Enigma Machine Structure and The Mathematical Principle of The Introduction and Self-experiment

Han Wang

Computer science and engineering school, University of New South Wales, Sydney NSW 2052

1279306866@qq.com

Abstract. The Enigma machine is a kind of advanced mechanical encryption system used by the Nazi German military during World War II, with the rotor structure as the main structure. Cryptographic machines generally take the form of a boxed structure. When encrypting a string of characters, the user enters the information into a machine or system and gets ciphertext. The original information can be obtained by reverse operation of the ciphertext. As the operator enters the message to be encrypted, a sequence of plain-length passwords can be recorded based on the sequence of letters lit up on the lamp board. This article will focus on the simple structure of the Enigma machine and the mathematics behind it, thus illustrating its importance and security in the history of human encryption. Then, to further explore the working principle of the Enigma machine and help to better understand its internal nature, the essay has provided a simple code that can realize the simple function of the Enigma, and also shows a simply equipped Enigma machine.

Keywords: enigma machine, encrypt, decrypt, password, mathematical principle.

1. Introduction

Those who are familiar with World War II history may know that the Enigma was an intelligence-encryption device used by Nazi Germany. It was a cryptographic machine used to encrypt and decrypt documents in order to prevent the Allies from deciphering secret German information. The essence of the Enigma machine algorithm is a symmetric encryption algorithm [1]. That is, after the user enters the information into the machine or system, a ciphertext can be obtained. Also, the original information can be obtained by reverse operation of the ciphertext. In a real application, in order for a message to be encrypted and decrypted correctly, the settings of the Enigma machine that sends and receives the message must be the same; The rotors must be exactly the same, and they must be arranged in the same order, starting positions and connections to the strips. All of these settings need to be determined before use and recorded in the password book. Since the Enigma machine has so many impressive functions in the field of cryptography, this paper will explore the specific working principle of the Enigma machine and its internal structure. The security of the Enigma machine will be proved by mathematical proof. In addition, based on this, the application of the Enigma machine in daily life will be probed further, such as the mathematical principle of using code to simulate the work of the Enigma machine and the internal series network of making a simple Enigma machine to dynamically restore its work.

Figure 1. Enigma machines were used to send messages in World War II [2].

2. Structure and mathematical principle of the enigma machine

The common the Enigma machine is mainly composed of five parts, which are plugboard, rotors, keyboard, lamp board and reflectors.

Figure 2. Plugboard [3].

Figure 3. Keyboard and Lamp board[4].

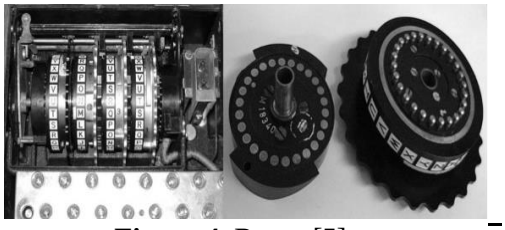

Figure 4. Rotors[5].

Figure 5. Reflector[5].

The outermost structures are keyboard and lamp board, which generally have 26 letters. Unlike keyboard, the letters on the lamp board can be lit. Every time you press the letter on the keyboard, the letter on the lamp board lights up. If you type ciphertext on the keyboard then the lit letters are in plain text. Generally speaking, there are only three rotors in the Enigma machine, and each rotor has 26 numbers corresponding to 26 letters. Each rotor is connected by electrically conductive metal rods to transmit a current signal. Inside each rotor, several wires are scrambled to ensure that every signal sent out is irregular. When the signal passes through the three rotors, it arrives at the reflector. Like the rotor, the reflector has 26 metal rods attached to the rotor and a lot of irregularly connected wires.

The current signal arrives at the reflector, then enters the leftmost rotor, and finally leaves the rightmost rotor. At the same time, each rotor can rotate in a different position, making it difficult to predict which letter will come out of and where it enters. When the electrical signal comes out of the right-most rotor, it will reach the front plugboard. There are many slots on the plugboard, and you can swap two letters by pulling or installing the plug. Suppose you want to switch the letters ‘A’ and ‘D’,

all you had to do was plug into the slot ‘A’ and slot ‘D’, and a connected circuit would form. By doing so, the final result is further encrypted, which also, of course, greatly improves security.

Figure 6. Insert the plug into the slot where you want to convert letters [6].

However, why is the Enigma machine so secure? As mentioned above, each part of the Enigma machine has many different combinations.

