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.

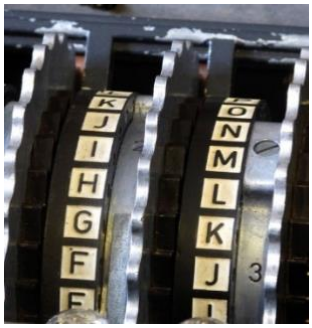

Figure 2: Enigma rotors
Source: commons.wikimedia.org/wiki/File:Enigma_rotors_with_alphabet_rings.jpg

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.

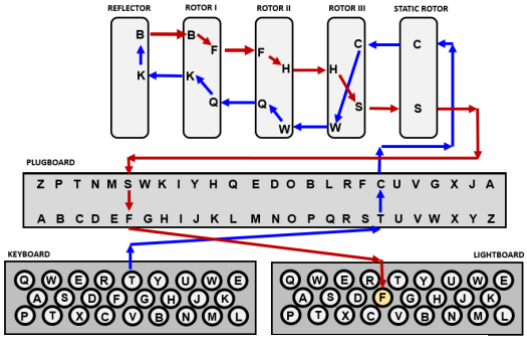

Figure 3: Letter pathway
Source: UNE Museum of Antiquities

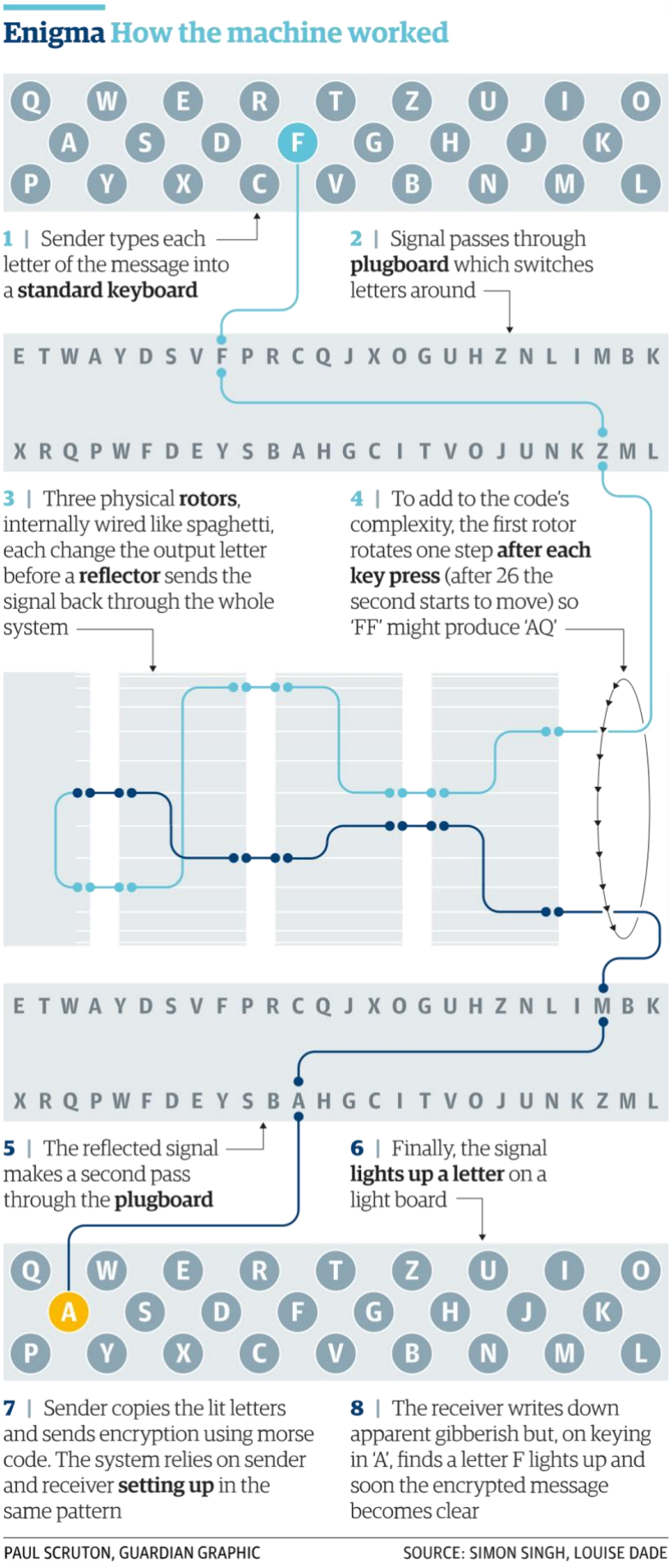

Figure 4: Letter pathway
