The Role of Permutations and Symmetric Groups in Breaking the Enigma Cryptosystem

Cole Hornbeck and Shivang Bokolia
Rochester Institute of Technology
April 2021

"[the cryptologist] tries to compensate his ignorance with long tests, imagination, and sometimes with an ounce of luck"

Marian Rejewski[3]

Contents

1 Introduction

Cryptography has long played a role in warfare, stretching back as far as the Roman Empire, and has only gotten more complex with time. There is no doubt that it's prevalence will only increase in the modern day, ruled by computers that can encrypt and decrypt thousands of messages each second. When looking into the history of cryptography, perhaps the most famous example is the German Enigma Machine. This famous cryptographic system ruled it's class during WWII, and, before it's numerous dials and rotors were cracked, allowed the German army to pass information with no fear of interception. This paper investigates the breaking of this Enigma Cipher, and the role that permutations and group theory played in this massive endeavour.

2 Abstract

This paper discusses about the fatal weakness of Enigma that was exploited by the Allies during WWII. A letter would never encipher to itself. Given enough messages, and a recurring word that could be found in all these messages, sets of permutations would begin to occur. With enough of these sets, the whole cipher would be able to be unravelled. The breaking of the German daily key was done through two critical breakthroughs. First, Marian Rejewski found patterns in the intercepted ciphertexts, and was able to reconstruct the inner workings of the Enigma machine from ciphers alone, without ever seeing a real machine.[3] Second, through intercepting encrypted German weather reports and Alan Turing's Bombe machine. The Allied code breakers could key off of the German 'wetterbericht', using the knowledge that a letter would never encrypt to itself to find where it could likely be, and using enough captured reports they would be able to discover the rotor character permutations, and thus break the key. Even with the introduction of permutation analysis though, breaking of the cipher would have been incredibly difficult or impossible without the capture of a German code book showing the connections and wiring of an Enigma plugboard.[3] This paper explores the discoveries and mathematical processes used by the first man to break the enigma code, Marian Rejewski, and the man who led the British war effort against Enigma, Alan Turing.

3 Definitions

3.1 Permutation

A permutation is an arrangement or rearrangement of a set. The collection of permutations of a set are every distinct way that set could be put together. For example, given the set $S = \{a, b, c, d\}$, three possible permutations of this set are

$$S_1 = \{b, c, a, d\}$$
$$S_2 = \{a, c, d, b\}$$
$$S_3 = \{d, c, b, a\}$$

Permutations play a major role in breaking the enigma cipher due to the rotors containing alphabetical permutations, or various permutations of the alphabetical set $P = \{abcdefghijklmnopqrstuvwxyz\}$. For any set of n distinct objects, there exist $n!$ possible permutations of that set.

3.2 Symmetric Group

The symmetric group S_n of degree n is the group of all permutations on n symbols. S_n is therefore a permutation group of order $n!$ and contains as subgroups every group of order n. Symmetric groups play an important role in many areas of mathematics.

4 History and Design of the Enigma Machine

The Enigma Machine was first invented sometime in the early 20th century. The first patent for it was given to Arthur Scherbius in 1918, and it started to sell in 1923. These machines were sold to businesses, banks, and even private citizens who could afford one. The most well-known customer was, however, the German military. There were different models of Enigma, with the most famous being the military models. These models contained the additional measure of a plugboard in the belly of the device. These boards added an extra layer of security to transmissions. In 1939, 5 weeks before the outbreak of World War II, famed Polish mathematician Marian Rejewski was the first person to crack the German Enigma code.

The enigma machine was made up of five major parts. The keyboard, plugboard, rotors, reflector, and the bulbs. When a key on the keyboard was pressed, it could connect an electrical circuit through these components and light up a bulb on the deck of the machine. This bulb would be the enciphered letter. An overview of the design is shown below in Figure 1.

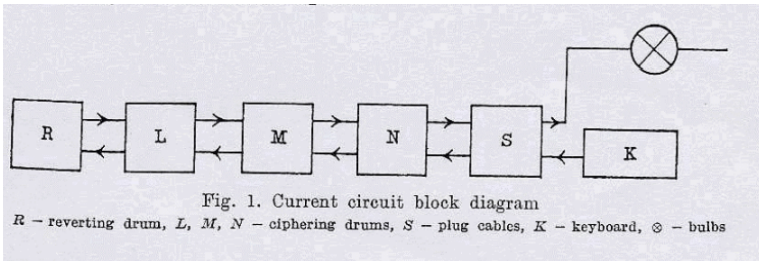

Fig. 1. Current circuit block diagram

R – reverting drum, *L*, *M*, *N* – ciphering drums, *S* – plug cables, *K* – keyboard, ⊗ – bulbs

