Could you have solved it? The maths behind the Enigma

By Teresa Holdsworth Ubeda

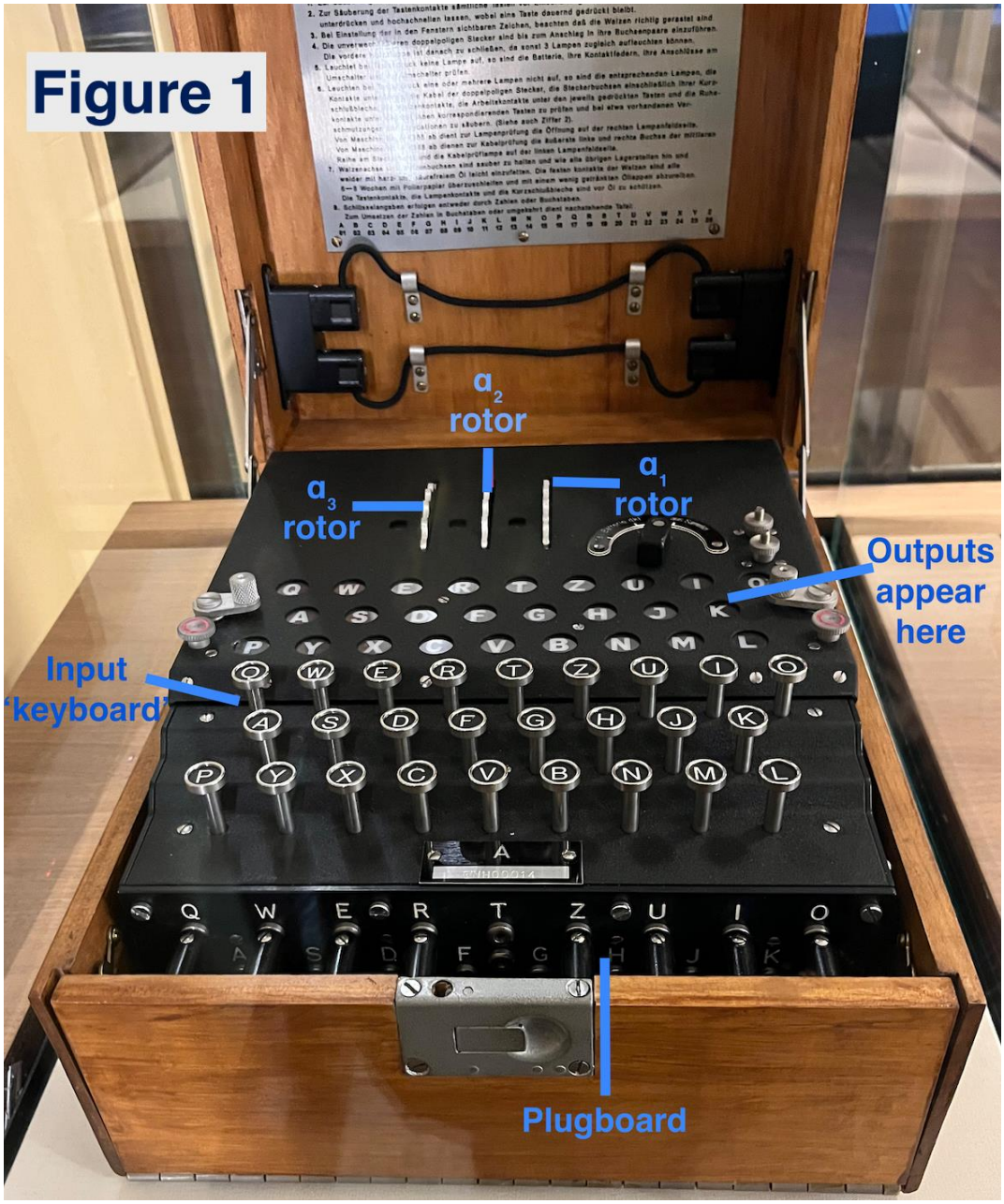
1 Introduction

The year is 1940 and the Second World War continues, as Nazi Germany invades Poland, then France and many other countries. Across the English Channel, Nazi bombs fall over Europe, causing huge damage. In the relative safety of the English countryside, orange leaves fall from trees as you travel past on a train. Along with a group of people from varying specialities, including engineers, chess grandmasters and linguists, you gather at a top-secret location called Bletchley Park. Warfare has always relied on sending secret military messages, but letting the opposition understand these would allow them to avoid or counter attacks. Encryption, scrambling a message so only recipients with prior knowledge can decrypt it, is therefore essential. Realising they needed an infallible encryption method, Nazi officials turned to the Enigma, a machine used to