Eric Roberts
CS 106A

Handout #29
February 3, 2016

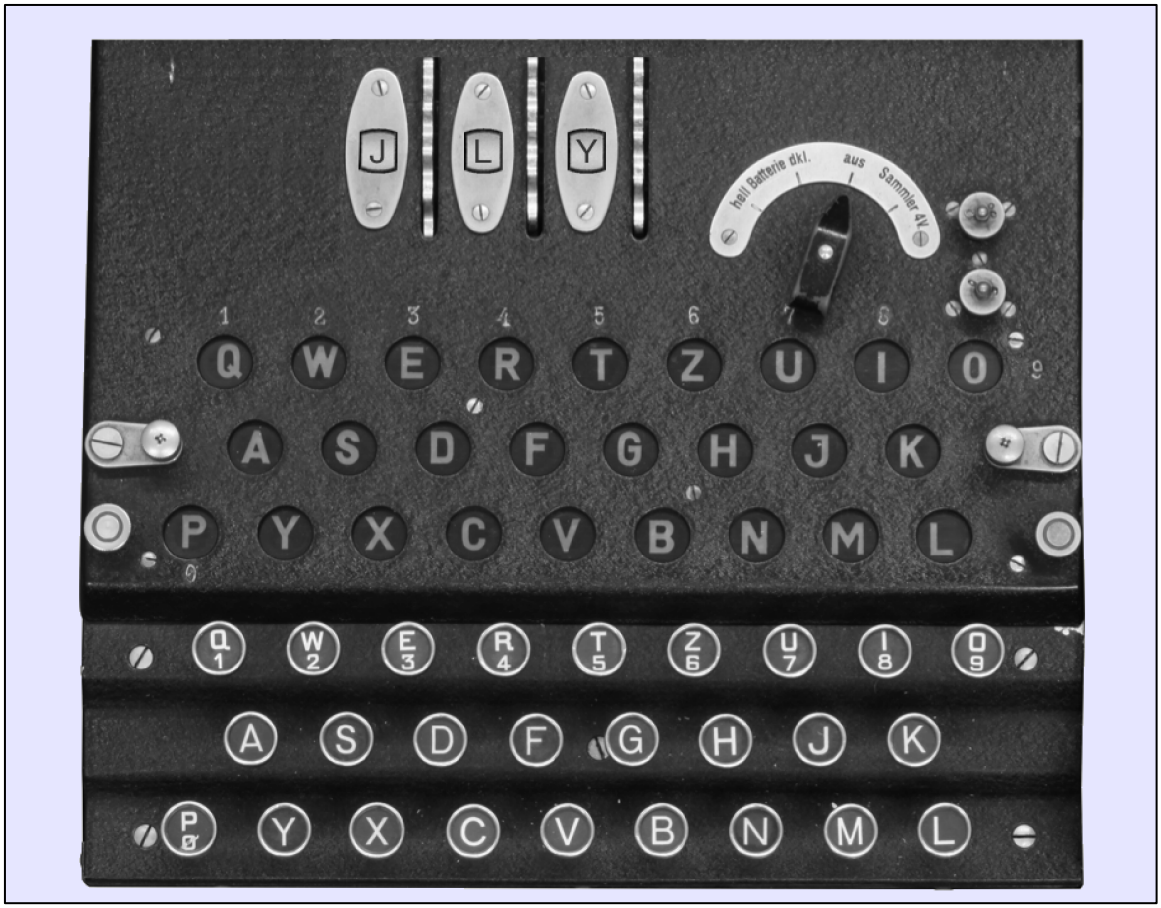
The Enigma Machine

In World War II, a team of British mathematicians working at a secret facility called Bletchley Park was able to break the German military code, which allowed the Allies to decipher German military communication. Breaking the German codes was an early application of ***cryptography,*** which is the science of creating and decoding messages. In the language of cryptography, the message you want to send is called the ***plaintext;*** the encoded message is called the ***ciphertext.*** Decoding the ciphertext typically depends on having the sender and receiver share a privileged piece of information called a ***key.***

In the early 1930s, the German military adopted a new encryption protocol based on an existing commercial device called ***Enigma.*** This handout describes the operation of the Enigma machine in enough detail for you to simulate its function in Assignment #4. A detailed account of how the Bletchley Park mathematicians broke Enigma is available on the CS 106A website.