Figure 1: A high-level block diagram of the enigma machine[3]

4.1 Plugboard

The plugboard was a simple design, yet produced an immense number of total combinations. The boards contained sockets that could connect two letters together. While some Enigma device plugboards contained only ten pairs of sockets, and thus could only swap 20 of the 26 available letters, many contained all 26 letters available in 13 sockets. The pairs could be rearranged as needed very easily, and performed a very basic swap operation on the key pressed. For example, if the 'Y' key was pressed, the plugboard could swap it with the 'G' key, or the 'R' key, or whichever new location it was plugged in to. This added an additional layer of security to the device.

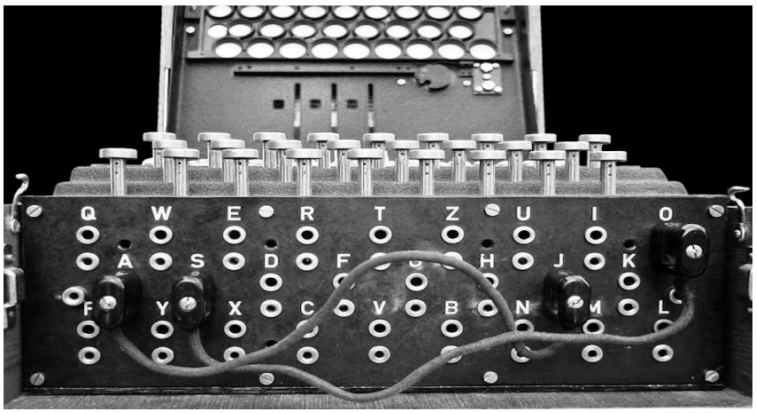

Figure 2: The 13-socket plugboard in an Enigma Machine[2]

4.1.1 Total Number of Combinations for the Plugboard

Assuming the use of a 26-letter, 13-socket plugboard, with only p total cables connecting two letters. There are 26! ways of arranging the 26-letters, but we don't want every combination of the 26-letters, we only want to make p pairs. This means that there are $26 - 2p$ number of letters left over which can left as it is. Since these are left as it is, they need to be divided by the total number of combinations:

$$Combinations = \frac{26!}{(26-2p)!} \tag{1}$$

There are a total of p pairs and the order of these pairs does not matter, hence they should be divided with the total combinations:

$$Combinations = \frac{26!}{(26-2p)! * p!} \tag{2}$$

There are a total of 2 letters in each pair and if the letters were swapped they would still be the same pair (eg. 'A' and 'B' would be the same pair as 'B' and 'A'). Hence we divide the total combinations by 2 but we do that for each pair, which would mean 2^p. Therefore, the total number of combinations of this plugboard can be calculated using Equation 3 below.

$$Combinations = \frac{26!}{(26-2p)! * p! * 2^p} \tag{3}$$

This is further proven in [2].

A very interesting property of this system is that, as shown in the table below, the maximum number of combinations is reached when $p = 11$, as opposed to what might be expected, such as when $p = 13$. This table is pulled from [2] and simply completes the calculation in Equation 3.

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

Figure 3: Table of plugboard combinations given p being the number of plug pairs occupied[2]

As shown in the above table, the number of possible plugboard layouts explodes into hundreds of trillions of possibilities. This is simply incalculable, and so determining the layout and wiring of the plugboard was incredibly important to breaking the cipher.

4.2 Rotors

The standard enigma machine could hold three 26-character rotors, although some more advanced models had the capability of containing up to 8 rotors for advanced encryption. The rotors were arranged in a sequential order, with the signal travelling through one, then the next, and finally the last, before being reflected and travelling back through the rotors. Every time a key was pressed, the first rotor would change by a single character, rotating very slightly. The first rotor was capable of changing 26 times, and upon completion of a full rotation, the second rotor would rotate by one character. Subsequently, once the second rotor had revolved fully, the third rotor would advance one character. Thus, it took a total of $26 * 26 * 26 = 26^3 = 17576$ key presses for a 3-rotor machine to return to it's original position. Also, the 3-rotor machine were chosen out of either 5 or 8 rotors, which increased the complexity for the enigma machine as the 3-rotors could be placed in ${}^5C_3$ or ${}^8C_3$ respectively. Figure 4 below shows a set of three rotors that would have been placed into the machine.