Source: www.theguardian.com/technology/2014/nov/14/how-did-enigma-machine-work-imitation-game

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.

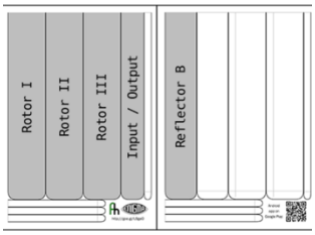

Figure 5: Basic rotor set up Source: wiki.franklinheath.co.uk/index.php/Enigma/Paper_Enigma

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.

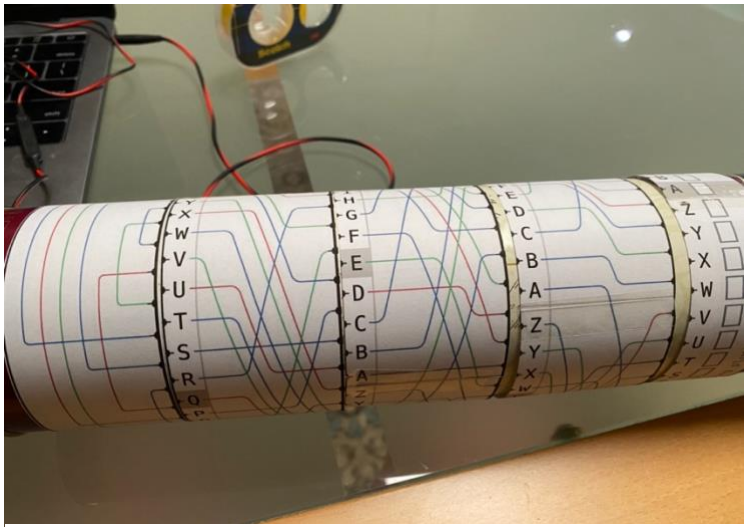

Figure 6: Finished Enigma machine Source: ACARA

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?