In plugboards, we assume that there are cables connected to the plugboard. ($0 \leq p \leq 13$), so, we get the total possible different combinations of sockets are C_{2p}^{26}.

Since each time you plug in the cable, the corresponding socket decreases by one so we can get the connection mode between cables and sockets as:

$$(2p-1)(2p-3)(2p-5)\ldots(3)(1) = (2p-1)!! \tag{1}$$

Hence, the number of different connections that have been made by an Enigma machine is:

$$C_{2p}^{26} \times (2p-1)!! = \frac{26!}{(26-2p)!\times p!\times 2^p} \tag{2}$$

Therefore, we get the total number of board combinations is:

$$\Sigma_{p=0}^{13} \frac{26!}{(26-2p)!\times p!\times 2^p} = 532{,}985{,}208{,}200{,}576 \tag{3}$$

Which is a very huge number

For the three ordered rotors, theoretically, there should be combinations of discs that can be built independently. This is because the disc in the middle of the rotor communicates input points and output points on both sides. However, considering that rotor rotation correction is required in every encoding or decoding process, you can select one of the discs from possibilities of leftmost, also, you can select one of them from of middle and one of them from of rightmost [5].

Hence, the number of all possible combinations is:

$$26! \times (26!-1) \times (26!-2) \tag{4}$$

The third variable is the initial rotation position of the three rotors, because each rotor corresponds to letters, so the total number of different initial positions is 26^3.

In addition, the movable ring on each rotor contains a notch. When the user encodes with an Enigma machine, the rightmost rotor rotates every time. Similar to the decimal mathematical calculation of one in ten. The gap in the right-most rotor causes the middle rotor to rotate once after each rotation cycle (usually 26 times from a certain letter). Similarly, when the middle rotor completes its rotation cycle, its notch forces the leftmost rotor to rotate every strokes of the key. Since the leftmost rotor is the last rotor, all possible combinations are $26^2 = 676$ [5].

The last variable is the reflector. Like the rotor, the reflector has 26 points of contact, but only one connecting surface. On one of the surfaces of the reflector. Unlike the rotor. The reflector is designed with wires connected internally to points of contact, so that electrical signals from the leftmost rotor enter the reflector and return through the rotor, but through different points of contact.

Similar to the position of the plug selected for the plugboard, when the first wire is connected to a random contact, there are 25 different connections at the other end. When the second wire is connected

in the same way as the first wire. There are 23 different connections. And so on, when all the contacts have been connected, we can get the total number of different connections for the reflector:

$$25!! = \frac{26!}{(13! \times 2^{13})} = 7{,}905{,}853{,}580{,}625 \quad (5)$$

Which is also a huge number.

It is obvious that every part of the Enigma machine can produce an unimaginable number of different combinations. Under the conditions of the time, it was difficult to decipher the Enigma machine with human hands or even machines.

3. Expansion

The research split this part into two parts, the first is the code presentation and the second is the self-made the Enigma machine.

In the code display section, the research tried to implement some of the most basic functions of the Enigma machine. Strip out keyboard and lamp board for input and output, and the code represents the rest of the working logic.

In “Plugboard” part, the research wants to randomly generate numbers that correspond to 26 letters between 0 and 25. Because a plug swaps two letters at the same time, the research wants to be able to extract two numbers at the same time for pairing. The role of the for loop is to match the first two digits in the scrambled number. The next step is a simple ASSII code conversion. The reflector section does this using a simple superclass extension.

```
class Plugboard:
    def __init__(self, seed: int, num: int) -> None:
        nums = list(range(26))
        random.seed(seed)
        random.shuffle(nums)
        self.code = {}

        for i in range(0, num * 2, 2):
         a, b = nums[i: i+2]
         self.code[a] = b
         self.code[b] = a

    def reflect(self, letter: str) -> str:
        assert len(letter) == 1
        index = ord(letter) - 65
        if index in self.code:
            return chr(self.code[index] + 65)
        return letter

class Reflector(Plugboard):
    def __init__(self, seed: int) -> None:
        super().__init__(seed, 13)
```

Figure 7. Plugboard code.

