Chapter 2 Classical Cryptosystems

Methods of making messages unintelligible to adversaries have been important throughout history. In this chapter we shall cover some of the older cryptosystems that were primarily used before the advent of the computer. These cryptosystems are too weak to be of much use today, especially with computers at our disposal, but they give good illustrations of several of the important ideas of cryptology.

First, for these simple cryptosystems, we make some conventions.

- *plaintext* will be written in lowercase letters and *CIPHERTEXT* will be written in capital letters (except in the computer problems).
- The letters of the alphabet are assigned numbers as follows:

  $$\begin{array}{cccccccccccccccc} a & b & c & d & e & f & g & h & i & j & k & l & m & n & o & p \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{array}$$

  $$\begin{array}{cccccccccc} q & r & s & t & u & v & w & x & y & z \\ 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 \end{array}$$

  Note that we start with $a = 0$, so z is letter number 25. Because many people are accustomed to a being 1 and z being 26, the present convention can be annoying, but it is standard for the elementary cryptosystems that we'll consider.
- Spaces and punctuation are omitted. This is even more annoying, but it is almost always possible to replace the spaces in the plaintext after decrypting. If spaces were left in, there would be two choices. They could be left as spaces; but this yields so much information on the structure of the message that decryption becomes easier. Or they could be encrypted; but then they would dominate frequency counts (unless the message averages at least eight letters per word), again simplifying decryption.

Note: In this chapter, we'll be using some concepts from number theory, especially modular arithmetic. If you are not familiar with congruences, you should read the first three sections of Chapter 3 before proceeding.

2.1 Shift Ciphers

One of the earliest cryptosystems is often attributed to Julius Caesar. Suppose he wanted to send a plaintext such as

$$\textit{gaul is divided into three parts}$$

but he didn't want Brutus to read it. He shifted each letter backwards by three places, so d became A, e became B, f became C, etc. The beginning of the alphabet wrapped around to the end, so a became X, b became Y, and c became Z. The ciphertext was then

$$DXRIFPAFSFABAFKQLQEOBBMXOQP.$$

Decryption was accomplished by shifting FORWARD by three spaces (and trying to figure out how to put the spaces back in).

We now give the general situation. *If you are not familiar with modular arithmetic, read the first few pages of Chapter 3 before continuing*.

Label the letters as integers from 0 to 25. The key is an integer κ with $0 \le \kappa \le 25$. The encryption process is

$$x \mapsto x + \kappa \pmod{26}.$$

Decryption is $x \mapsto x - \kappa \pmod{26}$. For example, Caesar used $\kappa = 23 \equiv -3$.

Let's see how the four types of attack work.

1. Ciphertext only: Eve has only the ciphertext. Her best strategy is an exhaustive search, since there are only 26 possible keys. See Example 1 in the Computer Appendices. If the message is longer than a few letters (we will make this more precise later when we discuss entropy), it is unlikely that there is more than one meaningful message that could be the plaintext. If you don't believe this, try to find some words of four or more letters that are

shifts of each other. Three such words are given in Exercises 1 and 2. Another possible attack, if the message is sufficiently long, is to do a frequency count for the various letters. The letter e occurs most frequently in most English texts. Suppose the letter L appears most frequently in the ciphertext. Since $e = 4$ and $L = 11$, a reasonable guess is that $\kappa = 11 - 4 = 7$. However, for shift ciphers this method takes much longer than an exhaustive search, plus it requires many more letters in the message in order for it to work (anything short, such as this, might not contain a common symbol, thus changing statistical counts).

2. Known plaintext: If you know just one letter of the plaintext along with the corresponding letter of ciphertext, you can deduce the key. For example, if you know $t(= 19)$ encrypts to $D(= 3)$, then the key is $\kappa \equiv 3 - 19 \equiv -16 \equiv 10 \pmod{26}$.

3. Chosen plaintext: Choose the letter a as the plaintext. The ciphertext gives the key. For example, if the ciphertext is H, then the key is 7.

4. Chosen ciphertext: Choose the letter A as ciphertext. The plaintext is the negative of the key. For example, if the plaintext is h, the key is $-7 \equiv 19 \pmod{26}$.

