DTiF
Digital Technologies in focus
Initiative of and funded by the Australian Government Department of Education, Skills and Employment

acara AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY

CLASSROOM IDEAS YEARS 7–10

The Enigma machine – background

The Enigma machine was invented for Germany by Arthur Scherbius in 1918. It is a cypher machine: a way of changing the letters of a message so that it appears to be scrambled or random letters. Enigma's main purpose was to protect commercial, diplomatic and military communication. A military model (Figure 1) was employed extensively by Nazi Germany during World War II, in all branches of the German military.

Figure 1: Military Enigma
Source: commons.wikimedia.org/wiki/File:Enigma MachineLabeled.jpg

Functionality

Each time a letter is typed, it appears as a different letter. The choices are *not* random. They are decided by a series of rotors (Figure 2) which are set daily to a new starting position. Each key press turns the rotors to a new position.

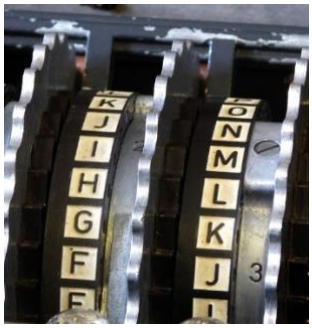

Figure 2: Enigma rotors
Source: commons.wikimedia.org/wiki/File:Enigma_rotors_with_alphabet_rings.jpg

Part mechanical and part electrical, Enigma has the appearance of an oversized typewriter. The first letter of a message was entered on the keyboard and a letter lit up showing what was replaced within the encrypted message. The human receiver at the other end followed the same process as the sender: however, they typed in the ciphertext and the letters which lit up were the decoded message.

Inside the box, the system is built around 3 physical rotors. Each takes in a letter and outputs it as a different one. That letter passes through all 3 rotors, bounces off a reflector at the end, and passes back through all 3 rotors in the other direction.

The Enigma message pathway is simplistic but remained one of the most secure ways for years for information to be sent. The machine function is further explained in figures 3 and 4. For example, a T might be pressed but the letter F would light up.

The power of the Enigma came from being simple for the operator to use but difficult to determine the encrypted letter for any input letter. The number of possible ways to jumble a message through an Enigma was nearly 159 quintillion.

