

BUILDING A BOMBE

The Mathematics and Circuitry that Bested German Encryption

Jonah Weinbaum

Contents

Chapter 1

Permutations

Definition 1.1. A **permutation** σ is a bijective function from a set S to the same set

$$\sigma: S \rightarrow S$$

A permutation can be thought of as “swapping” elements of our set while. Some simple examples of permutations are

Example 1.2.

- (a) $\sigma = \text{id}_S$ known as the **identity permutation**, which maps each element to itself
- (b) $S = \{a, b, c, d, \dots, z\}$ and σ defined by $\sigma(a) = b, \dots, \sigma(z) = a$. This shifts each letter by one place in the alphabet, wrapping around at z . We will call this the **Caesar permutation** and it will be denoted by θ_1

We will show that for a fixed rotor state the Enigma function is a permutation on the set of the latin alphabet. It is clear that the enigma machine sends this set to itself but it is not immediately apparent that such a mapping is bijective. To see this we must shift our description of the enigma to that of a composition of permutations. Let us first define some useful notation.

Definition 1.3. For a set S , the **symmetric group over S** is the group (G, \circ) of all permutations on S , where

$$G := \{f : S \rightarrow S \mid f \text{ is a bijection}\}$$

where the group operation is composition of functions. We denote this group $\text{Sym}(S)$.

It is clear that this group is closed with respect to composition, since the composition of bijections is itself a bijection. The identity element is simply the identity permutation described in CITE, and the inverse of an element is simply its inverse as a function, since the inverse of a bijection is itself a bijection this is well-defined.

Example 1.4. The Caesar cipher is one of the simplest and earliest encryption schemes. It involves shifting the set of letters by a fixed amount to encode a message. In Caesar's case he would shift each letter in a message by three places, sending $A \mapsto D, \dots, X \mapsto A, Y \mapsto B, Z \mapsto C$. In the context of permutations, this can be viewed as a repeated application of the Caesar permutation θ_1 described earlier CITE. For instance, to get Caesar's particular cipher we use $\theta_1 \circ \theta_1 \circ \theta_1$ (that is $\theta_1^3 \in \text{Sym}(\{A, \dots, Z\})$). For ease of notation we define

$$\theta_n := \theta_1^n \text{ for } n \in \mathbb{N}$$

Though this continues indefinitely, due to the nature of the permutation it is clear that $\theta_n = \theta_{(n+26)} \forall n \in \mathbb{N}$ so, in particular, there are only 26 distinct Caesar ciphers – a small keyspace indeed.

Section 1.1

Enigma as a Permutation

Suppose for now that we have fixed the rotors of the enigma machine in place. We want to examine how each letter is mapped through the machine. Recall from CITE

that the enigma machine maps each letter signal through the following transformations: the plugboard, three rotors, a reflector panel, back through the rotors, and back through the plugboard. Each of these components can themselves be viewed as a bijective mapping by construction. We will label the permutation associated with each as follows:

- (a) $R_{(1,r_1,\ell_1,g_1)}, R_{(2,r_2,\ell_2,g_2)}, R_{(3,r_3,\ell_3,g_3)}$ will be the rotors going from right to left where $r_i \in \{\text{I, II}, \dots, \text{VI}\}$ define which rotors we have selected, and $\ell_i, g_i \in \{a, \dots, z\}$ indicates the *Ringstellung* and *Grundstellung* respectively for that rotor.
- (b) P will be our plugboard setting
- (c) M_α will be our reflector setting where $\alpha \in \{A, B, C\}$ describes our reflector type

For the rotor and the reflector we include subscripts to describe their settings since their permutation is entirely determined by this short list of parameters. The plugboard, on the other hand, is itself (by the *Steckerverbindungen*) a description of its permutation so subscripts are unnecessary.

Supposing we are at ground position, the engima permutation E is as follows:

$$E = PR_1R_2R_3M_\alpha R_3^{-1}R_2^{-1}R_1^{-1}P^{-1}$$

We note that

$$\begin{aligned} E &= PR_{(1,r_1,\ell_1,g_1)}R_{(2,r_2,\ell_2,g_2)}R_{(3,r_3,\ell_3,g_3)}M_\alpha R_{(3,r_3,\ell_3,g_3)}^{-1}R_{(2,r_2,\ell_2,g_2)}^{-1}R_{(1,r_1,\ell_1,g_1)}^{-1}P^{-1} \\ &= (PR_{(1,r_1,\ell_1,g_1)}R_{(2,r_2,\ell_2,g_2)}R_{(3,r_3,\ell_3,g_3)}) \circ M_\alpha \circ (PR_{(1,r_1,\ell_1,g_1)}R_{(2,r_2,\ell_2,g_2)}R_{(3,r_3,\ell_3,g_3)})^{-1} \end{aligned}$$

Chapter 2

The UK Bombe

Section 2.1

Motivating Example

Section 2.2

Changes to Enigma

Starting in 1940, the German's enhanced the security of their key distribution. As discussed in CITE the *Grundstellung* rotor position was sent along with the daily key and an operator chose a *Spruchschlusse* to encode twice at the start of a message. Later iterations of this protocol removed the *Grundstellung* from key sheets.