2.2 Affine Ciphers

The shift ciphers may be generalized and slightly strengthened as follows. Choose two integers α and β, with $\gcd(\alpha, 26) = 1$, and consider the function (called an *affine function*)

$$x \mapsto \alpha x + \beta \pmod{26}.$$

For example, let $\alpha = 9$ and $\beta = 2$, so we are working with $9x + 2$. Take a plaintext letter such as $h(= 7)$. It is encrypted to $9 \cdot 7 + 2 \equiv 65 \equiv 13 \pmod{26}$, which is the letter N. Using the same function, we obtain

$$affine \mapsto \ CVVWPM.$$

How do we decrypt? If we were working with rational numbers rather than mod 26, we would start with $y = 9x + 2$ and solve: $x = \frac{1}{9}(y - 2)$. But $\frac{1}{9}$ needs to be reinterpreted when we work mod 26. Since $\gcd(9, 26) = 1$, there is a multiplicative inverse for $9 \pmod{26}$ (if this last sentence doesn't make sense to you, read Section 3.3 now). In fact, $9 \cdot 3 \equiv 1 \pmod{26}$, so 3 is the desired inverse and can be used in place of $\frac{1}{9}$. We therefore have

$$x \equiv 3(y - 2) \equiv 3y - 6 \equiv 3y + 20 \pmod{26}.$$

Let's try this. The letter $V(= 21)$ is mapped to $3 \cdot 21 + 20 \equiv 83 \equiv 5 \pmod{26}$, which is the letter f. Similarly, we see that the ciphertext $CVVWPM$ is decrypted back to *affine*. For more examples, see Examples 2 and 3 in the Computer Appendices.

Suppose we try to use the function $13x + 4$ as our encryption function. We obtain

$$input \mapsto \ ERRER.$$

If we alter the input, we obtain

$$alter \mapsto ERRER.$$

Clearly this function leads to errors. It is impossible to decrypt, since several plaintexts yield the same ciphertext. In particular, we note that encryption must be one-to-one, and this fails in the present case.

What goes wrong in this example? If we solve $y = 13x + 4$, we obtain $x = \frac{1}{13}(y - 4)$. But $\frac{1}{13}$ does not exist mod 26 since $\gcd(13, 26) = 13 \neq 1$. More generally, it can be shown that $\alpha x + \beta$ is a one-to-one function mod 26 if and only if $\gcd(\alpha, 26) = 1$. In this case, decryption uses $x \equiv \alpha^* y - \alpha^* \beta \pmod{26}$, where $\alpha\alpha^* \equiv 1 \pmod{26}$. So decryption is also accomplished by an affine function.

The key for this encryption method is the pair (α, β). There are 12 possible choices for α with $\gcd(\alpha, 26) = 1$ and there are 26 choices for β (since we are working mod 26, we only need to consider α and β between 0 and 25). Therefore, there are $12 \cdot 26 = 312$ choices for the key.

Let's look at the possible attacks.

1. Ciphertext only: An exhaustive search through all 312 keys would take longer than the corresponding search in the case of the shift cipher; however, it would be very easy to do on a computer. When all possibilities for the key are tried, a fairly short ciphertext, say around 20 characters, will probably correspond to only one meaningful plaintext, thus allowing the determination of the key. It would also be possible to use frequency counts, though this would require much longer texts.

2. Known plaintext: With a little luck, knowing two letters of the plaintext and the corresponding letters of the ciphertext suffices to find the key. In any case, the number of possibilities for the key is greatly reduced and a few more letters should yield the key.

   For example, suppose the plaintext starts with *if* and the corresponding ciphertext is PQ. In numbers, this means that 8 $(= i)$ maps to 15 $(= P)$ and 5 maps to 16. Therefore, we have the equations

$$8\alpha + \beta \equiv 15 \text{ and } 5\alpha + \beta \equiv 16 \pmod{26}.$$

Subtracting yields $3\alpha \equiv -1 \equiv 25 \pmod{26}$, which has the unique solution $\alpha = 17$. Using the first equation, we find $8 \cdot 17 + \beta \equiv 15 \pmod{26}$, which yields $\beta = 9$.

Suppose instead that the plaintext *go* corresponds to the ciphertext TH. We obtain the equations

$$6\alpha + \beta \equiv 19 \text{ and } 14\alpha + \beta \equiv 7 \pmod{26}.$$

Subtracting yields $-8\alpha \equiv 12 \pmod{26}$. Since $\gcd(-8, 26) = 2$, this has two solutions: $\alpha = 5, 18$. The corresponding values of β are both 15 (this is not a coincidence; it will always happen this way when the coefficients of α in the equations are even). So we have two candidates for the key: $(5, 15)$ and $(18, 15)$. However, $\gcd(18, 26) \neq 1$ so the second is ruled out. Therefore, the key is $(5, 15)$.

