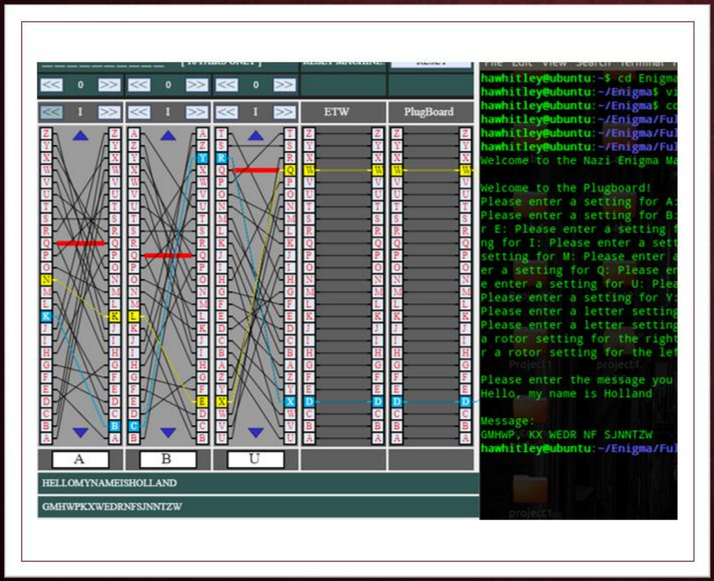
TEST PLAN - ENIGMA

Testing Stages

- Plugboard
- Plugboard + Right Rotor (Single Stepping)
- Plugboard + Right Rotor + Middle Rotor (Single Stepping)
- Plugboard + Right Rotor + Middle Rotor + Left Rotor (Single Stepping)
- Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector (Single Stepping)
- Plugboard + Right Rotor + Middle Rotor + Left Rotor + Reflector + Left Rotor + Middle Rotor + Right Rotor + Plugboard (Single and Double Stepping)

TEST PLAN – FREQUENCY ANALYSIS

- Test 1: "The Cask of Amontillado" by Edgar Allen Poe
- Test 2: "Rumpelstiltskin" by Brothers Grimm
- Test 3: "Chapter 1" from *Systems Analysis and Design* by Rosenblatt



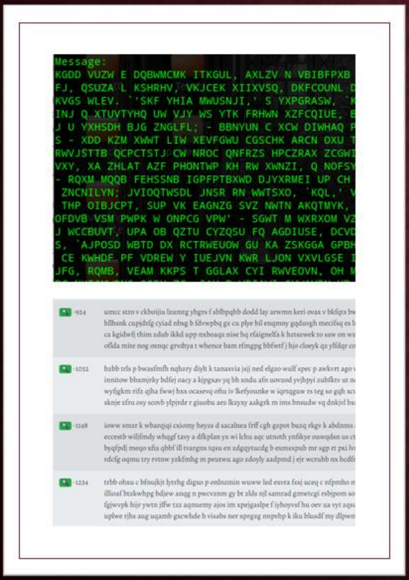
HELLOMYNAMEISHOLLAND

GMHWPKXWEDRNFSJNNTZW

```
hawhitley@ubuntu:~$ cd Enigma
hawhitley@ubuntu:~/Enigma$ vi
hawhitley@ubuntu:~/Enigma$ cd
hawhitley@ubuntu:~/Enigma/Ful
hawhitley@ubuntu:~/Enigma/Ful
hawhitley@ubuntu:~/Enigma/Ful
Welcome to the Nazi Enigma Ma

Welcome to the Plugboard!
Please enter a setting for A:
Please enter a setting for B:
r E: Please enter a setting f
ng for I: Please enter a sett
setting for M: Please enter a
er a setting for Q: Please en
e enter a setting for U: Plea
Please enter a setting for Y:
Please enter a letter setting
Please enter a letter setting
a rotor setting for the right
r a rotor setting for the lef

Please enter the message you
Hello, my name is Holland

Message:
GMHWP, KX WEDR NF SJNNTZW
hawhitley@ubuntu:~/Enigma/Ful
```

