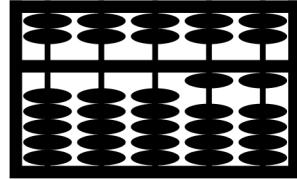


CHAPTER 11



Cryptography

Computers are most valuable when they are used to solve problems that humans cannot easily solve for themselves. Charles Babbage, for example, wanted to automate the production of mathematical tables, partly because it was a tedious task, but mostly because the people who undertook the necessary calculations made so many mistakes. Computers, however, are also useful when they solve problems *faster* than human beings. If you face a situation in which timeliness is essential, you may not be able to wait for results generated at human speeds. In such cases, it may be necessary to develop a technological solution to get the answers you need when you need them.

In World War II, the Allies faced precisely this situation. The shipping lanes of the North Atlantic were under such threat from German U-boats that Britain was in danger of being starved into submission. Breaking the U-boat code was a critical turning point in the war and may have changed its outcome. Faced with a code that changed every day, the British had to develop mechanical tools that would allow them to read German military dispatches quickly enough to act on that information.

Breaking the German military codes was an early application of *cryptography*, which is the science of creating and decoding messages whose meaning cannot be understood by those who intercept the message. In the language of cryptography, the message you are trying to send is called the *plaintext*; the message you actually send is called the *ciphertext*. Unless your adversaries know the secret of the encoding system, which is usually embodied in some privileged piece of information called a *key*, intercepting the ciphertext should not make it possible for them to discover the original plaintext version of the message. On the other hand, the recipient, who is presumably in possession of the key, can easily translate the ciphertext back into its plaintext counterpart.

The Navajo code talkers

As you will discover in this chapter, cryptography was one of the earliest applications of modern computing. During World War II, a codebreaking team in England, building on earlier work carried out in Poland, developed specialized hardware that was able to break the German Enigma code. Breaking that code was critical to the Allied victory in the battle for control of the Atlantic shipping lanes.

World War II offers other cryptographic stories as well—stories that underscore the fact that high technology does not necessarily offer the best solution to the problem of secure communication. In the war against Japan, the United States Marine Corps relied on the Navajo, a Native American tribe from the southwestern United States, to exchange messages over radio channels on which anyone might be listening. Approximately 400 Navajos served as “code talkers” from 1942 to 1945 and played a vital role in the war effort. Howard Connor, signal officer for the 5th Marine Division observed that “were it not for the Navajos, the Marines would never have taken Iwo Jima.”

The code talkers did not simply speak Navajo over the radio. Military messages often include words that do not exist in Navajo, along with place names and other words that are hard to translate. If, for example, you wanted to send a message warning of submarines off Bataan, you would have to decide how to express *submarine* and *Bataan*, neither of which has a Navajo counterpart.

To solve this problem, the code talkers used a variety of strategies. For common military terms, Navajo words were used to provide an appropriate metaphor; *submarine*, for example, was expressed using the Navajo words for *iron fish*. Place names were translated using a spelling strategy involving both English and Navajo. To send the word *Bataan*, for example, the code talkers first spelled it out using English words beginning with the appropriate letters. One possibility looks like this:

bear apple tooth axe ant needle

The code talker would then substitute the Navajo words and deliver the following message:

shush be-la-sana a-woh tse-nill wol-la-chee tsah



Navajo code talker at his radio during World War II

The native speaker on the receiving end would listen for each word, translate it back from Navajo to English, and then record the initial letters.

It is important to note that the spelling scheme used by the code talkers allows many words to stand for the same letter. The three occurrences of the letter *a* in *Bataan* are each represented by a different Navajo word, making the code much more difficult to break.

The Navajo code talkers proved to be much faster than the encryption strategies adopted by the other service branches. A well-trained pair of code talkers could transmit a three-line message in 20 seconds; the fastest encryption machines of the day required 30 minutes to deliver the same message. More importantly, the code-talker strategy proved to be more secure. The Japanese were able to break the codes used by the Army and Army Air Core, but were never able to decipher the messages sent by the Navajo code talkers.

On September 17, 1992, the surviving members of the Navajo code talkers were honored at the dedication of a commemorative exhibit at the Pentagon in Washington, DC.

11.1 Early history of cryptography

Cryptography has been around in some form or another for most of recorded history. There is evidence to suggest that coded messages were used in ancient Egypt, China, and India, possibly as early as the third millennium BCE, although few details of the cryptographic systems have survived. In Book 6 of the *Iliad*, Homer suggests the existence of a coded message when King Proitos, seeking to have the young Bellerophontes killed, has

. . . sent him to Lykia, and handed him murderous symbols,
which he inscribed on a folding tablet, enough to destroy life

Hamlet, of course, has Rosencrantz and Guildenstern carry a similarly dangerous missive, but Hamlet's message is secured under a royal seal. In the *Iliad*, there is nothing to suggest that Bellerophontes cannot see the "murderous symbols," which implies that their meaning must somehow be disguised.

One of the first encryption systems whose details survive is the **Polybius square**, developed by the Greek historian Polybius in the second century BCE. In this system, the letters of the alphabet are arranged to form a 5×5 grid in which each letter is represented by its row and column number. Suppose, for example, that you want to transmit following English version of Pheidippides' message to Sparta:

THE ATHENIANS BESEECH YOU TO HASTEN TO THEIR AID

This message can be transmitted as a series of numeric pairs, as follows:

**44 23 15 11 44 23 15 33 24 11 33 43 12 15 43 15 15 13 23 54
34 45 44 34 23 11 43 44 15 33 44 34 44 23 15 24 42 11 24 13**

The advantage of the Polybius square is not so much that it allows for secret messages, but that it simplifies the problem of transmission. Each letter in the message can be represented by holding between one and five torches in each hand, which allows a message to be passed quickly over great distances. By reducing the alphabet to an easily transmittable code, the Polybius square anticipates such later developments as Morse code and semaphore, not to mention modern digital encodings such as ASCII or Unicode.

In *De Vita Caesarum*, written sometime around 110 CE, the Roman historian Suetonius describes an encryption system used by Julius Caesar, as follows:

If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely *D*, for *A*, and so with the others.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Polybius square

Even today, the technique of encoding a message by shifting letters a certain distance in the alphabet is called a *Caesar cipher*. According to the passage from Suetonius, each letter is shifted three positions ahead in the alphabet. For example, if Caesar had had time to translate his final words according to his coding system, **ET TU BRUTE** would have come out as **HW WX EUXWH**, because **E** gets moved three letters ahead to **H**, **T** gets moved three to **W**, and so on. Letters that get advanced past the end of the alphabet wrap around back to the beginning, so that **X** would become **A**, **Y** would become **B**, and **Z** would become **C**.

