Puzzled: The Underlying Mathematics of the Enigma

New Mexico
Supercomputing Challenge

Final Report
April 4, 2007

Team 63
McCurdy School

Team Members

Sarah Armstrong
Joseph Koby
Juan-Antonio Vigil
Vanessa Trujillo

Teacher

Robert Dryja

Project Mentors

Bob Roby
Nick Bennett

Table of Contents

Executive Summary / Problem Statement

The team is creating a mathematical simulation of the Enigma machine. Its results are then compared to versions that have been confirmed to be accurate by the National Security Agency.

1. History

The "Enigma Machine" is a cryptographic machine that was created in the 1920's as a way for the German military to secure its communications. This cipher machine was used to encrypt and decrypt encoded messages. The German Army purchased these machines in 1925 from their original manufacturer, *Chiffriermaschinen Aktiengesellschaft.* The Enigma machine was extremely hard to crack, making it very valuable to the German Army. It wasn't until 1932 that three Polish mathematicians made progress on cracking the Enigma procedure. Once the procedure was cracked, this information was shared with the Allies, aiding them drastically in their victory over the Axis, especially in the Battle of the Atlantic.

Typical coded messages were disguised by weather reports. The coded messages usually consisted of: daily orders, troop movements, targets, and locations. Encoding and decoding messages was a process and usually took three to four people to complete the job; one person to read the encrypted message, one person to type, and one person to record the deciphered message. This is an Enigma Machine shown in figure 1.1

Fig. 1.1

2. Understanding the Enigma machine

In order to understand how there can be so many different ciphers with a three rotor Enigma Machine; one must understand the functions of the working parts and the math behind them. First we will explain what parts are in the Enigma Machine and how they work together to cipher or decipher a message.

There are six important components in the Enigma Machine that must be described before we explain how they all work together:

1. The keyboard with the twenty-six letters of the alphabet.
2. A plug board which contains 26 dual sockets for the plugs . It can contain from zero to thirteen dual-wired cables.

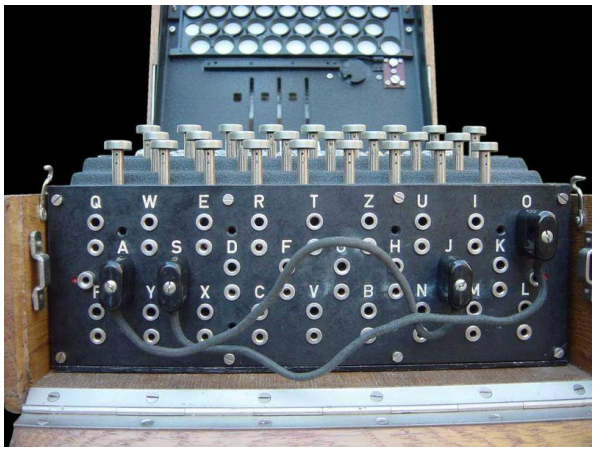

Fig. 2.1 Plug board and keys.

3. The rotors and the order they are set in. The rotors have twenty-six contact points that carry the electrical signal.

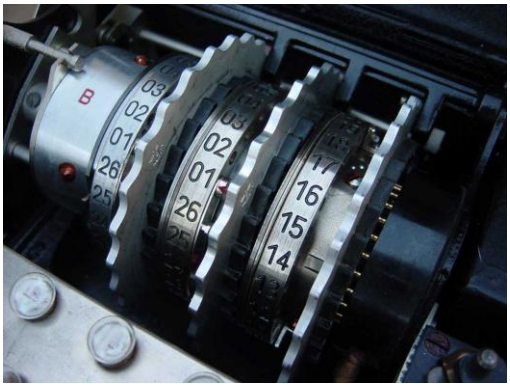

Fig. 2.2: Rotors and reflector on far left.

Fig. 2.3: Right side of rotor with 26 contact points.

Every time a key is pressed on the keyboard, the rotor rotates one position. Every time it makes a full revolution, the second rotor rotates once. Every time the second rotor makes a revolution, the third one will move once. There are about ten different models, or types, of rotors that can be used in any order. Different Enigma Machines have different rotor capacities. We are modeling a three rotor enigma.

4. The movable rings with notches on the sides of the rotors. These rings can be rotated to any of twenty-six positions. These controlled the movement of the rotor to the left of it.

Because we are using a three rotor Enigma Machine we only have two rings. This is because to the left of the last rotor is the reflector.

Fig. 2.4: Left side of rotor. The gray notched ring can be seen.

