The Turing Bombe and its Role in Breaking Enigma

Neil Sigmon
npsigmon@radford.edu
Department of Mathematics and Statistics
Radford University
U.S.A.

Rick Klima
klimare@appstate.edu
Department of Mathematical Sciences
Appalachian State University
U.S.A

Abstract: *The work of the codebreakers at Bletchley Park in breaking the German Enigma cipher during World War II was one of the most extraordinary events in human history. Led by Alan Turing, the codebreakers employed an electromechanical device known as the bombe to regularly cryptanalyze and read German encrypted communications throughout much of the war. This work likely helped the Allies to win the war much sooner than expected and saved countless lives. Due to the extraordinary number of combinations that the Enigma could be set to, the Germans believed that the Enigma was impenetrable. However, Turing and the codebreakers were able to use the bombe to exploit the part of the Enigma that the Germans thought gave the device its most security. This paper will describe the logic behind how the bombe exploited the Enigma cipher and the process involved.*

1. Introduction

In 1918, German electrical engineer Arthur Scherbius applied for a patent for a mechanical cipher machine. This machine, later marketed commercially under the name *Enigma*, was designed with electric current running through revolving wired wheels, called *rotors*. Scherbius offered his machine to the German military. Only after learning that their World War I ciphers had routinely been broken did the Germans adopt various models of the Enigma, which they used as their primary resource for encrypted communications throughout World War II.

Up until December, 1938, Polish codebreakers under the leadership of Marian Rejewski were able to successfully break and read German Enigma traffic. However, when German cryptographers made modifications to the Enigma, the security increase proved too much for the Poles to account for in a timely manner, so in July, 1939, just five weeks before Poland fell to the Germans, Rejewski and his team shared their work with cryptologists from Britain and France. In particular, British mathematician Alan Turing, identified weaknesses in the Enigma encryption process using patterns generated by cribs, which were made easier to find through the frequent mistaken use of standard salutations, titles, and addresses by German operators.

In this paper, we describe some of the various aspects of Turing and the Bletchley Park codebreakers' work in breaking the version of the Enigma used by the German Wermacht (army) in World War II. We begin with a brief description of the Enigma's components and the reasons behind the German's confidence of its security.

2. The Challenges of the Enigma

The Wermacht version of the Enigma was made up of various parts. Figure 1 shows a photo of the Enigma with important components labeled. A more detailed description of these components than described here can be found in [1]. When a letter is encrypted or decrypted on an Enigma, it is typed and travels from the keyboard to the plugboard, where it may or may not be swapped according to whether it has a cable connection to another letter. The plugboard output travels to and passes through

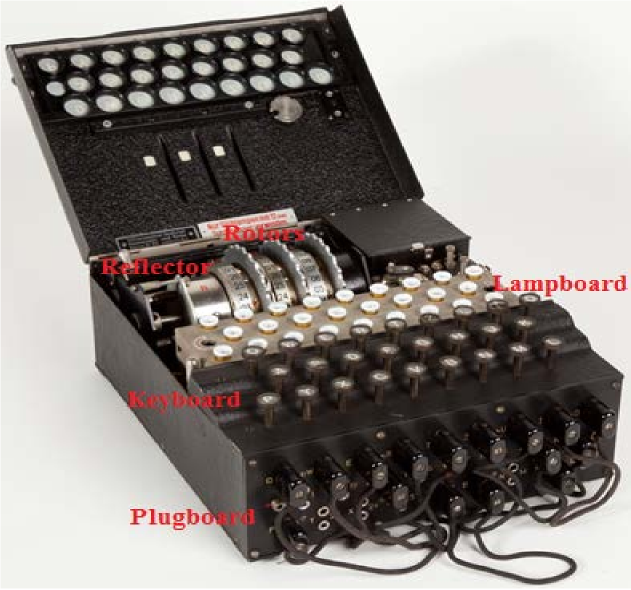

Figure 1: Enigma Cipher Machine

three rotors, where the letter is permuted to another letter by each rotor moving from right to left. The letter output by the leftmost rotor is then swapped to another letter by the reflector, and then returns and passes through the three rotors moving from left to right, being permuted again by each to another letter as it moves through each rotor. The letter output by the rightmost rotor travels back to the plugboard, where it may or may not be swapped according to whether it has a cable connection to another letter. The letter output by the plugboard travels where the result is lit up on the lampboard, representing the ciphertext or plaintext corresponding to the original letter typed.

The plugboard, rotors, and reflector each added to the overall security of the Enigma. We describe each factor next.

The Enigma Plugboard

The plugboard resembles a miniature old telephone switchboard. The plugboard was situated on the front of an Enigma, and had 26 open sockets, one to represent each possible letter. The plugboard sockets could either be left open or connected in pairs by short cables. If a pair of sockets was connected by a cable in the plugboard, then current designating either letter represented by the sockets would be converted at the plugboard to designate the other letter. If a socket was left open in the plugboard, then current designating the letter represented by the socket would leave the plugboard still designating the same letter.

There were many different choices for which plugboard sockets could be connected in an Enigma, with anywhere from zero to 13 cables used, and usually a very large number of possibilities for which pair of sockets could be connected by each cable. Varying the number of cables would have maximized security, but standard German operating procedure was to use a fixed number of cables. With a fixed number of cables, 11 cables would have maximized security, but for most of the war standard German operating procedure was to use 10 cables. Each Enigma provided for use in the field came with 12 cables, with two held in reserve in case any of the 10 in use became faulty.