Figure 1 shows the top view of a typical Enigma machine, expanded so that you can see the detail. At the bottom of the figure is a keyboard arranged in the standard German

Figure 1. Top view of a typical Enigma machine



layout. Above the keyboard is an array of lamps. Pressing a key lights one of the lamps, thereby indicating the encoded version of that letter. The mapping from keys to lamps is controlled by the three thumb wheels at the top of the diagram, which are called ***rotors.*** Each rotor—which were chosen from a stock of five rotors from which any three could be used in any order—can be set to any of 26 positions corresponding to the letters of the alphabet. The display windows at the top of Figure 1 show the letters **JLY**. The three letters together are called the ***rotor setting.***

Enigma rotors are three-dimensional, which unfortunately makes them difficult to represent on the printed page. The letters you see through the display windows are printed around the circumference of the rotor. The left and right edges of a rotor have 26 electrical contacts aligned with the 26 letters. Those contacts are wired across the rotor so that each contact on the left connects to a contact on the right in some scrambled arrangement. Each rotor therefore implements a reordering of the letters of the alphabet, which mathematicians call a ***permutation.***

Figure 2 offers two views of this wiring. The diagram on the left shows how the rotor would appear when viewed along its axis. The letters circle the rotor, and the contacts shown at the outer edge go through the rotor and connect to the contact in the interior circle, which are physically on the opposite side of the rotor itself. The contact **A** on this side of the rotor connects to contact **E** on the reverse side, as shown by the heavy line in the diagram. The diagram on the right offers a schematic view in which the connections across the rotor appear in a linear arrangement. This diagram has the same wiring, as indicated by the connection between **A** on the right and **E** on the left. When you look at this "unrolled" view, you have to keep in mind that the top and bottom of the rectangle are in fact connected together to form a circle.

