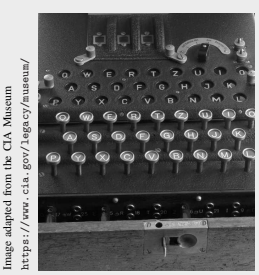
Computer Science Connections

The Enigma Machine and a First Computer

The Enigma machine was a physical cryptographic device used by the Germans during World War II to communicate between German high command and their military units in the field. The basic structure of the machine involved *rotors* and *cables*. A *rotor* was a 26-slot physical wheel that encoded a permutation π; when the wire corresponding to input i is active, the output wire corresponding to π_i is active. A *plugboard* allowed an arbitrary matching of keys on the keyboard to the inputs to the rotors—a *cable* was what actually connected a key to the first rotor. (The machine did not require any cables in the plugboard; if there was no cable, then the key pressed was what went into the rotor in the first place.) The basic encryption in the Enigma machine proceeded as follows (see Figure 9.35):



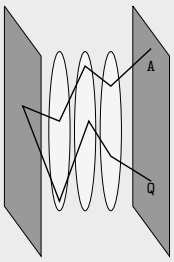


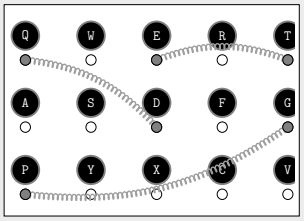
Figure 9.35 An Enigma machine, and a schematic of its operation. The operator types a letter (say, A), which goes through the plugboard, and is then permuted by rotor #1, rotor #2, rotor #3, the fixed permutation of the machine, rotor #3, rotor #2, and rotor #1. It then (after passing through the plugboard) lights up the output, say Q. The rotors advance by one notch, and encoding continues.

(1) The user pressed a key, say A, on the keyboard. If there was a cable from the A key, then the key would be remapped to the other end of the cable; otherwise the procedure proceeded using the A. (See Figure 9.36.)

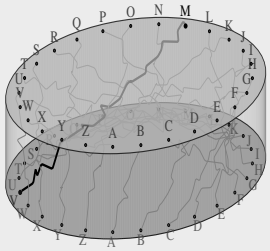
(2) The pressed key was permuted by rotor #1; the output of rotor #1 was permuted by rotor #2; the output of rotor #2 was permuted by rotor #3. (Again, see Figure 9.36.) The output of rotor #3 was "reflected" by a fixed permutation, and then the reflector's output pass through the three rotors, in reverse order and backward: the output of the reflector was permuted by rotor #3, then by #2, and then by #1.

(3) A light corresponding to the output of rotor #1, passed through the plugboard cable if present, lights up; the illuminated letter is the encoding.

The tricky part is that the rotors rotate by one notch, a bit like an odometer, when the key is pressed, so that the encoding changes with every keypress. The "secret key" that the two communicating entities needed to agree upon was which rotors to use in which order ($5 \cdot 4 \cdot 3 = 60$; there were 5 standard rotors in an Enigma), what the initial position of the rotors should be ($26^3 = 17{,}576$), and what plugboard matching to use ($\frac{26!}{13! \cdot 2^{13}} \approx 8 \times 10^{12}$ choices if all 26 letters were matched; see Example 9.34). Interestingly, almost all of the complexity came from the plugboards.



Each of the 26 keys is either mapped to itself (like W here), or is matched with another key (like Q ↔ D here). Pressing an unmatched key x yields x itself; pressing a matched key x yields whatever letter is matched to x.



Each rotor encodes a permutation of the letters; when the input letter i comes into the rotor, the output π_i comes out. (Here, for example, an input V turns into an output of M.) After each keypress, the top portion of the rotor rotates by a notch, so V would now turn into N.

Figure 9.36 The plugboard (above) and a rotor (below).