These new key sheets contained the following columns columns *Tag/Datum*, *Walzenlage*, *Ringstellung*, *Steckerverbindungen*, and *Kennggruppen*

Notice the removal of the *Grundstellung* as well as the addition of the *Kennggruppen*. The *Kennggruppen* were a set of four trigrams used to identify which setting was being used to encode a message, this is particularly useful if trying to decode a message using

a prior day's key. The operator would choose a trigram from the the *Kenngruppen*, append two letters to the front of the trigram, and this five letter combination (known as the *Buchstabenkenngruppe*) would precede the message being sent. If a message was sent in multiple segments, multiple *Buchstabenkenngruppe* were used to start each segment.

When sending a message the operator was to use the following protocol

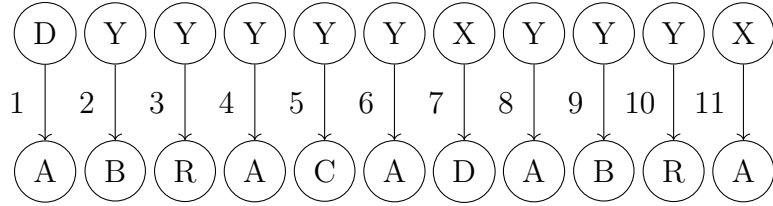
- I. The time at which the message was sent is listed
- II. The number of parts which the message contained is listed
- III. Which message part is being sent is listed
- IV. The length of the message part (not including *Buchstabenkenngruppe*) is listed
- V. A *Grundstellung* rotor position is chosen and listed
- VI. A *Spruchschlüssel* rotor position is chosen and encoded using the *Grundstellung*, this is listed
- VII. The *Buchstabenkenngruppe* is listed
- VIII. The message part encoded using the daily key and the *Spruchschlüssel* position is listed

It is clear that with this protocol, the Polish Bomba could no longer deduce the necessary details to decrypt enigma messages. All of the permutation information contained in the original key distribution protocol was removed and a new method needed to be derived for inferring information about the daily key.

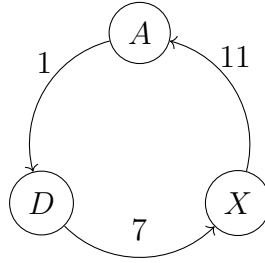
Section 2.3

Loops

The removal of the double encoded *Spruchschlüssel* does not mean that permutation information cannot be stored elsewhere in the message. For the sake of argument, let us say we knew that our encrypted message had plaintext encoding



Where the top row is the ciphertext and the bottom is the plaintext, further, the number in each mapping indicates how many steps away we are from the rotor positions when we began encoding the message.



Recall

$$E_1 = P^{-1}\theta_1 R_1^{-1}\theta_1^{-1} R_2^{-1} R_3^{-1} M R_3 R_2 \theta_1^{-1} R_1 \theta_1 P$$

$$E_7 = P^{-1}\theta_7 R_1^{-1}\theta_7^{-1} R_2^{-1} R_3^{-1} M R_3 R_2 \theta_7^{-1} R_1 \theta_7 P$$

$$E_{11} = P^{-1}\theta_{11} R_1^{-1}\theta_{11}^{-1} R_2^{-1} R_3^{-1} M R_3 R_2 \theta_{11}^{-1} R_1 \theta_{11} P$$

Then it follows that our loop can be represented by

$$\sigma = E_{11} \circ E_7 \circ E_1$$

and we see that all the intermediate plugboard settings cancel out. Lets isolate the plugboard settings by letting $\bar{\sigma}$ represent σ without the use of the plugboard for input and output, then

$$\sigma = P^{-1}\bar{\sigma}P$$

We have that

$$\begin{aligned}\sigma(A) &= A \\ \iff (P^{-1}\bar{\sigma}P)(A) &= A \\ \iff \bar{\sigma}(P(A)) &= P(A)\end{aligned}$$

