Enigma Machine

Michael Purcell

COMP2700

August 29, 2023

History

There are many models of the Enigma machine that were used throughout the mid twentieth century.

Various models of were used for both commercial and military applications by a wide variety of organisations.

All of these Enigma machines employed a set of *rotors* that were used to implement a polyalphabetic substitution cipher.

Most military versions of the Enigma machine also used a *plugboard* to significantly increase the size of the keyspace.

German Military Usage

Several versions of the Enigma machine were used by the German military forces during World War Two.

Notably, the Wermacht Enigma I was widely used by the German military throughout the 1930s. This model used a set of three interchangeable rotors, a plugboard, and a fixed reflector.

In 1938, the Germany army adopted two additional rotors for use in their existing Enigma machine, thereby increasing the size of the keyspace by a factor of ten.

Subsequent developments saw the introduction of additional rotors for use in existing three-rotor machines and the introduction of a four-rotor machine.

Polish Cipher Bureau

In 1932, Marian Rejewski, a mathematician at the Polish Cipher Bureau, successfully recovered message keys used to encrypt messages with an Enigma machine.

Between 1932 and 1939, the Polish Cipher Bureau developed mathematical techniques and novel mechanical devices to successfully cryptanalyse Enigma traffic.

During this time they developed the bomba kryptologiczna (cryptologic bomb) that could be used to recover Enigma keys in approximately two hours.

Shortly before the German invasion of Poland in 1939, the Polish Cipher Bureau shared their achievements with French and British military intelligence agencies.

The Government Code & Cipher School

GC&CS was the organisation that housed many of the allied forces code-breakers during World War Two.

GC&CS was housed at (the now famous) Bletchley Park.

Notable membners of GC&CS include Alan Turing, William Tutte, I.J. Goode, Gordon Welchman, and many others.

The team at GC&CS were able to continue the work of the Polish Cipher Bureau and successfully exploit Enigma traffic for all of World War Two.

Description

I/O Devices

Most Enigma machines were equipped with a keyboard input device and a glowboard output device.

To encrypt a message, a technician would use the keyboard to enter plaintext one character at a time.

Each time a key was pressed on the keyboard, a light on the glowboard would turn on, thereby indicating the corresponding ciphertext for that character.

Decryption worked similarly, with the technician entering ciphertext one character at a time and then reading off the corresponding plaintext from the glowboard.

Rotors

The heart of the Enigma machine is a set of three rotors, each of which can be understood as a hardwired implementation of a simple substitution cipher.

Each rotor has 26 inputs and 26 outputs, with a fixed mapping from inputs to outputs.

After each character is entered on the keyboard, one or more of the rotors will rotate forward one position.

The rightmost rotor always rotates. The other rotors behave a bit like an odometer, only rotating forward one step per complete revolution of the rotor to their right.

