Final Report

Abstract

*Enigma Machines were used in the main during World War II by the German military. It was a device which scrambled plain text into ciphered text. This project demonstrates both the enciphering and deciphering of the Enigma machine in a graphical simulation. The simulation demonstrates the movement of the rotors, within the machine, which presents a 3D graphical visualisation of the process of encrypting plain text into cipher text.*

Intro

With the outbreak of wireless communication in the early 1900s, there was a necessity for secure communication, particularly for military. With this came the invention of an Enigma machine in 1918, invented by a German engineer, Arthur Schebius, later the enigma machine patented in 1919. In the 1920s early models were used commercially, and later adopted by Nazi Germany before and during World War II. The Enigma machine was an electro-mechanical device which scrambled a plain text message into ciphered text using a letter substitution system. This enabled the military forces to communicate using coded messages.

In this project a graphical Enigma simulator was developed, which represented the inner working of the process of encryption, plain text to cipher text, as well as the process of decryption, ciphered text to plain text. The simulator will provide a greater detail of the processes in a 3-Dimensional graphical format. The simulation, developed by C++ language, also allowing users to encrypt their own text. In the background section, some historical information about the Enigma machine are presented, which enhance the understanding of Enigma machines.

Background

History

The Enigma machine was invented by a German engineer in 1918 and later adopted by Nazi Germany before and during World War II. The Enigma was a device used by the Germans to communicate with their allies using encrypted messages. The enigma consisted of a keyboard of 26 letters in the pattern of the normal German typewriter, but with no keys for numeric or punctuation characters. Behind the keyboard was a lamp board made up of 26 small circular windows, each bearing a letter in the same pattern as the keyboard, which could light up one at a time. Behind the lamp board was the scrambler unit consisting of a fixed wheel at each end, and a central space for three rotation wheels. Message were limited to a maximum of 250 letters to avoid the inner mechanism returning to the same position because the sequence would repeat itself after 17,576 (26x26x26) key rings. Had the messages not been limited then British code-breakers may have been able to break the encrypted messages. Thus potentially the number of cipher text alphabets was vast and this led German military authorities to believe in the absolute security of this cipher system [1]. The first Enigma machine was heavy and bulky.

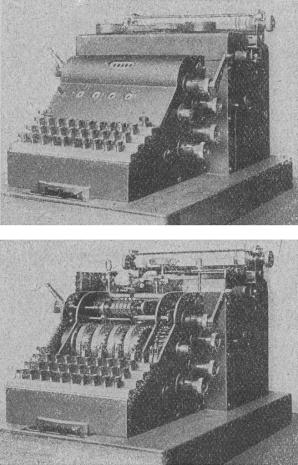


Figure 1 – Enigma A (Crypto Museum, 2008)

Various versions of the Enigma machine were developed, each with varying rotors. In 1926, a commercial version of the Enigma machine was purchased by the German Navy and adapted for military use. A special Enigma was developed by Chiffriermaschinen-AG, which had rotors that have the same contact alignment as the D rotors, but with teeth, multiple notches and are advanced cog wheels instead of pawls and ratchets. This model lead to the Enigma G. The Enigma G had different rotors with a zigzag pin placement and the counter on its right. Its rotors, which also had multiple notches, were moved by a system of gears.

In 1932 the Wehrmacht revised the commercial Enigma D and added the plugboard at the front of the machine. This version, known as Enigma I, became known as the Wehrmacht Enigma and was introduced on a large scale in the Army and public authorities. Initially this enigma came with three rotors, however from 1939 onwards they were equipped with five rotors. The Wehrmacht model was later adopted by the German Navy, with its securer plugboard and the extended set of rotors of eight. [2]

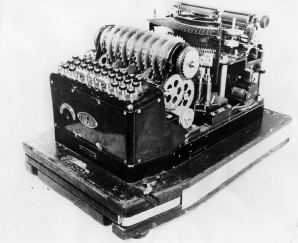


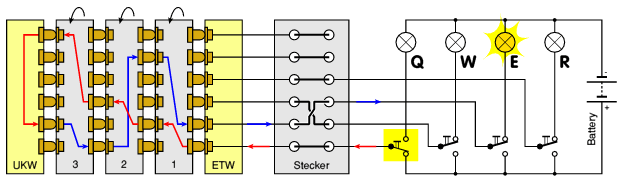
Figure 1-1 – The Wehrmacht Model (Crypto Museum, 2009)