encrypt banking transactions before WW2. Your job, alongside the unique people around you, will be to decipher the codes it produces, giving the Allies vital intelligence. As a mathematician, you hope to use cryptography to find a solution. The method I'll be detailing below was invented by Marian Rejewski, a Polish mathematician who never actually visited Bletchley Park during WW2, but his solution is brilliant and understandable, a huge feat given the complexity of the Enigma!

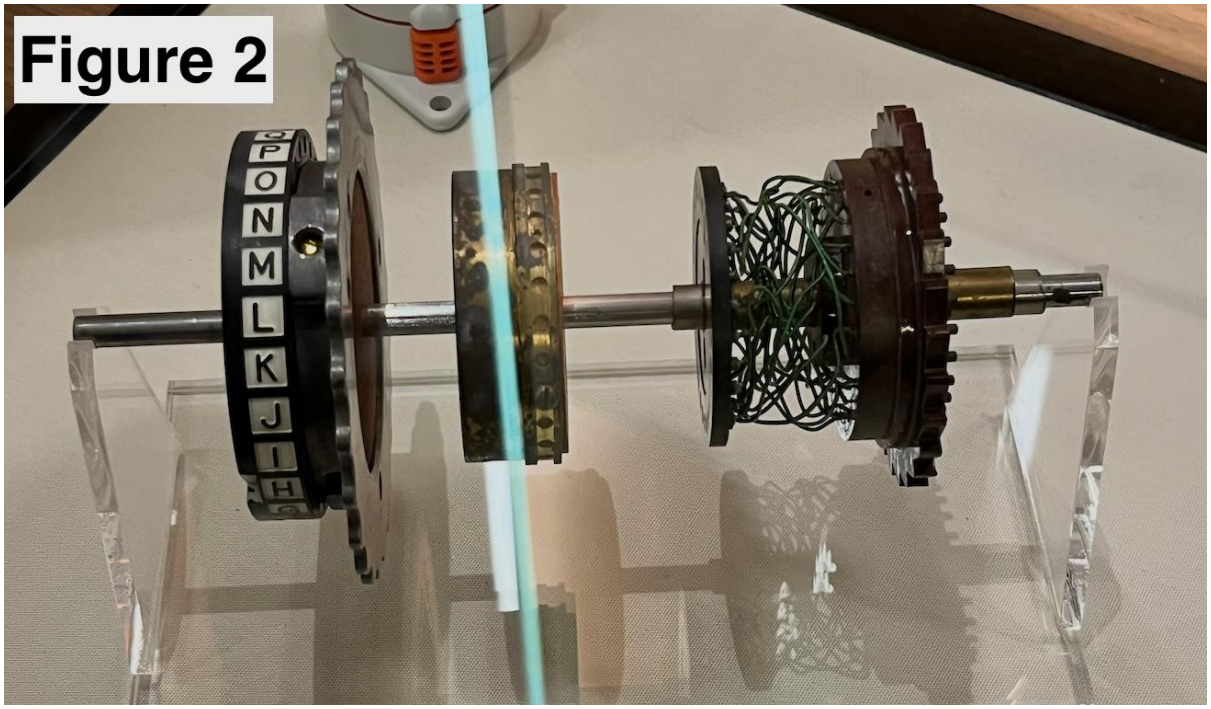
2 The Enigma

As you've accepted your position, here's your first briefing! Before you consider the maths you'll need for the decryption, you'll need to know what the Enigma is and how it works. I'll focus on the design used before WW2, as it's simpler to understand, allowing us to focus on the maths sooner.

As an electromechanical device, it uses both electricity and mechanics to encrypt the inputs. Here's a picture of an Enigma I took at Bletchley Park (Figure 1)!



