# Enigma

•••

Workshop

# Agenda

- How the enigma machine works
- How enigma was cracked
- Workshop

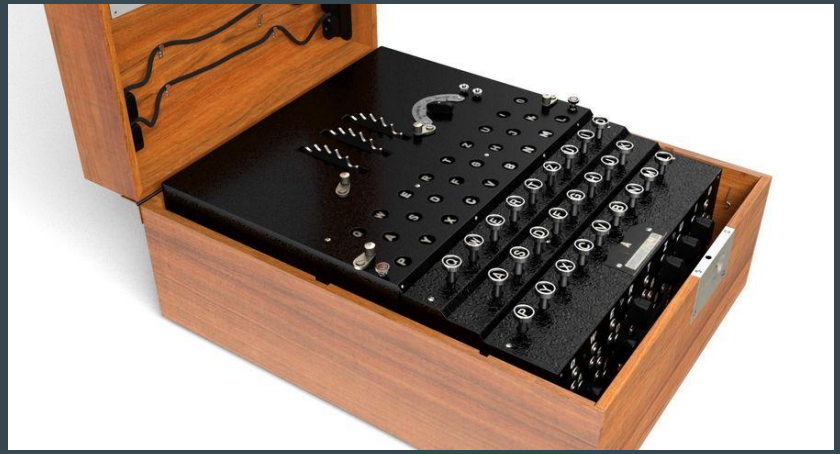
# What is the Enigma machine?

- The Enigma machine is a cipher device developed by a german engineer
- Early models where used commercially, but it was later used by the Nazis during WW2 to encrypt messages
- Entering a message into enigma will output an encoded message which can only be decrypted by knowing which settings where used when encrypting the message

# How does it work?

The main components of Enigma are:

1. Keyboard
2. Lightboard
3. Plugboard
4. Rotors
5. Reflector



# Keyboard

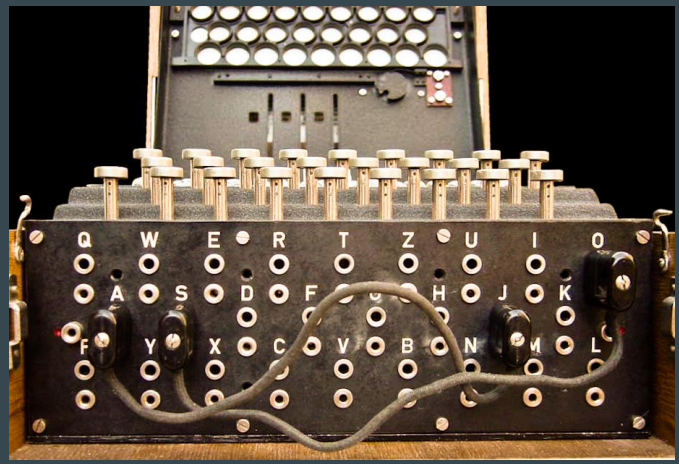
- Standard qwerty keyboard
- When a key is pressed, a current runs through the machine and lights one of the lamps
- The lamp indicates the encrypted letter

# Lampboard

- The operator would press a key and write down the encrypted or decrypted letter

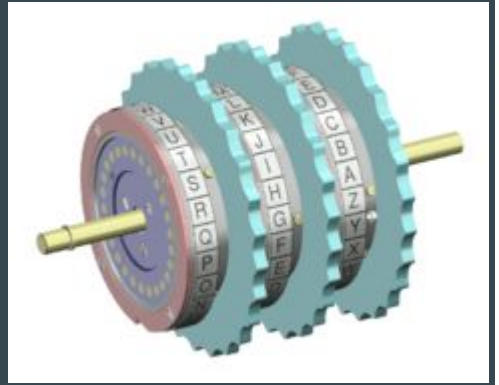
# Plugboard

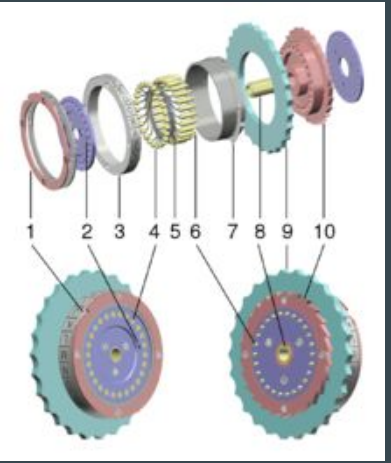
- The plugboard is a simple substitution cipher
- Each machine came with 10 cables
- Connecting A to J would simply substitute A for J