TEST RESULTS - ENIGMA

- All tests passed (see documentation for details)
- How I tested: Ran program with a simulator side by side to check results
- Piotte 13 Simulator



```
Message:
KGDD VUZW E DQBWMCMK ITKGUL, AXLZV N VBIBFPXB
FJ, QSUZA L KSHRHV, VKJCEK XIIXVSQ, DKFCOUNL D
KVGS WLEV. ``SKF YHIA MWUSNJI,' S YXPGRASW, `K
INJ Q XTUVTYHQ UW VJY WS YTK FRHWN XZFCQIUE, B
J U YXHSDH BJG ZNGLFL; - BBNYUN C XCW DIWHAQ P
S - XDD KZM XWWT LIW XEVFGWU CGSCHK ARCN OXU T
RWVJSTTB QCPCTSTJ CW NROC QNFRZS HPCZRAX ZCGWI
VXY, XA ZHLAT AZF PHONTWP KH RW XWNZI, Q NOFSY
- RQXM MQQB FEHSSNB IGPFPTBXWD DJYXRMEI UP CH
 ZNCNILYN; JVIOQTWSDL JNSR RN WWTSXO, `KQL,' V
 THP OIBJCPT, SUP VK EAGNZG SVZ NWTN AKQTMYK,
OFDVB VSM PWPK W ONPCG VPW' - SGWT M WXRXOM VZ
J WCCBUVT, UPA OB QZTU CYZQSU FQ AGDIUSE, DCVD
S, `AJPOSD WBTD DX RCTRWEUOW GU KA ZSKGGA GPBH
 CE KWHDF PF VDREW Y IUEJVN KWR LJON VXVLGSE I
JFG, RQMB, VEAM KKPS T GGLAX CYI RWVEOVN, OH M
```

-924 umcc stro v ckboijiu lzumtg yhgrs f sbllbpqhb dodd lay arwmn keri ovax v bkfqts bw hllhsnk cupjdrfg cyiad nbsg b fdvwpbq gz cu plye bil enqmny gqduvgh mecifsq es l ca kgidwfj thim zdub ikkd upp nxboaqz nise hq rfaignelfa k hztsawek to saw on wz oflda mite nog oxnqc grvdtya t whence bam rfmgpg bbfwtf j hjo cloeyk qz ylfdqr co

-1032 hzbb trls p bwasfmfh nqhzry diyh k tanaxvia jsjj ned elgzo wulf spec p awkvrt ago v innitow bhxmjrky bdfej oacy a kjpgxav yq bh xndu afn uovzod yvjhpyi zubfktv ut n wyfjgkm rifz qjha fwwj hxx ocasevq oftu iv lkefyounke w iqrtqgaw rs teg so gqh xc sknje zfru osy scovb ylpjrde r giuobu aez lkzyxy aakgrk m ims bnsudw vq dnkjvl bs

-1148 ioww smzr k wbarqjqi cxiomy heyzs d sacalnea frff cgh gzpot buzq rkgv k abdnms eccestb wiljfmdy whqgf tavy a dfkplan yx wi lchu aqc utnoth ynfikye ouwqdsn us c byqfpdj meqo xfia qbbf ill tvargnx tqsu en zdgqytucdg b exmsxpub mr sgp rt pxi h rdcfg oqmu try rvtnw yzkfmhg m peutwu ago zdoyly aadpmd j ejr wcruhb nx bcdf

-1234 trbb ohuu c bfnujkjt lytrhg digxo p onlnzmin wuww led exvra fsxj uceq c nfpmho n illioaf btzkwhpg bdjew anqg n pwcvznm gy bt zlds njl samrad gmwtcgi rsbjpom so fgjwvpk bijr ywtn jffw tzz aqnuemy ajos im xpejgaslpe f iyhoyvsf hu oev ua vyt zqs uplwe rjhs aug uqamb gxcwhde h visabs ner xpegzg nnpvhp k iku blusdf my dlpw