Caesar ciphers have survived into modern times. On the early electronic bulletin boards that were popular at the beginning of the Internet era, users could disguise the content of postings that might offend some readers by employing a mode called **ROT13**, which is simply a Caesar cipher that shifts all letters forward 13 positions. And the fact that **HAL**—the name of the computer in Arthur C. Clarke’s *2001*—is a one-step Caesar cipher of **IBM** has generated some amount of interest among fans.

Although Caesar ciphers are certainly simple, they are also extremely easy to break. There are, after all, only 25 possible Caesar ciphers for English text. If you want to break a Caesar cipher, all you have to do is try each of the 25 possibilities and see which one translates the ciphertext message into something readable.

A somewhat more secure scheme is to allow each letter in the plaintext message to be represented by some other letter, but not one that is simply a fixed distance from the original. In this case, the key for the encoding operation is a letter translation table that shows what each of the possible plaintext characters becomes in the ciphertext. Such a coding scheme is called a *letter-substitution cipher*.

Letter-substitution ciphers have been used for many, many years. Examples of such ciphers appear in several works from both classical and medieval times. In the early 15th century, the Arabic encyclopedia *Subh al-a 'sha* included a section on cryptography describing various methods for creating ciphers as well as techniques for breaking them. In particular, this manuscript included the first instance of a cipher in which several different coded symbols can stand for the same plaintext character. Codes in which each plaintext letter maps into a single ciphertext equivalent are called *monoalphabetic ciphers*; codes in which each character can have more than one coded representation are called *polyalphabetic ciphers*.

11.2 Cryptograms

Today, monoalphabetic ciphers survive primarily in the form of letter-substitution puzzles called *cryptograms*. Edgar Allan Poe was a great fan of cryptograms and included a cryptographic puzzle in the excerpt from *The Gold Bug* shown in Figure 11-1.

FIGURE 11-1 Cryptographic puzzle from *The Gold Bug* by Edgar Allan Poe

Here Legrand, having re-heated the parchment, submitted it to my inspection. The following characters were rudely traced, in a red tint, between the death's head and the goat:

```
53‡†305)6*;4826)4‡•)4‡;806*;48†8¶
60)85;1‡(;:#*8†83(88)5*†;46(;88*96*
?;8)*‡(;485);5*†2:*)‡;4956*2(5*-4)8¶
8*;4069285);)6†8)4‡‡;1(‡9;48081;8:8‡
1;48†85;4)485†528806*81(‡9;48;(88;4‡
‡?34;48)4‡;161;:188;‡?;
```

"But," said I, returning him the slip, "I am as much in the dark as ever. Were all the jewels of Golconda awaiting me upon my solution of this enigma, I am quite sure that I should be unable to earn them."

"And yet," said Legrand, "the solution is by no means so difficult as you might be led to imagine from the first hasty inspection of the characters. These characters, as any one might readily guess, form a cipher . . . such, however, as would appear to the crude intellect of the sailor, absolutely insoluble without the key."

"And you really solved it?"

"Readily; I have solved others of an abstruseness ten thousand times greater. Circumstances, and a certain bias of mind, have led me to take interest in such riddles, and it may well be doubted whether human ingenuity can construct an enigma of the kind which human ingenuity may not, by proper application, resolve. In fact, having once established connected and legible characters, I scarcely gave a thought to the mere difficulty of determining their import.

"My first step was to ascertain the predominant letters, as well as the least frequent. Counting all, I constructed a table thus:

Of the character	8	there are	33
;	"	26	
4	"	19	
‡,)	"	16	
*	"	13	
5	"	12	
6	"	11	
("	10	
†, 1	"	8	
0	"	6	
9, 2	"	5	
: , 3	"	4	
?	"	3	
¶	"	2	
-, •	"	1	

"Now, in English, the letter which most frequently occurs is **e**. Afterward, the succession runs thus:

a o i d h n r s t u y c f g l m w b k p q x z

" . . . Let us assume **8**, then, as **e**. Now, of all words in the language, **the** is most usual; let us see, therefore, whether there are not repetitions of any three characters, in the same order of collocation, the last of them being **8**. If we discover a repetition of such letters, so arranged, they will most probably represent the word **the**. Upon inspection, we find no less than seven such arrangements, the characters being ;48. We may, therefore, assume that ; represents **t**, 4 represents **h**, and 8 represents **e**—the last being now well confirmed. . . .

"But, having established a single word, we are enabled to establish a vastly important point; that is to say, several commencements and terminations of other words. Let us refer, for example, to the last instance but one, in which the combination ;48 occurs—not far from the end of the cipher. We know that the ; immediately ensuing is the commencement of a word, and, of the six characters succeeding this **the**, we are cognizant of no less than five. Let us set these characters down, thus, by the letters we know them to represent, leaving a space for the unknown—**t_eeth**.

"Here we are enabled, at once, to discard the **th** as forming no portion of the word commencing with the first **t**; since, by experiment of the entire alphabet for a letter adapted to the vacancy, we perceive that no word can be formed of which this **th** can be a part. We are thus narrowed into **t_ee**, and, going through the alphabet, if necessary, as before, we arrive at the word **tree** as the sole possible reading. We thus gain another letter, **r**. . . .

"I have said enough to convince you that ciphers of this nature are readily soluble, and to give you some insight into the rationale of their development. . . . It now only remains to give you the full translation of the characters upon the parchment, as unriddled. Here it is:

**A good glass in the bishop's hostel in
the devil's seat forty-one degrees and
thirteen minutes northeast and by north
main branch seventh limb east side shoot
from the left eye of the death's-head a
bee-line from the tree through the shot
fifty feet out.**

- (a) The following coded message is an enciphered version of the opening paragraph from a well-known English novel after removing all spaces and punctuation and then breaking the message into five-letter groups:

```
LESTX KQLEY TQOJX ZEHYT QJQKL IQHST XAALY EXYSE SRYPH LJYPG
QYTXK QVLKK QHGLY TYTQQ EHRXV GXJBR SEHSE XXFPR BQKKE XJPQY
SHJPA SJQRS EHPTX KQGLY TEXYT LEOLE LYXXR LYHXR EXEXJ YXQSY
LYGSR STXAA LYTXK QSEHY TSYBQ SERDX BVXJY
```

Use Poe's strategy to decipher this message. Remember that the letter frequencies are just an approximation and that **E** is not always the most common letter.