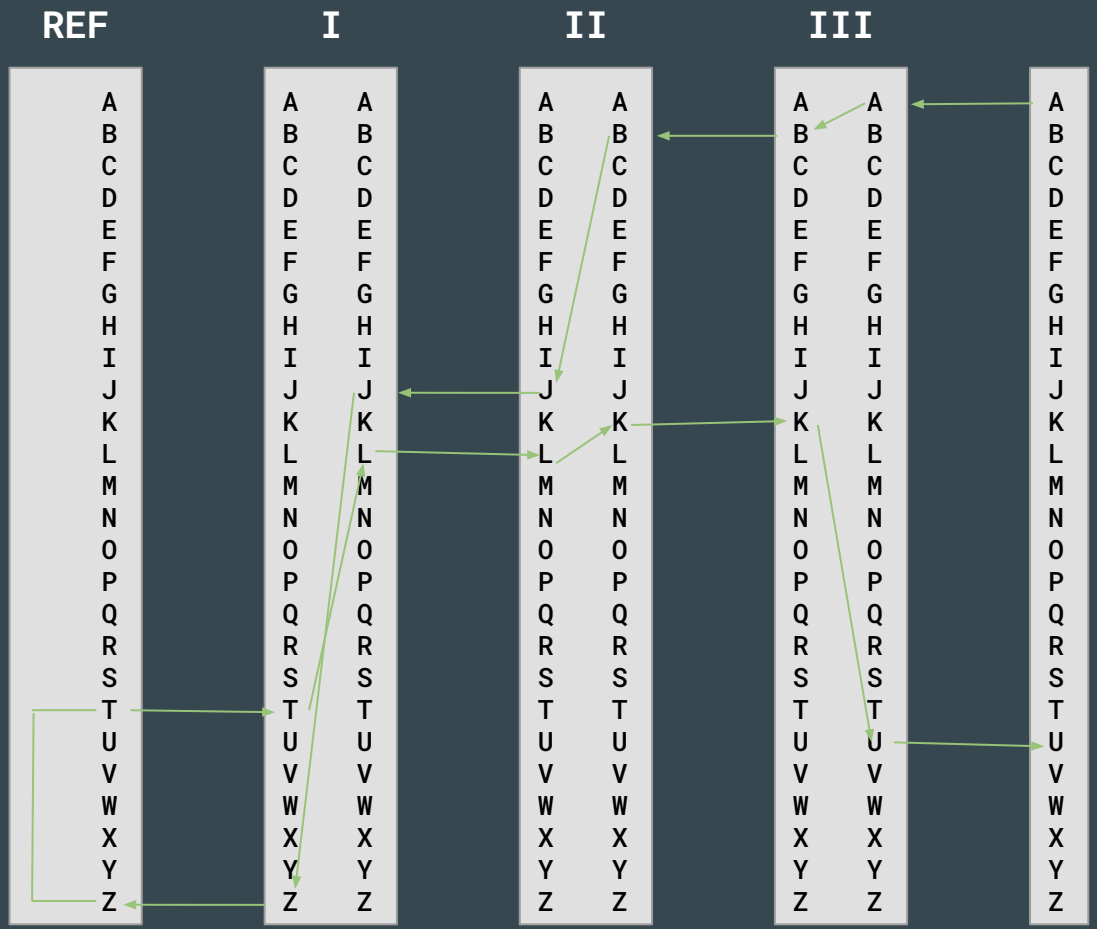
# Rotors

- Enigma has three slots for rotors
- Each rotor has a specific cipher
- There are multiple rotors to choose from
- The order matters
- When a key is pressed the first rotor is turned
- When the first rotor reaches its “notch position” it will turn the next rotor on the next keypress