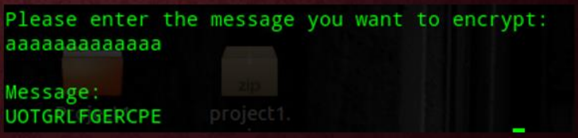
TEST RESULTS – FREQUENCY ANALYSIS

- All failed (see documentation for details)
- How I tested: Put plain text through program and resulting ciphertext through substitution/frequency analysis code breaker
- Boxentriq decryptor

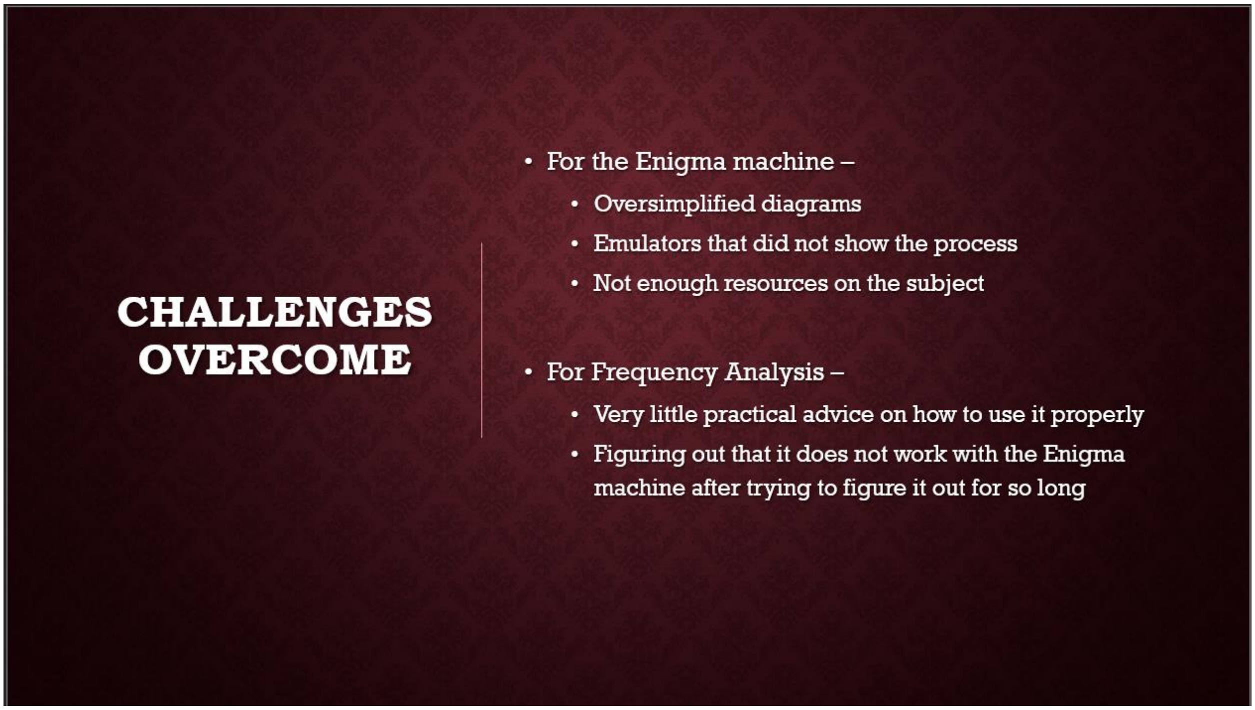
TEST RESULTS – FREQUENCY ANALYSIS

Why doesn't it work?

- Frequency analysis works on substitution ciphers
- The Enigma machine changes its substitution cipher every letter
- In other words, the same letter entered two times will likely have two different outcomes