When connecting the cables, the total number of possible plugboard connections when p cables are connected to $2p$ sockets chosen from 26 total is given by the formula

$$N_p = \frac{26!}{(2p)!(26-2p)!} \cdot \left[(2p-1)\cdot(2p-3)\cdots 3\cdot 1\right] \tag{2.1}$$

Using equation (2.1), we can see that the number of ways the Germans could plug p = 10 cables in pairs to the 20 chosen sockets is given by

$$N_{10} = \frac{26!}{20!\cdot 6!} \cdot \left[19\cdot 17\cdots 3\cdot 1\right] = 150738274937250$$

The Enigma Rotors

After leaving the plugboard, current went through a system of rotors that was situated in the back of an Enigma. Each individual rotor was a circular disk about the size of a hockey puck. All the Army versions of the German Enigma used 3 rotors.

We will call the flat sides of a rotor the *right* and *left* sides, since rotors could only be placed in an Enigma standing on end with each side facing in a particular direction. Both flat sides of a rotor contained 26 contact points, one to represent each letter, with the letters considered in alphabetical order around both sides of the rotor clockwise (when the rotor is viewed from the right). The contacts on the right side of a rotor were wired to the contact points on the left, but not necessarily straight across. The idea was that current could enter one side of a rotor at one of the contact positions, representing a letter, and pass through and exit the rotor on the other side at a different contact position representing another letter.

Besides having contact points representing each letter on both sides of the Enigma rotor, the Enigma rotor also had an etched ring containing the 26 letters (some rotors instead contained the numbers 1-26) oriented clockwise when looking at the rotor from the right. For each rotor slot in an Enigma, a small window was cut to show the letter (or number) at a particular location on the ring. We will call this letter the *window letter*. The window letter for a rotor is designed to help indicate the orientation of the rotor.

There were many different choices for how rotor contacts could be connected in an Enigma, but because rotors had to be hard-wired and changing the wiring was a very difficult thing to do, rotors with only a very small number of different wirings were ever produced and used in the field. Rotors with only five different wirings were produced for Wehrmacht Enigmas.

Besides the fact that the rotor can be oriented to any position with respect to its twenty six contact letters in the machine, there was a movable ring around the rotor that could be rotated to any of the twenty six letter positions. To describe how much the outer ring is rotated, we use a number called the *ring setting.* The ring setting indicates the position of the ring on a rotor. The rotors revolved within the machine during the actual encryption and decryption processes. Encrypting and decrypting messages with an Enigma was done one letter at a time, and each time an input letter was pressed on the keyboard, the rightmost rotor would immediately (before the current reached the rotors) rotate one position counterclockwise (when the rotor was viewed from the right). In addition, for each Enigma rotor **I–V**, there was a notch on the ring around the rotor. Since each notch was on the ring, its position in the rotor slot at any time could be identified solely by the window letter. For each notch, there was one particular position in the rotor slot, identified by a window letter called the *notch letter*, for which if the rotor rotated one position counterclockwise, the notch would cause the rotor to the left, if it were one of the rotors **I–V**, to also rotate one position counterclockwise.

The three rotors contribute to the total number of keys possible in the Wehrmacht Enigma in the following manner. First, the way the rotors are arranged makes a contribution. Recall that rotors with five different wirings were produced for Wehrmacht Enigmas, with three in use in the machine at a time. Hence, the number of ways that 3 rotors can be can be arranged in the machine from 5 total is given by $5 \cdot 4 \cdot 3 = 60$ total. Also, recall that around each Enigma rotor was a movable ring that could be rotated into any of 26 different positions while the wired part of the rotor was held fixed. Hence, with three rotors in use in the machine at a time, the number of possible ring settings for all of the rotors in a Wehrmacht Enigma is $26 \cdot 26 \cdot 26 = 17576$. Finally, recall that before a rotor was placed in an Enigma, it could be rotated into any of 26 possible orientations, each yielding a unique window letter. With three rotors in use in the machine at a time, the number of possible initial window letters for all of the rotors in a Wehrmacht Enigma is $26 \cdot 26 \cdot 26 = 17576$. Hence, the 3 rotors contribute a factor of

$$60 \cdot 17576 \cdot 17576 = 18534946560$$

to the total number of Enigma combinations.

The Enigma Reflector

Although rotors could only be situated with each side facing in a particular direction, current could pass through the rotors in either direction. The reason for this is that while current always initially passed through the rotors from right to left, to the left of the rotor slots was a reflector which sent the current back through the rotors from left to right. The reflector was always placed in the Enigma in only one way and did not rotate. In addition, the reflector was itself like half a rotor in the sense that on its right side there were 26 contact points, one to represent each possible letter, but on its left side there were no contacts. The contacts on the right side of a reflector were wired to each other in 13 pairs. Unlike plugboard sockets, reflector contacts were always fully connected.

Since reflectors (labeled B and C) with two different wirings were produced for Wehrmacht Enigmas, with one in use in the machine at a time, the number of different ways in which a reflector can be chosen for a Wehrmacht Enigma and contribute to the total key combination was 2.

The Total Number of Key Combinations for the Full Enigma Machine

Given the fact that there were 150738274937250 ways 10 cables can be plugged into 20 plugboard sockets, 18534946560 key settings produced by the 3 rotors, and 2 possible reflectors, the number of different key settings produced by a Wehrmacht Enigma is given by

$$150738274937250 \cdot 18534956560 \cdot 2 = 5587851741017032206720000\,.$$

This number, which is approximately 5.5879×10^{24}, more than five million billion billion, was much too large for a brute force attack on the Enigma to have been possible during World War II. However, through the work of Turing and the Bletchley Park codebreakers, this astronomical number was overcome. We describe Turing’s work next.

3. Using Cribs to Construct Menus

Turing was not interested in extending Rejewski's methods for breaking Enigma, because he correctly anticipated that the Germans would make improvements in Enigma's security that would continue to make the methods not feasible. This led to the use of cribs, which represent a small plaintext portion that corresponded to part of the ciphertext, to find weaknesses in the German's use of the Enigma machine

To demonstrate, suppose the crib **FOLLOWORDERSTO** was known to encrypt to part of the ciphertext message **NUENTZERLOHHBTDSHLHIY** that is underlined. As a way to describe this possible match, we start by numbering the crib letters with this ciphertext assignment in the following manner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Crib:	F	O	L	L	O	W	O	R	D	E	R	S	T	O
Cipher:	E	N	T	Z	E	R	L	O	H	H	B	T	D	S

Since the Enigma uses the same settings for both encryption and decryption, position 1 represents the case of where letter F is encoded as letter E and vice versa, letter E is encoded as letter F. Similarly, position 2 represents the case of where letter O is encoded as letter N and letter N is encoded as letter O. Geometrically, these pairing relationships can be expressed in Figure 2.