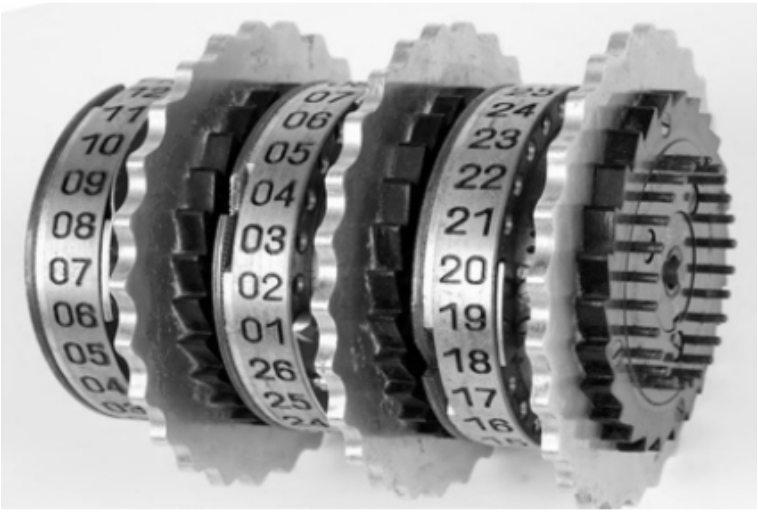

Figure 4: A set of three rotors[2]

These rotors were able to be turned by the user, in order to set the key for the machine. Each day, the 'daily key' would be read from a prepared sheet of keys for the month, and the rotors would be set to that combination of numbers. From there, the message could be typed out and encrypted. Similarly, knowing the daily key allowed for decryption, as setting the rotors correctly and typing in the ciphertext of a message would return the plaintext message. Some models of Enigma also contained a static wheel before the three rotors, which functioned very similar to the rotors, but without rotating, to add yet another level of scrambling. This paper will be covering machines without the static rotor.

Signals are passed through the rotors by means of a complex nest of wiring inside the rotors that transmitted the electrical signal through to the next rotor while changing which letter it appeared to come from. Figure 5 shows a simplified version of the wiring of the rotors, visualizing the electric transfer from one side of the rotor to the other.

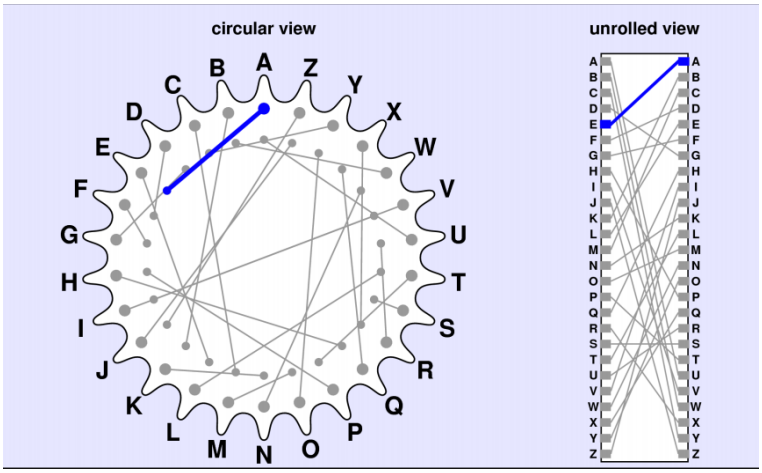

Figure 5: Design of the wiring in a single rotor, connecting inputs to outputs[4]

4.3 Reflector

The reflector was similar to a rotor, but simpler, and it did not rotate. While a rotor passed signals from one side to the other, scrambling them, the reflector simply bounced the signal back from the third rotor to itself, mixing the letter in the process, and allowing all rotors to again perform their permutation. It was very important for the reflector to switch the letter as well, as otherwise, reflecting the signal back through the rotors would simply decrypt the half-encrypted character. It could be thought of as a one-sided rotor. Adding the reflector component was a very simple way of doubling the number of encryption steps that the input letter receives.

Figure 6 below shows a combined view of three rotors with the reflector, indicating how it would be hooked into the system. As can be seen, it serves the purpose of redirecting the signal back through the rotors.