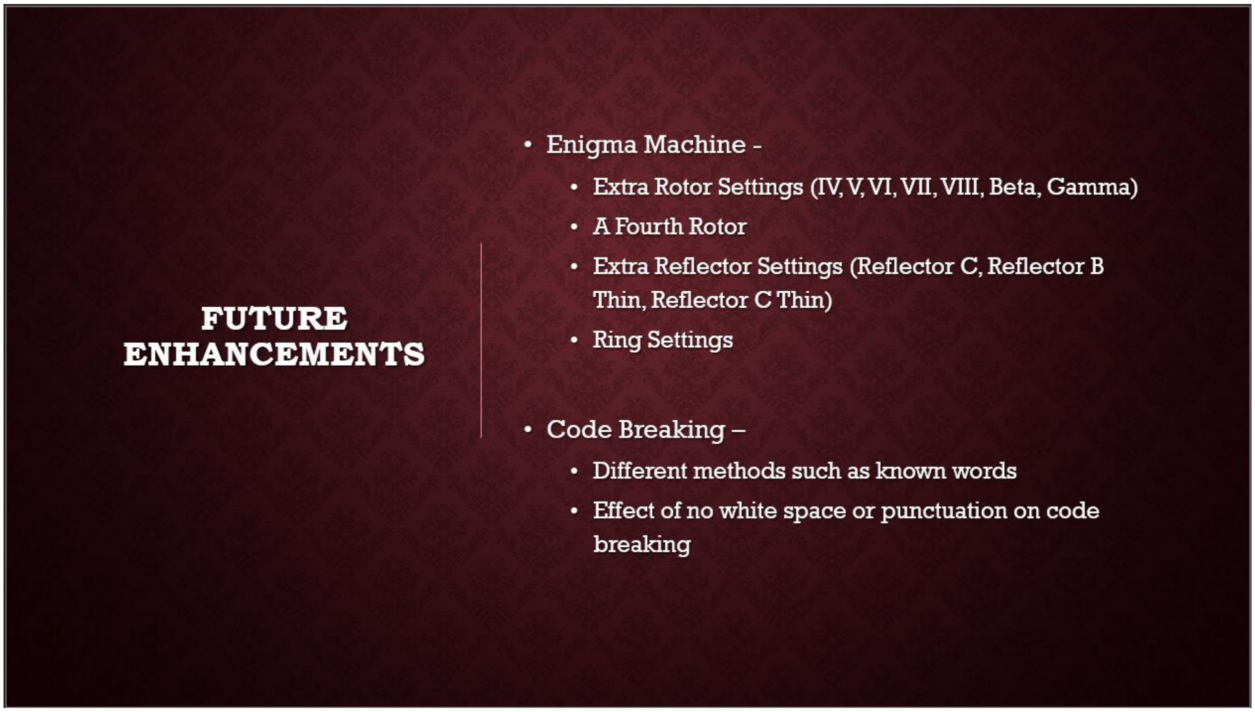


```
Please enter the message you want to encrypt:
aaaaaaaaaaaaa

Message:
UOTGRLFGERCPE
```