The preceding procedure works unless the gcd we get is 13 (or 26). In this case, use another letter of the message, if available.

If we know only one letter of plaintext, we still get a relation between α and β. For example, if we only know that g in plaintext corresponds to T in ciphertext, then we have $6\alpha + \beta \equiv 19 \pmod{26}$. There are 12 possibilities for α and each gives one corresponding β. Therefore, an exhaustive search through the 12 keys should yield the correct key.

3. Chosen plaintext: Choose ab as the plaintext. The first character of the ciphertext will be $\alpha \cdot 0 + \beta = \beta$, and the second will be $\alpha + \beta$. Therefore, we can find the key.

4. Chosen ciphertext: Choose AB as the ciphertext. This yields the decryption function of the form $x = \alpha_1 y + \beta_1$. We could solve for y and obtain the encryption key. But why bother? We have the decryption function, which is what we want.

2.3 The Vigenère Cipher

A variation of the shift cipher was invented back in the sixteenth century. It is often attributed to Vigenère, though Vigenère's encryption methods were more sophisticated. Well into the twentieth century, this cryptosystem was thought by many to be secure, though Babbage and Kasiski had shown how to attack it during the nineteenth century. In the 1920s, Friedman developed additional methods for breaking this and related ciphers.

The key for the encryption is a vector, chosen as follows. First choose a key length, for example, 6. Then choose a vector of this size whose entries are integers from 0 to 25, for example $k = (21, 4, 2, 19, 14, 17)$. Often the key corresponds to a word that is easily remembered. In our case, the word is *vector*. The security of the system depends on the fact that neither the keyword nor its length is known.

To encrypt the message using the k in our example, we take first the letter of the plaintext and shift by 21. Then shift the second letter by 4, the third by 2, and so on. Once we get to the end of the key, we start back at its first entry, so the seventh letter is shifted by 21, the eighth letter by 4, etc. Here is a diagram of the encryption process.

$$\begin{array}{rcccccccccccccccc} \text{(plaintext)} & h & e & r & e & i & s & h & o & w & i & t & w & o & r & k & s \\ \text{(key)} & 21 & 4 & 2 & 19 & 14 & 17 & 21 & 4 & 2 & 19 & 14 & 17 & 21 & 4 & 2 & 19 \\ \text{(ciphertext)} & C & I & T & X & W & J & C & S & Y & B & H & N & J & V & M & L \end{array}$$

A known plaintext attack will succeed if enough characters are known since the key is simply obtained by subtracting the plaintext from the ciphertext mod 26. A chosen plaintext attack using the plaintext $aaaaa\ldots$

will yield the key immediately, while a chosen ciphertext attack with $AAAAA\ldots$ yields the negative of the key. But suppose you have only the ciphertext. It was long thought that the method was secure against a ciphertext-only attack. However, it is easy to find the key in this case, too.

The cryptanalysis uses the fact that in most English texts the frequencies of letters are not equal. For example, e occurs much more frequently than x. These frequencies have been tabulated in [Beker-Piper] and are provided in Table 2.1.

Table 2.1 Frequencies of Letters in English

a	b	c	d	e	f	g	h	i	j
.082	.015	.028	.043	.127	.022	.020	.061	.070	.002
k	l	m	n	o	p	q	r	s	t
.008	.040	.024	.067	.075	.019	.001	.060	.063	.091
u	v	w	x	y	z				
.028	.010	.023	.001	.020	.001				

Table 2.1 Full Alternative Text

Of course, variations can occur, though usually it takes a certain amount of effort to produce them. There is a book Gadsby by Ernest Vincent Wright that does not contain the letter e. Even more impressive is the book La Disparition by George Perec, written in French, which also does not have a single e (not only are there the usual problems with verbs, etc., but almost all feminine nouns and adjectives must be avoided). There is an English translation by Gilbert Adair, A Void, which also does not contain e. But generally we can assume that the above gives a rough estimate of what usually happens, as long as we have several hundred characters of text.

If we had a simple shift cipher, then the letter e, for example, would always appear as a certain ciphertext letter, which would then have the same frequency as that of e in the original text. Therefore, a frequency analysis would probably reveal the key. However, in the preceding example of a Vigenère cipher, the letter e appears as both I and X. If we had used a longer plaintext, e would probably have been encrypted as each of Z, I, G, X, S, and V, corresponding to the shifts 21, 4, 2, 19, 14, 17. But the occurrences of Z in a ciphertext might not come only from e. The letter v is also encrypted to Z when its position in the text is such that it is shifted by 4. Similarly, x, g, l, and i can contribute Z to the ciphertext, so the frequency of Z is a combination of that of e, v, x, g, l, and i from the plaintext. Therefore, it appears to be much more difficult to deduce anything from a frequency count. In fact, the frequency counts are usually smoothed out and are much closer to 1/26 for each letter of ciphertext. At least, they should be much closer than the original distribution for English letters.