Figure 2. Two views of the wiring of an Enigma rotor

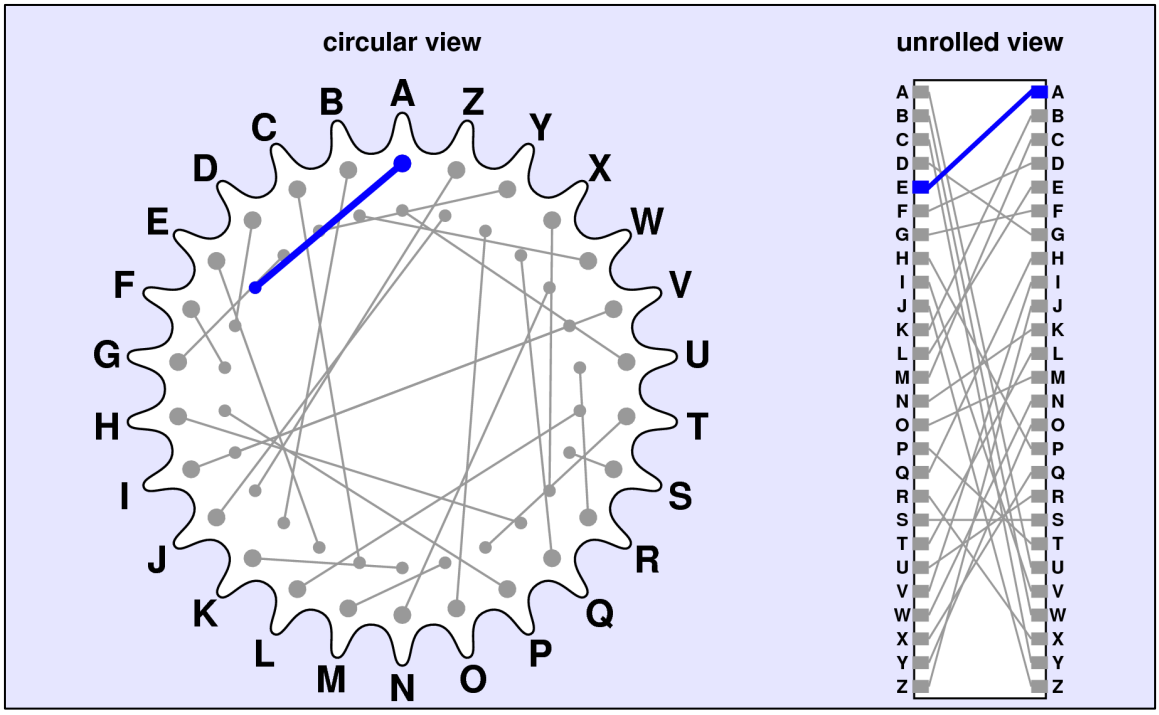


Figure 3 shows the internal structure of the Enigma machine, focusing on how the wiring of the three rotors makes it possible to encrypt information as it passes from one side of the machine to the other. Typing a character on the keyboard automatically advances the rotor on the right, thereby changing the pattern of connections inside the machine. When the rightmost rotor has completed a full revolution, the middle rotor advances one step; in much the same way, completing a revolution of the middle rotor advances the leftmost rotor. The rotors therefore advance in a fashion reminiscent of the odometer on a car. The right rotor advances on every character and is therefore called the ***fast rotor.*** The middle rotor advances once every 26 characters and is called the ***medium rotor.*** The left rotor advances only once every 256 (26×26) characters and is called the ***slow rotor.***

Figure 4 at the top of the next page shows what happens if the operator types the **A** key. Pressing the key advances the fast rotor, which changes the rotor setting from **JLY** to **JLZ**. The Enigma machine then applies a current to the wire leading from the **A** key at the right edge of the diagram and, at the same time, disconnects the **A** lamp so that only the encrypted version of the letter appears. The current flows from the **A** key through each of the three rotors, moving from right to left. It then passes into a circuit element called the ***reflector,*** which implements a fixed permutation. From the reflector, the current flows back across the rotors in the opposite direction. As shown in the diagram, the current initiated by typing **A** ends up on the wire labeled **B**, which causes the **B** lamp to light. Thus, given the rotor setting **JLZ**, the ciphertext form of the letter **A** is **B**.

The encryption patterns generated by the Enigma machine are difficult to break because the encoding changes on every character. If the operator types a second letter **A** immediately after the first, the fast rotor again advances, which this time completes the