Suppose that our initial rotor position was correct, then certainly our $\bar{\sigma}$ is correct. We can make a hypothesis that A is steckered to K in the plugboard. Suppose we find that $\bar{\sigma}(K) \neq K$, then $\sigma(A) \neq A$ and our loop is broken, breaking our assumptions, thus A must not be steckered to K . But this will actually eliminate more hypotheses than just A being steckered to K . We know that $\bar{\sigma}(K)$ is some letter which is not K . So we continue with a new hypothesis that A is steckered to $\bar{\sigma}(K)$ and if we find $\bar{\sigma}(\bar{\sigma}(K)) \neq \bar{\sigma}(K)$, then we have further eliminated this possibility. Each new hypothesis suggests that A is steckered to $\bar{\sigma}^i(K)$ which will be shown to be false if $\bar{\sigma}^{i+1}(K) \neq \bar{\sigma}^i(K)$. What if we find that $\bar{\sigma}^{i+1}(K) = \bar{\sigma}^i(K)$ at some point? This cannot happen since

$$\begin{aligned}\bar{\sigma}^{i+1}(K) &= \bar{\sigma}^i(K) \\ \Rightarrow \bar{\sigma}^{-i} \circ \bar{\sigma}^{i+1}(K) &= \bar{\sigma}^{-i} \circ \bar{\sigma}^i(K) \\ \Rightarrow \bar{\sigma}(K) &= K\end{aligned}$$

which by supposition is false. Then we can continue in our hypotheses until we eventually reach a cycle where $\bar{\sigma}^i(K) = K$. Then we gather a set of impossible steckerings, that is

$$P(A) \notin \{ \bar{\sigma}^i(K) \mid i \in \mathbb{N} \}$$

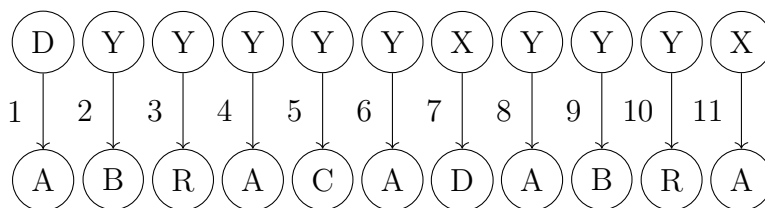
The notation we are using can be simplified significantly. The set $\{ \bar{\sigma}^i(K) \mid i \in \mathbb{N} \}$ is equivalent to the orbit of K via the group action of the cyclic subgroup $\langle \bar{\sigma} \rangle$ which can be denoted $\langle \bar{\sigma} \rangle \cdot K$.

We then have several cases

- (a) If $|\langle \bar{\sigma} \rangle \cdot K| = 26$, then A cannot be steckered to anything which is clearly impossible, thus our rotor position must be incorrect.
- (b) If $|\langle \bar{\sigma} \rangle \cdot K| = 25$, then A can only be steckered to the remaining letter $\{A, \dots, Z\} - \langle \bar{\sigma} \rangle \cdot K$
- (c) If $|\langle \bar{\sigma} \rangle \cdot K| = 1$, in this case we must have initially had $\bar{\sigma}(K) = K$ so we have not eliminated any possibilities.

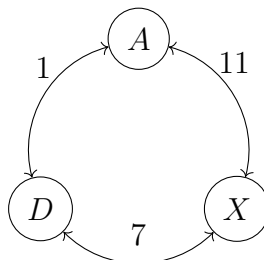
Motivating Example

Suppose we knew the plaintext which had been enciphered into a particular enigma transmission. Consider the following mapping,



where the top row indicates our enciphered message, the bottom row indicates the plaintext, and then indices on the arrows indicate how many steps forward our enigma machine has moved while enciphering this message. Our goal is to determine which enigma settings were used to encipher the message. In order to achieve this, we will examine which settings maintain the relationships between the enciphered and plaintext letters.

For example, any setting which maintains the above pairing must encipher A to D from the first position of the machine, then at the seventh position, it must encipher D to X , and at the eleventh position X must be enciphered back to A . It follows that if we had three enigma machines connected in series, with an offset of 1, 7, and 11, from our initial position, then inputting A on the first machine would result in an output of A on the third machine. We visualize this loop as follows



To express this mathematically we denote the permutation represented by the enigma at position i as σ_i . Since these each use the same plugboard we will also note the enigma at position i not using the plugboard as $\overline{\sigma}_i$, that is $\sigma_i = P\overline{\sigma}_iP$. Then our loop is expressed by the fact that $\sigma_{11} \circ \sigma_7 \circ \sigma_1$ has a fixed point at A . We also note that the intermediate plugboard settings cancel out, that is

$$\begin{aligned}
\sigma_1 \circ \sigma_7 \circ \sigma_{11} &= P\overline{\sigma_1}P \circ P\overline{\sigma_7}P \circ P\overline{\sigma_{11}}P \\
&= P \circ \overline{\sigma_1} \circ \overline{\sigma_7} \circ \overline{\sigma_{11}} \circ P
\end{aligned}$$

We will condense this notation by defining

$$\sigma := \sigma_1 \circ \sigma_7 \circ \sigma_{11}$$

and

$$\overline{\sigma} := \overline{\sigma_1} \circ \overline{\sigma_7} \circ \overline{\sigma_{11}}$$

And thus we have shown $\sigma = P\overline{\sigma}P$.

