Cracking Enigma: Chronology and Cryptogrophy

Brais Macknik-Conde

August 18th, 2024

The Enigma machine was a portable, German-made machine for encrypting and decrypting messages. It took the simple concept of key switches opening up electrical pathways from the battery to a corresponding lamp, but individually scrambled which lamp the battery would connect to every time a key was pressed (Fig 1).

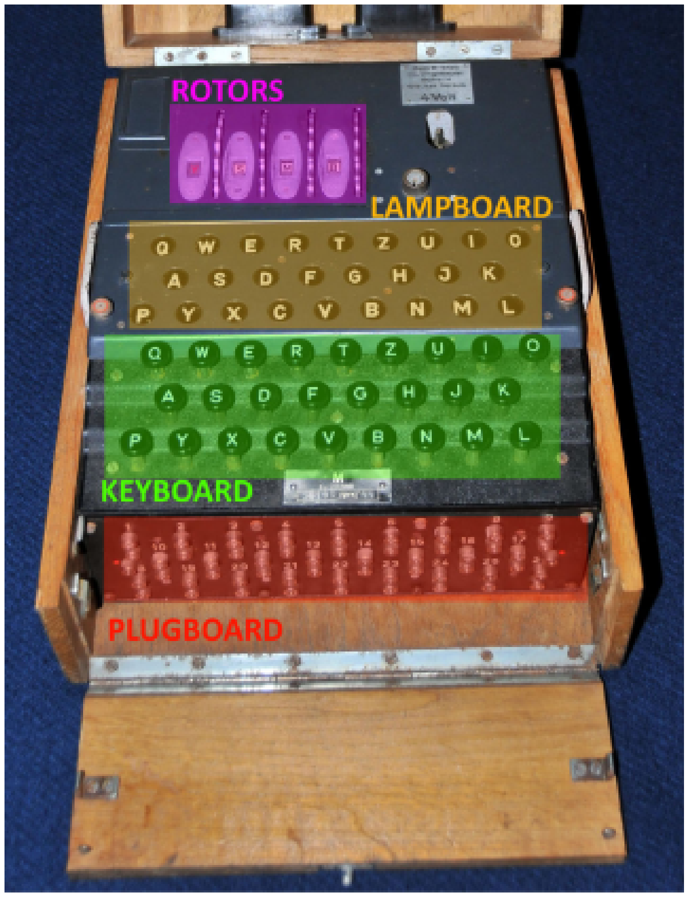
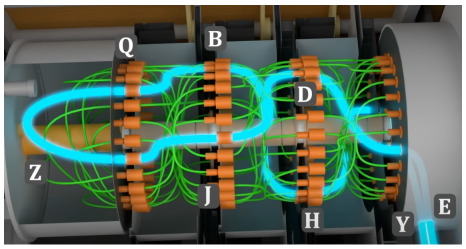


Figure 1: Diagram of the different parts of Enigma, Åvald Åslaugson Sommervoll & Leif Nilsen (2020): Genetic algorithm attack on Enigma's plugboard, Cryptologia, DOI: 10.1080/01611194.2020.1721617

This was done with 3 stepping rotors labeled I, II, and III. Later versions had 4 or more rotors to increase sophistication, but for the purposes of this paper, I will solely describe the 3-rotor model, as the principles remain the same irrespective of the number of rotors. As I will describe later, the differently numbered rotors could be placed in different order in the device to change the encryption. What was fundamental to the device irrespective of the settings was that the rightmost rotor would step every key press (the “fast rotor”), the middle rotor would step once every full rotation of the rightmost rotor and the leftmost rotor would step once every rotation of the middle rotor (that is, the rotors performed counting in base 26). Each rotor had two sides, each with 26 electrical contacts. Internally, the contacts on the right and left side were connected by a carefully constructed but randomized jumble of wires. This meant that every time the electrical signal went through a rotor, the corresponding lightbulb, signifying a letter, changed. (Fig.2, left). The order in which rotors I, II, and III were placed inside the machine could be changed to modify the encryption setting. Additionally, to keep track of what position the rotors start at, and to cycle the next rotor,, there was a ring having the 26 letters of the alphabet that could be attached to the outside of the rotor at different positions. There was also a pawl which would engage the next rotor in sequence by one step once every rotation. The position of this pawl was also movable and served as another encryption setting. When a key was pressed, after the electrical signal went through all three rotors right to left, the signal was reflected back through all 3 rotors from left to right by the reflector. These features were all standard on the Enigma machines made for businesses like banks, but the military version had an important addition. The plugboard connected pairs of keys together, intercepting the electrical signal once before the rotors, and once after going through the rotors. For example, if, on the plugboard, the keys F and G were connected, as well as the keys X and Y, a keypress on F would then go into the rotors as G. After going through the rotors, an example output of X would be changed to Y and then go to the lampboard.