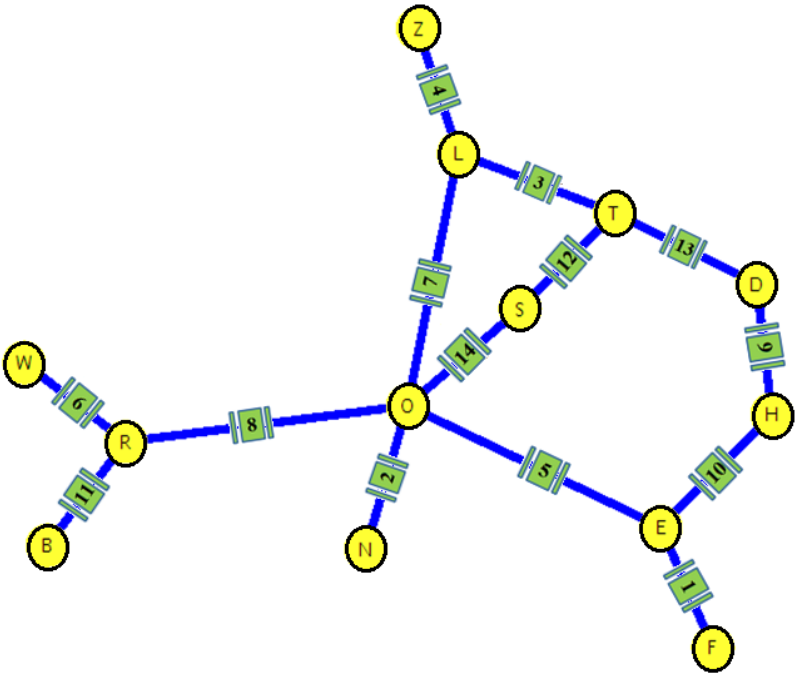

Figure 2: Crib-Ciphertext menu.

This geometric representation is known as a *menu*. In the Turing bombe, the links connecting each letter in the menu were actually cables that inside contained 26 individual wires, each representing one of the 26 alphabet letters.

In the Turing bombe, the plugboard, the rotors, and the reflector were connected in a "double-ended" fashion. The effect of going through an ordinary Enigma and a double-ended version is the same, except that current in the double-ended travels in the same direction. In the Turing bombe, Enigma rotors were not used but were emulated by cylindrical disks called *drums*. The combination of a letter passing through the rotors (drums) in both directions and the reflector we will denote as a *double scrambler*. On both sides of the double scrambler was a cable connection containing 26 wires, where each individual wire denoted a single letter of the alphabet. On the bombe, three drums emulating the three rotors in the double scrambler were mounted vertically on a sequence of shafts, with the top drum emulating the right most rotor of the Enigma, the middle drum emulating the middle rotor of the Enigma, and the bottom drum emulating the left most rotor.

Recall that of the approximately 5.5879×10^{24} initial configurations of the Wermacht Enigma, the plugboard contributed by far the largest factor, 150738274937250, of this number. We next discuss how Turing overcame this factor and used the plugboard as a weakness to exploit the Enigma.

4. Menu Loops and Logical Consistencies

Figure 2 illustrates when a input letter is encrypted with the Enigma, it travels through the plugboard, through the double scrambler, and back through the plugboard where it reaches its output letter. Since the plugboard connections where unknown, a letter, normally one in the menu that had the most links connected to it, was chosen. This chosen letter in the menu was known as the *central letter*.

Once a central letter was selected, next a possible plugboard partner was selected for the central letter. In Figure 1, suppose we select the letter O as our central letter and select and arbitrary letter, we will call for now α , as its plugboard partner. Physically in the bombe, the letter α would be activated by applying a voltage to its relay represented in a 26-relay device containing representations for all 26 letters known as the *indicator unit*. The indicator unit was connected to the central letter by a cable of 26 wires representing each alphabet letter. Consider the case on the menu in Figure 2 where the letter O is encrypted as the letter S. Since α is the plugboard partner of O, for the encryption process, it will transformed by the rotors and reflector of the Enigma (for the Turing bombe, the double scrambler encrypting or decrypting the letter at position 14), where upon completion results in another letter, which we call β. The letter β is then sent to the plugboard where it is either switched to or stays the same and becomes the letter S. Note that since the plugboard partner for the central letter is chosen, the possible plugboard partners of the other letters will be determined only by the result of the output of the double scrambler. This process is described in Figure 3.

The menu in Figure 3 has three sequences of links that formed closed loops. Starting with the letter O, one of the loops generates the closed letter sequence O → S → T → L → O with corresponding double scrambler link positions between these letters given by 14, 12, 3, 7. Continuing around the loop, the plugboard partner β of the letter S will be transformed by double scrambler 12 to the