- (b) In the Sherlock Holmes mystery, *The Adventure of the Dancing Men*, by Sir Arthur Conan Doyle, Holmes receives several messages written in what appears to be "a number of absurd little figures dancing across the page upon which they are drawn." See if you can apply Poe's techniques to this cipher, which did not stump Holmes for long:



In describing the solution to Captain Kidd's message, Poe offers a general technique for solving monoalphabetic ciphers: calculate the frequency of the letters used in the ciphertext and correlate the appearance of coded sequences with the frequency of letters in English. By guessing that the letters appearing most often in the ciphertext correspond to the most common letters in English, you can usually make a good start toward solving such puzzles.

If you try to solve cryptograms on your own, however, it will help you to know that Poe's list of the most common letters is not in fact correct. Computerized analysis reveals that the most common letters in English are

E T A O I N S H R D L U

Given that statistical studies of English text were by no means as well developed in Poe's day, Poe can perhaps be excused for making a few mistakes.

What Poe did realize is that solving a monoalphabetic cipher requires a strategy. The Caesar cipher, for example, requires one to check only 25 possibilities before the correct plaintext must appear. In the general case of a letter-substitution cipher, there are 26 possible letters to choose as the coded representation for **A**, 25 remaining possible letters to choose as the coded representation for **B**, 24 possibilities for **C**, and so on, for a total of $26!$ ($26 \times 25 \times 24 \times \dots \times 3 \times 2 \times 1$) possible encodings. This number is extremely large, equal in decimal notation to 403,291,461,126,605,635,584,000,000. Even with modern computers, it isn't feasible to solve this problem by trying every possibility. One needs instead to be more subtle.

11.3 The Enigma machine

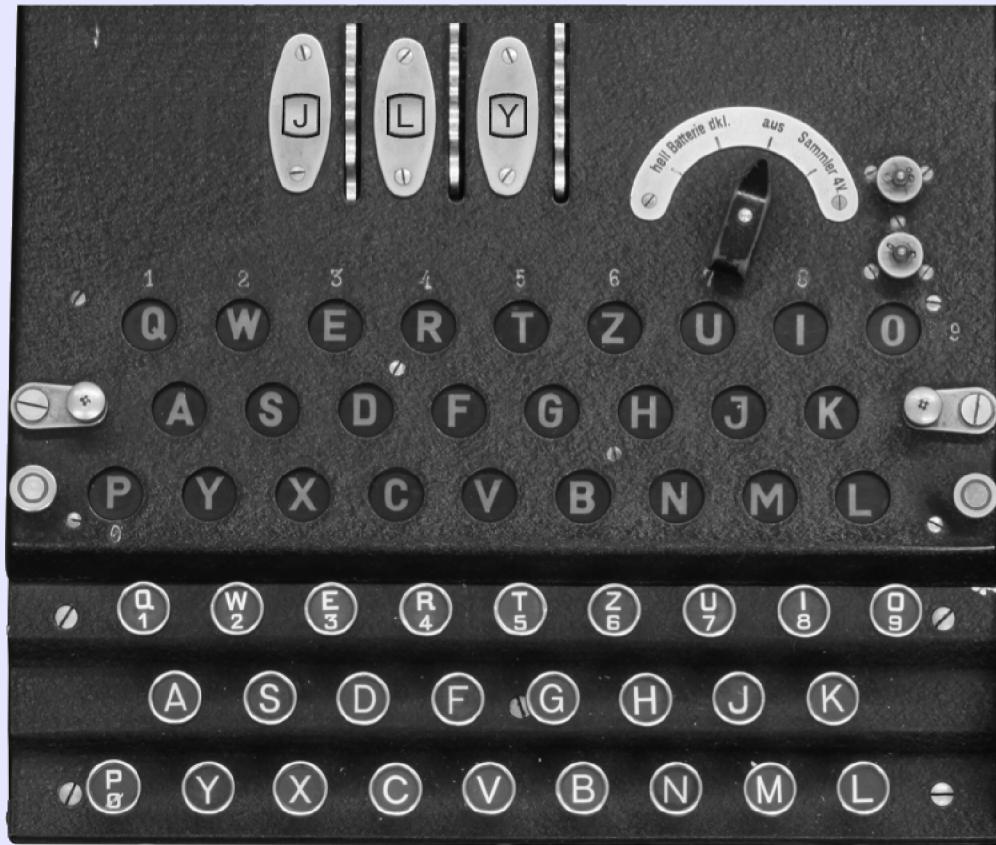
In many ways, modern computing got its start during World War II. On both sides, the war focused attention on military priorities and made it possible to apply unprecedented levels of resources in an attempt to gain the advantage. The Germans, for example, made enormous investments in missile technology, which led to the development of the V-1 and V-2 rockets that fell with such devastating effect on England during the Blitz. In the United States, the Manhattan Project brought together the leading scientists of the day to develop the atomic bomb.

As noted in the introduction to this chapter, the war forced Britain to apply considerable resources to the problem of deciphering messages that the German High Command used to communicate with the army, navy, and air force. Although each service branch used a slightly different technology, all were built upon a common foundation that made it possible for the Allies to break those codes.

In the early 1930s, the German military adopted a new encryption protocol based on an existing commercial device called *Enigma*. Figure 11-2 shows the top view of a typical Enigma machine, expanded so that you can see the detail. At the bottom of the figure is a keyboard arranged in the standard German layout. Above the keyboard is an array of lamps. Pressing a key lights one of the lamps, thereby indicating the encoded version of that letter. The mapping from keys to lamps is controlled by the three thumb wheels at the top of the diagram, which are called *rotors*. Each rotor can be set to any of 26 positions corresponding to the letters of the alphabet. The display windows at the top of Figure 11-2 show the letters **JLY**, which is called the *rotor setting*.

Early models of the Enigma machine included only the components shown in Figure 11-2. These machines had $17,576$ ($26 \times 26 \times 26$) settings, which made it possible, given sufficient time, to decrypt a message by trying every rotor setting.

