CRYPTOGRAPHY USING JAVA

THE ENIGMA MACHINE

Name| Course Title | Date

Introduction

The Enigma cipher was a field cipher used by the Germans during World War II. The Enigma is one of the better known historical encryption machines, and it actually refers to a range of similar cipher machines. The first Enigma machine was invented by a German engineer named Arthur Scherbius at the end of the first world war.[1] It was used commercially from the early 1920s on, and was also adopted by the military and governmental services of a number of nations — most famously by Nazi Germany before and during World War II.[2] A variety of different models of Enigma were produced, but the German military model, the Wehrmacht Enigma, is the version most commonly discussed.

If you would like to encrypt some of your own Enigma messages, have a look at the javascript example.

The Algorithm

This section will talk about the Enigma I aka Wehrmacht Enigma, other variants are similar in operation. The 'key' for the enigma consists of several elements:

1. The rotors and their order
2. The rotor start positions
3. The ringstellung, or ring settings
4. Steckerverbindungen, or plug board settings

For information on the procedures used by the Germans during WW2 when sending Engima messages, including how the indicators were set, see this description.

The Rotors

Assume that our rotors are I,II,III moving from left to right, and we are trying to encipher the letter 'A'. We will assume for now that as the letter 'A' is enciphered each rotor is in its start position ('AAA'). Since our rotors are I,II,III moving from left to right, the character A will first go through rotor III. Each rotor applies a simple substitution operation. The substitution table for rotor III can be seen below.

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

BDFHJLCPRTXVZNYEIWGAKMUSQO
```

B is replaced with D, C is replaced with F etc. So after the letter 'A' goes through the rotor, it comes out as a 'B'. The letter 'B' is now input through rotor II, where it is replaced by 'J' etc. This is best depicted using a table (for a full description of the rotor wirings for each rotor, see this wikipedia page):

III	II	I	Reflector	inv(I)	inv(II)	inv(III)
A -> B	B -> J	J -> Z	Z -> T	T -> L	L -> K	K -> U

After the letter goes through rotors III,II,I it then hits the reflector and undergoes another simple substitution. After coming out of the reflector, the letter is sent back through the rotors in the reverse direction (this means the inverse substitution is applied). We can see from the table that after the enciphered letter comes back out rotor III at the end, we are left with the letter U. One important step that I have not yet mentioned is the fact that the rotors increment before each letter is enciphered. If the rotor start positions are 'FEQ', then they will first be incremented to 'FER' before the first letter is enciphered.

Incrementing The Rotors

A common mistake when implementing the enigma is assuming the rotors act as a standard odometer, there are however a few key differences. Each rotor has a notch which causes the rotor to its left to step. Rotor I causes the next rotor to step on transition from Q to R, rotor II on the transition E to F etc. Rotors I through V are used in the Wermacht enigma, later more rotors were added which had two notches.

I	II	III	IV	V	VI	VII	VIII
Q	E	V	J	Z	Z,M	Z,M	Z,M

There is one extra confounding detail, that of 'double stepping'. When a rotor steps, it also causes the rotor to its right to step. This is not noticed when the second rotor steps, since the first rotor steps every key press. However, when the 3rd (left most) rotor steps, it causes the second rotor to step also. This means the machines period is not 26x26x26, but only 26x25x26.

The Ringstellung

The Ringstellung (ring settings) are generally provided as a 3 letter string e.g. 'FAM' (or alternatively as numbers between 1 and 26, representing the letters). In the previous discussion I have assumed that each rotor's simple substitution cipher was fixed. The ringstellung provide the capability of shifting the substitution cipher as follows. With a ring setting of 'A' (or 1), rotor I's substitution looks like this:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

EKMFLGDQVZNTOWYHXUSPAIBRCJ
```

With a ring setting of 'B' (or 2), rotor I's substitution looks like this:

```
ZABCDEFGHIJKLMNOPQRSTUVWXY

EKMFLGDQVZNTOWYHXUSPAIBRCJ
```

The Steckerverbindungen

The steckerverbindungen (plugboard) is an added layer of security which consists of 13 wires which plug into into sockets on the front of the enigma machine. Each wire connects 2 letters e.g. P to O. These pairings are specified as part of the key material. When a letter is typed, before it goes into the first rotor, it undergoes the substitution according to the plugboard, then after the letter comes out it is put through the plugboard substitution again before being output. An example plugboard setting is as follows: PO ML IU KJ NH YT GB VF RE AC (This means P and O are swapped, M and L are swapped etc.).

If we use the example above where the letter 'A' was encrypted with rotors I, II and III with the start positions AAA, we had the letter A encrypted as a U. If we now take into account the plugboard, using the plugboard settings in the previous paragraph, the 'A' is first translated to a 'C' before encipherment. Encipherment continues as usual, this time the 'C' is output as a 'J'. This letter is then routed through the plugboard again to be substituted with 'K'. So now we have an 'A' being enciphered as a 'K' with the plugboard in use. The plugboard significantly increases the strength of the enigma cipher as a whole, more than adding another rotor could.