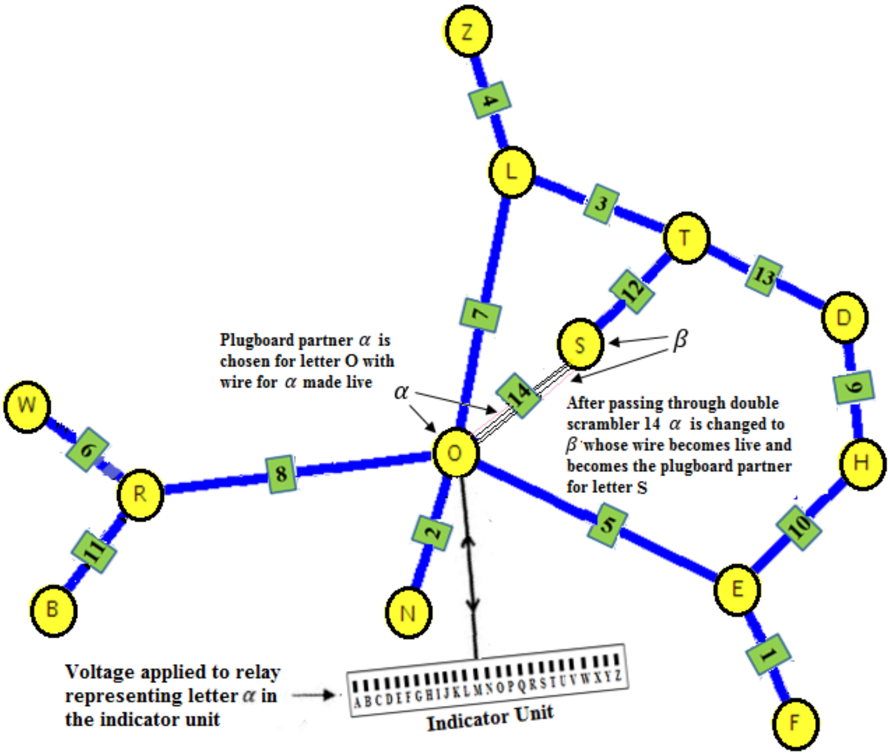
Z
4
L
3
T
13
D
7
12
S
β
9
Plugboard partner α is chosen for letter O with wire for α made live
α
14
After passing through double scrambler 14 α is changed to β whose wire becomes live and becomes the plugboard partner for letter S
W
6
R
8
O
H
11
2
5
10
B
N
E
1
F
Voltage applied to relay representing letter α in the indicator unit
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Indicator Unit

Figure 3: Menu illustrating plugboard partners between letters.

plugboard partner of the letter T, which we label as δ. Next, upon transformation by double scrambler 3 transforms δ to the plugboard partner, which we label as μ, of letter L. Figure 4 demonstrates how the double scramblers for this loop encipher the plugboard partners of each letter in the loop.

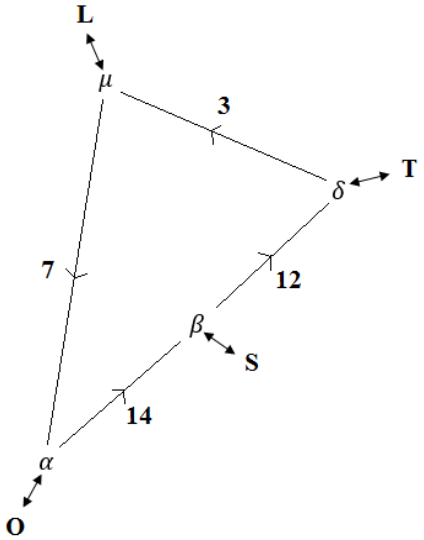

Figure 4: Closed loop containing plugboard pairs.

Because the double scramblers are only applied to the plugboard partner of each menu letter, the identity of the plugboard pairs β, δ, and μ are only dependent on the initial choice of α and the positions of the double scramblers, but not on any of the letters that occur in the original menu.

From the letters on menu loops and their corresponding plugboard pairs as described in Figure 4, Turing was able to formulate a working hypothesis about the rotor order, rotor positions, and reflector that was used when an intercepted message was encrypted. Recall for any encrypted message formed with the Enigma, each letter was assigned only one letter to be its corresponding plugboard letter (or left as itself), and this plugboard assignment was never changed in a message encryption or decryption. Suppose in Figure 1.8 we choose a plugboard partner for the letter O, say $\alpha = \text{A}$. Suppose in addition that we know the correct reflector, rotor order, initial window letters, and ring settings that was used in performing the message encryption. When applying double scramblers 14, 12, and 3 to α , we would generate plugboard partners β , δ , and μ of the menu letters S, T, and L, respectively. Next, we would apply double scrambler 7 to μ . Assuming the reflector, rotor order, initial window letters, and ring settings were correct, the result of applying double scrambler 7 to μ must be $\alpha = \text{A}$ to be logically consistent, for if not, then the fact that each menu letter had to have a unique plugboard partner would be contradicted. This scenario led to several possibilities about whether the choice of the plugboard partner $\alpha = \text{A}$, choice of the reflector, rotor order, or rotor configuration were correct, which we will describe next.

5. Searching for the Correct Settings

To search for correct settings, recall that there are only two possible reflectors and sixty possible ways to arrange the three rotors chosen from five in the Enigma. Both could be tested in a reasonable time by brute force. The number of possible initial window letter settings was $26^3 = 17576$ and number of ring settings was $26^3 = 17576$. However, it was not necessary to find the correct settings for both the initial window letters and ring settings by brute force. For the individual rotors, it was only necessary to find the *rotor core starting positions*. The rotor core starting positions are determined by the offsets defined by the initial window letters and ring settings. For example, suppose that for an Enigma encryption with three rotors, the initial window letters are TRA (alphabet letter numbers 20 18 1) and the ring settings are 10 15 21. Then, the rotor offsets for this encryption would be 10 15 6 since $20-10=10$, $18-15=3$, and $1-21=-20$ with $-20+26=6$. However, if we performed the same encryption with these rotors with initial window letters ZZZ (alphabet letter numbers 26 26 26) and ring settings 16 23 20, the offsets would be again 10 3 6 since $26-16=10$, $26-23=3$, and $26-20=6$. Then, assuming the same reflector was used, the same rotor order, and that there were no rotor turnovers of the middle and left rotors for either scenario, then the encryption using these rotors with initial window letters of TRA and ring settings 10 15 21 would give the exact same encryption with initial window letters of ZZZ and ring settings of 16 23 20. Therefore, to test for the correct rotor core starting positions, a setting for the initial window letters, in many cases ZZZ, was chosen, and the ring settings were found by brute force, resulting in $26^3 = 17576$ possibilities, until the correct rotor core starting positions were found. Therefore, to find a plugboard partner for the central letter that was logically consistent, a combination of a reflector (either B or C), a rotor order (60 possible), and a ring setting (17576 possible) was tested, giving a maximum total of $2 \cdot 60 \cdot 17576 = 2109120$ possible settings that required testing. Although significant, it was an astronomical reduction compared to the approximately 5.5879×10^{24} initial configurations that the Wermacht Enigma presented.