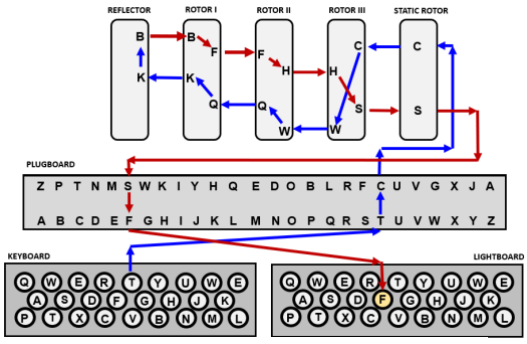

Figure 3: Letter pathway
Source: UNE Museum of Antiquities

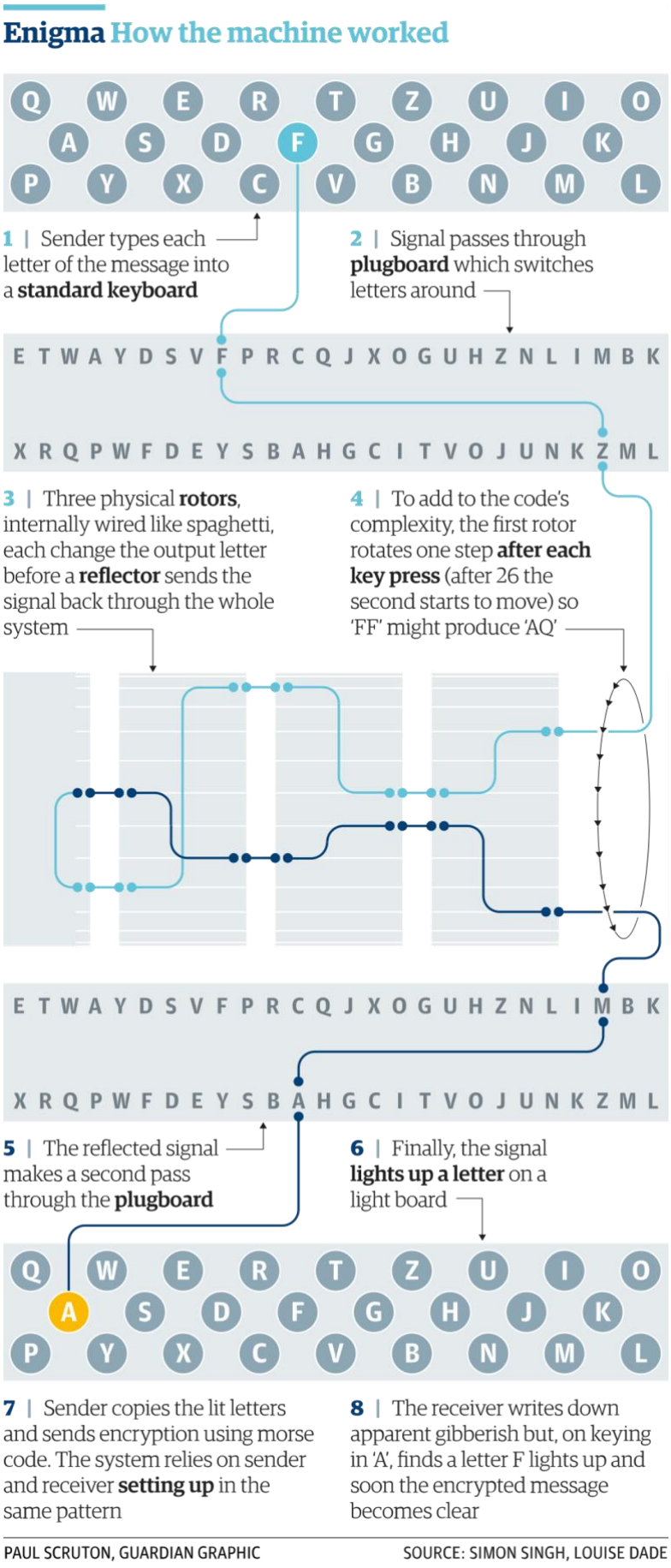

Figure 4: Letter pathway
Source: www.theguardian.com/technology/2014/nov/14/how-did-enigma-machine-work-imitation-game

Unplugged Enigma machine

Materials needed:

- Print these PDF templates on A4 paper (Tip: Do *not* select 'fit to page' or the dimensions of the Enigma machine will not be correct.)
- A tube (75mm in diameter and at least 225mm long; similar to the tube for a popular potato chip brand)
- Clear sticky tape
- A pair of scissors

Assembly:

- Cut each strip of paper along the black lines. You should end up with 5 strips titled: Rotor I, II, III, IV, V; 2 Reflector strips, B and C; and an Input / Output strip.
- You can start with the basic Enigma machine using 3 rotors, 1 reflector and the input/output. See the shaded strips in figure 5.
- Fasten these around the tube in this order, from left to right: Reflector B, Rotor I, Rotor II, Rotor III, Input / Output. See figure 6.

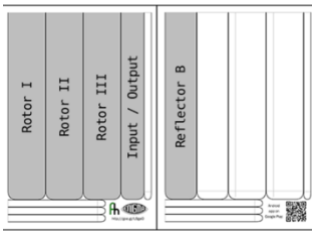

Figure 5: Basic rotor set up Source: wiki.franklinheath.co.uk/index.php/Enigma/Paper_Enigma

Set up:

1. Make sure the grey bars on the reflector and the Input / Output strips line up; this shows the start position of your Enigma machine and lets you track the turnover positions of the rotor.
2. You need to start by setting the 'key'. Turn the rotors so that the 3 letters of your key are in line with the grey bars, for example A, B, C.
3. For each letter in your message, turn the right-hand rotor one step, making sure that the other rotors and the Input / Output stay still. You must do this before you read the letter, even the first one.
4. Find the letter from your message on the Input /Output at the right-hand side, and trace the line through all 3 rotors, into the reflector, out again back through all 3 rotors and into the Input / Output. Write down the letter you end up on.

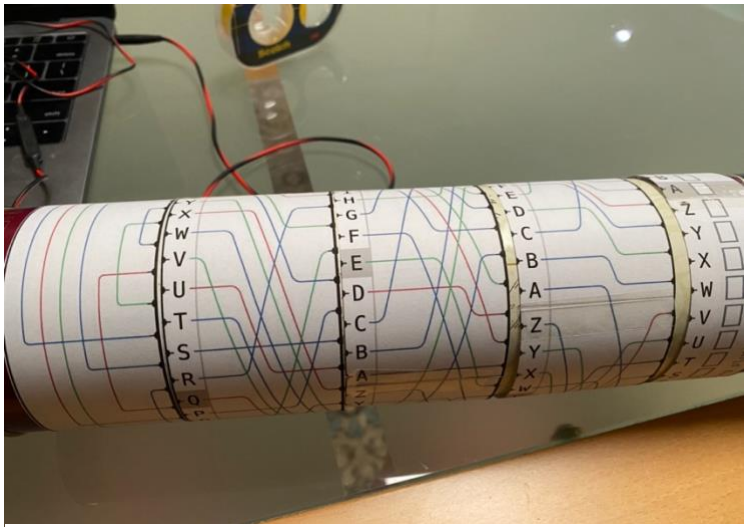

Figure 6: Finished Enigma machine Source: ACARA

This YouTube clip demonstrates the set up and functionality of the Enigma machine: www.youtube.com/watch?v=pZsuxZXN33g (8 min)