Here is a more substantial example. This example is also treated in Example 4 in the Computer Appendices. The ciphertext is the following:

VVHQWVVRHMUSGJGTHKIHTSSEJCHLSFCBGVWC
RLRYQTFSVGAHW
KCUHWAUGLQHNSLRLJSHBLTSPISPRDXLJSVEEG
HLQWKASSKUWE
PWQTWVSPGOELKCQYFNSVWLJSNIQKGNRGYBWL
WGOVIOKHKAZKQ
KXZGYHCECMEIUJOQKWFWVEFQHKIJRCLRLKBIE
NQFRJLJSDHGR
HLSFQTWLAUQRHWDMWLGUSGIKKFLRYVCWVSP

GPMLKASSJVOQXE
GGVEYGGZMLJCXXLJSVPAIVWIKVRDRYGFRJLJSLV
EGGVEYGGEI
APUUISFPBTGNWWMUCZRVTWGLRWUGUMNCZVI
LE

The frequencies are as follows:

A	B	C	D	E	F	G	H	I	J	K	L	M
8	5	12	4	15	10	27	16	13	14	17	25	7
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
7	5	9	14	17	24	8	12	22	22	5	8	5

Note that there is no letter whose frequency is significantly larger than the others. As discussed previously, this is because e, for example, gets spread among several letters during the encryption process.

How do we decrypt the message? There are two steps: finding the key length and finding the key. In the following, we'll first show how to find the key length and then give one way to find the key. After an explanation of why the method for finding the key works, we give an alternative way to find the key.

2.3.1 Finding the Key Length

Write the ciphertext on a long strip of paper, and again on another long strip. Put one strip above the other, but displaced by a certain number of places (the potential key length). For example, for a displacement of two we have the following:

		V	V	H	Q	W	V	V	R	H	M	U	S	G	J	G
V	V	H	Q	W	V	V	R	H	M	U	S	G	J	G	T	H
														*		
T	H	K	I	H	T	S	S	E	J	C	H	L	S	F	C	B
K	I	H	T	S	S	E	J	C	H	L	S	F	C	B	G	V
				*												
G	V	W	C	R	L	R	Y	Q	T	F	S	V	G	A	H	· · ·
W	C	R	L	R	Y	Q	T	F	S	V	G	A	H	W	K	· · ·
				*												

Mark a $*$ each time a letter and the one below it are the same, and count the total number of coincidences. In the text just listed, we have two coincidences so far. If we had continued for the entire ciphertext, we would have counted 14 of them. If we do this for different displacements, we obtain the following data:

displacement:	1	2	3	4	5	6
coincidences:	14	14	16	14	24	12

We have the most coincidences for a shift of 5. As we explain later, this is the best guess for the length of the key. This method works very quickly, even without a computer, and usually yields the key length.

2.3.2 Finding the Key: First Method

Now suppose we have determined the key length to be 5, as in our example. Look at the 1st, 6th, 11th, ... letters and see which letter occurs most frequently. We obtain

A	B	C	D	E	F	G	H	I	J	K	L	M
0	0	7	1	1	2	9	0	1	8	8	0	0
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
3	0	4	5	2	0	3	6	5	1	0	1	0

The most frequent is G, though J, K, C are close behind. However, $J = e$ would mean a shift of 5, hence $C = x$. But this would yield an unusually high frequency for x in the plaintext. Similarly, $K = e$ would mean $P = j$ and $Q = k$, both of which have too high frequencies. Finally, $C = e$ would require $V = x$, which is unlikely to be the case. Therefore, we decide that $G = e$ and the first element of the key is $2 = c$.

We now look at the 2nd, 7th, 12th, ... letters. We find that G occurs 10 times and S occurs 12 times, and the other letters are far behind. If $G = e$, then $S = q$, which should not occur 12 times in the plaintext. Therefore, $S = e$ and the second element of the key is $14 = o$.

Now look at the 3rd, 8th, 13th, ... letters. The frequencies are

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	0	3	3	1	3	5	1	0	4	10	0
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	1	2	3	5	3	0	2	8	7	1	0	1

The initial guess that $L = e$ runs into problems; for example, $R = k$ and $E = x$ have too high frequency and $A = t$ has too low. Similarly, $V = e$ and $W = e$ do not seem likely. The best choice is $H = e$ and therefore the third key element is $3 = d$.