To begin the process, as was mentioned before, a central letter was selected (usually one with the most links attached to it), a reflector and rotor order was chosen, and a ring setting was tested. A voltage was applied to one of the twenty six relays in the indicator unit connected to the central letter representing a possible plugboard partner to that letter. If there was a loop available in the menu, a test was made to see if the menu was logically consistent. We describe the start of this process in the following example.

Example 1: Consider the following crib-ciphertext match.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Crib:	F	O	L	L	O	W	O	R	D	E	R	S	T	O
Cipher:	E	N	T	Z	E	R	L	O	H	H	B	T	D	S

We will assume that this encryption was produced by a Wermacht Enigma with reflector C and rotor order I V III. We also assume that the initial window letter settings are ZZZ, which corresponds to the initial drum letter settings on the bombe, and ring settings are 16 23 18. Given the fact that when a key is pressed to encrypt a letter on the Enigma, the right rotor will advance one letter (we will assume there will be no turnover for the middle and left rotors), then the window letters for the encryption performed between each crib-ciphertext assignment will be as follows.

	ZZA	ZZB	ZZC	ZZD	ZZE	ZZF	ZZG	ZZH	ZZI	ZZJ	ZZK	ZZL	ZZM	ZZN
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Crib:	F	O	L	L	O	W	O	R	D	E	R	S	T	O
Cipher:	E	N	T	Z	E	R	L	O	H	H	B	T	D	S

Figure 5 shows the menu for this crib-ciphertext pair with the window letters with double scrambler numbers expressed with each link. Table 1 on the next page shows the resulting Enigma encryption of every alphabet letter for the given settings.

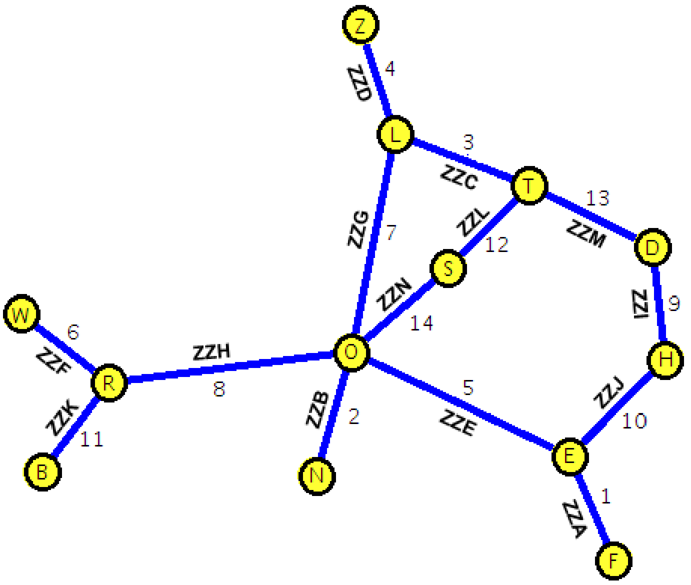

Figure 5: Triple loop menu with encryption window letters.

Consider the loop in the menu given by the letters O S T L O. Suppose we choose the letter A to be the plugboard partner of the letter O. If a voltage is applied to the letter A and this letter is input into double scrambler 14 with drum letter setting ZZN, the output using Table 1 by looking where

	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**
ZZA	I	C	B	E	D	Q	J	P	A	G	R	W	V	X	U	H	F	K	Z	Y	O	M	L	N	T	S
ZZB	M	R	X	K	H	N	Z	E	O	P	D	V	A	F	I	J	W	B	T	S	Y	L	Q	C	U	G
ZZC	D	Z	K	A	G	H	E	F	N	M	C	T	J	I	U	W	V	Y	X	L	O	Q	P	S	R	B
ZZD	F	N	H	P	L	A	K	C	Z	V	G	E	U	B	W	D	Y	S	R	X	M	J	O	T	Q	I
ZZE	L	F	P	H	S	B	X	D	J	I	T	A	U	Z	V	C	Y	W	E	K	M	O	R	G	Q	N
ZZF	M	J	D	C	O	S	I	W	G	B	T	U	A	V	E	Y	R	Q	F	K	L	N	H	Z	P	X
ZZG	H	E	K	O	B	G	F	A	S	T	C	M	L	U	D	V	W	X	I	J	N	P	Q	R	Z	Y
ZZH	N	F	G	L	Y	B	C	S	T	W	O	D	V	A	K	X	Z	U	H	I	R	M	J	P	E	Q
ZZI	B	A	Q	P	I	R	S	X	E	Y	L	K	U	Z	V	D	C	F	G	W	M	O	T	H	J	N
ZZJ	X	D	M	B	R	T	I	W	G	N	O	S	C	J	K	V	U	E	L	F	Q	P	H	A	Z	Y
ZZK	E	M	W	S	A	G	F	U	Z	K	J	R	B	P	T	N	Y	L	D	O	H	X	C	V	Q	I
ZZL	U	L	Q	H	Z	M	P	D	T	V	S	B	F	O	N	G	C	X	K	I	A	J	Y	R	W	E
ZZM	S	L	D	C	X	N	M	V	J	I	R	B	G	F	U	W	T	K	A	Q	O	H	P	E	Z	Y
ZZN	G	K	T	R	J	P	A	L	W	E	B	H	N	M	S	F	V	D	O	C	Y	Q	I	Z	U	X

Table 1: Double Scrambler menu results for reflector C, rotor order I V III, ring settings 16 23 18.