Analysing an interactive Enigma machine – using Scratch code

The goal of analysing the Enigma machine is to better understand the workings of a device that played an important role in the history of computing. It is also an excellent system to better understand some of the design decisions we make when creating a code representation of a problem.

The intention is to replicate some of the encryption mechanisms of the original Enigma. The basic idea is to start with a plaintext input (typed by the operator) and apply a rotating cipher to encrypt it, resulting in a ciphertext output:

PLAINTEXT ⟹ ENIGMA ⟹ CIPHERTEXT

1. Open Scratch online scratch.mit.edu/
2. Open the predesigned Scratch project to begin exploring what algorithms are required for each of the functioning parts of the Enigma machine scratch.mit.edu/projects/572099314/
3. Ask students:
   - What do you notice about the code blocks? See Figure 7.
   - Where could you make changes to enhance the project?
   - What would you need to change about the code blocks to have a 3-rotor Enigma machine?

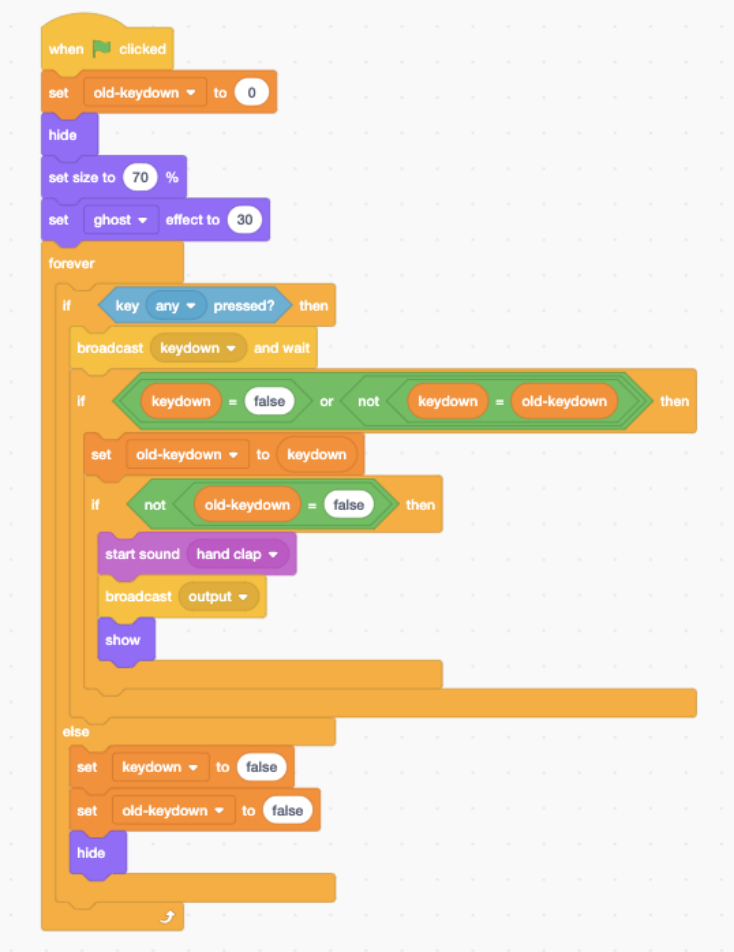

Figure 7: Explore the code blocks
Source: scratch.mit.edu/projects/572099314/

Analysing an interactive Enigma machine – using Python code

The Python program at this website trinket.io/python/d3ca641052?showInstructions=true allows you to encode and decode messages using the Enigma encryption.

You can apply your own Enigma settings by editing lines 3 to 9 of this code. See Figure 8.

```
# ----------------- Enigma Settings -----------------
rotors = ("I","II","III")
reflector = "UKW-B"
ringSettings ="ABC"
ringPositions = "DEF"
plugboard = "AT BS DE FM IR KN LZ OW PV XY"
# ---------------------------------------------------
```

Figure 8: Python code lines 3–9 for editing
Source: trinket.io/python/d3ca641052?showInstructions=true

- How could we simplify the rotor rotation code? What would happen if it was replaced with the code shown at Figure 9?

```
def rotate(self, offset=1):
    self.rotations = offset
    self.alphabet = self.alphabet[offset:] + self.alphabet[:offset]
```

Figure 9: Python code *Source: ACARA*

- Does this change any other aspects of the program?