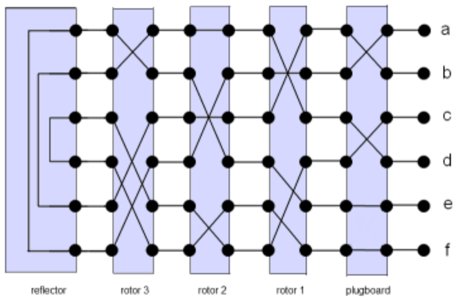


Figure 2: (left) Internal electricity flow of the rotors and reflector, “How did the Enigma Machine work?” by Jared Owen Animations

(right) Example Signal Pathway on an Enigma with 6 letters, Alan Turing and the Enigma Machine

On top of these encrypting measures, the Enigma machine had a variety of different and independent settings that could be used. The starting rotation of the rotors (ground setting), as well as which keys were connected on the plugboard, could be changed. Therefore, all of these settings had to be shared by the sender and recipient of an Enigma message: the order of the rotors, the ring setting (the position of the ring with respect to the internal wiring, the pawl position, and the plugboard wire connections. The internal wiring within individual rotors, however, and the reflector, stayed unchanged within the machine.

The Enigma had a few flaws with its design. For example, a pressed key could never output itself for the cyphertext. This was intentional, because in each individual configuration of enigma, the 26 letters of the alphabet were turned into 13 pairs, that meant that if a B encrypted as a Y, then if the machine was reset to how it was before, if Y was pressed, B would be outputted. This made Enigma able to encrypt and decrypt messages. This also was helpful for cracking the code, as commonly said phrases could be lined up against the ciphertext, and any positions where a ciphertext character and a plaintext character matched could be discounted.

Secondly, the order on which the rotors were placed could be determined, because the pawl on the alphabet ring was positioned at different letters on different rotors. This meant that cryptanalysts could determine which of the rotors was moving every key press by determining how far into the message it would cycle the second rotor.

Lastly, the plugboard changes signals forwards and backwards, meaning that if keys X and Y were connected, a pressed X would change to a Y, but a pressed Y must also change to an X.

The first codebreakers working on Enigma were Polish Mathematicians in 1932. Based on just the ciphertexts and a commercial Enigma they had bought, they were able to deduce the internal wirings of the rotors in the military Enigma, even without a military Enigma machine of their own. Using this knowledge, they created working replicas of Enigma in 1933. They could not yet decipher messages with this knowledge, as they did not know the daily settings used by Enigma operators. Each officer had a book of settings for the day. This included the start positions of the rotors. They would send the start encoded positions twice at the beginning of each message. For example, a start position XYZ (also known as rotor position 1) could be written in plaintext XYZXYZ at the beginning of each message. They encoded this message using Enigma itself, returning an example ciphertext of ABCDEF (this string of letters at the beginning of a message is known as the message indicator). They did not know it, but this practice was allowing the Polish to understand their messages. The ciphertext ABCDEF told the Polish that when a letter was encoded at rotor position 1, it returned A, and at rotor position 4, the same letter returned D.