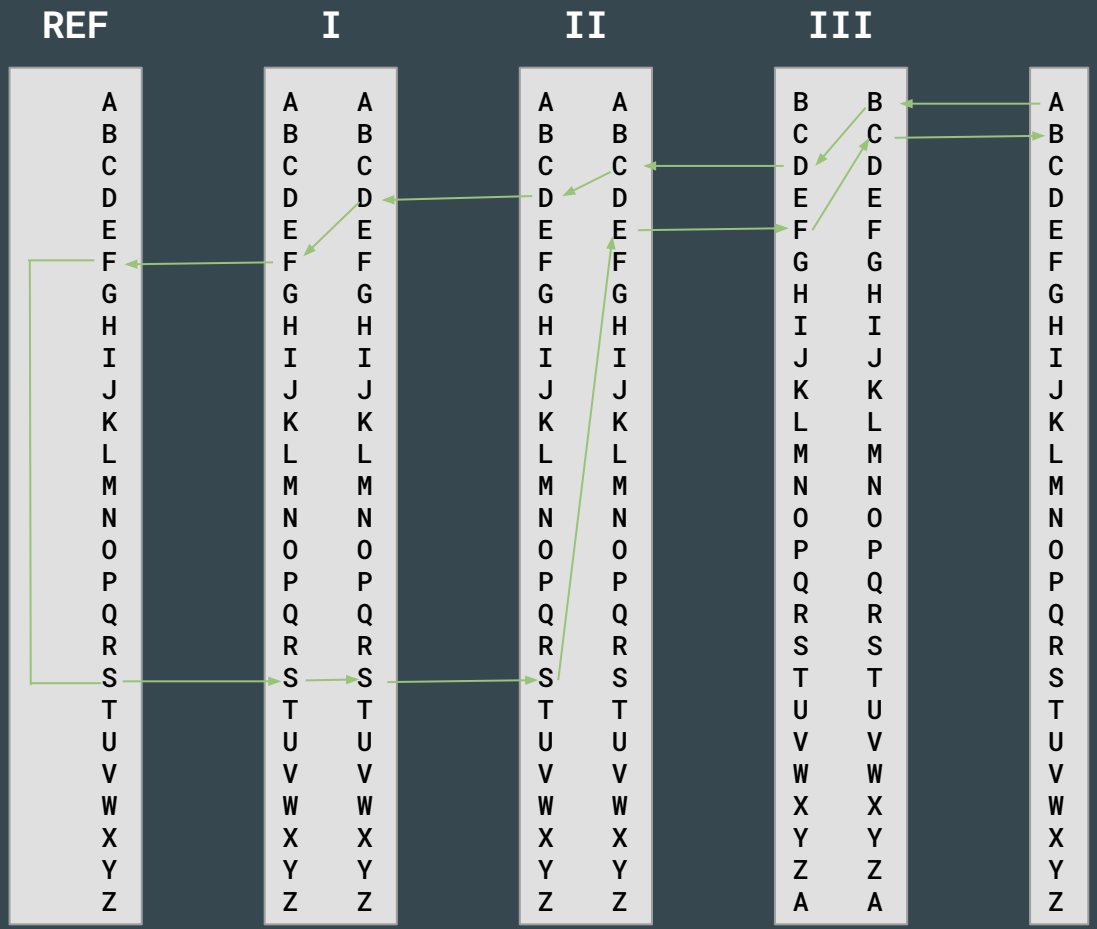




# Example:



# Example:

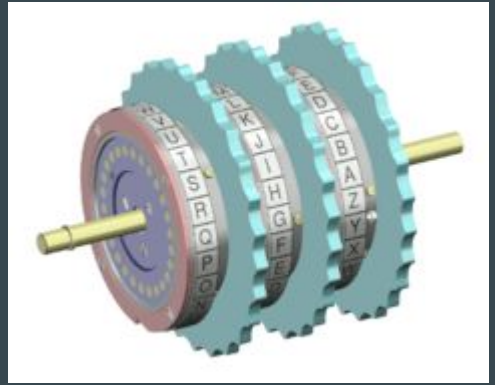

REF
I
II
III
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

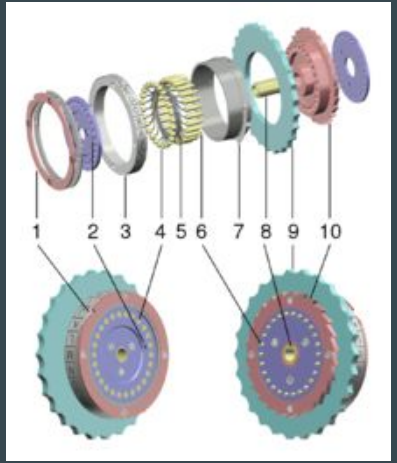
# Notch position

- When a rotor is in its notch position, the next time it rotates, it will also rotate the next rotor
- Video