Figure 3. Structural diagram of the Enigma machine showing the rotor setting JLY

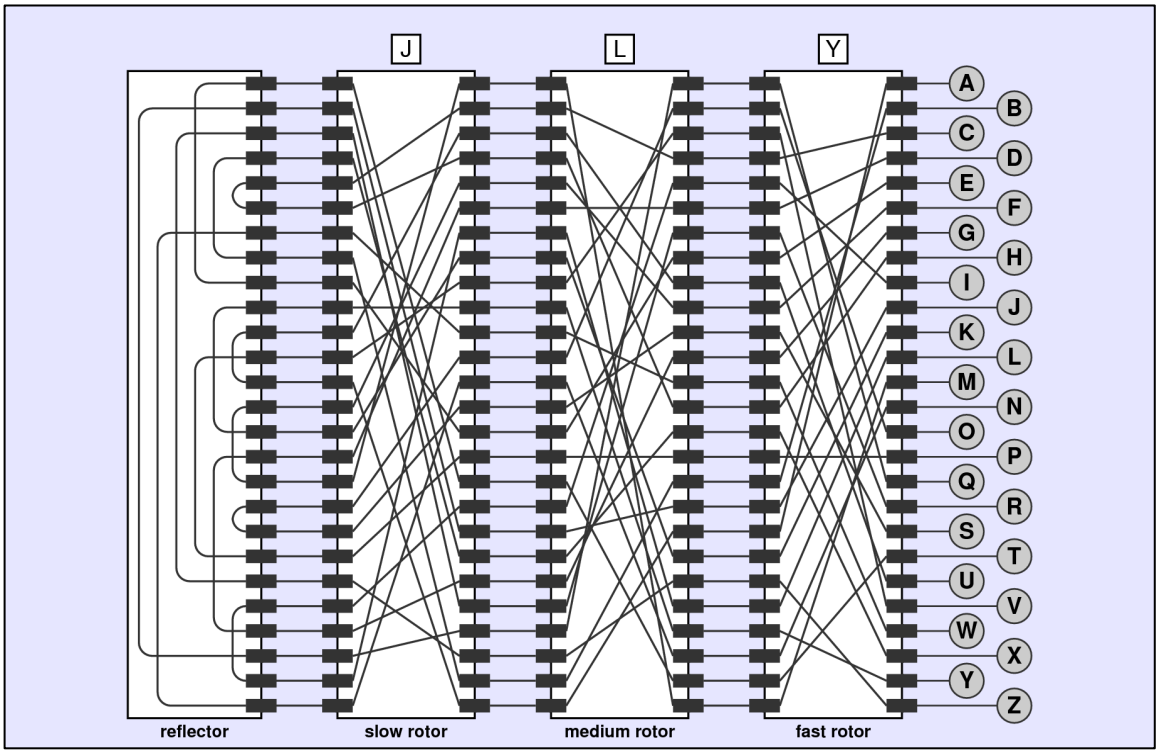