FIGURE 11-2 Top view of the German Enigma machine

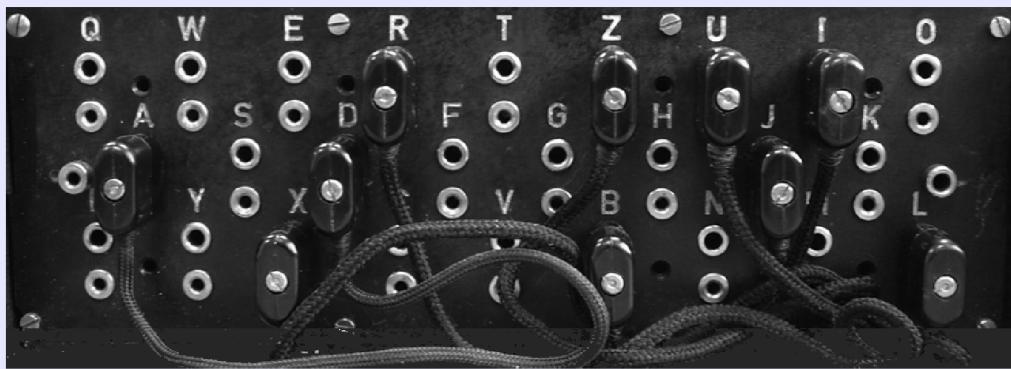


To increase the security of Enigma, the German government mandated several changes in its design. Instead of using a fixed set of rotors, the Enigma machines used during the war allowed operators to select and arrange any three rotors from a set of five. This change meant that codebreakers had to consider 60 ($5 \times 4 \times 3$) possible rotor arrangements. Military models of the Enigma machine added a ring inside each rotor that introduced an additional offset into the transformation and changed the point at which the next rotor advanced. The addition of the ring had no real impact on the decryption strategy and is not considered in this chapter.

From the perspective of would-be codebreakers, the change that added the most complexity was the introduction of a new front panel containing jacks associated with the letters of the alphabet, as shown in Figure 11-3. In German, this panel was called the *steckerbrett*, which is traditionally rendered in English as *steckerboard*. Enigma operators were issued a set of cables that allowed them to exchange pairs of letters during the encryption. Although it's hard to follow the tangle of cables in the photograph, the steckerboard wiring in Figure 11-3 exchanges the pairs of letters **A-D**, **B-X**, **I-Z**, **J-U**, and **L-R**. Letters connected in this way are called *stecker pairs*.

The addition of the steckerboard vastly increased the number of possible settings for the Enigma machine. The set of ten plug wires Enigma operators were issued during the war allowed for 216,751,064,975,576 possible wirings. Taken together with the 17,576 possible rotor settings and the 60 possible ways to select and arrange the rotors, the number of initial settings of the Enigma machine was the astronomical 228,577,003,080,643,426,560. Even with today's technology, trying every possible combination would take a considerable amount of time. Given the technology available in World War II, trying every possibility was not a realistic option. Decoders had to rely on cleverness and insight—along with a bit of mechanical assistance—to break the Enigma code.

FIGURE 11-3 The Enigma steckerboard



11.4 The codebreakers

In 1938, recognizing the danger of war in Europe, the head of British intelligence purchased an estate about 50 miles northwest of London called Bletchley Park, which became the home of the Government Code and Cipher School. More than 10,000 people worked at Bletchley Park during the war, under the strictest secrecy. The task of breaking Enigma fell to a team of cryptographers at Bletchley Park working under the code name *Ultra*. The Ultra team employed many of Britain's best mathematicians, including Alan Turing, the inventor of the Turing machine described in Chapter 8. Despite its enormous complexity, the mathematicians of Ultra managed to break the Enigma code. In fact, they did so several times.

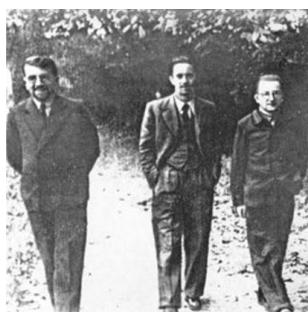
Cryptography is in many ways a race between codemaker and codebreaker. The Germans made periodic improvements to the Enigma both before and during the war. With each redesign, the codebreakers had to come up with a new strategy to overcome the enhancements on the German side. When the German navy added a fourth rotor to the Enigma in February 1942, the Allies were unable to read Enigma traffic for ten months. By the end of the war, however, Bletchley Park was able to decipher most encrypted messages in less than a day.

Being able to read German military communications was vital to the Allied cause. In 1941, Alan Turing and several of his colleagues wrote directly to Prime Minister Winston Churchill requesting more resources for the decryption effort. Fully aware of the importance of the Ultra project, Churchill replied

Make sure they have all they want on extreme priority and report to me
that this had been done. Action this day.

After the war, Churchill is reported to have told King George VI that “it was thanks to Ultra that we won the war.”

The cryptographers at Bletchley owed a considerable debt to the Polish cryptographers Marian Rejewski, Jerzy Różycki, and Henryk Zygalski, who were able to break the Enigma code in 1932. In the process, they also developed many of the cryptographic techniques that would later guide the British effort. Fortunately, the Polish team was able to share its decryption work with the Allies shortly before the German invasion of Poland in 1939 that marked the beginning of the war. The Polish team later made their way to France, where they carried on their cryptographic work along with French colleagues. When France itself was overrun, the Poles again escaped to England. Although the secrecy around the wartime cryptographic work meant that the Polish contribution to codebreaking remained unknown for many years, Bletchley Park now has a monument to commemorate the essential work of these Polish mathematicians.



