# BUILDING A BOMBE

## The Mathematics and Circuitry that Bested German Encryption

Jonah Weinbaum

# Contents

# Chapter 1

# Permutations

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

$$\sigma\colon S \to S$$

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

*Example* 1.2.

(a) $\sigma = \mathrm{id}_S$ known as the **identity permuation**, which maps each element to itself

(b) $S = \{a, b, c, d, \ldots, z\}$ and $\sigma$ defined by $\sigma(a) = b, \ldots, \sigma(z) = a$. This shifts each letter by one place in the alphabet, wrapping around at $z$. We will called this the **Caesar permutation** and it will be denoted by $\theta_1$

We will show that for a fixed rotor state the Enigma functions 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 discription 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

$$G \coloneqq \{f : S \to S \mid f \text{ is a bijection}\}$$

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

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 Caeser 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 $A \mapsto D, \ldots, X \mapsto A, Y \mapsto B, Z \mapsto C$. In the context of permuations, this can be viewed as a repeated application of the Caesar permuation $\theta_1$ described earlier CITE. For instance, to get Caesar's particular cipher we use $\theta_1 \circ \theta_1 \circ \theta_1$ (that is $\theta_1^3 \in \mathrm{Sym}(\{A, \ldots, Z\})$). For ease of notation we define

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

Though this continues indefinitely, due to the nature of the permuation 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 $r_i \in \{\mathrm{I}, \mathrm{II}, \ldots, \mathrm{VI}\}$ define which rotors we have selected, and $\ell_i, g_i \in \{a, \ldots, z\}$ indicates the *Ringstellung* and *Grundstellung* respectively for that rotor.

(b) $P$ will be our plugboard setting

(c) $M_\alpha$ 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

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

# 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 *Kenngruppen*

Notice the removal of the *Grundstellung* as well as the addition of the *Kenngruppen*. The *Kenngruppen* 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 preceed 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

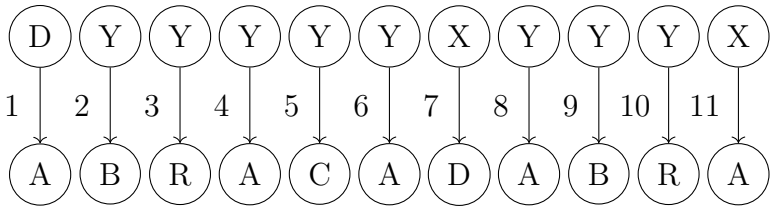
VIII. The message part encoded using the daily key and the *Spruchschlüssel* position is listed

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 infering information about the daily key.

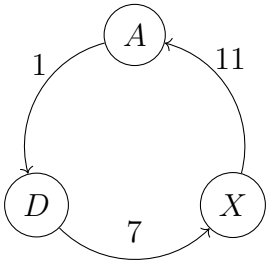
Section 2.3

# 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

$$E_1 = P^{-1}\theta_1 R_1^{-1}\theta_1^{-1}R_2^{-1}R_3^{-1}MR_3R_2\theta_1^{-1}R_1\theta_1 P$$

$$E_7 = P^{-1}\theta_7 R_1^{-1}\theta_7^{-1}R_2^{-1}R_3^{-1}MR_3R_2\theta_7^{-1}R_1\theta_7 P$$

$$E_{11} = P^{-1}\theta_{11} R_1^{-1}\theta_{11}^{-1}R_2^{-1}R_3^{-1}MR_3R_2\theta_{11}^{-1}R_1\theta_{11} P$$

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 $\overline{\sigma}$ represent $\sigma$ without the use of the plugboard for input and output, then

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

We have that

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

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

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

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

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

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

We then have several cases

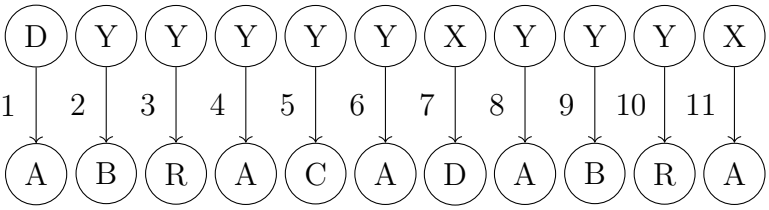
(a) If $|\langle\overline{\sigma}\rangle \cdot K| = 26$, then $A$ cannot be steckered to anything which is clearly impossible, thus our rotor position must be incorrect.