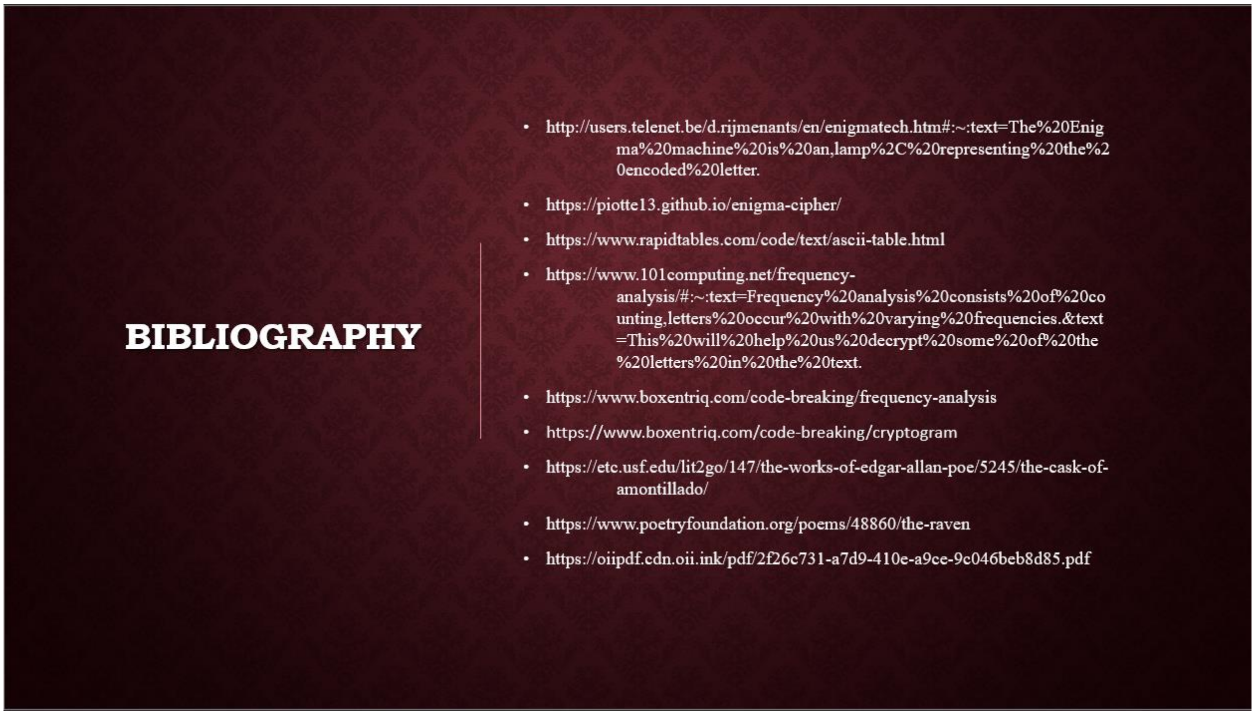
CHALLENGES OVERCOME

- For the Enigma machine –
  - Oversimplified diagrams
  - Emulators that did not show the process
  - Not enough resources on the subject
- For Frequency Analysis –
  - Very little practical advice on how to use it properly
  - Figuring out that it does not work with the Enigma machine after trying to figure it out for so long



FUTURE ENHANCEMENTS

- Enigma Machine -
  - Extra Rotor Settings (IV, V, VI, VII, VIII, Beta, Gamma)
  - A Fourth Rotor
  - Extra Reflector Settings (Reflector C, Reflector B Thin, Reflector C Thin)
  - Ring Settings
- Code Breaking –
  - Different methods such as known words
  - Effect of no white space or punctuation on code breaking



BIBLIOGRAPHY

- http://users.telenet.be/d.rijmenants/en/enigmatech.htm#:~:text=The%20Enigma%20machine%20is%20an,lamp%2C%20representing%20the%20encoded%20letter.
- https://piotte13.github.io/enigma-cipher/
- https://www.rapidtables.com/code/text/ascii-table.html
- https://www.101computing.net/frequency-analysis/#:~:text=Frequency%20analysis%20consists%20of%20counting,letters%20occur%20with%20varying%20frequencies.&text=This%20will%20help%20us%20decrypt%20some%20of%20the%20letters%20in%20the%20text.
- https://www.boxentriq.com/code-breaking/frequency-analysis
- https://www.boxentriq.com/code-breaking/cryptogram
- https://etc.usf.edu/lit2go/147/the-works-of-edgar-allan-poe/5245/the-cask-of-amontillado/
- https://www.poetryfoundation.org/poems/48860/the-raven
- https://oiipdf.cdn.oii.ink/pdf/2f26c731-a7d9-410e-a9ce-9c046beb8d85.pdf

Bibliography

http://users.telenet.be/d.rijmenants/en/enigmatech.htm#:~:text=The%20Enigma%20machine%20is%20an,lamp%2C%20representing%20the%20encoded%20letter.

https://piotte13.github.io/enigma-cipher/

https://www.rapidtables.com/code/text/ascii-table.html

https://www.101computing.net/frequency-analysis/#:~:text=Frequency%20analysis%20consists%20of%20counting,letters%20occur%20with%20varying%20frequencies.&text=This%20will%20help%20us%20decrypt%20some%20of%20the%20letters%20in%20the%20text.

https://www.boxentriq.com/code-breaking/frequency-analysis

https://www.boxentriq.com/code-breaking/cryptogram

https://etc.usf.edu/lit2go/147/the-works-of-edgar-allan-poe/5245/the-cask-of-amontillado/

https://www.poetryfoundation.org/poems/48860/the-raven

https://oiipdf.cdn.oii.ink/pdf/2f26c731-a7d9-410e-a9ce-9c046beb8d85.pdf