5. The reflector. It has twenty-six contact points that touch the contact points of the rotor to the right of it. Inside the reflector, there are thirteen pairs of wires that connect two contact points together. The wires will carry the electric signal that is reflected back through the rotors to the plug board. The reflector doesn't rotate like the rotors, but it can be adjusted before the operator starts typing.
6. The twenty-six lights are for every letter in the alphabet.

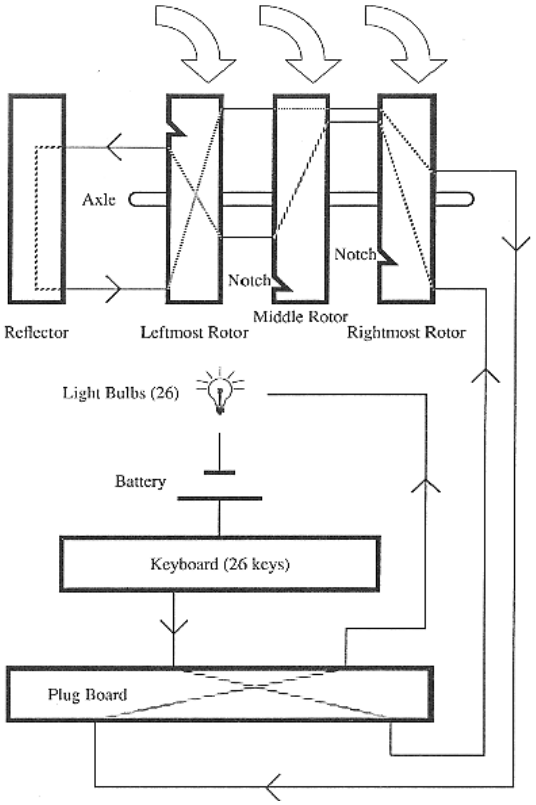

Fig. 2.5

Let's say someone is going to cipher a message. [Use Figure 2.5 as a visual aide]. First he positions the rotors by rotating them to a set. For example, he sets them all at "1". Then he can plug any or all of the thirteen cables in the sockets. Now he can begin to cipher a message. The first word begins with "A", so he types "A", which sends an electrical signal from the keyboard to the plug board. The cable that is connected to "A" on one end is connected to "J" on the other end, so the signal will go to "J". The "J" does not mean that the ciphered message will come out as "J", it is only a label on the plug board. From the cable in the plug board, the signal goes through the rotors and into the reflector. The reflector sends the signal back through the rotors to the plug board. The signal returns through different cables than before the first time, which lights up a letter; let's say "X". This is the ciphered letter. Something to keep in mind is that if the operator keeps pressing

"A", it will never come out as the same letter because the rotors rotate every time a key is pressed. Now that we have explained the Enigma Machine's mechanical operation, we will explain the math which determines the number of different cipher combinations that are possible. This also has six steps.

The first variable is the number of plug board combinations. There are 26 sets of sockets on the plug board and 26 plugs paired on thirteen cables. With the choice of p cables ($0 \leq p \leq 13$) there are $\binom{26}{2p}$ different combinations. Now we determine how many ways those p cables can be positioned on the plug board with the $2p$ selected sockets. When one end of the cable is inserted, the other end has $2p$-1 sockets available. The next cable has $2p$-3 available. The third has $2p$-5 and so on. This continues until there is no room left. Knowing this, the number of plug board combinations is: $\binom{26}{2p}$x(2p-1)x(2p-3)x(2p-5)x…x 1 = $\frac{26!}{(26-2p)!\times p!\times 2^p}$. Now the number of plugs used must be factored in. The number of combinations for p is mutually exclusive and the maximum number of plugs that can be used is 13. When these are taken into account the equation turn into: $\sum_{p=0}^{13} \frac{26!}{(26-2p)!\times p!\times 2^p}$ = 532,985,208,200,576.

The second part of the equation is the three rotors. The 26 contact points of the rotors yield 26! possible rotors models. Each rotor can be any one of those 26!. The middle rotor can be any one of 26!-1 rotors. The third could be any one of 26!-2. Given these facts, the possible number of rotor combinations is 26!x(26!-1)x(26!-2) = $6.5\text{x}10^{79}$. This huge number made it impossible for the Germans to create this many different models of rotors. The

Germans only made a total of ten different types of rotors during the war. If the Allies had no idea how many rotors the Germans had or how those rotors were wired, they would have had to use this number to consider every possible combination that could have been manufactured.