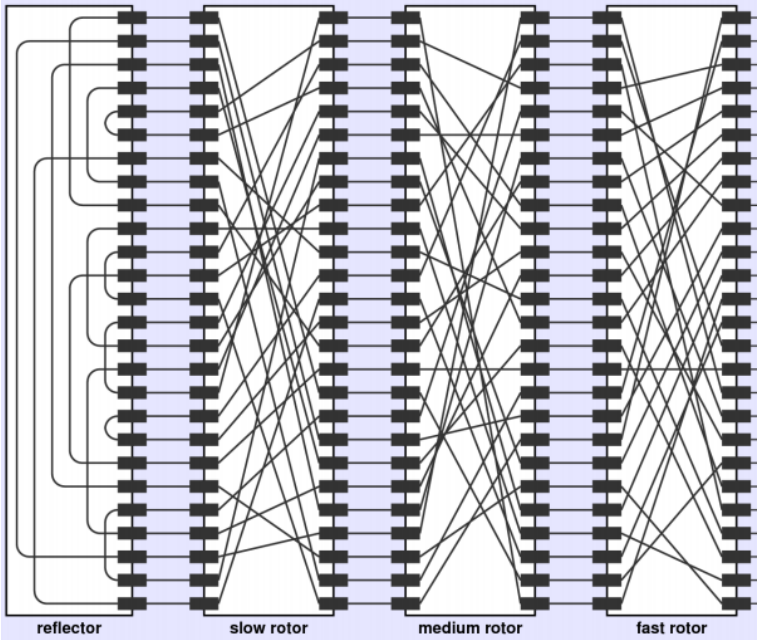

Figure 6: Combination of three rotors with the reflector[4]

5 Use of Permutations and Symmetric Groups

The German military used two slightly different versions of Enigma. The Army and Air force used the straightforward method, with a daily key that was passed around on sheets, one month of keys at a time. This worked well, but was broken through the use of repeating messages, as will be shown below. The German Navy used a slightly different system. The daily key would be used to encrypt a 3-digit 'message key'. This message key would be encrypted twice and added to the front of the message for a 6-character message key. The recipient would receive the message and have to decode the message keys before they could read the whole message. Both of these approaches will be studied and explained in the below sections, with the Daily Code playing a large role in breaking the Message Code.

5.1 The Polish Bomba

The Poles were the first to break the military variant of the Enigma in 1932. A young Polish mathematician Marian Rejewski had recovered the wiring of the military Enigma machine and later worked with two other mathematicians Henryk Zygalski and Jerzy Rozyki to recover the daily Enigma keys. In 1938, Rejewsi came up with a solution called the "Bomba Kryptologiczna" (cryptologic bomb), often abbreviated to Bomba. The Bomba was based on the principle that the random 3-letter message key was sent twice at the beginning of each message and that every now and then, a particular plaintext letter, yields the same ciphertext letter three positions further on. The main mechanism has been explained in the section below.

5.2 Marian Rejewski's Characteristic Sets

The message key consisted of a 3-character key that was enciphered twice using the daily key, creating a 6-character cipher key. This key was placed at the beginning of the message. Due to the nature of the Enigma machine, this resulted in the 6-character key $ABCDEF$ containing

permutations related to AD, BE, and CF, since the first, second, and third characters in each 3-digit key would be known to be the same. This gave code breakers a stepping point from which to begin.

As an example, consider the collection of captured 6-character message keys

$$k_1 = dmqvbn$$
$$k_2 = vonpuy$$
$$k_3 = pucfmq$$

From the first and fourth characters of the keys, it can be seen that d is substituted for v, v is substituted for p, and p is substituted for f. This reveals the beginnings of the permutation AD, containing $dvpf$. Further analysis on these keys reveals

$$AD = dvpf$$
$$BE = oumb$$
$$CF = cqny$$

Collection of further messages containing keys from that day could be used to create the entire permutations, called "the characteristic set or, directly, the characteristics of a given day"[3].

Recalling Figure 1, the block names of the various components of the enigma machine will be used to denote their position in the overall permutation. S denotes the plugboard permutation, L, M, and N are the rotor permutations, and R is the permutation of the reflector at the end of the chain. A preliminary permutation[3] can be set up of

$$C = SNMLRL^{-1}M^{-1}N^{-1}S^{-1} \tag{4}$$

The inverted permutations are derived from the signal returning from the reflector to pass once again through the rotor permutations, although backwards this time. As such, it can be seen that in this configuration, if not for the reflector, the entire system would cancel itself out and return the exact message. Marian Rejewski, one of the first people to reason out the internal workings of the Enigma machine, used these calculations further to create a replica Enigma for study. His equations assumed only one rotor turning during use, and that motion was seen as the permutation $P = \{abcdefghijklmnopqrstuvwxyz\}$ to transform each letter to the next. As the signal leave the keyboard, the plugboard operates normally, but the first rotor has turned and thus interacts with P^{-1}. Similarly, on the way back, the rotor was already turned and so the first rotor interacts simply with $P^{-1^{-1}} = P$, but the plugboard now goes through P^{-1}. The second and third rotor are seemingly not effected by the first rotor shift[3]. As such, the permutation in Equation 4 becomes