ZZN in the last row matches A in the first column gives the letter G, which would be the plugboard partner for menu letter S. The letter G input into double scrambler 12 with drum letter setting ZZL yields the letter P, which represents the plugboard partner of the menu letter T. Next, inputting the letter P into double scrambler 3 with drum letter setting ZZC yields the letter W, which represents the plugboard partner of the menu letter X. Finally, inputting letter W into double scrambler 7 with drum letter setting ZZG yields the letter Q, which represents the plugboard partner of the menu letter O. In summary, we have the following menu letter/plugboard partners for this loop.

$$\begin{array}{rccccccccc}
\text{Drum Setting :} & & \text{ZZN} & & \text{ZZL} & & \text{ZZC} & & \text{ZZG} & \\
\text{Menu Letter :} & \text{O} & \to & \text{S} & \to & \text{T} & \to & \text{X} & \to & \text{O} \\
\text{Plugboard Partner :} & \text{A} & \to & \text{G} & \to & \text{P} & \to & \text{W} & \to & \text{Q}
\end{array}$$

Since the central letter O has two plugboard partners A and Q, a logical inconsistency has occurred. Hence, the letter A is not a plugboard partner for O, the wrong reflector, or the wrong ring settings have been chosen. If the letter S is chosen as the initial plugboard partner of the central letter O, we have the following menu letter/plugboard partners for the loop.

$$\begin{array}{rccccccccc}
\text{Drum Setting :} & & \text{ZZN} & & \text{ZZL} & & \text{ZZC} & & \text{ZZG} & \\
\text{Menu Letter :} & \text{O} & \to & \text{S} & \to & \text{T} & \to & \text{X} & \to & \text{O} \\
\text{Plugboard Partner :} & \text{S} & \to & \text{O} & \to & \text{R} & \to & \text{I} & \to & \text{S}
\end{array}$$

This loop has no logical inconsistencies. This indicates the possibility that a correct plugboard partner (the letter S) for the central letter O has been found. □

Once a chosen letter produced a logical inconsistency, Turing found it more efficient, instead of searching the rest of the alphabet to find a possible letter that was logically consistent for the given loop, to instead let the plugboard letter output from the loop for the central letter to serve as the next input to the loop, letting this letter transpire through the loop, and performing this task over and over again until a cycle was produced. Each non-plugboard partner was recorded by the indicator unit and this method gives a way of eliminating possible plugboard pairs for the central letter. The next example demonstrates this process.

Example 2: Recall in Example 1 we saw that when letter A was selected as a plugboard letter for the central letter O, the following menu letter/plugboard partners for the loop O S T L O was given by the following result.

$$\begin{array}{rccccccccc}
\text{Drum Setting :} & & \text{ZZN} & & \text{ZZL} & & \text{ZZC} & & \text{ZZG} & \\
\text{Menu Letter :} & \text{O} & \to & \text{S} & \to & \text{T} & \to & \text{X} & \to & \text{O} \\
\text{Plugboard Partner :} & \text{A} & \to & \text{G} & \to & \text{P} & \to & \text{W} & \to & \text{Q}
\end{array}$$

We take the output plugboard partner Q for the central letter O and reinsert it into the loop. Using Table 1, this gives the following menu letter/plugboard partners.

$$\begin{array}{rccccccccc}
\text{Drum Setting :} & & \text{ZZN} & & \text{ZZL} & & \text{ZZC} & & \text{ZZG} & \\
\text{Menu Letter :} & \text{O} & \to & \text{S} & \to & \text{T} & \to & \text{X} & \to & \text{O} \\
\text{Plugboard Partner :} & \text{Q} & \to & \text{V} & \to & \text{J} & \to & \text{M} & \to & \text{L}
\end{array}$$

Next, taking the output plugboard partner L for the letter O and reinserting it into the loop gives the following menu letter/plugboard partners.

$$\begin{array}{rccccccccc}
\text{Drum Setting :} & & \text{ZZN} & & \text{ZZL} & & \text{ZZC} & & \text{ZZG} & \\
\text{Menu Letter :} & \text{O} & \to & \text{S} & \to & \text{T} & \to & \text{X} & \to & \text{O} \\
\text{Plugboard Partner :} & \text{L} & \to & \text{H} & \to & \text{D} & \to & \text{A} & \to & \text{H}
\end{array}$$

We continue until we obtain the original plugboard partner A, of the letter O, giving these results.

$$\text{A} \to \text{Q} \to \text{L} \to \text{H} \to \text{Y} \to \text{O} \to \text{K} \to \text{J} \to \text{E} \to \text{W} \to \text{M} \to \text{N} \to \text{A}$$

We represent the previous result as the cycle (AQLHYOKJEWMN). These are letters that are not legitimate plugboard partners for the central letter O.

The cycle that can be generated using the loop is dependent on the initial plugboard partner chosen for the central letter O. The following is a representation of all of the cycles that can be generated by the loop O S T L O by using the letters A, B, C, P, S, and Z as initial plugboard partners.

(AQLHYOKJEWMN) (BRGDIXF) (CVU) (PT) (S) (Z) □

As seen by the last example, using the loop O S T L O in Figure 5 to generate the cycle (AQLHYOKJEWMN) gives a sequence of letters that would not be plugboard partners of the central letter O. However, this one cycle does not generate all of the letters that are not plugboard partners. One method to find other non-plugboard partners is to pick another letter not in the cycle (AQLHYOKJEWMN) and see if it generates another cycle with a logical inconsistency. However, a more efficient way of doing this for menus that have more than one loop is to generate a cycle using another loop and using the loops together to generate more non-plugboard partners. We demonstrate how this can work in the following example.