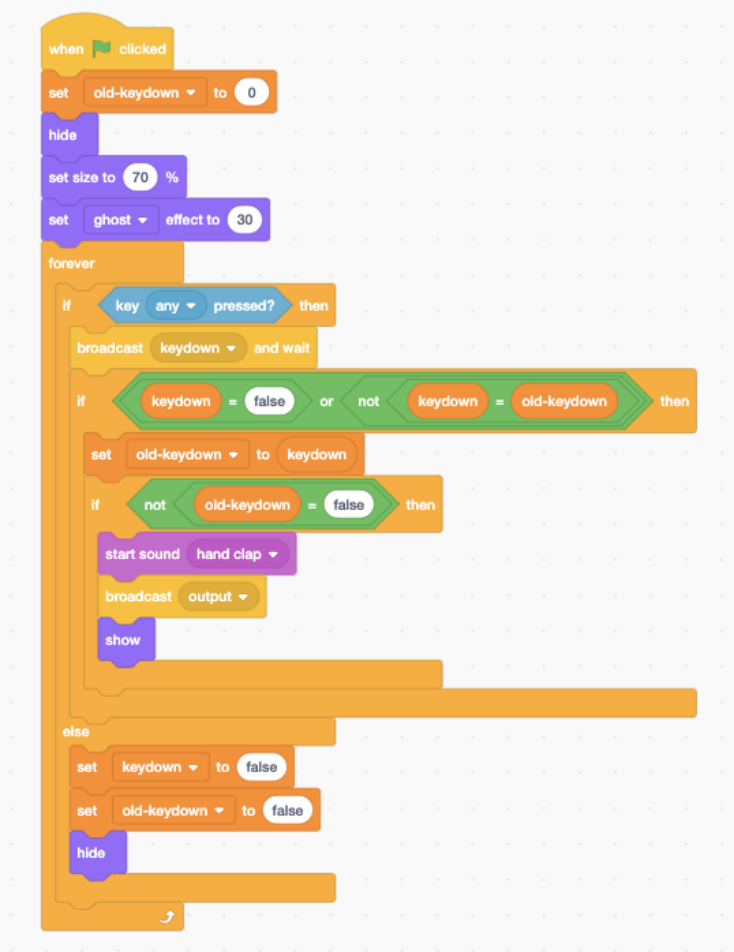

Figure 7: Explore the code blocks
Source: scratch.mit.edu/projects/572099314/

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 (v8.4)

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 (v8.4)

<table>
<tr><td>Digital Technologies

Achievement standard</td><td colspan="3">By the end of Year 8, students distinguish between different types of networks and defined purposes. They explain how text, image and audio data can be represented, secured and presented in digital systems.
Students plan and manage digital projects to create interactive information. They define and decompose problems in terms of functional requirements and constraints. Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability. They analyse and evaluate data from a range of sources to model and create solutions. They use appropriate protocols when communicating and collaborating online.</td></tr>
<tr><td>Strands</td><td colspan="3">Digital Technologies processes and production skills
Collecting, managing and analysing data
Evaluating
Collaborating and managing</td></tr>
<tr><td>Content descriptions</td><td colspan="3">Acquire data from a range of sources and evaluate authenticity, accuracy and timeliness (ACTDIP025)
Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)
Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032)</td></tr>
<tr><td rowspan="2">Key concepts</td><td rowspan="2">data collection
data interpretation</td><td>Key ideas</td><td>Thinking in Technologies
Systems thinking</td></tr>
<tr><td>General capabilities</td><td>ICT capability
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 (v8.4)

<table>
<tr><td>Digital Technologies

Achievement standard</td><td colspan="3">By the end of Year 10, students explain the control and management of networked digital systems and the security implications of the interaction between hardware, software and users. They explain simple data compression, and why content data are separated from presentation.

Students plan and manage digital projects using an iterative approach. They define and decompose complex problems in terms of functional and non-functional requirements. Students design and evaluate user experiences and algorithms. They design and implement modular programs, including an object-oriented program, using algorithms and data structures involving modular functions that reflect the relationships of real-world data and data entities. They take account of privacy and security requirements when selecting and validating data. Students test and predict results and implement digital solutions. They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise. They share and collaborate online, establishing protocols for the use, transmission and maintenance of data and projects.</td></tr>
<tr><td>Strands</td><td colspan="3">Digital Technologies processes and production skills
• Collecting, managing and analysing data
• Generating and designing
• Evaluating
• Collaborating and managing</td></tr>
<tr><td>Content descriptions</td><td colspan="3">• Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements (ACTDIP036)
• Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data (ACTDIP037)
• Evaluate critically how student solutions and existing information systems and policies, take account of future risks and sustainability and provide opportunities for innovation and enterprise (ACTDIP042)
• Plan and manage projects using an iterative and collaborative approach, identifying risks and considering safety and sustainability (ACTDIP044)</td></tr>
<tr><td rowspan="2">Key concepts</td><td rowspan="2">• data collection
• data interpretation
• abstraction
• algorithms
• specification</td><td>Key ideas</td><td>Thinking in Technologies
• systems thinking
• computational thinking</td></tr>
<tr><td>General capabilities</td><td>• ICT capability
• 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)