$$C = SPNP^{-1}MLRL^{-1}M^{-1}PN^{-1}P^{-1}S^{-1} \tag{5}$$

The unknown permutations A through F can now be written as

$$A = SP^1NP^{-1}MLRL^{-1}M^{-1}P^1N^{-1}P^{-1}S^{-1}$$
$$B = SP^2NP^{-2}MLRL^{-2}M^{-2}P^2N^{-2}P^{-2}S^{-2}$$
$$C = SP^3NP^{-3}MLRL^{-3}M^{-3}P^2N^{-3}P^{-3}S^{-3}$$
$$D = SP^4NP^{-4}MLRL^{-4}M^{-4}P^2N^{-4}P^{-4}S^{-4}$$
$$E = SP^5NP^{-5}MLRL^{-5}M^{-5}P^2N^{-5}P^{-5}S^{-5}$$
$$F = SP^6NP^{-6}MLRL^{-6}M^{-6}P^2N^{-6}P^{-6}S^{-6}$$

And will produce the characteristic set permutations of

$$AD = SP^1NP^{-1}MLRL^{-1}M^{-1}P^1N^{-1}P^3NP^{-4}MLRL^{-1}M^{-1}P^4N^{-1}P^{-4}S^{-1}$$
$$BE = SP^2NP^{-2}MLRL^{-1}M^{-1}P^2N^{-1}P^3NP^{-5}MLRL^{-1}M^{-1}P^5N^{-1}P^{-5}S^{-1}$$
$$CF = SP^3NP^{-3}MLRL^{-1}M^{-1}P^3N^{-1}P^3NP^{-6}MLRL^{-1}M^{-1}P^6N^{-1}P^{-6}S^{-1}$$

In these three permutation sets, the characteristic sets are known, as is P, and the job of the cryptographer, and in the first case of Marian Rejewski, is to find the value of S, L, M, N, and R. Substitution of Q for the repeated value $MLRL^{-1}$ brings the number of unknowns down to only 3, which becomes more solvable.

Marian Rejewski in [3] utilizes the Theorem on the Product of Transpositions, which states *If two permutations of the same degree consist only of disjoint transpositions, then their product contains an even number of disjoint cycles of the same length.* He uses this, along with it's reverse theorem, to arrive at the knowledge of the left side of the equations

$$A = SP^1NP^{-1}QPN^{-1}P^{-1}S^{-1}$$
$$B = SP^2NP^{-2}QPN^{-1}P^{-2}S^{-1}$$
$$C = SP^3NP^{-3}QPN^{-1}P^{-3}S^{-1}$$
$$D = SP^4NP^{-4}QPN^{-1}P^{-4}S^{-1}$$
$$E = SP^5NP^{-5}QPN^{-1}P^{-5}S^{-1}$$
$$F = SP^6NP^{-6}QPN^{-1}P^{-6}S^{-1}$$

At this point, the process stalled until the capture of a German code book dictating the locations of the plugboard spokes in S, which allowed it to be known and the equation made solvable.

5.3 Turing-Welchman Bombe

Based on the information presented by the Polish code breakers, the British Mathematician Alan Turing developed a machine that was capable of recovering the key settings even if the Germans would drop the double encryption of the message key at the beginning of each message. This machine was called the Bombe (later known as The Turing-Welchman Bombe). The Bombe was designed to carry out a systematic search to determine the following components of an Enigma Key: the rotor order, the rotor core starting positions, and some of the plugboard pairings. Although the Polish Bomba concept was known, Turing decided on taking a different approach. Bearing in mind

the massive number of possible keys, if a search was to be carried out then it was utmost importance to find a way that would greatly reduce the size of the 'key space' that was involved. Turing devised such a procedure that depended on a 'crib' of the part of the plaintext of the enciphered message. A 'crib' can be considered a sequence of letters from the plain-text of a message that can be matched one to one with some of the letters from the ciphertext. Another big flaw of the Enigma machine that was exploited by Turing was that a letter never encrypted to itself (eg. A will never encrypt to A or B to B).