Example 3: The menu in Figure 5 has three loops, which we can label as O S T L O, O S T D H E O, and O L T D H E O. In Example 1.2 on page 16, we generated a sequence of cycles representing potential plugboard partners and non-plugboard partners for the central letter O using the loop O S T L O. Using Table 1, we can generate similar cycle sequences of for the loops O S T D H E O and O L T D H E O. For various initial plugboard partners, the following summarizes the cycles generated.

Loops	**Cycles**
O S T L O	C_1 = (AQLHYOKJEWMN), C_2 = (BRGDIXF),
	C_3 = (CVU), C_4 = (PT), C_5 = (S), C_6 = (Z)
O S T D H E O	D_1 = (ABHVOKGCNQWUFYJZELMP), D_2 = (DXRT),
	D_3 = (I), D_4 = (S)
O L T D H E O	E_1 = (AQBYVFRHMUNPDCOJGT), E_2 = (EZ),
	E_3 = (IX), E_4 = (K), E_5 = (LX), E_6 = (S)

If the letter S is chosen as a plugboard partner for the letter O, then the single element cycle (S) occurs for all loops. This fact says that the letter S could be a plugboard partner for the letter O.

However, if a wrong plugboard partner is initially chosen, the correct possible plugboard partner can be determined in the following manner. If the letter A is chosen as the initial plugboard partner to O, the cycles C_1 = (AQLHYOKJEWMN) and D_1 = (ABHVOKGCNQWUFYJZELMP), and E_1 = (AQBYVFRHMUNPDCOJGT) together indicate that the twenty three letters A, B, C, D, E, F, G, H, J, K, L, M, N O, P, Q, R, T, U, V, W, Y, and Z are non-plugboard pairs of the letter O. Since cycles D_1 and E_1 indicate the letter B is a non-plugboard partner, cycle C_2 is activated, which contributes the two additional letters I and X as non-plugboard partners of the letter O. Hence, as can be seen in the following table, when the letter A is designated as a plugboard partner of the letter O, all of the twenty-five false possible plugboard partners will be generated.

Test Letters	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
Input Letter A	A																									
Cycle C_1	A				E			H		J	K	L	M	N	O		Q						W		Y	
Cycle D_1	A	B	C		E	F	G	H		J	K	L	M	N	O	P	Q				U	V	W		Y	Z
Cycle E_1	A	B	C	D		F	G	H		J			M	N	O	P	Q	R		T	U	V			Y	
C_1, D_1, E_1	A	B	C	D	E	F	G	H		J	K	L	M	N	O	P	Q	R		T	U	V	W		Y	Z
B input to C_2		B																								
Cycle C_2		B		D		F	G		I									R						X		
C_1, D_1, E_1, C_2	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R		T	U	V	W	X	Y	Z

The same result can be obtained by choosing other letters as the initial plugboard partners. The only letter not reached is the letter S, which would represent the only letter relay in the indicator unit not receiving a voltage. Hence, this represents another method for determining that the letter S is a potential plugboard partner for the letter O. □

It should be emphasized that the letter S found in Example 2 is only a candidate plugboard partner for the central letter O. Normally, when the Enigma machine was used in the field, 10 plugboard cables were used to switch 20 total letters in pairs. Hence, once the first potential plugboard partner of a letter was found, it was necessary to find 9 other letters that had plugboard partners, plus the six other letters that were not swapped by a cable. A device known as the *checking machine* was designed to determine if the potential plugboard partner of the central letter was correct and to look for the other plugboard partners. In fact, when finding an initial plugboard of the central letter, it was common for the Turing bombe to find other logically consistent plugboard candidates, which could result in a *false stop*. In most scenarios, however, no plugboard partner was found that produced a logically consistent result. Normally, these three cases would occur.

- The reflector, rotor order, ring settings, and initial plugboard partner chosen for the central letter in the menu have no logical inconsistencies that occur for any loops. Only one relay in the indicator unit containing the letter with the potential plugboard partner will have voltage applied to it. The bombe will stop and the information will be recorded and sent to the checking machine for further analysis. In this case, all loops tested are logically consistent.

- The reflector, rotor order, ring settings and the initial plugboard partner chosen for the central letter in the menu produce logical inconsistencies for twenty-five letters. In this case, the twenty-five letters that are incorrect plugboard partners will have voltage applied to their corresponding relays in the indicator unit, and the relay letter containing the potential plugboard partner will have no voltage applied to it. The bombe will stop and the information will be recorded and sent to the checking machine for further analysis.

- The reflector, rotor order, or ring settings are incorrect. No matter what initial plugboard partner used, all outcomes will be logically false and all letters will not be plugboard partners. In this case, all twenty six relays representing letters in the indicator unit will have voltage. In this case, the bombe ignores this setting and proceeds by testing the next setting.

We describe the checking machine analysis for the potential plugboard partner next.

6. The Checking Machine

The checking machine had three drums designed similarly to the German Enigma rotors and a fourth drum that functioned as a reflector. Each drum had a ring of letters on the outside designed for indicating the menu letter positions and an adjustable ring designed to designate a ring setting. The machine had twenty-six letter keys designed to input letters and twenty-six lamps to indicate the corresponding output letters. When a bombe stop occurred, the settings were recorded and used for the selection of the drums, ring settings, and reflector on the checking machine. Using the plugboard letter of the central letter producing the stop, the checking machine drums would be moved to positions corresponding to the menu link positions required to transform the plugboard letter of the central letter to the plugboard letter of another menu letter. The process was continued for other links with the goal of finding plugboard partners of all menu letters. Since the Germans typically used ten cables to connect twenty of the letters, the goal of the entire process was to find the twenty letters that were plugboard partners. Once these letters were found, the other six letters were assumed to not have plugboard partners. However, the results were only used if no letters with multiple plugboard partners in the menu occurred. If a logical inconsistency occurred, the bombe stop was assumed to be false and the results disregarded. The bombe was restarted to look for other settings that would decipher the message. If the menu was logically consistent, the information was recorded and the message sent for decipherment. We illustrate how this process worked next.