Thirdly, we must determine the number of possible primary rotor settings. There are 26 contact points and three rotors, therefore, the equation is $26^3 = 17{,}576$.

The fourth variable is the moveable ring on the two rotors. Those can be set to any of the 26 positions. The first ring turns the second rotor after 26 keystrokes. The second ring moves the third rotor after 26^2 keystrokes. The third rotor does not have a ring because the reflector, which does not move, is to the left of it. The number of possible positions of the rings is $26^2 = 676$.

The fifth variable is the reflector, which has 26 contact points that are grouped into 13 pairs. The internal wiring could be changed by connecting two ends of a wire to two different contact points. When one end of a wire is connected, there are 25 remaining contact points. When both ends are connected, it leaves 24 points for the next wire and so on. It is similar to the possibilities for the rotors. While the number of remaining contact points decreases, the huge number of possibilities increases. The number of different combinations in the reflector is: $\frac{26!}{13! \times 2^{13}} = 7.9 \text{x} 10^{12}$. Again, as with the rotors, the Germans could not have produced this many reflectors, but the Allies still had to consider this number to try to break the code.

Now that we have all of the mathematical information, we can determine the number of possible combinations. The final equation is:

$$\left(\sum_{p=0}^{13}\frac{26!}{(26-2p)!\times p!\times 2^{p}}\right)\times\left[\ 26!\quad 26!-1\quad 26!-2\ \right]\times\ 26^{3}\ \times\ 26^{2}\ \times\frac{26!}{13!\times 2^{13}}=$$

3,283,883,513,796,974,198,700,882,6,882,752,878,37955,261,095,623,685,444,055,315,2,006,433,615,627,409,666,33,182,371,154,802,769,920,000,000,000

or about **3x10^114**.

In an effort to put this huge number in perspective, there are only about 1x10^80 atoms in the observable universe. This is why it would have been nearly impossible for the Allies to break the code had it not been for the help of defectors and spies that gave them key information. Because they did have some information, by the end of the war, the Allied cryptanalysts only had to work with 107,458,687,327,250,619,360,000, or about $1x10^{23}$, which is about one hundred thousand billion billion.

3. Computer Model

3.1 Objectives

The goal behind the computer program is to replicate the encoding system used by the Enigma Machine. This replication is accomplished using a program written in Java. In order to replicate the actual Enigma code, the components (namely the rotors) of the virtual encoder need to exactly match the ones used in the real machine. Although each rotor had its letters arranged in a specific way, one can also come up with their own way to arrange the letters on the rotors. Additionally there can be many more rotors on this virtual machine than on the real ones.

3.2 How the Program Works

The core of the program makes use of multiple 'class' commands, which produces multiple variables that represent the rotors, as well as the reflector and the character map. Each of these rotor variables consists of an array that is 2 by 26 units (26 units for 26 letters of the alphabet, and 2 units for 2 sides of each rotor.) The letters on each side of the rotor are linked to each other (the command that does this is called SetWiring.) After these dimensions are established, there is a series of commands that govern how the rotors advance.

The actual letters are defined by creating a character map. Like the real machine, the electronic version uses only the 26 upper case letters (no punctuation, or numbers.) Each of the letters is defined as a number between 0 and 25 (0 is A, 1 is B, etc.) It should be noted that the character map could easily be modified to include upper and lower case letters as well as various symbols (!@#$%^&*_+-=). Finally, the reflector is defined as an array similar to those used for the rotors. Unlike the rotors, this array is 1 unit by 26, and does not advance (because the reflector has one "side" and is stationary.)

The code operates by reading these three classes into the main java package. While doing this, the program also imports various tests which determine if any limitations are violated (such as use of a character that is not in the character map.) Next, a simple command is given that capitalizes all the letters in the message, and substitute the spaces with "XZ." The program then reads the message to be encoded. The first thing that happens to the message is it is formatted from plaintext to conditioned plaintext, as can be seen in figure 0.1

Hello I am a computer nerd.
To
HELLOXZIXZAMXZAXZCOMPUTERXZNERDXZ

Fig. 3.1

After this a specific number of rotors is created (anywhere from three, to several dozen), and the letter configuration / starting position are defined. The code for the first rotor is shown below:

```
Rotor[] rotor = new Rotor[3];
      try {
         rotor[0] = new Rotor(
            new int[] {19,  0,  6,  1, 15,  2, 18,  3, 16,  4, 20,  5, 21,
                    13, 25,  7, 24,  8, 23,  9, 22, 11, 17, 10, 14, 12},
21);
```

Fig. 3.2

In this string of code, Rotor[3] tells the program to create three rotors and the numbering {19, 0, 6…12} is where the letters are configured. The 21 just before the last parenthesis closes is the location of the "notch," which tells the next rotor when to turn. Next the reflector wiring is set up using the same numbering technique for the rotors. The difference is that the numbers are in pairs (because the reflector was wired in pairs.) The starting positions of the rotors are then assigned, and then the program prints out the original message and the version configured for encoding.

