

BUILDING A BOMBE

The Mathematics and Circuitry that Bested German Encryption

Jonah Weinbaum

Contents

1	The UK Bombe	1
1.1	Motivating Example	1
1.2	Changes to Enigma	1
1.2.1	Scanning Methods	5
1.3	The Bombe	7

Chapter 1

The UK Bombe

Section 1.1

Motivating Example

Section 1.2

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.

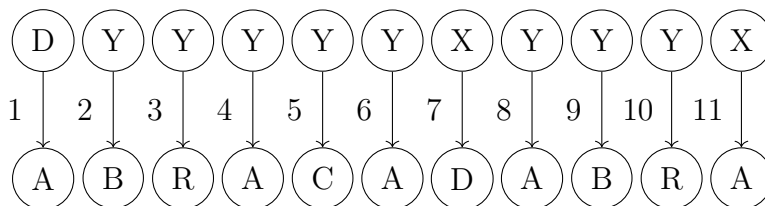
When sending a message the operator was to use the following protocol

- 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.

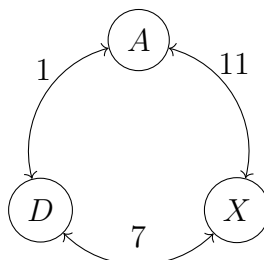
Motivating Example

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 $\bar{\sigma}_i$, that is $\sigma_i = P\bar{\sigma}_iP$ (conversely, $\bar{\sigma}_i = P\sigma_iP$). 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

$$\begin{aligned}\sigma_1 \circ \sigma_7 \circ \sigma_{11} &= P\bar{\sigma}_1P \circ P\bar{\sigma}_7P \circ P\bar{\sigma}_{11}P \\ &= P \circ \bar{\sigma}_1 \circ \bar{\sigma}_7 \circ \bar{\sigma}_{11} \circ P\end{aligned}$$

We will condense this notation by defining

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

and

$$\bar{\sigma} := \bar{\sigma}_1 \circ \bar{\sigma}_7 \circ \bar{\sigma}_{11}$$

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

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

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

and so we derive

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

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

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.

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

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

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 $\langle \bar{\sigma} \rangle$ 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.

1.2.1. Scanning Methods

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

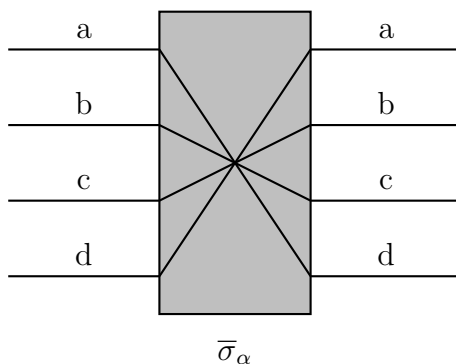
- (a) If we examine a particular hypothesis, say A is steckered to K , we can rule out this steckering if we find that K is not in a 1-cycle, that is if $\bar{\sigma}(K) \neq K$. If we mechanize this process we can eliminate rotor positions which do not satisfy this singular hypothesis. Turing called 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 perform single line scanning in sequence, that is, for each steckering hypothesis, we can rule out rotor positions which have all steckering hypotheses invalid. Turing called this method **serial scanning**.
- (c) Serial scanning requires a separate 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 $\bar{\sigma}$ 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 mechanize 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.” Effectively, spider-scanning is like a form of simultaneous scanning which is restricted to examining only one cycle at a time.

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. 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.

Section 1.3

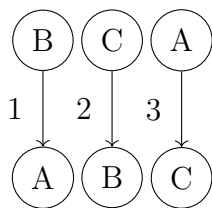
The Bombe

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).

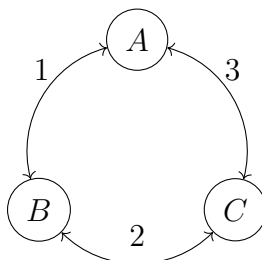


A couple quick notes about this abstraction. First as these lines are simply wires, current can flow in either direction, left-to-right, or right-to-left. Second, we can apply current to multiple wires concurrently, for example, applying current at a and c will cause d and b to be live on the other side of the machine. Finally, the choice to use lower case letters will become clear further in this section as we want to separate the plugboard letters from the actual plaintext-ciphertext letters.