Examine the Python code at this website: starcoder.org/hacking/post-enigma-machine/

- What changes could you make to simplify it?
- Are all aspects of the original Enigma machine covered in the program?

Links to the Australian Curriculum

Tables 1 and 2 show related aspects of the Australian Curriculum.

Table 1: Links from the task to the Australian Curriculum: Digital Technologies Years 7–8

<table>
<tr><td>Digital Technologies

Achievement standard</td><td colspan="3">By the end of Year 8, students develop and modify creative digital solutions, decompose real-world problems, and evaluate alternative solutions against user stories and design criteria. Students acquire, interpret and model data with spreadsheets and represent data with integers and binary. They design and trace algorithms and implement them in a general-purpose programming language. Students select appropriate hardware for particular tasks, explain how data is transmitted and secured in networks, and identify cyber security threats. They select and use a range of digital tools efficiently and responsibly to create, locate and share content; and to plan, collaborate on and manage projects. Students manage their digital footprint.</td></tr>
<tr><td>Strands

Sub-strands</td><td colspan="3">Digital Technologies knowledge and understanding
Digital systems
Digital Technologies processes and production skills
Creating designed solutions
Generating and designing
Privacy and security</td></tr>
<tr><td>Content descriptions</td><td colspan="3">investigate how data is transmitted and secured in wired and wireless networks including the internet AC9TDI8K02
design algorithms involving nested control structures and represent them using flowcharts and pseudocode AC9TDI8P05
explain how multi-factor authentication protects an account when the password is compromised and identify phishing and other cyber security threats AC9TDI8P13</td></tr>
<tr><td rowspan="2">Technologies core concepts</td><td rowspan="2">systems
computational thinking
systems thinking</td><td>Digital Technologies core concepts</td><td>specification
algorithms
implementation
digital systems
privacy and security</td></tr>
<tr><td>General capabilities</td><td>Digital Literacy
Literacy
Numeracy</td></tr>
<tr><td>Cross-curriculum priorities</td><td></td><td>Learning area or subject connections</td><td>History
Mathematics</td></tr>
</table>

Table 2: Links from the task to the Australian Curriculum: Digital Technologies Years 9–10

<table>
<tr><td>Digital Technologies
Achievement standard</td><td colspan="3">By the end of Year 10, students develop and modify innovative digital solutions, decompose real-world problems, and critically evaluate alternative solutions against stakeholder elicited user stories. Students acquire, interpret and model complex data with databases and represent documents as content, structure and presentation. They design and validate algorithms and implement them, including in an object-oriented programming language. Students explain how digital systems manage, control and secure access to data; and model cyber security threats and explore a vulnerability. They use advanced features of digital tools to create interactive content, and to plan, collaborate on, and manage agile projects. Students apply privacy principles to manage digital footprints.</td></tr>
<tr><td>Strands
Sub-strands</td><td colspan="3">Digital Technologies knowledge and understanding
Digital systems
Digital Technologies processes and production skills
Creating designed solutions
Generating and designing
Privacy and security</td></tr>
<tr><td>Content descriptions</td><td colspan="3">investigate how hardware and software manage, control and secure access to data in networked digital systems AC9TDI10K01
design algorithms involving logical operators and represent them as flowcharts and pseudocode AC9TDI10P05
develop cyber security threat models, and explore a software, user or software supply chain vulnerability AC9TDI10P13</td></tr>
<tr><td rowspan="2">Technologies core concepts</td><td rowspan="2">systems
computational thinking
systems thinking</td><td>Digital Technologies core concepts</td><td>specification
algorithms
implementation
digital systems
privacy and security</td></tr>
<tr><td>General capabilities</td><td>Digital Literacy
Literacy
Numeracy</td></tr>
<tr><td>Cross-curriculum priorities</td><td></td><td>Learning area or subject connections</td><td>History
Mathematics</td></tr>
</table>

Resource
PDF templates for unplugged Enigma machine:
https://fhcouk.files.wordpress.com/2012/05/pringlesenigma3a4.pdf

Useful links
- A basic guide to making your own Enigma machine: cyber.org/enigma
- How Alan Turing cracked the Enigma machine: www.iwm.org.uk/history/how-alan-turing-cracked-the-enigma-code
- How the Enigma machine worked: www.youtube.com/watch?v=G2_Q9FoD-oQ

Acknowledgement
This activity was developed as part of the University of New South Wales – Graduate Certificate in Cyber Security (Security Engineering Course)