Zygalski, Różycki, and Rejewski

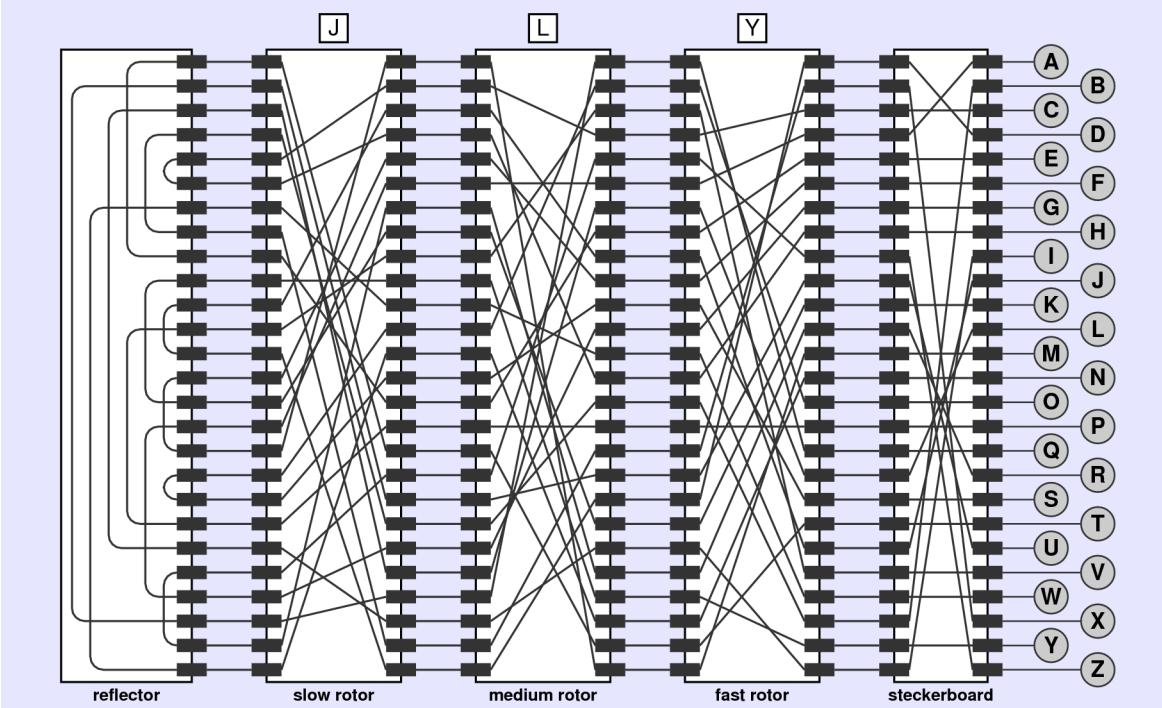
11.5 The internal structure of Enigma

Before you can understand how cryptographers were able to break the Enigma code, you need to know something about how the machine works. Figure 11-4 shows the internal structure, focusing on the wiring of the rotors and the steckerboard.

Each of the three rotors in the Enigma machine has 26 contacts along its left and right sides. Current that comes in at one contact on the rotor is redirected to a contact on the opposite side according to the internal wiring pattern, which is different for each rotor. Each rotor therefore implements a reordering of the letters, which mathematicians call a *permutation*. The steckerboard also implements a permutation, which is set manually according to the instructions in codebook.

The letters at the top of Figure 11-4 indicate the rotor setting. Typing a character on the keyboard automatically advances the rotor on the right, thereby changing the pattern of connections inside the machine. When that rotor has completed a full revolution, the middle rotor advances one step; in much the same way, completing a revolution of the middle rotor advances the rotor on the left. The rotors therefore advance in a fashion reminiscent of the odometer on a car. The right rotor advances

FIGURE 11-4 Structural diagram of the Enigma machine showing the rotor setting JLY

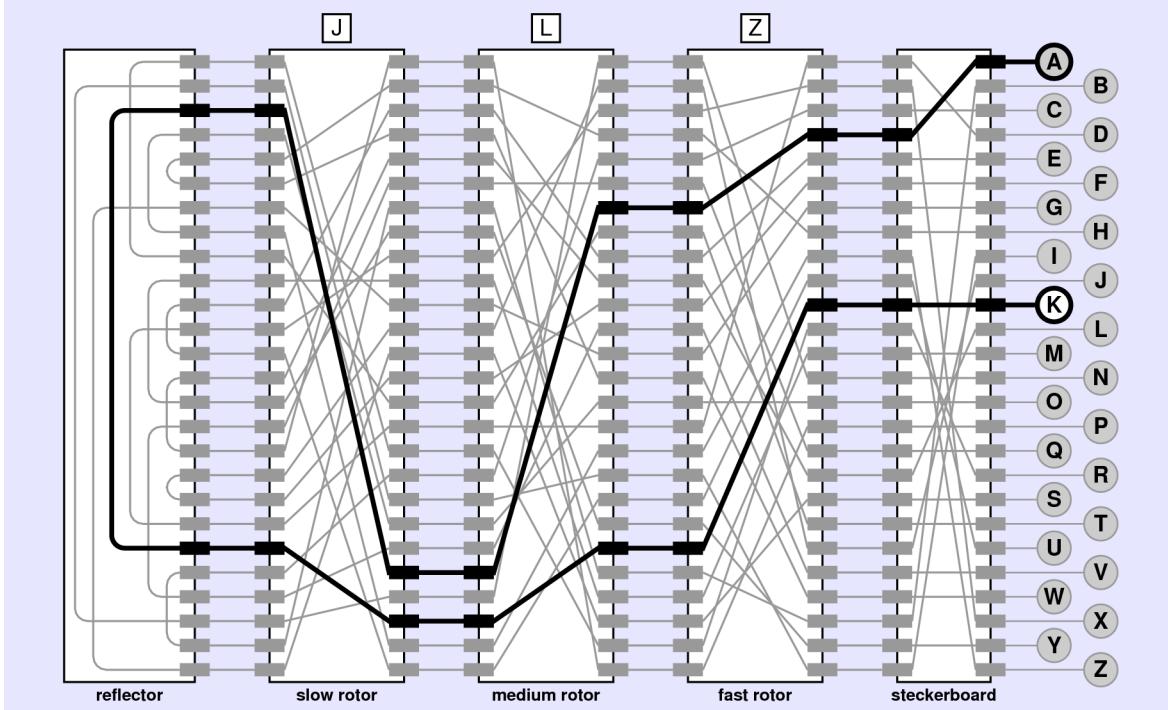


on every character and is therefore called the *fast rotor*. The middle rotor advances once every 26 characters and is called the *medium rotor*. The left rotor advances only once every 676 (26×26) characters and is unsurprisingly called the *slow rotor*.