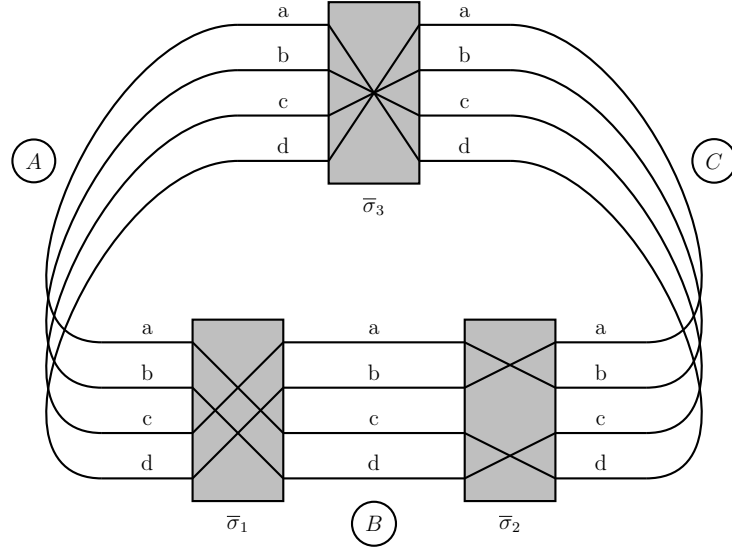
As described in our motivating example, we may want to connect Engima permutations in series to capture an underlying relationship between a ciphertext-plaintext pairing. Suppose we had plaintext ciphertext pairing



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

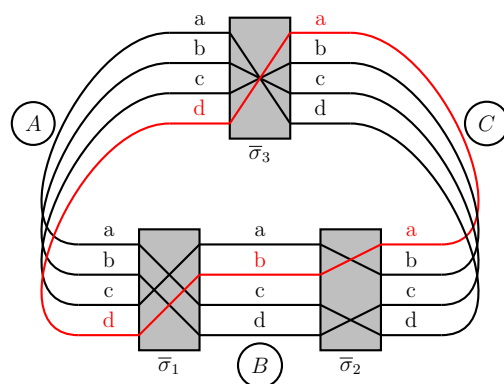
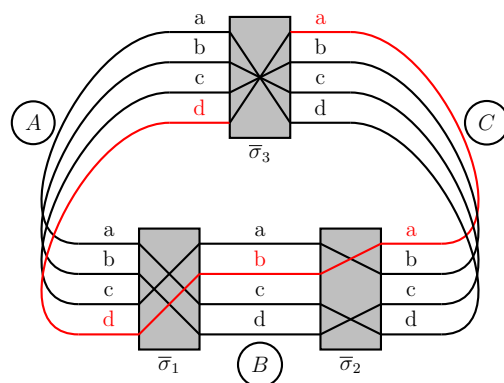
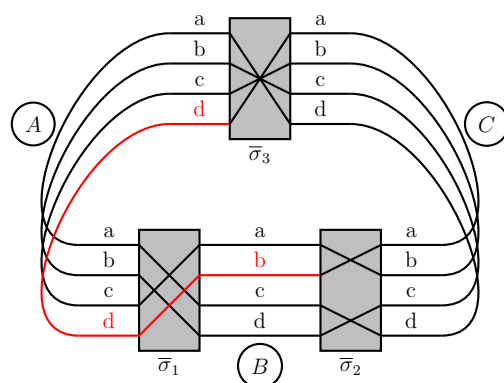
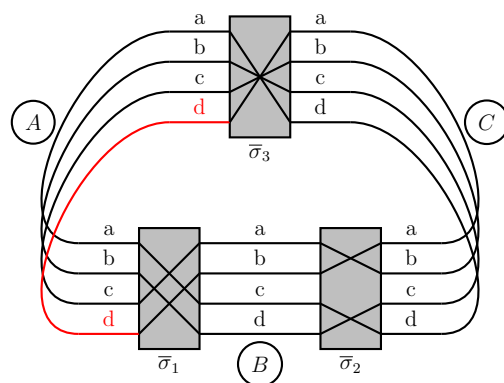