Breaking the code

The Germans believed that the messages being sent to their allies were not breakable. However, the code breakers based at Bletchley Park cracked the secret messages being broadcasted, which played a crucial role in the defeat of Germany. The Polish were the first people to come close to cracking the Enigma code. Marian Rejewski, Henryk Zygalski and Jerzy Rozicki were three mathematicians who successfully cracked the Enigma. They also developed an electro-mechanical machine, called the Bomba, to speed up the code breaking processing. [2] With the invasion of Poland looming, the Poles shared their information with the British, who in turn established the Government Code and Cipher School at Bletchley Park, however it was only in 1941 where their work began to pay off meaningfully when they were able to gather evidence of the planned invasion of Greece. [3]

Bletchley Park

Bletchley Park is the home of the Government Code and Cipher School (GC&CS) based in Milton Keynes, UK, where the Enigma was initially broken. This location was chosen as it is 45 miles north of London with direct railway connections to here, as well as to Cambridge and Oxford, which allowed scientists and army personnel to travel inconspicuously. Alan Turing developed the Bombe, not to been confused by the Bomba which it was in fact based on. He developed a more universal method based on cribs, pieces of guessed plain text, due to the Polish method of exploiting the German vulnerability of the double-enciphered message indicator which could no longer be used. The value of this codebreaking machinery was recognised by the British Prime Minister, Winston Churchill, who introduced a new level of secrecy to supersede all other levels, known as ULTRA. Three additional rotors used exclusively by the Navy and not shared with any other parts of the army. In 1941 Turing discovered the procedure of the additional wheels. [4]

Figure 1-2 Circuit Diagram (Crypto Museum, 2009)

Rotor Wiring

Each rotor had 26 positions, one for each letter of the alphabet. After a key has been pressed the rotor rotates so the next letter is visible. If the letter ‘A’ was active on the key press then on the next key press the letter ‘B’ will be active. Once a full cycle is complete the next rotor will rotate one notch. Once the first rotor reaches the letter ‘Z’ it will ensure the letter ‘A’ is active for the next key press but also if the next rotor was on the letter ‘Q’ then the letter ‘R’ would be active during the next key press. The same would occur for the third rotor once the second rotor has completed a cycle. This results in 17,576 (26 x 26 x 26) possibilities. In addition to the rotors there was a reflector (Umkehrwalze) added on the end and a plugboard (Steckerbrett or Stecker) was introduced to the first Wehrmacht version of the Enigma machine. The reflector redirected the current back to the rotors by a different route. With the exception of the beta and gamma reflectors, each letter was pair with one another. For example ‘E’ and ‘Q’ were paired together on reflector B, so when the rotor passed the current to ‘E’, ‘Q’ would be passed back to that rotor. The plugboard added an extra layer of complexity to the Enigma machine. It was situated at the front of the machine and enabled the key press to map to a different letter on the rotor.

Related Work

We are not aware of any simulators which represent the inner works of an Enigma machine in a 3D graphical representation. However various simulator which encipher and decipher text are available widely on the internet. One simulator in particular caught the attention. [5] It demonstrates the current path when a key is pressed through three rotors, in a simple form. This simulator provided a deeper understanding on how a simulator for this project could be developed.

Specification

*Project Management*

Project Specification

The purpose of this project is to develop a Graphical Enigma Simulator which will demonstrate the process of encryption and decryption. A particular aim is to visually demonstrate the principle of polyalphabetic substitution in operation in the rotors.

Project Plan

A plan of how much time to allocate to each aspect of the project was essential to facilitate good time management and to provide an understanding of how the project is progressing. The aspects of the project are as follow:

Background research

Requirement analysis

Project design

Ethical Approval

Code Implementation

Testing

Interim report

Evaluation  
Final report and Portfolio

A Gantt chart, which is shown in Appendix 1, was created with task list and timescales to ensure the developer could complete tasks on time.