The 4th, 9th, 14th, ... letters yield $4 = e$ as the fourth element of the key. Finally, the 5th, 10th, 15th, ... letters yield $18 = s$ as the final key element. Our guess for the key is therefore

$$\{2, 14, 3, 4, 18\} = \{c, o, d, e, s\}.$$

As we saw in the case of the 3rd, 8th, 13th, ... letters (this also happened in the 5th, 10th, 15th, ... case), if we take every fifth letter we have a much smaller sample of letters on which we are doing a frequency count. Another letter can overtake e in a short sample. But it is probable that most of the high-frequency letters appear with high frequencies, and most of the low-frequency ones appear with low frequencies. As in the present case, this is usually sufficient to identify the corresponding entry in the key.

Once a potential key is found, test it by using it to decrypt. It should be easy to tell whether it is correct.

In our example, the key is conjectured to be $(2, 14, 3, 4, 18)$. If we decrypt the ciphertext using this key, we obtain

themethodusedforthepreparationandreadingofcodemess
agesis

simpleintheextremeandatthesametimeimpossibleoftransl
atio

nunlessthekeyisknowntheeasewithwhichthekeymaybech
angedis

anotherpointinfavoroftheadoptionofthiscodebythosedesi
rin

gtotransmitimportantmessageswithouttheslightestdange
roft

heirmessagesbeingreadbypoliticalorbusinessrivalsetc

This passage is taken from a short article in Scientific American, Supplement LXXXIII (January 27, 1917), page 61. A short explanation of the Vigenère cipher is given, and the preceding passage expresses an opinion as to its security.

Before proceeding to a second method for finding the key, we give an explanation of why the procedure given earlier finds the key length. In order to avoid confusion, we note that when we use the word "shift" for a letter, we are referring to what happens during the Vigenère encryption process.

We also will be shifting elements in vectors. However, when we slide one strip of paper to the right or left relative to the other strip, we use the word "displacement."

Put the frequencies of English letters into a vector:

$$\mathbf{A}_0 = (.082,\ .015,\ .028,\ \ldots,\ .020,\ .001).$$

Let $\mathbf{A}_i$ be the result of shifting $\mathbf{A}_0$ by i spaces to the right. For example,

$$\mathbf{A}_2 = (.020,\ .001,\ .082,\ .015,\ \ldots).$$

The dot product of $\mathbf{A}_0$ with itself is

$$\mathbf{A}_0 \cdot \mathbf{A}_0 = (.082)^2 + (.015)^2 + \cdots = .066.$$

Of course, $\mathbf{A}_i \cdot \mathbf{A}_i$ is also equal to .066 since we get the same sum of products, starting with a different term. However, the dot products of $\mathbf{A}_i \cdot \mathbf{A}_j$ are much lower when $i \neq j$, ranging from .031 to .045:

$\lvert i-j \rvert$	0	1	2	3	4	5	6
$\mathbf{A}_i \cdot \mathbf{A}_j$	.066	.039	.032	.034	.044	.033	.036
	7	8	9	10	11	12	13
	.039	.034	.034	.038	.045	.039	.042

The dot product depends only on $|i-j|$. This can be seen as follows. The entries in the vectors are the same as those in $\mathbf{A}_0$, but shifted. In the dot product, the ith entry of $\mathbf{A}_0$ is multiplied by the jth entry, the $(i+1)$st times the $(j+1)$st, etc. So each element is multiplied by the element $j-i$ positions removed from it. Therefore, the dot product depends only on the difference $i-j$. However, by reversing the roles of i and j, and noting that $\mathbf{A}_i \cdot \mathbf{A}_j = \mathbf{A}_j \cdot \mathbf{A}_i$, we see that $i-j$ and $j-i$ give the same dot products, so the dot product only depends on $|i-j|$. In the preceding table, we only needed to compute up to $|i-j| = 13$. For example, $i-j=17$ corresponds to a shift by 17 in one direction, or 9 in the other direction, so $i-j=9$ will give the same dot product.

The reason $\mathbf{A}_0 \cdot \mathbf{A}_0$ is higher than the other dot products is that the large numbers in the vectors are paired with large numbers and the small ones are paired with small. In the other dot products, the large numbers are paired somewhat randomly with other numbers. This lessens their effect. For another reason that $\mathbf{A}_0 \cdot \mathbf{A}_0$ is higher than the other dot products, see Exercise 23.