We can then think of this loop as a series of Enigma permutations

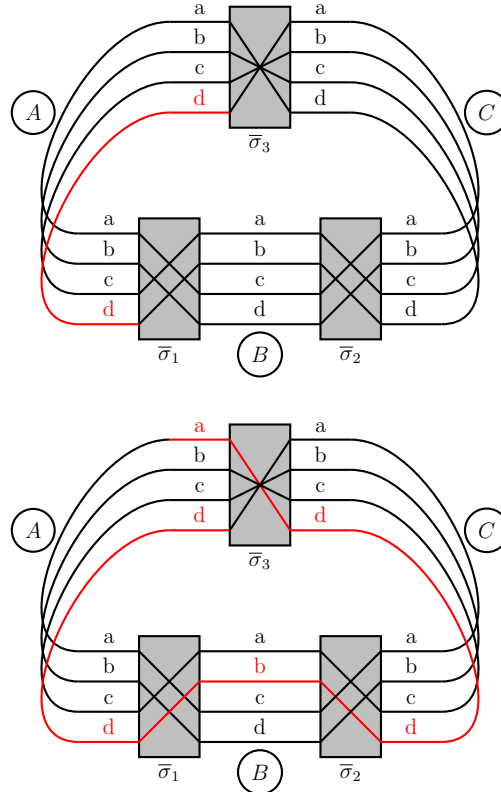


As before, let us make a hypothesis regarding steckering. Suppose $P(A) = D$. We know that A must map to A via $P\bar{\sigma}_3\bar{\sigma}_2\bar{\sigma}_1P$. In our diagram, however we do not see a plugboard so how can we represent the plugboard's involvement? In order to achieve this *we* will function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

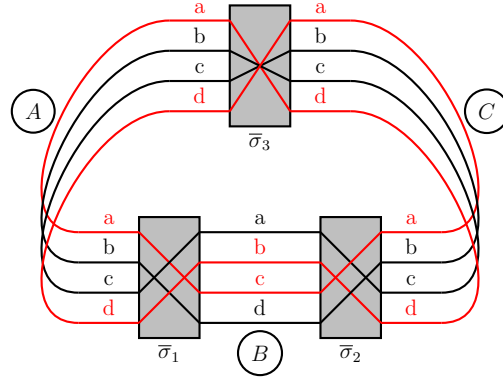
We will denote each wire by a capital and lowercase letter which indicates both the cable and specific wire to which we refer. For instance, line d on cable A will be denoted Ad . Any time a wire Xy is live, implicitly this means that there is an intermediary plugboard which mapped $P(X) = Y$ or $P(Y)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



Following our hypothesis that $P(A) = D$ we find that after sending current through Ad we arrive after mapping through $\bar{\sigma}$ 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 $\bar{\sigma}$ and repeat this process which may end up in the following situation



In this example, following our steckering hypothesis $P(A) = D$, results in Aa becoming live after mapping through $\bar{\sigma}$ 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. Thus we can eliminate the steckering possibilities that $P(A) \in \{A, D\}$. Effectively, this process is a mechanization of finding the cycle containing a particular letter. In the above diagram sending current from Ad until we arrive back at Ad is akin to repeatedly applying $\bar{\sigma}$ to D until we return to D . This can be visualized in our diagram as



Where we find that in $\bar{\sigma}$ we have the cycle (AD) since Ad and Aa are the only two live wires in the A cable after allowing current to reach a steady-state. If we had instead applied current to Ab we would find that Ab and Ac become live and give us the full permutation $\bar{\sigma} = (AD)(BC)$. The beauty of this design is that it is able to deduce the elements in a cycle of $\bar{\sigma}$ nearly instantaneously since it only requires the circuit to reach a steady-state.

Equipped with this tool spider-scanning becomes quite trivial. The goal of spider-scanning is to eliminate a rotor position by checking if the permutation has a 26-cycle. In our diagram, with four lines, this would be analogous to testing a steckering hypothesis (any hypothesis) and finding that the entire cable becomes live. Then all elements lie within the same cycle of $\bar{\sigma}$ and thus all steckering possibilities are immediately eliminated.