5.4 Turing-Welchman Bombe Mechanism

Every morning, the Germans would send a weather report. This report was generally of the same format almost everyday other than the weather itself. A word that occurred commonly on the report was "weather report" or "wetterbericht" in German. The word wetterbericht is considered as the crib in this case. The word is compared to an enigma code to check if the word "wetterbericht" fits in the message. The factor Turing was exploiting here was that a letter cannot encrypt to itself. Consider the enigma code to be: "...jxatqbggywcrybgdt...".

The word "wetterbericht" was placed in parallel with "...jxatqbggywcrybgdt..." and at letter-4 both the plaintext and the enigma code have the same letter 't'. The plaintext is moved by a position compared to the enigma code, hence "wetterbericht" was then placed parallel with "xatqbggywcrybgdt...". But again the letter t on position 3 matches for both the texts and hence the word "wetterbericht" is moved by another position. The enigma code at this point was "atqbggywcrybgdt..." and the word "wetterbericht" did not match any letter with the enigma code. At this point, it was assumed that "atqbggywcrybgdt..." could be considered to be "wetterbericht". This was were breaking of the enigma machine began.

The Bombe machine tried to work out the plug board for the Enigma Machine[1]. When a letter was pressed, the signal goes through the plugboard first and then through the three rotors and then loops back and goes through the three rotors again but in reverse order and finally goes through the plugboard one more time and lights up the code for that letter. For the word "wetterbericht", take the second letter for both the enigma code and the plaintext i.e. "t" becomes "e". This can be used to work out the plugboard.

Assume that the letter 't' was connected 'a' on the plugboard and can be written as Assume(ta). Looking at the above mechanism of the signal channel, the letter 't' goes through the clip board and becomes 'a'. The 'a' goes through the rotors and since the wiring of the rotors is known, the letter at the end can be calculated and in this case can be considered to be 'p'. The letter 'p' goes through the plugboard and is assumed to become 'e'. It can be deduced that 'p' is connected to 'e' on the plugboard and can be written as Deduce(pe). When the same is done again and again using the wetterbericht crib, several deduce statements can be made: Deduce(kq), Deduce(xb), Deduce(tg). The last deduce statement received is a problem. It states that the letter 't' and 'g' are connected on the plugboard, but at the starting of this test it was assumed that 't' and 'a' were connected on the plugboard, this is a contradiction and the assumption for 't' and 'a' was wrong.

Now we assume the next 't' and 'b' or 't' and 'c' or 't' and 'd' and so on for all the 26 options. If all the 26 options were wrong that would mean the rotor position was wrong and the rotor is changed by one and the whole process is repeated again.This would take a large amount of time but Turing came up with two ways to make this slightly quicker. The first one was that once

a mistake has been found like 'ta' and 'tg' in the above example that would mean all the other deduced statements related to these are also wrong ('kq' and 'xb' in the above example) and do not need to be checked. The second way was doing the whole process with electrical circuits and that's where the Bombe machine came in. The Bombe machine did all of the computation using electrical circuits. It applied an electrical current to the assumption 'ta' and flowed through the machine to find 'tg' and other deductions and found all the deductions that were wrong instantaneously and then moved the rotor by one. All of this was done by the Bombe machine in about 20 minutes.

Shortly afterwards, the Bombe was improved by adding the diagonal board which was an invention by Gordan Welchman a fellow code-breaker with Alan Turing. The diagonal board greatly reduced the number of steps needed for the code-breaking efforts.

6 Conclusion

Regarding the rotors and components of the enigma machine as nothing more than mathematical permutations was a huge step forward in the fight against the German codes. With Alan Turing's Bombe machine breaking the German daily codes each day, and Marian Rejewski's method to intercept messages to recreate the enigma machine, mathematics played a pivotal role in the struggle against the Axis powers. It's unknown what may have transpired without these brilliant minds working around the clock on a mathematical problem that was widely considered, at the time, impossible.

References

[1] Lee A. Gladwin. *Alan Turing, Enigma, and the Breaking of German Machine Ciphers in World War II*. Tech. rep. American Government Archives, 1997.

[2] Kalika Prasad and Munesh Kumari. *A Review on Mathematical Strength and Analysis of Enigma*. Tech. rep. Department of Mathematics, Central University of Jharkhand, India, 2020.

[3] Marian Rejewski. "An Application of the Theory of Permutations in Breaking the Enigma Cipher". In: *Applicaciones Mathematicae* 16.4 (1980).

[4] Eric Roberts and Jerry Cain. *The Enigma Machine*. Tech. rep. Department of Computer Science, Standford University, 2017.