Reflector

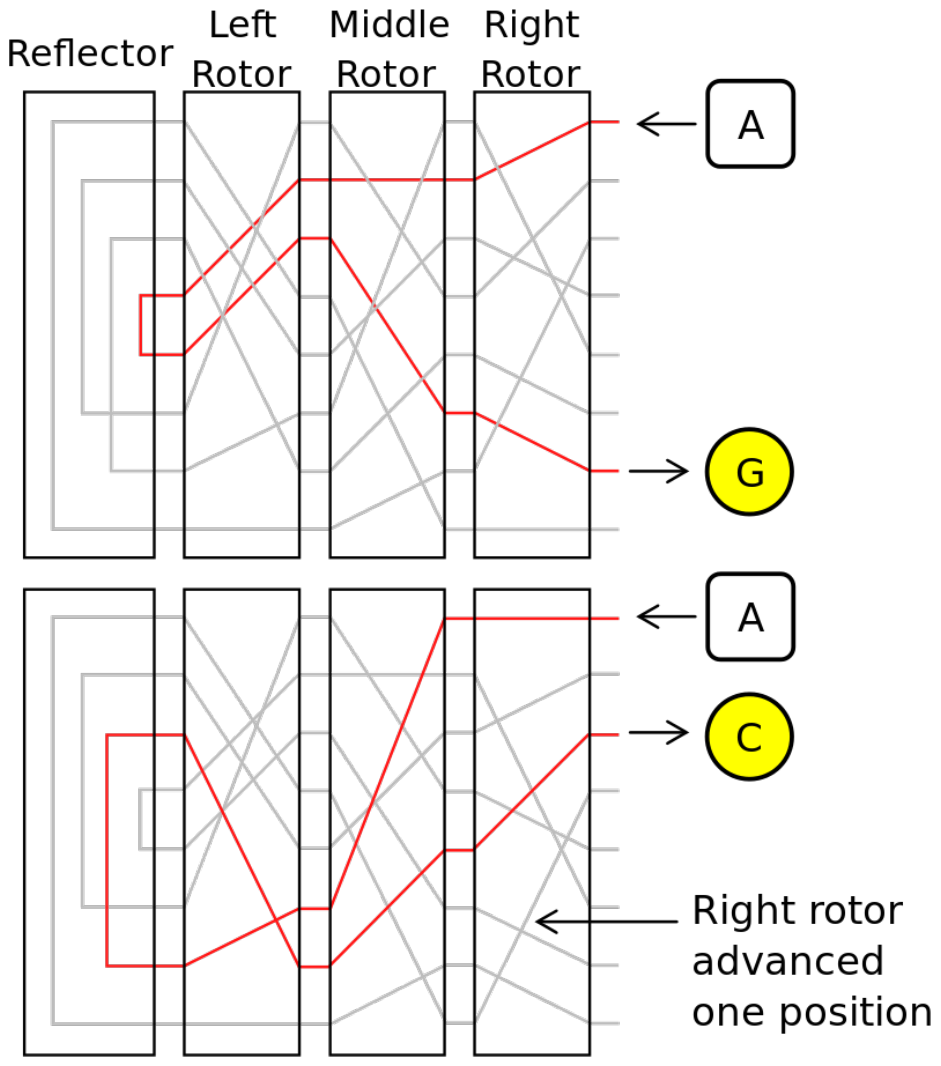
The reflector connects the leftmost rotor to itself.

After passing through the reflector, a signal travels back through the rotors via a different path than it took on the forward pass.

The reflector ensures that the settings that were used to encrypt a message can also be used to decrypt that message.

The reflector is a source of one of the biggest flaws in the design of the Enigma machine. That is, no character can ever be encrypted to itself!

Rotors and Reflector



Plugboard

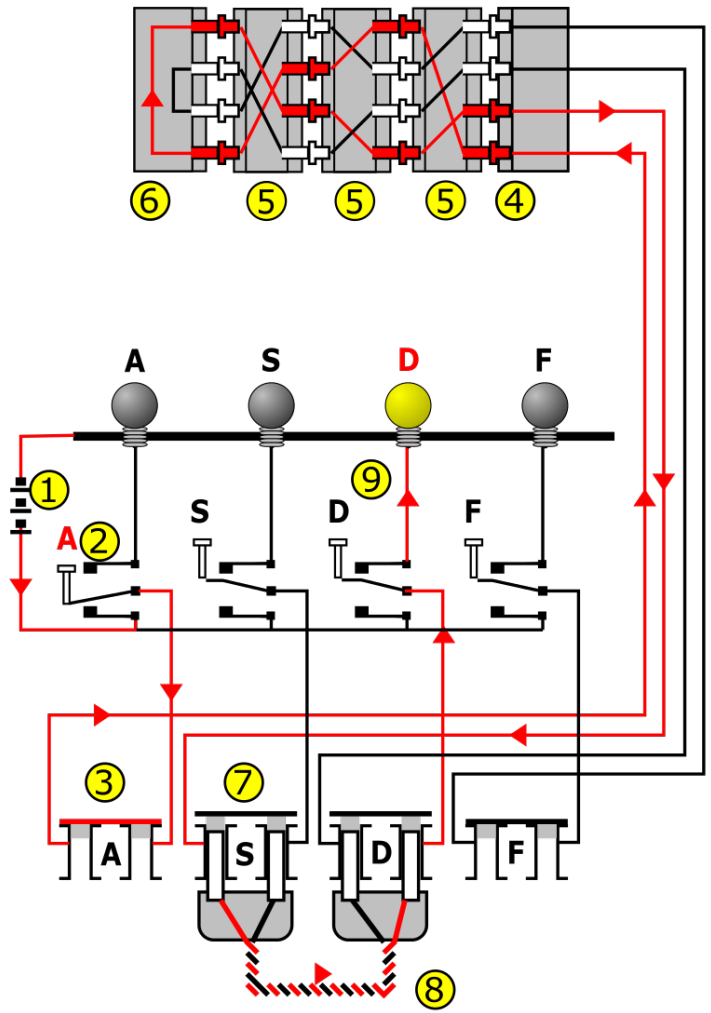
The plugboard is a set of plugs on the front of the enigma machine that can be connected by cables.