It has a ‘keyboard’ where letters can be inputted, and above it are embedded circles with letters on them and three rotors. The output is shown when a corresponding letter illuminates after the original letter is entered. In front is the plugboard, which I wont talk about in this essay because of how intricate it is. Electrical current travels through wires which connect components together inside the machine. Pressing a letter on the keyboard causes an electrical current to flow to the rotors (Figure 2):



The rotors are cylinders with a jumble of wires within them. Twenty-six copper pins are arranged in a circle on both faces, serving as connections between rotors and the rest of the machine, allowing the current to pass through. Every pin on one side is attached to a wire, which is connected to a pin in another position on the opposite side. This scrambles the output letter. For example, 'X' could enter rotor α_1(Figure 1) and exit as 'S'. This repeats in the other two rotors before reaching the reflector, which comprises twenty-six wires forming loops (pairing up letters) so the current can be sent back through rotor α_3. Due to the reflector, a letter can't be encrypted as itself. The current then travels back through the three rotors, before lighting up a bulb that shows the output letter. The parts of the Enigma that cannot be changed by the operator are the internal wire structures of the rotors and reflector.

There were many different settings that the operator would have to modify to use the Enigma. Five different rotors were provided, of which three were chosen and arranged varyingly when inserted into the machine. Each rotor had 26 different rotation positions, due to the 26 connection pins on its circular faces. However, this description implies the Enigma is a monoalphabetic substitution cipher, with each input always having the same output, e.g. T is always encrypted to V. If this was the case, using frequency analysis (assuming the most common letter in the ciphered text is the most common letter in the original language), or trial and error would easily solve it.

This is where the mechanical aspect of the Enigma comes in. Every time you press a letter, the rightmost rotor, α_1, rotates forwards by one place (1/26th). This changes the connections between the rotors and the rest of the machine, as each pin moves to the position in front of it while the wires that provide the input/output stay in the same position. Therefore, inputting the same letter can produce a range of outputs

except for the input letter (due to the reflector). Every rotor has a notch, allowing the rotor to its left to rotate forward one position once per every 26 of its own rotations. The position of this notch can be shifted, altering when the second rotor rotates forwards within the first 26 key presses. For example, it can cause the second rotor to rotate on the 3rd key press and then every 26n + 3 key presses. The third rotor will rotate every 26^2 = 676 letters, but when it rotates within the first 676 letters is determined by the position of the notch on $\alpha_1$and α_2. However, in the rest of this essay, we will assume that only the first rotor rotates.

2.1 Using the Enigma

Decrypting Enigma messages worked by setting up the machine with the exact configuration as the sending machine. Typing the encrypted message causes the output to be the original text. This is because the reflector's loops allow it to be its own inverse, so if A is mapped to D, D is also mapped to A.

Nazi officials distributed new machine configurations daily, which operators used to encrypt their messages. Therefore, you and the team at Bletchley Park only have one day to decipher messages before configurations change. Time is of the essence; hence the need for the efficiency provided by Rejewski's ideas.

3 Mathematical application!

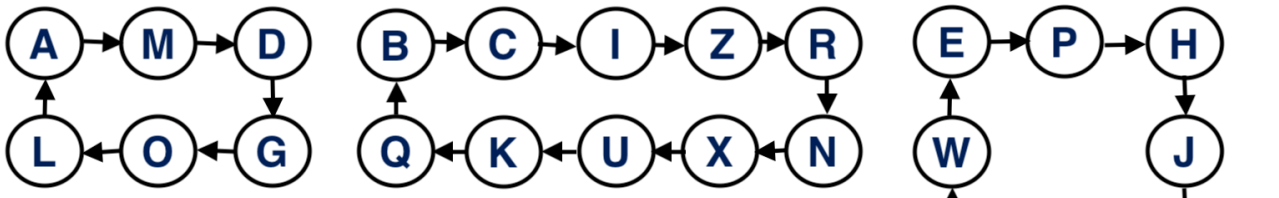
Now you understand the Enigma's function, it's time to use your mathematical knowledge to decrypt it! Firstly, we must identify which area of maths to use. The set we're altering is the alphabet, which we'll denote as Γ = {A, B ... Z}. In the rotors, reflector and the Enigma itself, elements of Γ are mapped to different elements within Γ. 'A' can become 'V', 'R' can become 'W' and so on. This is an example of an injective function, because no inputs have the same output, and a surjective function because every input is mapped to another letter in the alphabet. Therefore, the function is bijective! As it maps a finite set (26 letters) to itself, it is a permutation. Permutations are written in many ways as shown in Figure 3:

