Enigma explanation

This explanation will focus on what happens inside of the enigma instead of how it happens, I will assume you have a basic understanding of how the enigma works.

Table of Contents

Single Letter Encyption

Stage 1: The Plugboard

When a letter is pressed on the keybaord, an electrical current flows through the plugboard, or "Steckbrett" in German. Here, two letters can be switched and the remaining part of the encryption will be performed with a different letter. Note that only two letters at a time can be added to the plugboard. Let's say for example that we connect the plugs for corresponding to the letters 'A' and 'U', if we now press the key 'A' on the keyboard, the remaining part of the ecnryption will use a 'U', instead of an 'A'. Note that 'U' will also become 'A', so a situation where 'A' becomes 'B', but 'B' becomes something different, is impossible. After we (potentially) swaped two letters, the electircal current continues to the next stage.

Stage 2: The ETW

ETW stands for "Eintrittswalze", which is German and means something like "entry roll". This is another chance for our letter to change before the actuall encryption itsell. Besides the name "Walze", this does not rotate. However, we can change the Grundstellung and the Ringstellung, we will talk about theese later tough. Every Walze has a wiring table, this basicly tells you what letter becomes what after passing through the rotor. You can find most known wiring tables on wikipedia, along with more details about the physical design of rotors in the enigma. The ETW is, in most cases just a duplicate of the alphabet, unless the Ringstellung or Grundstellung is changed, a letter stays itself.

Stage 3: The Walzen

This is where most of the acutall encryption happens. In a typical enigma, there are 5 rotors, 3 of thoose actually turn, so it's normaly refered to as an enigma with 3 rotors. The M4 enigma had an aditional rotor, but that additional rotor also didn't turn. The current runs through the rotors 2 times, right now we will focus on the first run. At this point it should be noted that the rotors turn every time a key is pressed, the rotors turn before the letter is acutally encrypted.

Every wiring table represents this relationship:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
EKMFLGDQVZNTOWYHXUSPAIBRCJ
```

So, when a signal enters a rotor, we have to look at the top row and find the corresponding letter. Then we look for what letter that encrypts to and repeat the process for every rotor. If we take the letter 'H' as an example, in this case (This wiring table is for the rotor I), and 'H' corresponds to a 'Q'. Now we go on to the next rotor and repeat the process.

Stage 4: The UKW

UKW stands for "Umkehrwalze", buts it's more commonly refered to as the "Reflector". As the name suggest, this reflects the singal and is very easy to understand. A UKW also has a wiring table, but this one is a little bit more special. When before we essentailly had a monoalphabetic substitution, now it's similar to the plugboard, when letter x corresponds to letter y, letter y has to correspond to letter x. A wiring table for a UKW could look like this:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
YRUHQSLDPXNGOKMIEBFZCWVJAT
```

Notice the relationships here:

```
A→Y Y→A
B→R R→B
L→G G→L
```

So if, after all 3 rotors, we had the letter 'Y', we would now reflect the letter 'A'

Stage 5: The Walzen, again

As the name "Reflector" suggests, the signal now goes backwards through the Walzen, back to the plugboard. There is one minor difference though, we now have to look at the bottom row and find the corresponding letter in the top row. The singal also goes through the ETW, again.

Stage 6: The Plugboard, again

We now go through the plugboard again, the concept is exactly the same as the first time.

Stage 7: Let there be light

The signal obviously doesn't go back to the keyboard, instead, every letter has a light bulb assotioated with it, the light bulb corresponding to the encrypted letter will now light up.