The next focus is on the “Rotor”. The previous part, like plugboard, needs to introduce and generate random numbers. At this point, two functions ‘forward’ and ‘backward’ are introduced, representing forward and reverse passing through the rotor respectively. Then the research added a ‘step’ function to determine whether the rotor needed to carry or not, and when it completed 26 turns, the Boolean value was ‘True’ to advance one, and vice versa.

```
ass Rotor:
  def __init__(self, seed: int, shift: int = 0) -> None:
      self.code = list(range(26))
      self.shift = shift
      random.seed(seed)
      random.shuffle(self.code)

  def forward(self, letter: str) -> str:
      assert len(letter) == 1
      index = ord(letter) - 65
      index = (index + self.shift) % 26
      return chr(self.code[index] + 65)
```

```
def backward(self, letter: str) -> str:
    assert len(letter) == 1
    index = ord(letter) - 65
    index = (self.code.index(index) - self.shift) % 26
    return chr(index + 65)

def step(self) -> bool:
    self.shift += 1
    if self.shift == 26:
        self.shift = 0
        return True
    return False
```

Figure 8. Rotor code.

The research ended up creating a class called "Enigma". The ‘process’ function represents the process of decoding or encoding. Of course, the "process" function alone is not enough. The research also needs

a ‘process_appendix’ function to get more stuff passed in each time, instead of typing it again and again. Finally, "step" each time press a key. It is using a variable called "carry" to see if it's running.

```
class Engima:
    def __init__(self, rotors, reflector, plugboard) -> None:
        self.rotors = rotors
        self.reflector = reflector
        self.plugboard = plugboard

    def process(self, letter: str) -> str:
        assert len(letter) == 1
        letter = self.plugboard.reflect(letter)
        for rotor in self.rotors:
            letter = rotor.forward(letter)
        letter = self.reflector.reflect(letter)

        for rotor in reversed(self.rotors):
            letter = rotor.backward(letter)
        letter = self.plugboard.reflect(letter)

        self.step()
        return letter

    def process_appendix(self, sentence: str) -> str:
        return "".join([self.process(letter) for letter in sentence])

    def step(self) -> None:
        for rotor in self.rotors:
            carry = rotor.step()
            if not carry:
                return
```

Figure 9. Enigma code.

Testing the code:

```
message = 'HITOM'

def get_enigma():
        rotor1 = Rotor(1225, 5)
        rotor2 = Rotor(1226, 3)
        rotor3 = Rotor(1227, 1)
        refactor = Reflector(1228)
        plugboard = Plugboard(1229, 6)
        engima = Engima([rotor1, rotor2, rotor3], refactor, plugboard)
        return engima


engima1 = get_enigma()
engima2 = get_enigma()

enc = engima1.process_appendix(message)
dec = engima2.process_appendix(enc)

print(enc)
print(dec)
```

```
SNXRQ
HITOM
```

Figure 10. Testing part.

After testing, the code works!

This part is about how a self-made Enigma machine work:

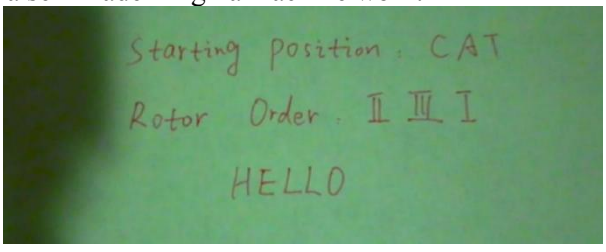

Figure 11. Self-made enigma machine test.

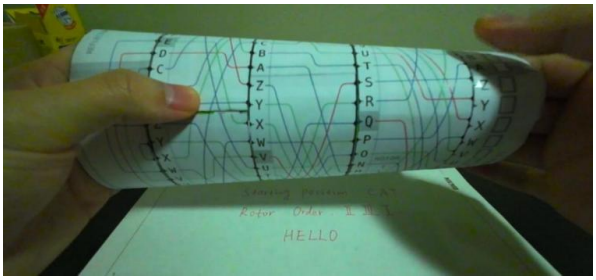

Figure 12. Self-made enigma machine test.

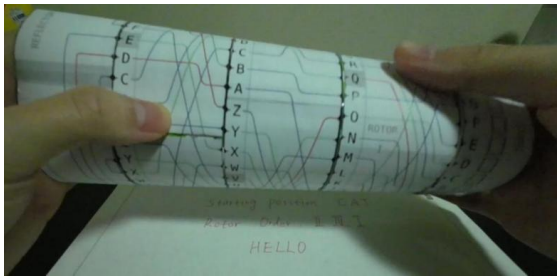

Figure 13. Self-made enigma machine test.