The program then takes the message and cycles it through the rotor and reflector arrays, and prints out the resulting encrypted message. For the "Hello I am a computer nerd" message the code prints out DLTBBSLXYEJQYIWIMJRKAYYLKOUUIGO (please, don't bother trying to pronounce it.) Although the work is technically done, the program then resets the rotors and sends this jumble through in the opposite direction, again like the real machine. This decodes the message and proves the program writes out something it can then decipher. The whole process is shown below for one letter.

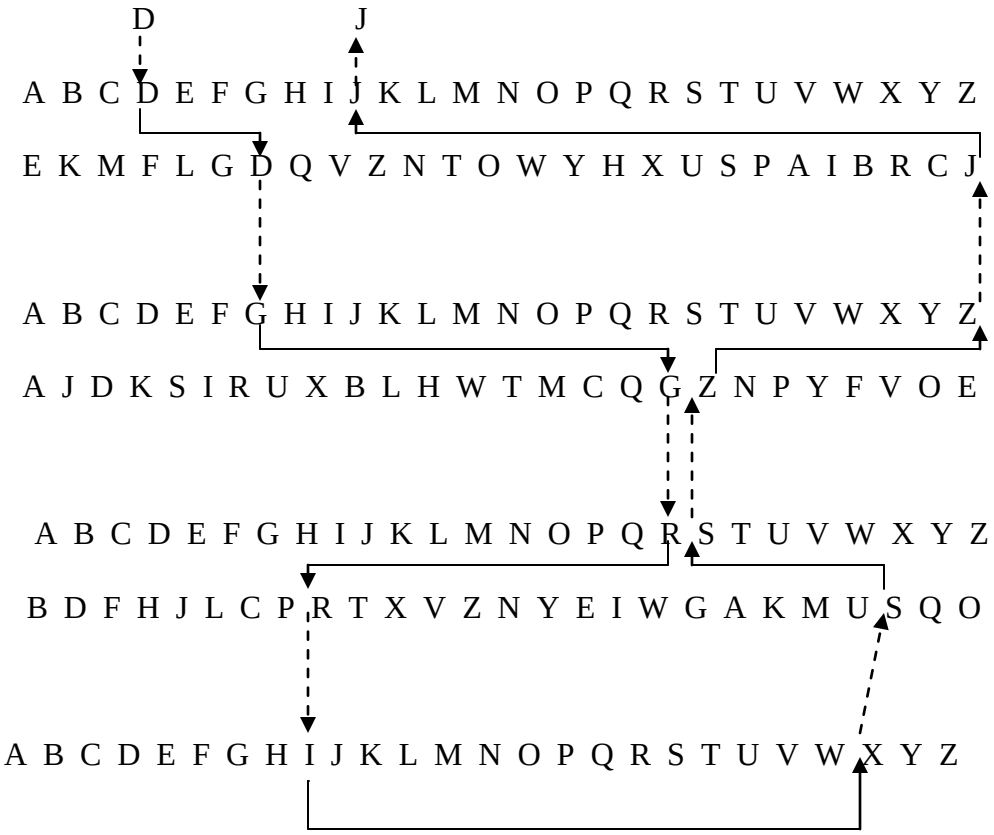

Fig. 3.3

4. Conclusions

It is entirely possible to accurately replicate the Enigma machine's operation on a computer. A big advantage of using a computerized version of the Enigma machine is that it

only takes one person to operate, instead of three. Furthermore, a computer allows for one to make a great deal of modifications to the Enigma code. It is possible to use so many characters and rotors that the statistical probability of deciphering a message gets *extremely* low. This program has enormous potential for future research and testing.

5. Acknowledgements

- Miller, A. Ray, The Cryptographic Mathematics of Enigma, Fort Meade, MD: National Security Agency
- Mowry, David P., German Cipher Machines of World War II, Fort Meade, MD: NSA Center for Cryptologic History
- Bennett, Nick
- Lorenzen, Paul