(b) If $|\langle\overline{\sigma}\rangle \cdot K| = 25$, then $A$ can only be steckered to the remaining letter $\{A, \ldots, Z\} - \langle\overline{\sigma}\rangle \cdot K$

(c) If $|\langle\overline{\sigma}\rangle \cdot K| = 1$, in this case we must have intially had $\overline{\sigma}(K) = K$ so we have not eliminated any possibilities.

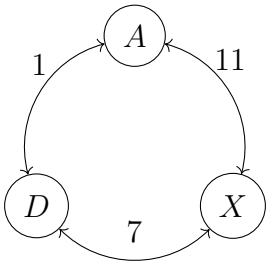
## Motivating Example

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 ouput 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 $\sigma_i$. 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 $\sigma_i = P\overline{\sigma_i}P$ (conversely, $\overline{\sigma}_i = P\sigma_i P$). 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

$$\sigma \coloneqq \sigma_1 \circ \sigma_7 \circ \sigma_{11}$$

and

$$\overline{\sigma} \coloneqq \overline{\sigma_1} \circ \overline{\sigma_7} \circ \overline{\sigma_{11}}$$

And thus we have shown $\sigma = P\overline{\sigma}P$ (conversely, $\overline{\sigma} = P\sigma P$).

Let us hypothesize that $A$ is steckered in the plugboard to $\alpha$ – that is, $P(A) = \alpha$ (conversely, $P(\alpha) = A$). It then follows that for a fixed $i \in \mathbb{N}$

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

and so we derive

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

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 $\alpha$ 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 $|\langle\overline{\sigma}\rangle \cdot \alpha| > 1$ 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 $\alpha$ since we would similarly find that the orbit of that element was not a singleton. Then we now have

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

thus eliminating several elements that $A$ could be steckered to.

By representing $\overline{\sigma}$ in its cycle notation we can quickly see whether certian hypotheses are possible. For example, suppose we found that

$$\overline{\sigma} = (ABCDEF)(GHIJK)(L)(MNOPQRSTUVWXYZ)$$

If we suppose that $A$ is steckered to any element in the cycle $(ABCDEF)$ we find that this element has an orbit of length 6 in $\langle\overline{\sigma}\rangle$ and thus $A$ cannot be steckered to any element in this cycle. Then it is clear that $A$ can only be steckered to $L$ in this case.

### 2.3.1. Scanning Methods

Turing describes various methods of mechanising the above analysis of cycle-type to determine when we can eliminate rotor positions.

(a) If we examine a particular hypothesis, say $A$ is steckered to $K$, we can rule out this steckering if we find that $K$ is not in a 1-cycle, that is if $\overline{\sigma}(K) \neq K$. If we mechanize this process we can eliminate rotor positions which do not satisfy this singular hypothesis. Turing called this method **single line scanning**. Note, however, that this method may eliminate rotor positions which do have valid steckerings, just not the particular steckering that we hypothesized.

(b) If we find $\overline{\sigma}$ does not contain a 1-cycle must then have that no steckerings of $A$ are valid and thus the hypothesized rotor position must be incorrect. If we wanted to rule out certain rotor positions one method would be mechanising the above description. Examining each possible steckering of a letter sequentially and ruling out any rotor positions which have no valid steckerings. Turing called this method **serial scanning**.

(c) Serial scanning requires a seperate examination of each steckering. Turing proposed a machine which could concurrently examine all steckering possibilities and eliminate rotor positions which had no valid steckerings. Turing called this method **simultaneous scanning**.

(d) If we find $\overline{\sigma}$ has a 26-cycle, then we must have that there are no 1-cycles and thus no valid steckerings. It then follows that the rotor position is incorrect. If we mechanism this process we can eliminate *some* rotor positions which do not have valid steckerings. We will call this method **spider scanning**. Note, however, that this method would not, for example, detect that a $13, 13$-cycle contains no valid steckerings. As Turing explained, "The ideal machine that Welchman was aiming at was to reject any position in which a certain fixed-for-the-time Stecker hypothesis led to any direct contradiction... The spider does more than this in one way and less in another. It is not restricted to dealing with one Stecker hypothesis at a time, and it does not find all direct contradictions."

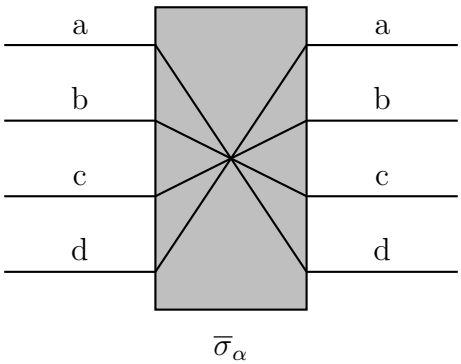
Iterations of each scanning methods were proposed or designed, but in the end we find that the spider scanning method was used in the implementation of the Bombe.At a high level, we encode the structure of the ciphertext-plaintext pairings, we input a hypothesis that a particular letter (e.g $A$) is steckered to another (e.g. $\alpha$), and we electrically produce the elements that must also be steckered to our test letter (e.g. $\langle\overline{\sigma}\rangle \cdot \alpha$) 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.