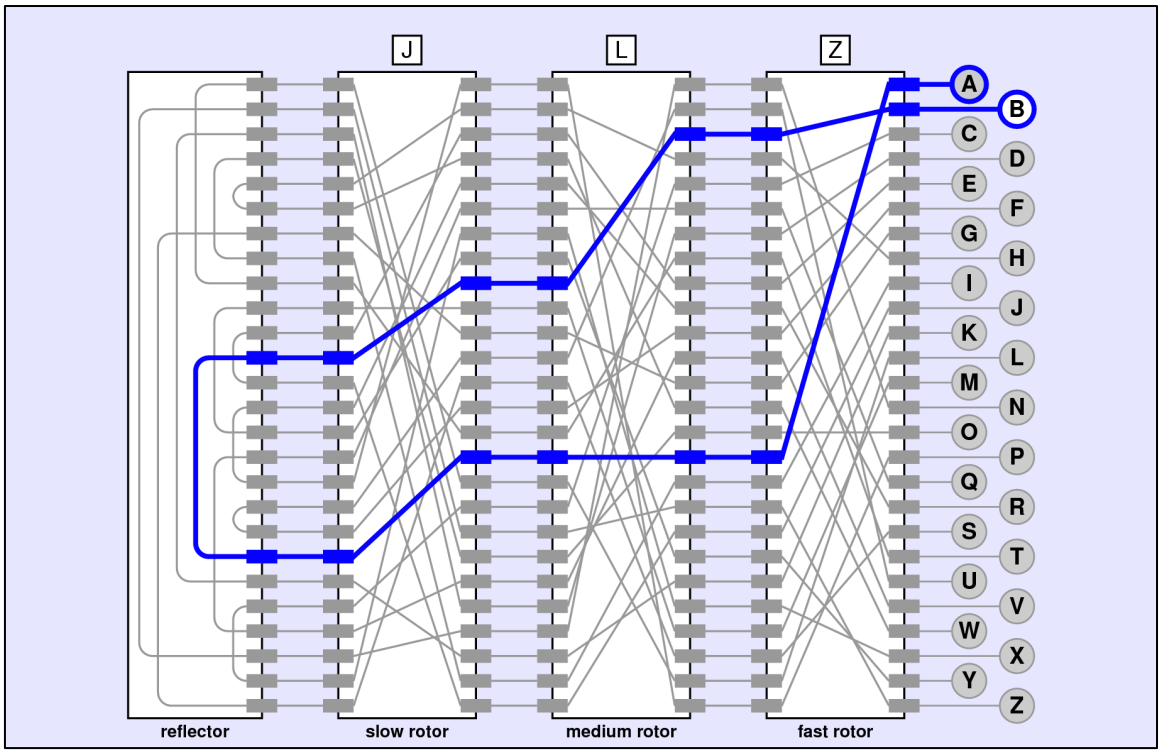
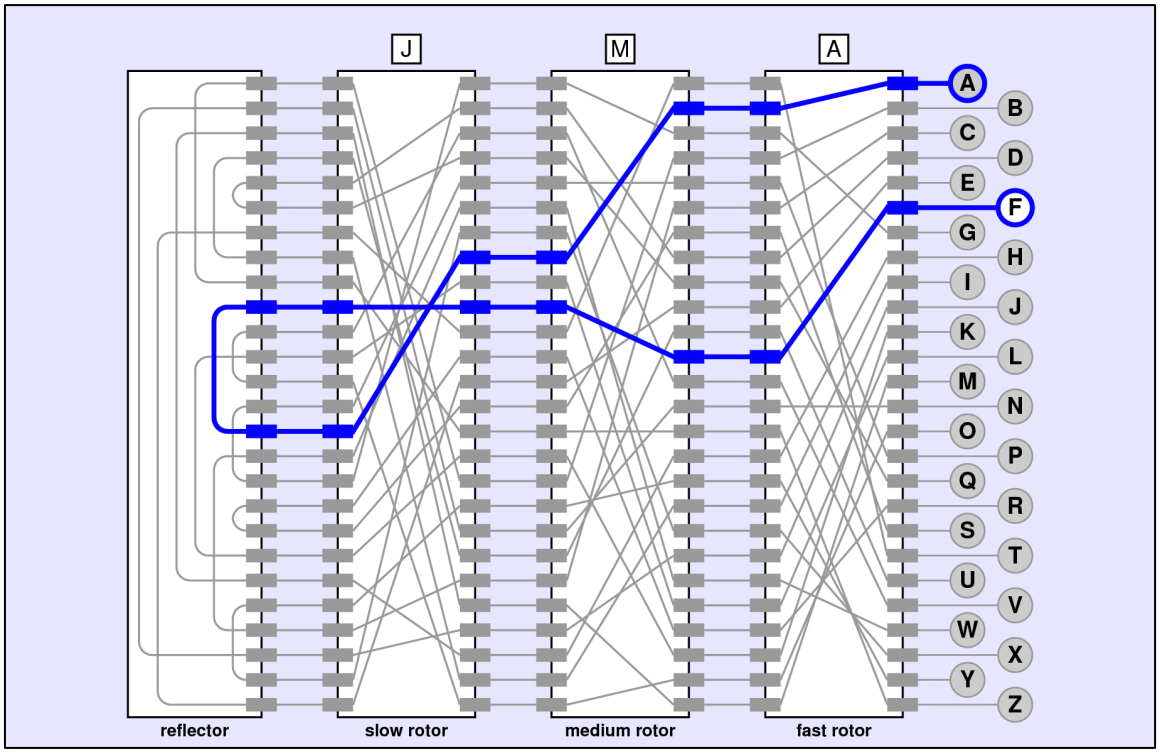