Encryption of string

Encrypting a string is basicly just encrypting a lot of letters, in between we must rotate the Walzen. Every Walze has one or multiple notch postions, this means, when they reach that position, they will turn the next rotor with them. Here, an anomaly called "double stepping" occours. Wheater this behaviour was intendet or not is not clear, but it basicly makes the middle rotor step twice. Here is a simple explanation in the form of pseudo-code:

```
if (The first rotor has rotated to the notch position of that rotor):
    rotate the second rotor
if (The second rotor has rotated to the notch position of that rotor):
    rotate the second rotor
    rotate the third rotor
In any case rotate the first rotor
```

To easily simulate the rotation of a rotor programaticly, we just keep track of the rotations in form of a number, every time the rotor rotates, this number increases by one. Now, to find what letter would have come out of the encryption, had we actually rotated the rotor, do theese three things:

1. Add the number of rotations to your current letter, A=0, B=1; So if we, for example, want to pass the letter 'A' through the rotor, and the rotation of that rotor is 3, we would add 0 (A) to 3, which is 3, 3 corresponds to a 'D', so the new letter is D.
2. Do what is described in Stage 3
3. Subtract the number of rotations from your current letter to get the final result, our rotation is still 3, our current letter is a 'D'. Let's say a 'D' corresponds to a 'J' on our wiring table, we need to subract 3 (Number of rotations) from 9 (J), we get 6, this corresponds to a 'G'. The rotor has changed our input from the letter A to a G.

Grund- & Ringstellung

This is both set before the actuall encryption. This is what you agree upon with the person you want to send the message to before any acutall encryption happens, along with the plugboard pairs. Both the Grund- as well as the Ringstellung change the second row of our wiring table. Every wiring table you will find on the wikipedia site has a default Ring- & Grundstellung of "A". When changing the Grundstellung we essentialy just rotate the rotor one time for every letter that the desired Grundstellung letter is away from "A". For example, if we wanted Grundstellung F, we would have to rotate our rotor 5 times. Changing the Ringstellung is a lot more complicated.

Ringstellung:

On every rotor, there originally was a dot. We need to know the position of this dot. It's always where the ring setting letter is (So, if the ring setting is A, it's where the A is). Since all the wiring tables on Wikipedia assume a ring setting of A, we can find the dot position by finding the position of the "A", in case of rotor I, it's the 20. letter. We need to remember this position for later.

The next step is to shift all the letters up the alphabet by one for each letter that we go up in the alphabet. So, if we want ring setting C, we go up by 2 in the alphabet (Start out at A, B = 1; C = 2). Shifting up the alphabet means, an A becomes a B, a B a C, and so on. Each time we shift up the alphabet, we need to add one to the dot position we remembered. We do this addition mod 26 (Or what ever the lenght of your alphabet is).

The last step is to rotate the new wiring, until our ring setting letter is at the postion of our dot.

Example:

Our original wiring is "EKMFLGDQVZNTOWYHXUSPAIBRCJ" We want to get it to the Ringstellung "C".

First, we get the dot position, so we look for the A, which is at position 20. We remember this.

Now we shift all the letters up the alphabet by one for each letter that we go up the alphabet from A to get to our Ringstellung, and add one to our dot position:

```
A → B:
EKMFLGDQVZNTOWYHXUSPAIBRCJ
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
FLNGMHERWAOUPXZIYVTQBJCSDK

dot position + 1 = 21


B → C:
EKMFLGDQVZNTOWYHXUSPAIBRCJ
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
GMOHNIFSXBPVQYAJZWURCKDTEL

dot position + 1 = 22
```

Our new wiring look like this: "GMOHNIFSXBPVQYAJZWURCKDTEL" and our dot position is 22, our ring setting is C, so we want the letter C to be at position 22 in the wiring.

```
Rotation 1:

GMOHNIFSXBPVQYAJZWURCKDTEL
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
LGMOHNIFSXBPVQYAJZWURCKDTE
```

```
Rotation 2:

LGMOHNIFSXBPVQYAJZWURCKDTE
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
ELGMOHNIFSXBPVQYAJZWURCKDT
```

C is now at position 22, we have completed all the steps. The final wiring table looks like this:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
ELGMOHNIFSXBPVQYAJZWURCKDT
```

Here are a few more examples.