Section 2.4

## The Bombe

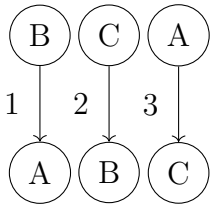
In the section, we outline the construction of a rudementary Bombe. Our goal is to construct a machine using the above insight to quickly elimate a rotor position based on contradictory hypotheses. For clarity sake, we will imagine that our Engima machine operates on an alphabet of only four letters $\{A, B, C, D\}$. We must shift from the above mathematical construction to an electrical one. We first abstract the idea of the Engima 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 denoted $\overline{\sigma}_\alpha$ (the bar indicates we are not yet considering the plugboard).



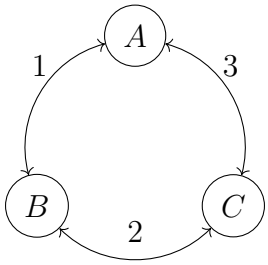

A couple quick notes about this abstraction. First as these lines are simply wires, current can flow in either direction, left-to-right, or right-to-left. Second, we can

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. Finally, the choice to use lower case letters will become clear further in this section as we want to seperate the plugboard letters from the actual plaintext-ciphertext letters.

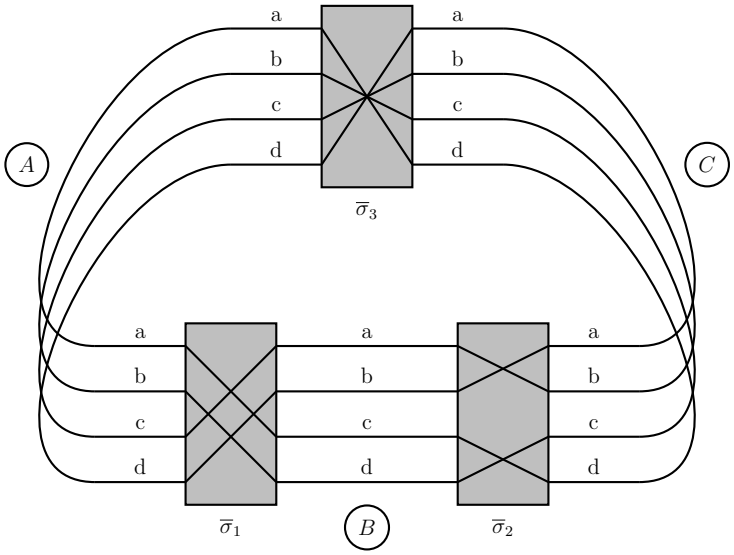
As described in our motivating example, we may want to connect Engima permutations in series to capture an underlying relationship between a ciphertext-plaintext pairing. Suppose we had plaintext ciphertext pairing