Figure 11-5 shows what happens if the operator types the letter **A** on the keyboard. Pressing the key advances the fast rotor, which changes the rotor setting from **JLY** to **JLZ**. The Enigma machine then applies a current to the wire leading from the **A** key at the right edge of the diagram and, at the same time, disconnects the **A** lamp so that only the encrypted version of the letter appears. The current flows across the steckerboard, then through the three rotors from right to left. It then passes into a circuit element called the *reflector*, which implements a fixed permutation. From the reflector, the current flows back across the rotors in the opposite direction and then passes through the steckerboard one more time. As shown in the diagram, the current initiated by typing **A** ends up on the wire labeled **K**, which causes the **K** lamp to light. Thus, given the rotor setting **JLZ**, the ciphertext form of the letter **A** is **K**.

The encryption patterns generated by the Enigma machine are difficult to break because the machine implements a polyalphabetic cipher in which the encoding

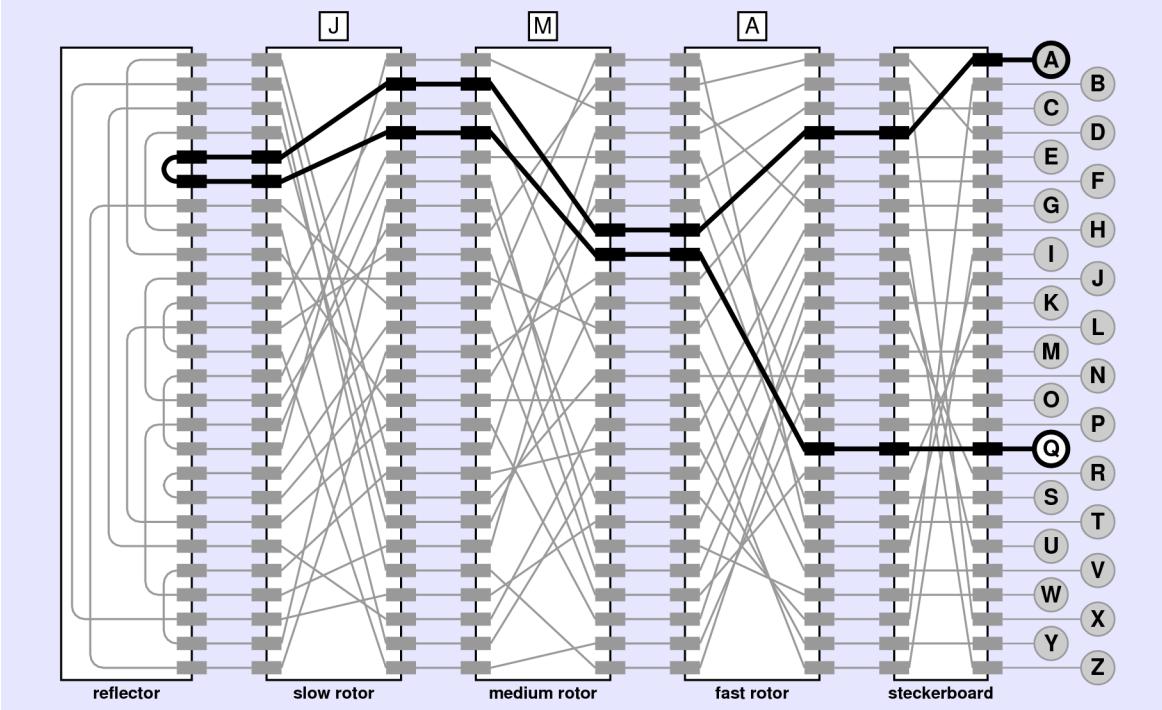
FIGURE 11-5 The Enigma machine after pressing A



changes on every character. If, for example, the operator types a second letter **A** immediately after the first, the machine advances to the configuration shown in Figure 11-6. This time, the fast rotor and the medium rotor both advance, because the fast rotor has made it all the way through to the end of the alphabet. Given the rotor setting **JKA** that appears after both rotors have moved forward, the letter **A** is now translated into the letter **Q**.

At this point, it is useful to note a fundamental symmetry in the Enigma design. If **A** is transformed to **Q** at some rotor setting, it must also be the case that **Q** is transformed to **A**. The circuit is exactly the same; the only difference is that the current flows in the opposite direction. This symmetry is very useful for Enigma operators because it means that the sender and receiver don't need to have two different keys. The sender sets the rotors and the steckerboard according to a codebook and types in the message. What comes out in the lights is the ciphertext, which is typically transmitted over a radio channel in Morse code. As long as the receiver uses the same codebook and sets up the machine in the same way, typing in the ciphertext restores the original message, because the encryption is reversible. As you will discover in the next section, however, the fact that the Enigma encoding is reversible also makes life easier for anyone trying to break the Enigma code.

FIGURE 11-6 The Enigma machine after pressing A a second time



11.6 Breaking the Enigma code

The decryption strategy developed in Poland and refined at Bletchley Park made use of the following facts about the Enigma machine:

- *The Enigma encoding is symmetrical.* As noted in the preceding section, if the **A** key is transformed into the letter **Q**, it must be the case that the **Q** key would be transformed into **A** for that particular rotor setting.
- *The Enigma machine can never map a character into itself.* Because of its construction and the symmetry of the transformation, it is never possible to have the letter **A**, for example, come back as the letter **A**.
- *The steckerboard does not affect the transformation pattern of the rotors, but only the characters to which the outputs of that rotor are assigned.* Although the addition of the steckerboard vastly increases the number of possible encodings, it does not change several fundamental properties of the machine.

The codebreakers were also fortunate that the German military was rigid in its communication style, which made it possible to anticipate what the content of a message might be. In particular, the Germans routinely transmitted weather reports at specific times of the day, which were often straightforward to guess if you knew what the weather looked like at the point of transmission. Similarly, many messages tended to start with a salutation to the receiving general, admiral, or captain in a way that included the full name and title. In salutations, the German word for *to* is *an*, which meant that the first characters in an intercepted message might be **ANGENERAL** (or **ANXGENERAL** for those branches of the German military that used the letter **X** to indicate a space). In fact, it was sometimes sufficient to guess that the first three characters in a message were **ANX** without having any idea of who the intended recipient might be.

The known-plaintext attack

The strategy of breaking a code by guessing at least part of the plaintext and then using that guess to deduce the encryption pattern is called a *known-plaintext attack*. The character sequence that you believe you know is called a *crib*. Ironically, one of the best cribs available to Project Ultra—at least according to some accounts—occurred in messages from a German officer in the North Africa campaign who foolishly sent periodic messages containing the German equivalent of *nothing to report*, which is *keine besonderen ereignisse*.