Let us hypothesize that A is steckered in the plugboard α – that is, $P(A) = \alpha$.

It then follows that

$$\begin{aligned}
\overline{\sigma}^i(\alpha) &= P \circ \sigma^i \circ P(\alpha) \\
&= P \circ \sigma^i(A) \\
&= P(A)
\end{aligned}$$

and so we derive

$$P(A) = \alpha \Rightarrow P(A) = \overline{\sigma}^i(\alpha) \quad \forall i \in \mathbb{N}$$

Then we have that A must be steckered to all values in the set $\{\overline{\sigma}^i(\alpha) \mid i \in \mathbb{N}\}$. We note that this set is that orbit of the element α under the group action of the subgroup $\langle \overline{\sigma} \rangle$ – that is, $\langle \overline{\sigma} \rangle \cdot \alpha$.

By construction of the enigma machine, A cannot be steckered to more than one value at a time, so if $|\langle \bar{\sigma} \rangle \cdot \alpha| > 1$ our initial hypotheses that $P(A) = \alpha$ must have been incorrect. Further, the above argument also illustrates that A cannot be steckered to any element in the orbit of α since we would similarly find that the orbit of that element was not a singleton. Then we now have

$$|\langle \bar{\sigma} \rangle \cdot \alpha| > 1 \Rightarrow P(A) \notin \langle \bar{\sigma} \rangle \cdot \alpha$$

thus eliminating several elements that A could be steckered to.

In fact, if $|\langle \bar{\sigma} \rangle \cdot \alpha| = 26$ then A cannot be steckered to *any* element. Of course, by construction, A must be steckered to *some* element, so it must be the case that this rotor position is impossible since no plugboard setting could produce the loop relationship in our ciphertext-plaintext pairing.

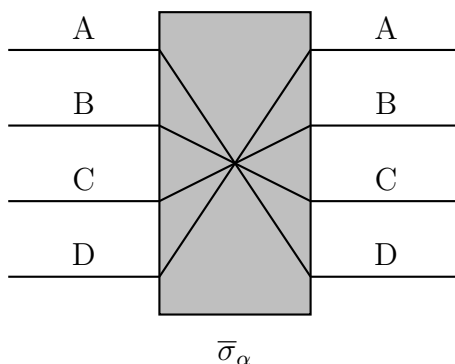
This is the key idea in the construction of the Bombe. At a high level, we encode the structure of the ciphertext-plaintext pairings, we input a hypothesis that a particular letter (e.g. A) is steckered to another (e.g. α), and we electrically produce the elements that must also be steckered to our test letter (e.g. $\langle \bar{\sigma} \rangle \cdot \alpha$) if our hypothesis were to be true. If we find that this production generates a set of all letters and thus disqualifies this rotor position from producing the plaintext we observed, then we can continue on to the next rotor position until we have no contradictory results from our hypothesis.

Before describing the function of the machine more concretely, we will first examine how we as humans can examine such permutations and deduce hypotheses quickly.

Section 2.4

The Bombe

In the section, we outline the construction of a rudimentary Bombe. Our goal is to construct a machine using the above insight to quickly eliminate a rotor position based on contradictory hypotheses. For clarity sake, we will imagine that our Enigma machine operates on an alphabet of only four letters $\{A, B, C, D\}$. We must shift from the above mathematical construction to an electrical one. We first abstract the idea of the Enigma machine and do away with the input output system of keys and lamp lights. Rather, we imagine the Enigma machine as black-box that takes in 4 cables encoding, via current, each of the 4 letters, and outputs currents on the corresponding letter after applying the Enigma permutation. Imagine the following set of wires encoding a possible Enigma permutation we will denote $\bar{\sigma}_\alpha$ (the bar indicates we are not yet considering the plugboard).



A couple quick notes about this abstraction. First as these lines are simply wires, current can flow in either direction, left-to-right, or right-to-left. Second, we can apply current to multiple wires concurrently, for example, applying current at A and C will cause D and B to be live on the other side of the machine.

As described in our motivating example, we may want to connect Enigma permutations

in series to capture an underlying relationship between a ciphertext-plaintext pairing. Suppose analysis found that we have three Enigma permutations represented by $\bar{\sigma}_\alpha, \bar{\sigma}_\beta$, and $\bar{\sigma}_\gamma$ which form a loop having a fixed point at A .