Figure 3

Example of a permutation, α, which could have been produced by a rotor:

x	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
α(x)	M	C	I	G	P	Y	O	J	Z	F	Q	A	D	X	L	H	B	N	W	V	K	T	E	U	S	R

This can also be shown in graph form, with each loop being a disjoint cycle, as they do not share any elements



These graphs can be written in cycle notation, e.g (A M D G O L) but because it is a loop, notation like (D G O L A M) is equally valid. The differences in how this is writen is known as the rotation of the cycle

Reflector permutations on the other hand are made up of 13 disjoint cycles of length two, like this one!

The length of a disjoint cycle is basically how many nodes it has in it, like this one has a length of 8

The reflector is an interesting type of permutation: A proper involution, as it's made up of transpositions (cycles of length two). This means it's the same as its own inverse: $\beta = \beta^{-1}$.

3.1 Permutations and the Enigma

While the Enigma is a permutation, it's also a series of permutations (the rotors and reflector). We'll denote the permutation produced by the Enigma as θ, so the first permutation can be written as θ_1

As a series of permutations: $\theta_1 = \alpha_1^{-1}\alpha_2^{-1}\alpha_3^{-1}\beta\alpha_3\alpha_2\alpha_1$

Where α are the rotors and β is the reflector, showing the order the current passes through the rotors and reflector. Note that a permutation with a negative power is the inverse of the original permutation, while the absolute value of the power is the number of times that permutation is applied. Half the rotor permutations have a negative power here because the current travels through them in opposite direction after going through the reflector. Inputting the encoded letter results in the original letter when the Enigma has been set up in the same configuration (so it applies the same permutation), meaning that it is a proper involution permutation like the reflector.

The second permutation applied by the Enigma is different to the first, as $\alpha_1$has rotated forward by 1/26th, meaning that each 26-connection pin has moved one place away from their original positions around the rotor. This is where we introduce the 'shift forwards' permutation, δ , which maps each alphabet letter to the following one, like $\delta(A) = B$ and $\delta(Z) = A$. As the rotor rotates, the pins will shift 'forwards', but inputs from the wires (attached to the plugboard) will connect to them in the same position. Thus, each connection has essentially shifted forwards. Likewise on the other side, between $\alpha_1$and α_2, the pins have shifted forwards on $\alpha_1$but remain the same on α_2, meaning the connections have 'shifted backwards', the inverse of the δ permutation. Thus, the permutation produced by the once rotated $\alpha_1$is $\delta^{-1}\alpha_1\delta^1$.

Therefore, θ_p (the permutation applied by the Enigma on the *p*th letter entered), can be written as:

$$\theta_p = \delta^{-p}\alpha_1^{-1}\delta^p\alpha_2^{-1}\alpha_3^{-1}\beta\alpha_3\alpha_2\delta^{-p}\alpha_1\delta^p$$