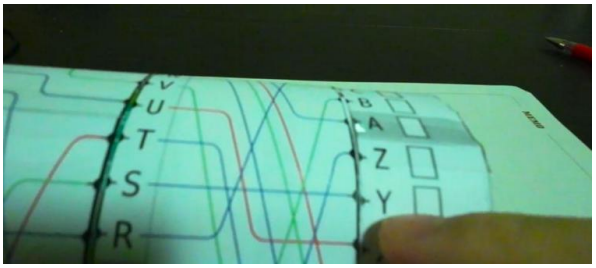

Figure 14. Self-made enigma machine test.

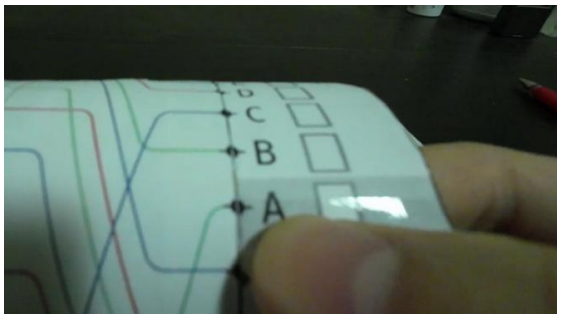

Figure 15. Self-made enigma machine test.

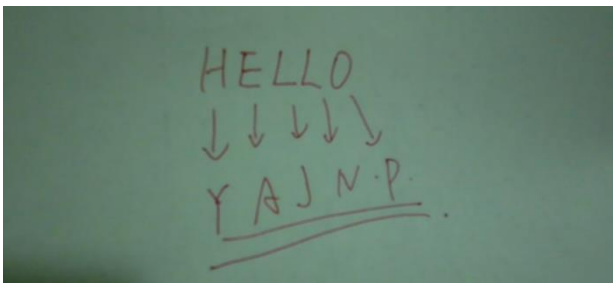

Figure 16. Self-made enigma machine test.

4. Conclusion

In conclusion, the Enigma machine as one of the greatest coding machines of the last century has indeed greatly influenced the future development of cryptography. The mechanical components and circuit components are used to change the input electrical signal to different degrees. Physically, the same input has tens of thousands of different possibilities of output under the coordination and change of different components. This was a remarkable innovation in the evolution of cryptography, and it led to the renewal and acceleration of cryptographic methods.

In this paper, the re-exploration of the structure of the Enigma machine is the integration and induction of the principle of the Enigma machine based on predecessors. From a popular point of view explain the Enigma machine in the physical encryption process and working mode. Since every structure has more or less movable components and different combinations. Therefore, taking this as an opportunity, this paper continues to deeply explore its internal mathematical principles, in a more intuitive way to explain the security of Enigma machine. Of course, what is described in this paper is only to discuss the possibility of different combinations of existing components of Enigma machine, because of the problem of equipment and funds. It has not been discussed in a more in-depth way, which is worthy of attention in the subsequent research.

A wonderful part of this paper is that it has made a basic and valuable restoration for the practical application of Enigma machine in life. Using the code to reveal the working principle of the Enigma

machine, and can carry out some simple encryption and decryption. A simple Enigma machine was made with the materials around, and a decoding process was successfully completed with it. These are the most remarkable points in this study. However, the study of Enigma machine does not end there. In the next stage of research, we will summarize the shortcomings of this experiment and the areas that can be improved, and further explore the problems that have not been involved in this project.

References

[1] Enigma | Definition, Machine, History, Alan Turing, & Facts | Britannica 2022. Enigma. [online] Available at: [Accessed 25 July 2022].

[2] The Machine That Changed the Course of World War II | by Andrei Tapalaga ■ | History of Yesterday 2020. The Machine That Changed the Course of World War II. [online] Available at: [Accessed 25 July 2022].

[3] Enigma Machine | Brilliant Math & Science Wiki 2022. Enigma Machine. [online] Available at: [Accessed 25 July 2022].

[4] The Enigma Machine: keyboard and light panel (arizona.edu) 2022. The Enigma Machine: keyboard and light panel. [online] Available at: [Accessed 25 July 2022].

[5] K, Prasad and M, Kumari (2020), A review on mathematical strength and analysis of Enigma< [2004.09982] A review on mathematical strength and analysis of Enigma (arxiv.org)>

[6] Enigma Steckerbrett (cryptomuseum.com) 2022. Enigma Steckerbrett. [online] Available at: [Accessed 25 July 2022].