Figure 4. The Enigma machine after pressing the A key



alphabet and cycles back to **A**. Completing the cycle causes a "carry" to the medium rotor, which advances from **L** to **M**. Given the rotor setting **JMA**, the letter **A** is translated into the letter **F**, as shown in Figure 5.

Figure 5. The Enigma machine after pressing the A key a second time



At this point, it is useful to note a fundamental symmetry in the Enigma design. If **A** is transformed to **F** at some rotor setting, it must also be the case that **F** is transformed to **A**. The circuit is exactly the same; the only difference is that the current flows in the opposite direction. This symmetry is very useful for Enigma operators because it means that the sender and receiver don't need to have two different keys. The sender sets the rotors according to a codebook and types in the message. What comes out in the lights is the ciphertext, which is typically transmitted over a radio channel in Morse code. As long as the receiver uses the same codebook and sets up the machine in the same way, typing in the ciphertext restores the original message, because the encryption is reversible. As you will discover in the next section, however, the fact that the Enigma encoding is reversible also makes life easier for anyone trying to break the Enigma code.

Breaking the Enigma code

Although you don't need it for the assignment, the decryption strategy first developed in Poland and later refined at Bletchley Park is fascinating. The decryption technique relied on the following facts about the Enigma machine:

- *The Enigma encoding is symmetrical.* As noted in the preceding section, if the **A** key is transformed into the letter **F**, it must be the case that the **F** key would be transformed into **A** for that particular rotor setting.
- *The Enigma machine can never map a character into itself.* Because of its construction and the symmetry of the transformation, it is never possible to have the letter **A**, for example, come back as the letter **A**.

The codebreakers were also fortunate that the German military was rigid in its communication style, which made it possible to anticipate what the content of a message might be. In particular, the Germans routinely transmitted weather reports at specific times, which were easy to guess if you knew the weather at the point of transmission.

The known-plaintext attack

The strategy of breaking a code by guessing at least part of the plaintext and then using that guess to deduce the encryption pattern is called a ***known-plaintext attack.*** The character sequence that you believe you know is called a ***crib.*** Ironically, one of the best cribs available to the decoders at Bletchley Park—although the story may be apocryphal—occurred in messages from a German officer in the North Africa campaign who foolishly sent periodic messages containing the German equivalent of *nothing to report,* which is *keine besonderen ereignisse.*

Suppose that one of the Allied listening posts in North Africa had intercepted the following coded message:

UAUNFYRLPZSWMEDS I NFKRJXFSXKJCAXKEZ

If the sender is behaving in his usual way, you suspect that this message contains the plaintext sequence

KE I NEBESONDERENERE I GN I SSE

If you can figure out where in the message this sequence occurs, you can then use the pattern of letters to make deductions about the settings of the Enigma machine. If these deductions allow you to determine the rotor pattern, you've broken Enigma for that day.

Aligning the crib with the ciphertext

The first challenge in implementing the known-plaintext attack consists of figuring out where in the ciphertext the suspected crib might occur. Fortunately, many of the potential positions for the crib can be ruled out simply by taking note of the fact that the Enigma machine never translates a letter to itself. For example, the crib cannot occur at the beginning of the ciphertext because the letter **N** would map to itself in the fourth character position (as would the letter **E** further on), as shown in the following diagram:

```
U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
K E I N E B E S O N D E R E N E R E I G N I S S E
```

The codebreakers at Bletchley used the word ***crash*** to refer to positions at which a letter in the ciphertext matches its counterpart in the crib. The first step in the decryption process is to slide the crib under the ciphertext until no crashes occur.

Figure 6 on the next page shows what happens if you carry out this process for every possible alignment of the crib and ciphertext. There are only two possible alignments that produce no crashes, which arise from shifting the crib five and six characters to the right, respectively. If the crib is correct, it must be in one of those two positions.

After eliminating the alignments ruled out because of crashes, the cryptographers at Bletchley would then try each of the possible alignments to see whether any of the remaining possibilities gave rise to a consistent rotor setting.