There is one plug for each letter.

If two letters are connected by a cable, then those letters are transposed before being sent to the rotors.

The plugboard is largely responsible for the huge size of the Enigma keyspace.

Encryption Schematic



Settings

There are a large number of machine settings, each of which corresponds to a different cryptographic key.

Some settings were generally changed relatively infrequently. These include the arrangement of the rotors and the placement of the cables in the plugboard.

Other settings were changed on a per message basis. These were generally limited to the initial position of the rotors.

Message Settings

Individual cipher clerks were responsible for choosing the initial position of the rotors for each message that they sent. To facilitate decryption, these settings would have to be communicated to the recipient of the each message.

This was done by using a default setting for the rotors. This was known as the Grundstellung was among the settings that was changed relatively infrequently.

The cipher clerk would choose choose an initial position for each rotor and encode that setting as a three-character code word. They would then use the Grundstellung to encrypt their chosen code word twice.

They would then reset the rotors to the positions indicated by their code word and encrypt the remainder of the message.

Cryptanalytic Properties

There are several properties that the allied code breakers used to break Enigma.

The first property is intrinsic to the design of Enigma. As described above, because of the way the reflector is used no character is ever encrypted to itself.

The second property, is really a property of the protocols that the German cipher clerks used to transmit messages. By encrypting redundant information in the code words used to indicate initial rotor positions, they inadvertently revealed information about the machines settings.

Cribs

Why is it bad that no character can be encrypted to itself?

Essentially, in this case the problem was that it allowed allied code breakers to identify *cribs* that they could use.

A crib is a snippet of plaintext that a cryptanalyst knows (or just assumes) corresponds to some stretch of ciphertext.

This property of Enigma allows us to efficiently identify stretches of ciphertext that may correspond to various common cribs that had been observed in previous messages.

Example

Index	0	1	2	3	4	5	6	7	8
Ciphertext	r	q	u	t	b	z	p	w	h
Plaintext		a	t	t	a	c	k		

Here we see that the crib “attack” cannot correspond to the snippet of ciphertext “qutbzp” because both have a ‘t’ in as their third character.

Loops

Suppose that we have identified a crib and a snippet of cipher text to which it corresponds.

How can we use that to attack Enigma?

One way, which is similar to what was done at Bletchley park using the British Bombes, involves identifying “loops”.

A loop in this context is a series of triples $\{(j_i, x_i, y_i)\}_{i=0}^{n}$ such that $E_j(x_i) = y_i$ for all i, $y_i = x_{i+1}$, and $y_{n+1} = x_0$.

Example

Consider the following ciphertext/crib pair.

Index	0	1	2	3	4	5	6	7	8	9
Ciphertext	R	E	W	K	V	P	A	Z	E	T
Plaintext	A	T	T	A	C	K	T	H	E	R

This yields the loop

$$(0, R, A) \rightarrow (6, A, T) \rightarrow (9, T, R)$$

Using Loops

To use a loop, we first observe that a loop in the ciphertext/plaintext characters will also be a loop if remove the plugboard from the equation.

For the preceding example, let ρ, α, τ be the stecker partners of R, A, T respectively.

Then we have the loop

$$(0, \rho, \alpha) \rightarrow (6, \alpha, \tau) \rightarrow (9, \tau, \rho)$$

Notice that this loop does not depend on the plugboard.

Fishing for Contradictions

To use this loop, we will test every setting for the rotors and every stecker pair for the first ciphertext character in our loop.

Let R be the function corresponding to our guess for the rotor settings and let ρ' be our guess for the stecker partner for R. Finally, let $R_{j_0}(\rho') = \alpha'$, $R_{j_1}(\alpha') = \tau'$, and $R_{j_2}(\tau') = \rho''$.

Observe that if our guess is correct, then we must have $\rho'' = \rho'$. So, if $\rho'' \neq \rho'$, then our guess must be wrong.

This process allows us to efficiently eliminate possible settings. We would finish the attack by exhaustively testing all of the settings that we failed to eliminate by examining all of the loops that our cribs exposed.