This also applied for positions 2 and 5, and 3 and 6. With this info, and the fact the Enigma paired letters, we know that if we input A into Enigma at rotor position 1, we return ?, and entering ? into Enigma at rotor position 4, returns D. If we look at multiple message indicators that were made using the same ground setting (finding messages with the same ground setting meant just looking at messages from the same day), we can develop cycles. On top of ABCDEF, we could have message indicators DMLGOI and GHKAVT. This gives us a cycle of (ADG) by looking at the first and fourth characters of every string. With enough indicators, an entire alphabet could be reconstructed out of cycles. This could look like (BEHVPMOUQWR)(CFLIKTJSX)(ADG)(ZN)(Y). The lengths of these cycles are called the characteristic. This example has a characteristic of 11,9,3,2,1. Every setting on the Enigma had a characteristic, and the Polish were able to find these characteristics, even without the German message indicators. These characteristics were also independent of the plugboard, any setting on the plugboard with the same rotor settings was grouped under one characteristic. With this knowledge, the Polish built a machine called the cyclometer, which fundamentally took an input and ran it through just the rotors at position 1 and took that output and ran it through another set of rotors at position 4. Using this machine, the Polish found the characteristics for every rotor position for all six rotor orders, or $26 \cdot 26 \cdot 26 \cdot 6 = 105456$ characteristics. This was also done for positions 2 and 5, and 3 and 6. Finding the characteristics for all of the settings took a year, but by 1937, they could find the characteristic of a German message and, using the cracked database, find what settings were used to create it. Using these settings, all other unknowns could be deduced. The Germans changed the reflector wiring on 11/1/1937, and all the characteristic work had to be redone. Shortly after the new database was finished, the Germans changed their method for creating message indicators, and the Polish efforts were rendered useless.

The new method for making the message indicators was to have the operator create their own ground setting from 3 letters and send this in plaintext. Using this ground setting, they would encrypt their message setting as before, by sending it to be encrypted twice. The old method for creating databases of characteristics would not work, as one would need to be made for every message with new ground setting. A new message indicator could look like MOK INDICA. The Polish did know that the first and fourth character of the encrypted message setting portion of the indicator must be the same character in plaintext. This was also true for the second and fifth character, as well as the third and sixth. When in the encrypted message setting one of these two positions would match, it was called a female. There was a chance of a message indicator having a female, as the chance of any one pair having a female was about . To work on their new decrypting method, the Polish needed to find 3 message indicators with a female of the same letter in each of the 3 possible positions. For example, for females of letter I, MOK INDICA, ROM KITBIT, and HUB MUIDKI would work. We now know that with ground setting MOK, at rotor positions 1 and 4, the same input has the same output I. We also know that with ground setting ROM, at rotor positions 2 and 5, the same input has the same output I. Finally, we know that with ground setting HUB, at rotor positions 3 and 6, the same input has the same output I. With this knowledge the Polish built another machine, the "bomba kryptologiczna" or the cryptological bomb. The bomba essentially had 6 sets of enigma rotors, two of them set at ground setting MOK, another two set at ground setting ROM, and the last two at ground setting HUB. The bomba would individually check every combination of rotor rotations until inputting W would achieve the same output on both of the rotor set that are set to the same ground setting. This output did not matter, as a plugboard could change this output, and the plugboard settings are unknown. What doing this found out was the ring setting used. Because there are also 6 ways to arrange the rotors ({I,II,III},{I,III,II},{II,I,III},{II,III,I},{III,I,II},{III,II,I}), there had to be 6 bomby (plural of bomba) machines. Now that the ring setting and rotor order were known, the plugboard could be found manually.

Using bomby machines and their manual methods, the Polish could read about 75% of German messages encrypted using Enigma by 1938. In December 1938, the Germans added 2 new rotors into the usage of the Enigma, such that 3 out of 5 rotors would be used. This increased the number of rotor orders from 6 (3!) to 60 (5!2!). This would have made 60 bomby machines needed. Then in another month the Germans increased the number of pairs on the plugboard from six to ten. Because of this, and the fact that they knew they were going to be invaded soon due to intercepting German message traffic, the Polish gave all their intel, bomby machines, and Enigma replicas to the British in a secret meeting.

Perhaps the most famous figure when it comes to the cracking of Enigma is Alan Turning. He is also often referred to as the father of computer science. He started his work at Bletchley Park, the then facility of GCCS (the Government Code and Cypher School) the day after UK declared war on Germany. Turing was running Hut 8 (a building within Bletchley Park focused on cracking Naval Enigma messages). The Polish had already tried to crack these messages, but they could not figure out how the message indicator was written, and so once Turing started his work, he was by himself. He alone found that the plugboard connections and ground setting on Enigma were changed every day, and the rotor order and the ring setting were changed on odd numbered days. Importantly, he found how the message indicator was written. In the message preamble, there are two important 3-character strings written. These are the key indicator group, and the message indicator group. Let's have these be ABC and FGH. To send these, the operator would write them down and offset them by one.