Methodology

Throughout the development of the project the iterative development cycle was utilised. Waterfall and agile methodologies were considered however neither were suitable for this project because the waterfall approach did not allow revision to the project and as for the agile approach not enough information on the structure of the simulator was known at that stage. The iterative approach allowed revision of other parts of the project in stages.



Figure 2 – Iterative model (Voltreach, 2011)

Source Control

To reduce the risk of file corruption and deletion from the developer’s local machine, GitHub was used. GitHub is a repository hosting service, which also offers revision control. This allows regular backups to be made and what aspects of the project has been done.

Ethical Approval

User participation was required to carry evaluation therefor ethical approval from the School of Computing Ethics Committee was required. The ethics form and approval letter can be found in Appendix 2 and Appendix 3.

Meeting with Supervisor

Weekly meetings were arranged with the project supervisor. During meetings, the tasks that had been carried out the previous week was discussed. The supervisor provided suggestions and improvements to the project to ensure the requirements could be fulfilled. Insight was also provided on difficulties encountered by the developer and the suggested next steps for the following week.

Requirement Elicitation

The first step in the development lifecycle was requirements. The main source for gathering requirements was from meetings with the supervisor.

Functional Requirements

A set of functional requirements were established detailing each of the functions the simulator should facilitate.

Main Menu

The simulator shall have a main menu.

Main Menu – options

The main menu shall contain three options: Encrypt, Decrypt and Exit.

Encrypt Option

The simulator shall allow for encryption process to be simulated after selecting Encrypt from the Main Menu.

Decrypt Option

The simulator shall allow for decryption process to be simulated after selecting Decrypt from the Main Menu.

Exit Option

This option shall allow the simulator to close once after selecting Exit from the Main Menu.

Encryption

The simulator shall scramble plain text into cipher text.

One Rotor – Encryption

The simulator shall demonstrate the operation of encryption in one rotor.

Three Rotors – Encryption

The simulator may demonstrate the operation of encryption in three rotors.

Decryption

The simulator shall unscramble cipher text into plain text.

One Rotor – Decryption

The simulator shall demonstrate the operation of decryption in one rotor.

Three Rotors – Decryption

The simulator may demonstrate the operation of decryption in three rotors.

Visual Representation

The simulator shall visually demonstrate the principle of poly-alphabetic substitution in operation in the scrambling unit of an enigma machine.

Animation

The simulation shall be demonstrated using animation.

Attack Method

The simulator may include an attack method, which could become a game, where the user would guess the encrypted plain text.

Non-Functional Requirements

A set of non-functional requirements were established detailing requirements of the implementation.

Graphical User Interface

The interface shall be presented in a graphical format.

Operating System – Windows

The simulator shall be compatible on Windows Operating Systems.

Operating System – Mac/Linux

The simulator may be compatible on Mac/Linux Operation Systems.

Development

The simulator should be developed using C++ and Visual Studio IDE.

Design

Implementation and Testing

Considerations

Evaluation

Usability

Other Criteria

Summary and Conclusions

Acknowledgment

References

[1] **Royal Naval Museum**. ‘The Enigma Machine’ [Online].

Available at:

http://www.royalnavalmuseum.org/info\_sheets\_enigma.htm

(Accessed at: 27/10/2014)

[2] **Rijmenants D.** ‘The German Enigma Cipher Machine’ [Online].

Available at:

http://users.telenet.be/d.rijmenants/en/enigma.htm

(Accessed at: 27/10/2014)

[3] **History** ‘Code Breaking’ [Online].

Available at:

http://www.history.co.uk/study-topics/history-of-ww2/code-breaking

(Accessed at: 11/11/2014)

[4] Enigma History [Online]

Available at:

http://www.cryptomuseum.com/crypto/enigma/hist.htm

(Access at: 17/03/2015)

[5] EnigmaCo.de [Online]

Available at:

http://www.Enigmaco.de

(Access at: 17/03/2015)