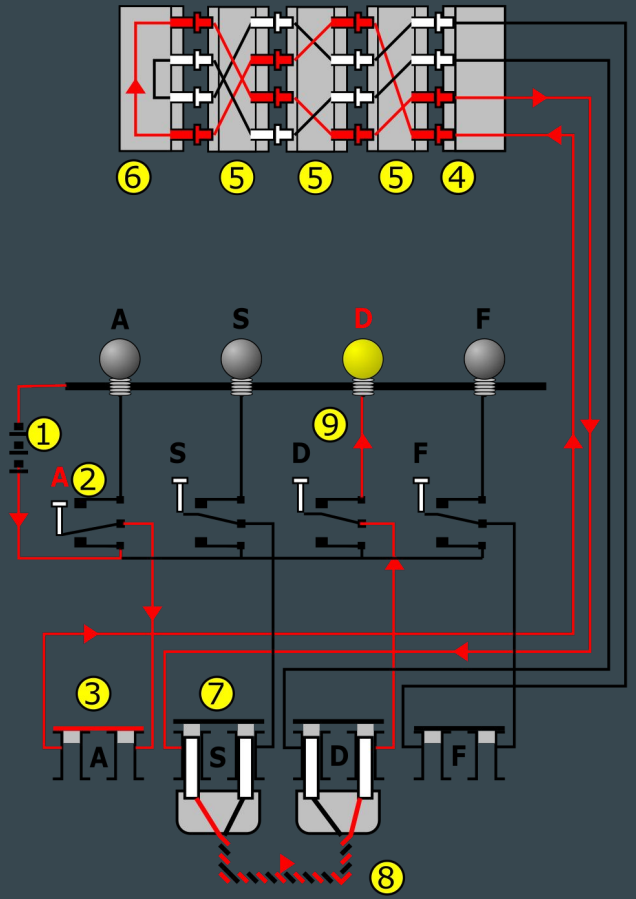
# Reflector

- When the signal has passed through the rotors, it is first encrypted by the reflector and then reflected back through the rotors





# Signal chain



# Enigma settings

- Rotors used and in what order (e.g. I IV III)
- Starting position of rotors
- Rotor ring position
- Plugboard settings (e.g. AS FG KJ LE)
- Reflector (A, B or C)

# How many settings are there?

Rotors selection: $$\frac{5!}{(5-3)!} = 60$$

Rotor settings: $$26^3 = 17,576$$

Plugboard settings: $$\frac{26!}{(26-20)! \cdot 2^{10} \cdot 10!} = 150,738,274,937,250$$

Result: $$158,962,555,217,826,360,000$$

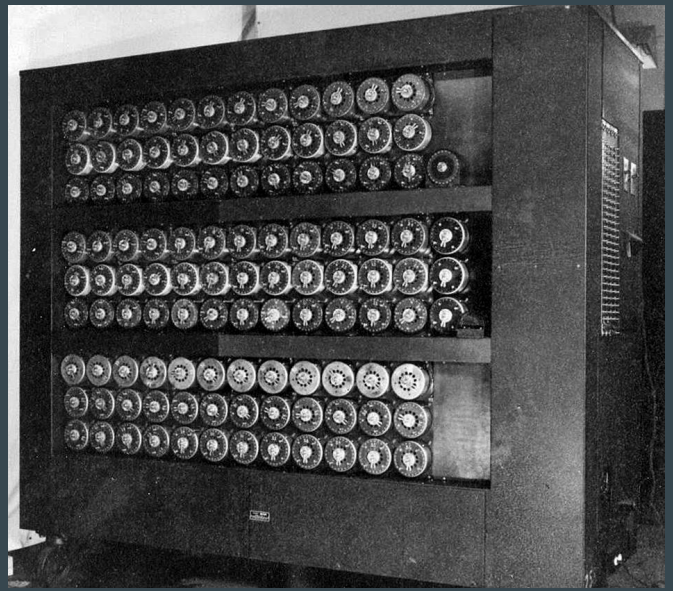
# Cracking Enigma

- The polish where the first to crack enigma using the “Bomba” machine
- Bomba only worked if three conditions were met
  - Message key was repeated in the encrypted message (*)
  - Rotors available was limited to three
  - A majority of the plugboard cables would be unused