$$\begin{array}{c} \quad\text{A B C} \\ \text{F G H}\quad \end{array}$$

From here, two random characters are put in the spaces

X A B C

F G H Y

This creates 4 vertical bigrams of characters. These are XF, AG, BH, and CY. The operator would have a table which would change a 'encrypt' and 'decrypt' bigrams. For example, if XF becomes HO, HO will become XF. Our four bigrams could be encoded as HO, DL, WY, NM respectively. There were 9 of these bigram tables, to be used on different days. These four bigrams would be sent at the beginning of the message. This would look like HODLWYNM. The receiving operator would decode these bigrams with their bigram table, and first read the key indicator group, which was determined from a settings book. If they had the same key indicator group, they were using the same settings on the machine. Then the operator would read the message indicator group, and input it into their machine at ground setting. The resulting string of 3 characters, DSP, for example, would be the message key for the rest of the message.

In Hut 8, they were able to start putting together the bigram tables. Their methods of constructing the bigram tables would then change, because the HMS Griffin captured the German trawler Polares. The bigram tables themselves were not found, but settings lists were retrieved. These lists allowed the further construction of the bigram tables.

An important concept is what it means to be in depth. For two messages to be in depth, the rotor positions at a certain character must be the same, even if the starting rotor positions were not. For example, if the rotors start at position ABK on a certain message after the preamble, and ABA on a different message after the preamble. The rotor positions will match on a 3-character offset.

```
Message key ABA: POFHEVPOBMFEYMNVYUKPLVXOHUGV
Message key ABK:   AOVAVNMSOVNQFWHIUKCOPMVHGQPH
```

In the above example, the section where the characters line up vertically is called in depth. With random characters, like the ones above, we expect 1 in 26 to match, and that is what we see. In German, this would be more like 1 in 13. It was determined however, in German Naval messages that 1 in 17 would match. Alan Turing developed a method of looking at how many match, taking into account strings of matches in a row, and scoring them in a way that determines their probability of actually being in depth or not.

The Good-Turing method sets a score to certain bigrams in two messages. To see where messages would align, they would take two sheets of paper and lay them on top of each other, with holes where letters in a message were. Where two letters matched, the score given could be calculated by Bayes Theorem.

$$O(A|B) = \frac{P(B|A)}{P(B|\overline{\mathrm{A}}}O(A)$$

This can be thought of as the probability of a set being in depth given a match is the probability of that match given the message is in depth over the probability it is a match given the message is not in depth all times the probability the set is in depth. Because we know the letter frequencies in Naval Enigma messages, this equation just turns into

$$O(A|B) = \frac{17}{26}(\text{Length of in depth section})$$

We also need to calculate how much each match decreases the odds of the message being in depth. This would be determined by

$$O(A|\overline{\mathrm{B}}) = \frac{P(\overline{\mathrm{B}}|A)}{P(\overline{\mathrm{B}}|\overline{\mathrm{A}}}O(A) = \frac{\frac{16}{17}}{\frac{25}{26}} = \frac{416}{425}O(A)$$

When there is a string of more than one matching characters, the likelihood of the message being in depth increases faster than if the characters individually matched. The increase in likelihood can be determined using MLE (Maximum Likelihood Estimation). Using this MLE method works for just one character, but it is much easier to use Bayes' Theorem. The MLE equation for this use-case is

$$G_k \equiv \frac{k+1}{m-k}|S_{k+1}.$$

In simpler terms, the MLE is equivalent to k(the frequency of a string of letters of the length of the matching section occurring) plus 1 divided by m (possibility of any string of letters occurring) - k, all multiplied by then all that multiplied by essentially the length of the in depth portion.

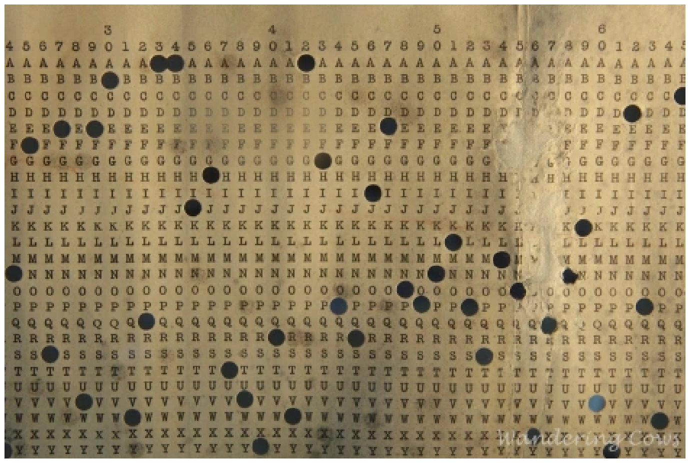


Figure 3: A Banbury sheet, of which two could be laid on top of each other to check for matching character positions in a message, wordpress.com