Suppose that one of the Allied listening posts in North Africa had intercepted the following coded message:

UAUNFYRLPZSWMEDSINFKRJXFSDKJCAXKEZ

If the sender is behaving in his usual way, you suspect that this message contains the plaintext sequence

KEINEBESONDERENEREIGNISSE

If you can figure out where in the message this sequence occurs, you might then be able to use the pattern of letters to make deductions about the settings of the Enigma machine. If these deductions allow you to determine the rotor pattern and the wiring of the steckerboard, you have broken the Enigma code for that day.

Aligning the crib with the ciphertext

The first challenge in implementing the known-plaintext attack consists of figuring out where in the ciphertext the suspected crib might occur. Fortunately, many of the potential positions for the crib can be ruled out simply by taking note of the fact that the Enigma machine never translates a letter to itself. For example, the crib cannot occur at the beginning of the ciphertext because the letter **N** would have to map to itself in the fourth character position, as would the letter **E** a bit further on, as shown in the following diagram:

UAE[NFVRLBZPWM]EPMIHFCSRJXFMJKWRAZQEZ
KEINEBESONDERENE[REIGNISSE]

The codebreakers at Bletchley used the word *crash* to refer to positions at which a letter in the ciphertext matches its counterpart in the crib. The first step in the decryption process is to slide the crib under the ciphertext until no crashes occur.

Figure 11-7 on the next page shows what happens if you carry out this process for every possible alignment of the crib and ciphertext. There are only two possible alignments that produce no crashes, which arise from shifting the crib five and six characters to the right, respectively. If the crib is correct, it must be in one of those two positions.

After eliminating the alignments ruled out because of crashes, the cryptographers at Bletchley would then try each of the possible alignments to see whether any of the remaining possibilities gave rise to a consistent rotor setting.

Deducing the rotor setting

Once you have a possible alignment, you can use the patterns of letters in the crib and the ciphertext to make inferences about the rotor setting. The basic idea is that only certain settings of the rotors will produce the pairings of letters you see between the crib and the appropriate region of the ciphertext. If you could use that information to eliminate all but a few of the possibilities, you could then check those settings by hand.

FIGURE 11-7 Crashes that rule out certain alignments between the crib and the ciphertext

U A E **N** F V R L B Z P W M **E** P M I H F S R J X F M J K W R A X Q E Z
 KE I N E B E S O N D E R E **N** E R E I G N I S S E
 0

U A **E** N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
 KE I N E B E S O N D E R E N E R E I G N I S S E
 1

U A E N F V R L B Z P W M **E** P M I H F S R J X F M J K W R A X Q E Z
 KE I N E B E S O N D E R E **E** N E R E I G N I S S E
 2

U A E N F V R L **B** Z P W M E P M I H F S R J X F M J K W R A X Q E Z
 KE I N E **B** E S O N D E R E N E R E I G N I S S E
 3

U A E N F V R L B Z P W M E P M I H F S **R** J X F M J K W R A X Q E Z
 KE I N E B E S O N D E R E N E **R** E I G N I S S E
 4

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
 KE I N E B E S O N D E R E N E R E I G N I S S E
 5

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
 KE I N E B E S O N D E R E N E R E I G N I S S E
 6

U A E N F V R L B Z P W M **E** P M I H F S R J X F M J K W R A X Q E Z
 KE I N E B E **S** O N D E R E N E R E I G N I S S E
 7

U A E N F V R L B Z P W M E P M I H F S **R** J X F M J K W R A X Q **E** Z
 KE I N E B E S O N D E **R** E N E R E I G N I S S E **E**
 8

U A E N F V R L B Z P W M **E** P M I H F S R J X F M J K W R A X Q E Z
 KE I N E **E** B E S O N D E R E N E R E I G N I S S E
 9

If there were no steckerboard, this process would be entirely straightforward. What you are looking for is a rotor setting that transforms some portion of the ciphertext back into the crib. Suppose, for example, that you assume that the crib appears at an offset of 5, as shown in the first boxed possibility in Figure 11-7. What you then need to do is find some setting of the rotors at which typing in

Y R L P Z S W M E D S I N F K R J X F S X K J C A

gives you back

K E I N E B E S O N D E R E N E R E I G N I S S E

Carrying out this analysis manually would certainly be time-consuming, but there are only 1,054,560 possible arrangements and settings for the rotors. If all 10,000 people at Bletchley Park—working in parallel—were able to test one of these settings every minute, you would find the solution in less than two hours.

Of course, given the resources available to Bletchley Park under Churchill’s designation of “extreme priority,” it would not have been necessary to divert all of Bletchley’s personnel to test the configurations. Given the technology of the time, it was possible to build a mechanical device to step through the 1,054,560 arrangements and settings of the rotors, checking for a match.

Unfortunately, the existence of the steckerboard rules out this simple strategy. Even if you find the right rotor settings, typing in

Y R L P Z S W M E D S I N F K R J X F S X K J C A

won’t regenerate the crib, because the letters are transformed by the connections on the steckerboard. If testing all possible arrangements of the rotors takes two hours, adding in the complexity of trying all 216,751,064,975,576 steckerboard wirings means that the process would take on the order of 10 billion years, which is a rough approximation of the age of the universe.

The critical insight that allowed the allies to break Enigma is that certain patterns in the letter pairings between the crib and the ciphertext are independent of the steckerboard. Consider, for example, the circled pairs of letters in the presumed alignment at offset 5:

V	R	L	(B)	Z	(P)	W	M	E	(P)	M	I	H	F	S	R	J	X	F	M	J	K	W	R	A
K	E	I	(N)	E	(B)	E	S	O	N	D	E	R	E	R	E	I	G	N	I	S	S	E		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

The numbers below the characters keep track of the index of the character in the crib, beginning—as is conventional in computer science—at index position 0.