(*) Germans would encrypt a message using e.g. ring settings AGK and include the setting twice  in the message. Then the would encrypt it again using the daily settings

# Cracking Enigma

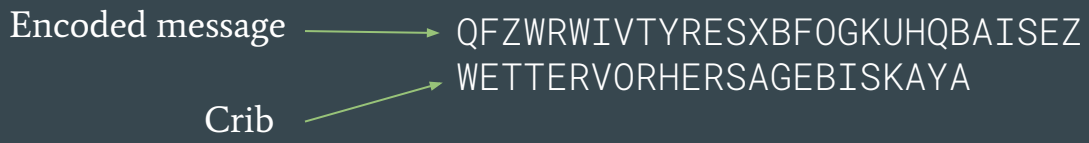
- Alan Turing design a machine called “Bombe” which didn’t have the same limitations as the polish “Bomba”
- Bombe performed a “Known-plaintext attack”
    - The phrase used for the attack is called a **Crib**



# The flaw in Enigma

Enigma emulator

# Known-plaintext attack



# Known-plaintext attack

QFZWRWIVTY**R**ESXBFOGKUHQBAISEZ
WETTERVO**R**HERSAGEBISKAYA

# Known-plaintext attack

QFZWRWIVTYRESXBFOGKUHQBAISEZ
WETTERVORHERSAGEBISKAYA

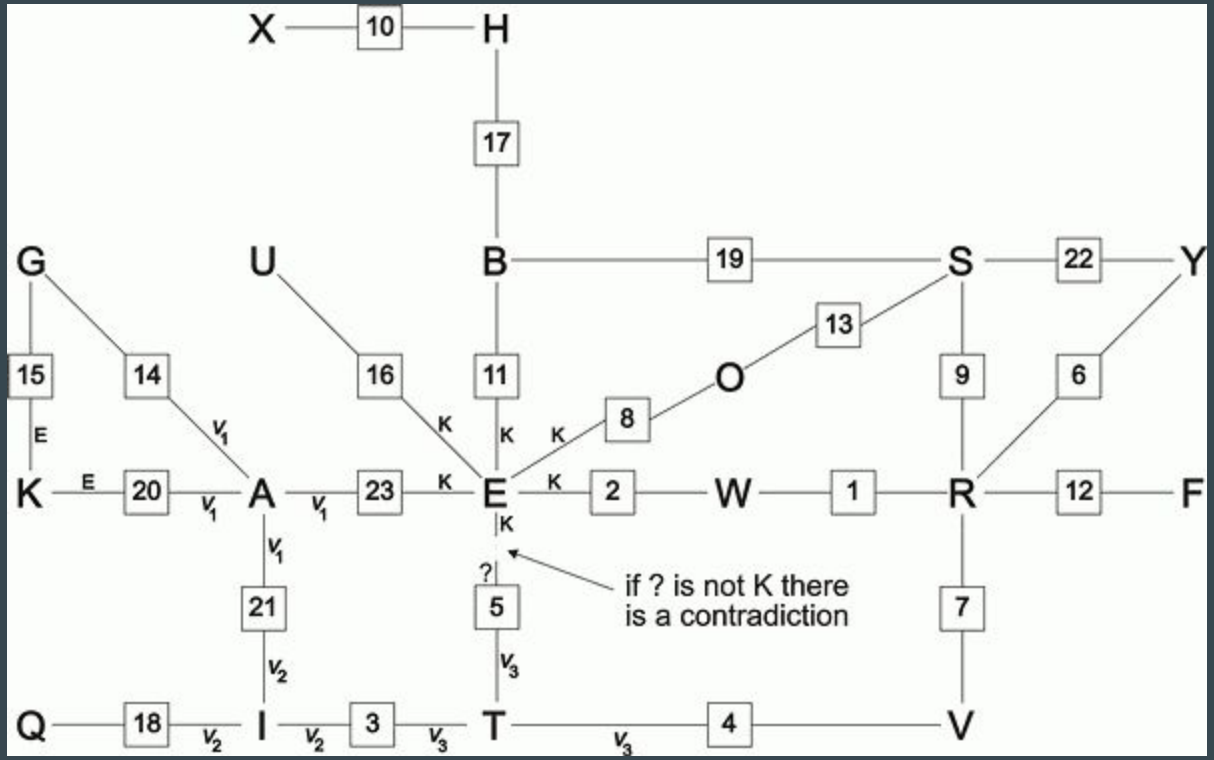
# Known-plaintext attack

QFZWRWIVTYRESXBFOGKUHQBAISEZ
WETTERVORHERSAGE

# Known-plaintext attack

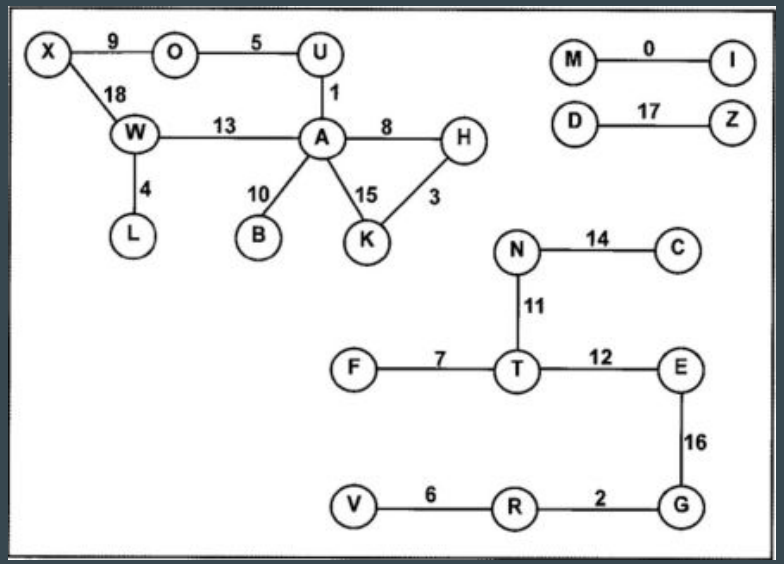
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | W | I | V | T | Y | R | E | S | X | B | F | O | G | K | U | H | Q | B | A | I | S | E |
| W | E | T | T | E | R | V | O | R | H | E | R | S | A | G | E | B | I | S | K | A | Y | A |

# Menu


X
10
H
17
G
U
B
19
S
22
Y
13
15
14
16
11
O
9
6
8
E
K
E
20
A
23
K
E
2
W
1
R
12
F
?
21
5
7
if ? is not K there
is a contradiction
Q
18
I
3
T
4
V

# Known-plaintext attack

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I | U | G | H | L | U | V | F | A | O | B | N | E | W | N | A | G | Z | W |
| M | A | R | K | W | O | R | T | H | X | A | T | T | A | C | K | E | D | X |




There are two loops, XOUAW and AHK. We create the mapping:

- A → α
- H → β
- K → γ

We make a quess for α. We know that enigma outputs a H on the 8th step. So we input α eight times. The last letter is our guess for β. We repeat this for for β to get our guess for γ. Lastly we close the loop by repeating again for γ. To satisfy the loop, the encrypted letter for γ, let's call it ά. If α is equal to ά our guess is potentially correct. First we must check that it satisfies all loops.

If we can satisfy all loops, this means that we've found some of the plugboard settings. If not, our rotor settings must be wrong.

# Workshop

Implement the enigma machine

# Data and tips

Test data here: repo

- Start simple.
  - Encode a single letter with three *rotors* without rotating the *rotors* and without the reflector
  - Add rotations
  - Add the reflector
  - Add the plugboard
- Note: The rotors rotate when the key is pressed, meaning that the letter is encrypted in rotated position, not the initial position

# Resources

Repo

Video: How did the Enigma Machine work?

Enigma emulator

Rotor wiring example

Cribs and menus