Example 4: For the menu given in Figure 1, we saw in Example 3 when the reflector C, rotor order I V III, and ring settings 16 23 18 are tested, the letter S was a candidate plugboard partner of the central letter O. Using Table 1, we start at O and first proceed along the menu using these steps.

Drum Setting :		ZZB			ZZH			ZZF			ZZK	
Menu Letter :	O	→	N	O	→	R	R	→	W	R	→	B
Plugboard Partner :	S	→	T	S	→	H	W	→	W	H	→	U

These results indicate, besides the letters O ↔ S, that the letters N ↔ T, R ↔ H and B ↔ U are plugboard partners. The letter W has no plugboard partner. To find other partners, we proceed around the menu for the loops O S T L O and O L T H E O and get the following results.

Drum Setting :		ZZN		ZZL		ZZC		ZZG			ZZD	
Menu Letter :	O	→	S	→	T	→	L	→	O	L	→	Z
Plugboard Partner :	S	→	O	→	N	→	I	→	S	I	→	Z

Drum Setting :		ZZG		ZZC		ZZM		ZZI		ZZJ		ZZE	
Menu Letter :	O	→	L	→	T	→	D	→	H	→	E	→	O
Plugboard Partner :	S	→	I	→	N	→	F	→	R	→	E	→	S

Drum Setting :		ZZO	
Menu Letter :	E	→	F
Plugboard Partner :	E	→	D

No logical inconsistencies for plugboard partners of menu letters moving through these two loops. Confirming that the loop O S T D H E O produces no logical inconsistencies will be left as an exercise. These results give the additional information that the letters I ↔ L and D ↔ F are plugboard partners. The letters E and Z have no plugboard partners. In summary, assigning the letter S as a plugboard partner to central letter O produces no logical inconsistencies in the menu. The checking machine process confirms that the letters O ↔ S, N ↔ T, R ↔ H, B ↔ U, I ↔ L, and D ↔ F are plugboard partners. The letters W, E, and Z have no plugboard partners.

Unfortunately, if we assume that when messages where encrypted on the Enigma that the Germans used 10 cables for the plugboard to assign partners to 20 of the letters, the results are incomplete. We have found only six pairs of letters (12 total) that are switched by the plugboard with three letters with no plugboard partners. However, by using the crib/ciphertext encipherment plus decrypting other parts of ciphertext with the plugboard settings already discovered, the other plugboard assignments can be found. Going back to the crib and the ciphertext, the crib **FOLLOWORDERSTO** was known to encrypt to part of the ciphertext message **NUENTZERLOHHBTDSHLHIY** that is underlined. Using the plugboard assignments already found and Table 1, we can set up the following table assignment.

	ZZY	ZZZ	ZZA	ZZB	ZZC	ZZD	ZZE	ZZF	ZZG	ZZH	ZZI	ZZJ	ZZK	ZZL
Cipher:	N	U	E	N	T	Z	E	R	L	O	H	H	B	T
Plugb:	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
	T	B	E	T	N	Z	E	H	I	S	R	R	U	N
Dscram:	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
	A	S	D	S	I	I	S	W	S	H	F	E	H	O
Plugb:	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Plain:	.	O	F	O	L	L	O	W	O	R	D	E	R	S

	ZZM	ZZN	ZZO	ZZP	ZZQ	ZZR	ZZS
Cipher:	D	S	H	L	H	I	Y
Plugb:	↕	↕	↕	↕	↕	↕	↕
	F	O	R	I	R	L	.
Dscram:	↕	↕	↕	↕	↕	↕	↕
	N	S	P	B	L	K	.
Plugb:	↕	↕	↕	↕	↕	↕	↕
Plain:	T	O	.	U	I	.	.

If we assume that paired with the second plaintext letter O a two letter word is formed, candidate words would be DO, GO, NO, SO, and TO. However, the letters D, N, S, and T already have plugboard partners. Hence, the likely first plaintext letter would be G, which says that the letters A ↔ G, are plugboard partners. Now consider the next unknown plaintext letter produced by the setting ZZO. Since the plaintext letters that follow are U and I, a good guess would be P ↔ Q are plugboard partners. This would spell out the phrase QUI. Assuming the last two missing plaintext letters spells out the word QUICK, it can be shown that K ↔ C and J ↔ Y are plugboard partners, giving the plaintext of **GO FOLLOW ORDERS TO QUICK** (more plaintext letters would likely follow in this message). Hence, we have found that ten plugboard partner assignments are A ↔ G, B ↔ U, C ↔ K, D ↔ F, H ↔ R, I ↔ L, J ↔ Y, N ↔ T, O ↔ S, and P ↔ Q. The letters E, M, V, W, X and Z have no plugboard assignments. □

7. Conclusion

In this paper, we have described the basic steps in how the Turing bombe was used to overcome and use the most significant factor, the plugboard, in order, to decipher messages with the Enigma World War army cipher. However, many more aspects, including working with menus with fewer loops, when rotor turnovers occur, and false stops have not been discussed. More details can be found in [1] and [3]. A detailed discussion on how a symbolic manipulator such as Maple can be used to search for and verify potential settings will be given in [3].

References

[1] Carter, F. (2008). *The Turing Bombe*. Bletchley Park Trust, 2008.

[2] Klima, R. and Sigmon, N. (2012). *Cryptology Classical and Modern with Maplets*. Boca Raton, FL: Taylor & Francis Group, LLC.

[3] Klima, R. and Sigmon, N. (to be published in 2018). *Cryptology Classical and Modern, Second Edition*. Boca Raton, FL: Taylor & Francis Group, LLC.