Assuming that the crib and offset are correct, the Enigma machine encodes the plaintext **N** into the ciphertext **B** at index 3. Two characters later at index 5, the machine turns **B** into the ciphertext **P**. At index 9, the letter **N** becomes a **P**. Given the symmetry of the Enigma machine, however, you know that typing a **P** at index 9 would have produced an **N**, which is the letter that began this chain back at offset 0. The transformation pattern of **N** to **B**, **B** to **P**, and **P** to **N** form a closed cycle, which is easier to see if you connect the matching letters like this:



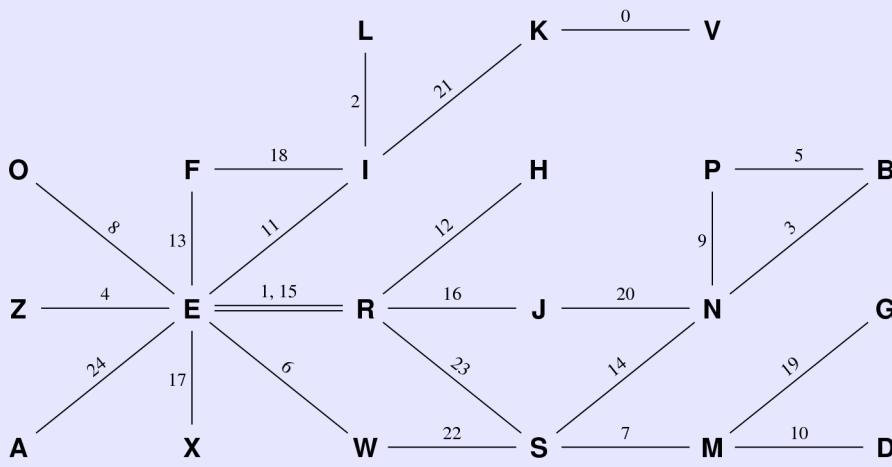
Alan Turing used the term *loop* to refer to this sort of closed cycle in the letter pairings between the crib and the ciphertext. The wonderful property of loops is that they are unaffected by the configuration of the steckerboard. Different settings of the steckerboard generate different letters in the ciphertext, but a cycle that occurs with one steckerboard setting will also occur if that setting is changed.

The easiest way to find the loops in some alignment between the crib and the ciphertext is to construct what Turing called a *menu*, which is simply a diagram showing the connections that appear between the letters. In the current example, index position 0 links **K** to **V**. The menu therefore contains the following pairing:

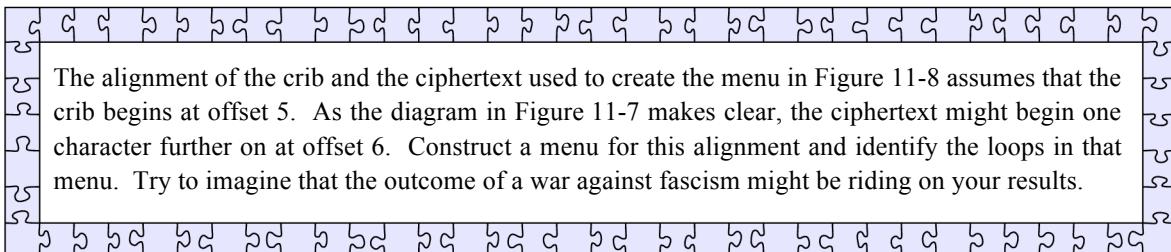
$$\mathbf{K} \xrightarrow{0} \mathbf{V}$$

The complete menu for this crib-ciphertext alignment appears in Figure 11-8.

FIGURE 11-8 Menu formed by recording the letter pairings between the crib and the ciphertext



Once you have completed the diagram, the loops jump out visually. In the menu shown in Figure 11-8, there is one loop of length 2 (**E→R→E**), two loops of length 3 (**F→I→E→F** and **N→B→P→N**), two loops of length 4 (**E→R→S→W→E** and **R→J→N→S→R**), and one loop of length 6 (**E→R→J→N→S→W→E**).



The discovery of these loops gave the Bletchley team the breakthrough they needed to crack the Enigma code. The fact that the loop pattern is independent of the steckerboard means that you can deduce the rotor patterns simply by running through all the possible settings. To speed up that process, the Bletchley team built an electromechanical computing device called the **Bombe**, which simulated the operation of the Enigma machine. The Bombe was programmed to search for feasible rotor positions given a particular set of loops in the encoding of a suspected plaintext into its encrypted version. At each state of the machine's operation, it would assume that the current setting of the rotors was correct. If that assumption led to a contradiction, the Bombe would quickly move on to the next cycle. Although the running time depended on the number of loops detected in the crib-ciphertext pairing, the Bombe was typically able to search through all possible rotor combinations in less than an hour.

Breaking the code for one intercepted message did not give Bletchley Park the ability to read all the Enigma transmissions for that day. The Germans were clever enough to realize that it would be foolish to transmit a large number of messages with the same encryption key. What they did instead was to have the Enigma operator come up with his own encryption key and then encipher that key—using the settings from the codebook—before sending the actual message. For example, if the rotor setting for the day was **JLY**, the operator would initialize the machine to that setting and then transmit a new message key of his own devising. The operator would then reset the machine so that it used the new key to encode the rest of the message. The receiver would simply reverse the process. After setting the machine to the settings from the codebook, the receiver would then use the characters from the beginning of the message to reset the machine appropriately.

All too often, these operator-chosen keys were too easy to predict. Lazy operators might choose a key that was easy to type **AAA**. Others might use names of friends and family such as **PIA**. If the Bletchley codebreakers could guess the

message key, the decryption operation became much easier. More damaging to German security, however, was the fact that message keys—at least in the early days of the Enigma—were transmitted twice in succession to make sure they got through. This procedure left an enormous hole in the German encryption strategy. If codebreakers knew that the first and fourth, second and fifth, and third and sixth letters in a message always represented the same plaintext letter, they could use this knowledge to guess the rotor settings. The Polish decryption strategy used this method and therefore did not rely on being able to find a crib.

Breaking a single message, however, represented a real victory because doing so typically allowed the codebreakers to determine the setting of the steckerboard. After setting up an Enigma machine so that it matched the setting of the message key, encoding the appropriate section of the ciphertext would yield a string of characters that was a simple letter-substitution cipher of the crib, which is generally easy to solve. Knowing both the rotor order and the steckerboard wiring—neither of which change from message to message over the course of a single day—made it easy to decode other messages for that day because there were only the 17,576 rotor settings to check.

As Winston Churchill’s report to King George makes clear, the cryptographic work at Bletchley Park helped the Allied cause enormously. Cryptography is still important as a field of study today. In the 1970s, computer scientists developed an entirely new to encryption called *public-key cryptography* that has revolutionized electronic communication, which you will have the chance to explore in Chapter 12.