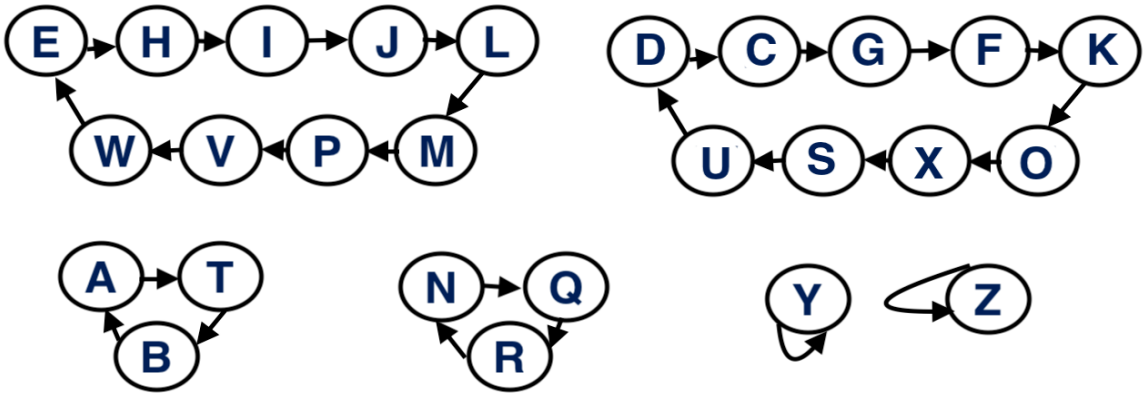
3.2 Deciphering the Enigma

To further reduce the possibility of frequency analysis techniques, operators were instructed to choose a random three lettered code and type it twice, before resetting the Enigma to its initial configuration. For example, if someone chose the code TOM, the output could be PGT VKE. Finding the permutation used to map the initial letter (which is unknown to the decoder) to P would help decrypt the message after the initial six letters. Here's how!

So, PGT VKE = $\theta_1(T)\ \theta_2(O)\ \theta_3(M)\ \theta_4(T)\ \theta_5(O)\ \theta_6(M)$

Remember, Enigma permutations are their own inverse and made up of cycles of 2 (transpositions). So for an unknown letter, $\theta_1(\varepsilon) = P$, where ε is the original first letter. Similarly, $\theta_4(\varepsilon) = A$. As they are their own inverse, $\theta_1(P) = \varepsilon$ so, $\theta_4\big(\theta_1(P)\big) = V$. This also applies to $\theta_2\theta_5(G) = K$ and $\theta_3\theta_6(T) = E$. Once we've intercepted enough messages on the same day, we could completely map out the permutations of $\theta_1\theta_4$ (an example is shown in Figure 4), $\theta_2\theta_5$and $\theta_3\theta_6$.

Figure 4



This is where things get interesting. In Rejewski's theorem, he stated that the product of two proper involutions (like θ_1 and θ_4) on the same elements can be expressed as a product of pairs of cycles of the same length. I'm omitting the proof due to space constraints, but it implies that the two elements in any transposition in $\theta_1$or θ_4 must be split between two cycles of equal lengths in the product of $\theta_4\theta_1$[1]. This will help us find the 13 disjoint transpositions of θ_1!

Firstly, we'll pair up cycles of the same length, which is easy when there are two. If there were four or six cycles of the same length, there would be more possibilities.

Using the example: (EHIJLMPVW) (DCGFKOXSU) are two disjoint cycles of the same length. These can pair up, e.g. E = $\theta_1(D)$ or E = $\theta_1(K)$, so there are nine possibilities (due to the 9 rotations of the cycle). It's much easier with cycles of length one, like (Y) and (Z), as there are no different rotation possibilities, so the only possibilities are $\theta_1(Z) = Y$ or $\theta_1(Y) = Z$, which are the same thing because the Enigma permutation is its own inverse. This means one of the transpositions will be (Z Y) The next part requires trial and error, but is made more efficient by noticing human errors.

While operators should've chosen their letters randomly, they mostly used messages like ABC, AAA and XYZ, with A usually being first. Assuming the first letter of the message is $\theta_1(A)$ could help find the correct permutation transpositions. If that failed, you would try other frequent first letters until you had the correct 13 transpositions of the Enigma permutation. Finding the exact permutations applied by the Enigma each key press allows you to map the internal rotor wiring and decode the rest of the message. This method worked effectively at the start of the war, so congratulate yourself if you made it this far!

4 Conclusion

This essay is a simplified look at the Enigma, as it increased in intricacy throughout the war, until complex machines like the Bombe, developed by Turing and Welchman, were the most efficient way of deciphering the messages.

Ultimately, the work of Rejewski and other mathematicians was vital deciphering Enigma messages, saving lives, resources and informing Allied plans. So what do you think? Would you have been an asset to Bletchley Park? Thank you for coming on this journey with me!

Sources used throughout this essay:

https://people.cs.uchicago.edu/~davidcash/284-autumn-21/02-enigma.pdf
https://cryptocellar.org/enigma/files/rew80.pdf

https://old.maa.org/press/periodicals/convergence/the-theorem-that-won-the-war-part-32-rejewskis-theorems

https://www.ebsco.com/research-starters/military-history-and-science/enigma-machine