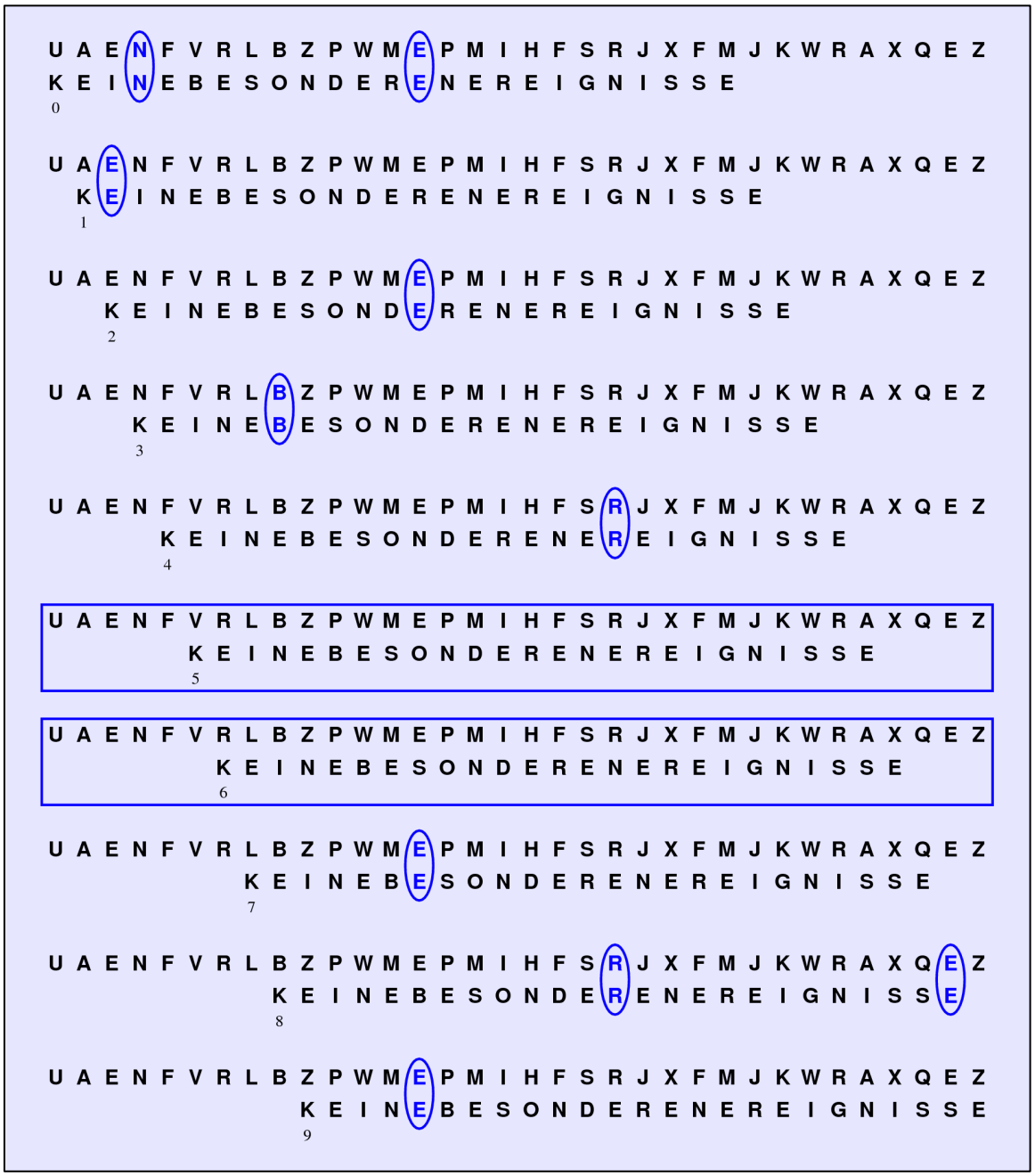
The critical insight that allowed the allies to break Enigma is that certain patterns in the letter pairings between the crib and the ciphertext are possible only with certain rotor settings. Consider, for example, the circled pairs of letters in the alignment at offset 5:

```
V R L B Z P W M E P M I H F S R J X F M J K W R A
K E I N E B E S O N D E R E N E R E I G N I S S E
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

The numbers below the characters keep track of the index of the character in the crib, beginning—as is conventional in computer science—at index position 0.

Assuming that the offset is correct, the Enigma machine encodes the plaintext **N** into **B** at index 3. At index 5, the machine turns **B** into **P**. At index 9, the letter **N** becomes a **P**. Given the symmetry of the Enigma machine, however, you know that typing a **P** at index 9 would have produced an **N**, which is the letter that began this chain back at offset 3. The transformation of **N** to **B, B** to **P**, and **P** to **N** forms a cycle, as follows:

Figure 6. All possible alignments of the crib and ciphertext, thereby showing the crashes



Alan Turing used the term ***loop*** to refer to this sort of closed cycle in the letter pairings between the crib and the ciphertext. Finding the loops and determining the pattern of the rotors is beyond the scope of this assignment, but you can read more about it in the chapter on cryptography from the book I'm writing on the great ideas of computer science. That chapter is available on the CS 106A website.