Now that we know the offset of two messages, and we know their respective message keys, we can write that A+2 =K. With this new knowledge, and the fact that the first two rotors are not in different rotations in each position of the in-depth portion, they can determine which rotor of 8 possible is the third, as some rotors turn over the next rotor at different points. When two rotors turned over at the same point, more methods had to be used to distinguish between. Once the right rotor is known, we can use the same principles to find the second and first. The banburisms saved checking all rotor orders.

Alan Turing's Bombe machine worked on simple logic and brute force. Alan Turing realized that a certain plaintext (a crib) could be lined up against a German message and a possible location it could be in the message's plaintext would be where none of the letters on the ciphertext match with the crib. This is because the Enigma cannot encipher a letter as itself. For example, with rotor positions labeled on top we could have the crib 'water'.

```
1 2 3 4 5
j k i h g t w r w m p c s a d e y m g u a g o c n
w a t e r
```

None of the letters match. What we know from this is that at rotor positions 1, j goes to w, a position 2, k to a, and so on. If we guess the first plugboard connection, we move on to the rotors, of which we know how they will encrypt our new letter, so long as the second rotor or third rotor does not move. Once we have the output from the rotors, we know what it must become, so we have a new plug board setting. We do this for all possible beginning plugboard settings, and where there are contradictions, we do not have to test large groups of the possible setting permutations. This is what the Bombe Machine did, it acted like multiple Enigma machines, and where there were logical errors in a certain rotor rotation, it would throw away that possibility and try a different rotation. Once the machine solves a solution where there are no contradictions and a single input goes to a single output, the machine stops. Banburisms stopped being used to determine rotor orders, as there were enough Bombe Machines that they could brute force the permutations faster than if banburisms were used.

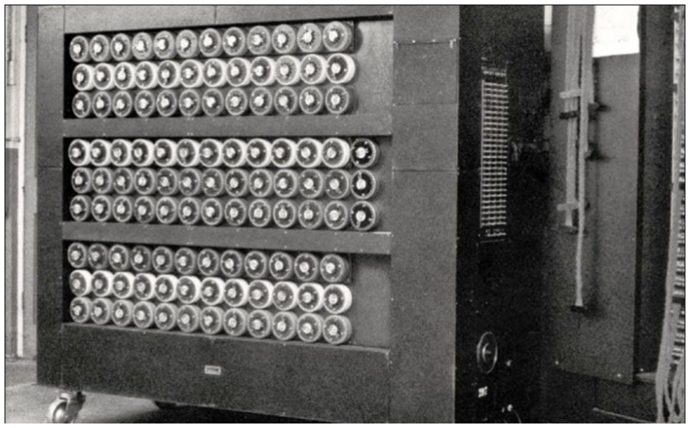


Figure 4: The Bombe Machine, heritage.humanists.uk

The work on Enigma is estimated by historians to have shortened the war by two years, and saved approximately 14 million lives.

Sources

Diagram of the different parts of Enigma, Åvald Åslaugson Sommervoll & Leif Nilsen (2020): Genetic algorithm attack on Enigma's plugboard, Cryptologia, DOI: 10.1080/01611194.2020.1721617

Internal electricity flow of the rotors and reflector, "How did the Enigma Machine work?" by Jared Owen Animations

Example Signal Pathway on an Enigma with 6 letters, Alan Turing and the Enigma Machine

Banbury sheet, wordpress.com

The Bombe Machine, heritage.humanists.uk

Enigma Message Procedures, ciphermachinesandcryptology.com

Turing Centenary Conference CiE 2012: How the World Computes, p.14

Bauer, Friedrich L., Decrypted Secrets: methods and maxims of cryptology, p.387

Church, K. and Gale, W., Enhanced Good-Turing and Cat-Cal: Two New Methods for Estimating Probabilities of Enlgish Bigrams, p.84-87

McAllester, D. and Schapire, R., On the Convergence Rate of Good-Turing Estimators, p.3-4

The Polish Bomba, cryptomuseum.com

Sigmon, N. and Klima, R., The Turing Bombe and its Role in Breakign Enigma

The Turing-Welchman Bombe, tnmoc.org