Then we find that there exists a loop in the pairing as follows

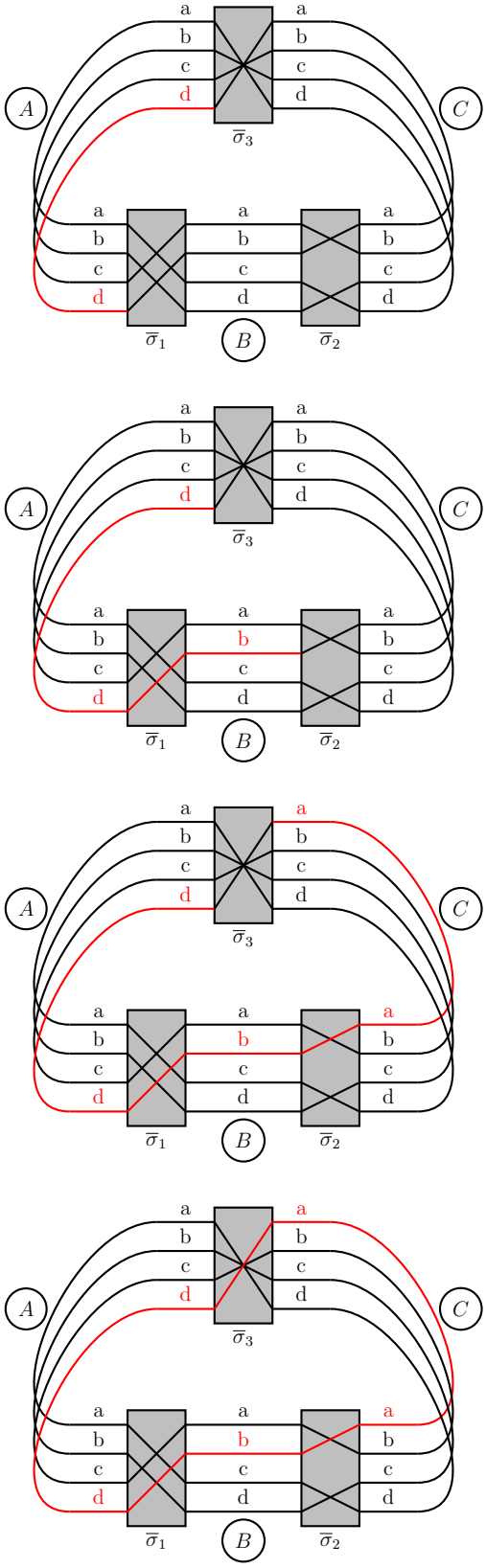



We can then think of this loop as a series of Enigma permutations

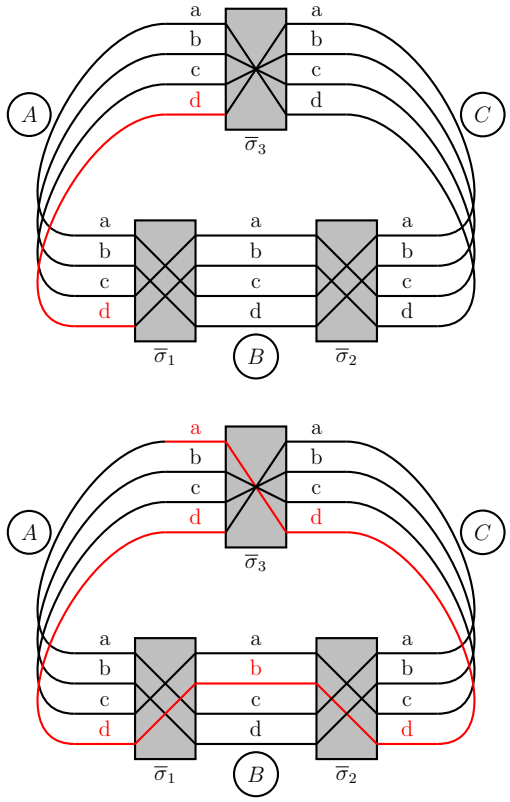



As before, let us make a hypothesis regarding steckering. Suppose $P(A) = D$. We know that $A$ must map to $A$ via $P\overline{\sigma}_3\overline{\sigma}_2\overline{\sigma}_1P$. In our diagram, however we do not see a plugboard so how can we represent the plugboard's involvement? In order to achieve this *we* will function as the plugboard by applying voltage to the $d$ line on the $A$ thus implicitly performing the a plugboard mapping in which $P(A) = D$. Then this will go through the map electrical mapping $\overline{\sigma}_1$ arriving on the $B$ cable, followed by $\overline{\sigma}_2$ arriving on the $C$ cable, followed by $\overline{\sigma}_3$ arriving back on the $A$ cable, where *we* as the implicit plugboard know that the output of this electrical mapping must go through the pluboard again to get a final output of $A$.

We will denote each wire by its a capital and lowercase letter which indicates both the cable and specific wire to which we refer. For instance, line $d$ on cable $A$ will be denoted $Ad$. Any time a wire $Xy$ is live, implicitly this means that there is an intermediary plugboard which mapped $P(X) = Y$. To see this electrical mapping occur let us visualize a current being sent through $Ad$.


a
b
c
d
A
C
$\overline{\sigma}_3$
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

Following our hypothesis that $P(A) = D$ we find that after sending current through $Ad$ we arrive after mapping through $\overline{\sigma}$ back at $Ad$ and this makes sense since we know that $D$ will map back to $A$ thus preserving the loop in our ciphertext-plaintext pairing. On the other hand, if we change $\overline{\sigma}$ and repreat this process which may end up in the following sitatuation




In this example, following our steckering hypothesis $P(A) = D$, results in $Aa$ becoming live after mapping through $\overline{\sigma}$ and thus we must have $P(A) = A$ as well in order for this to ultimately map back to $A$ and preserve our ciphertext-plaintext loop above.