Let's assume that the distribution of letters in the plaintext closely matches that of English, as expressed by the vector $\mathbf{A}_0$ above. Look at a random letter in the top strip of ciphertext. It corresponds to a random letter of English shifted by some amount i (corresponding to an element of the key). The letter below it corresponds to a random letter of English shifted by some amount j.

For concreteness, let's suppose that $i = 0$ and $j = 2$. The probability that the letter in the 50th position, for example, is A is given by the first entry in $\mathbf{A}_0$, namely .082. The letter directly below, on the second strip, has been shifted from the original plaintext by $j = 2$ positions. If this ciphertext letter is A, then the corresponding plaintext letter was y, which occurs in the plaintext with probability .020. Note that .020 is the first entry of the vector $\mathbf{A}_2$. The probability that the letter in the 50th position on the first strip and the letter directly below it are both the letter A is $(.082)(.020)$. Similarly, the probability that both letters are B is $(.015)(.001)$. Working all the way through Z, we see that the probability that the two letters are the same is

$$(.082)(.020) + (.015)(.001) + \cdots + (.001)(.001) = \mathbf{A}_0 \cdot \mathbf{A}_2.$$

In general, when the encryption shifts are i and j, the probability that the two letters are the same is $\mathbf{A}_i \cdot \mathbf{A}_j$. When $i \neq j$, this is approximately 0.038, but if $i = j$, then the dot product is 0.066.

We are in the situation where $i = j$ exactly when the letters lying one above the other have been shifted by the same amount during the encryption process, namely when the top strip is displaced by an amount equal to the key length (or a multiple of the key length). Therefore we expect more coincidences in this case.

For a displacement of 5 in the preceding ciphertext, we had 326 comparisons and 24 coincidences. By the reasoning just given, we should expect approximately $326 \times 0.066 = 21.5$ coincidences, which is close to the actual value.

2.3.3 Finding the Key: Second Method

Using the preceding ideas, we give another method for determining the key. It seems to work somewhat better than the first method on short samples, though it requires a little more calculation.

We'll continue to work with the preceding example. To find the first element of the key, count the occurrences of the letters in the 1st, 6th, 11th, ... positions, as before, and put them in a vector:

$$\mathbf{V} = (0, 0, 7, 1, 1, 2, 9, 0, 1, 8, 8, 0, 0, 3, 0, 4, 5, 2, 0, 3, 6, 5, 1, 0, 1, 0)$$

(the first entry gives the number of occurrences of A, the second gives the number of occurrences of B, etc.). If we divide by 67, which is the total number of letters counted, we obtain a vector

$$\mathbf{W} = (0, 0, .1045, .0149, .0149, .0299, \ldots, .0149, 0).$$

Let's think about where this vector comes from. Since we know the key length is 5, the 1st, 6th, 11th, ... letters in the ciphertext were all shifted by the same amount (as we'll see shortly, they were all shifted by 2). Therefore, they represent a random sample of English letters, all shifted by the same amount. Their frequencies, which are given by the vector $\mathbf{W}$, should approximate the vector $\mathbf{A}_i$, where i is the shift caused by the first element of the key.

The problem now is to determine i. Recall that $\mathbf{A}_i \cdot \mathbf{A}_j$ is largest when $i = j$, and that $\mathbf{W}$ approximates $\mathbf{A}_i$. If we compute $\mathbf{W} \cdot \mathbf{A}_j$ for $0 \leq j \leq 25$, the maximum value should occur when $j = i$. Here are the dot products:

$$\begin{gathered} .0250, .0391, .0713, .0388, .0275, .0380, .0512, .0301, .0325, \\ .0430, .0338, .0299, .0343, .0446, .0356, .0402, .0434, .0502, \\ .0392, .0296, .0326, .0392, .0366, .0316, .0488, .0349 \end{gathered}$$

The largest value is the third, namely .0713, which equals $\mathbf{W} \cdot \mathbf{A}_2$. Therefore, we guess that the first shift is 2, which corresponds to the key letter c.

Let's use the same method to find the third element of the key. We calculate a new vector $\mathbf{W}$, using the frequencies for the 3rd, 8th, 13th, ... letters that we tabulated previously:

$$\mathbf{W} = (0, .0152, 0, .0454, .0454, .0152, \ldots, 0, .0152).$$

The dot products $\mathbf{W} \cdot \mathbf{A}_i$ for $0 \leq i \leq 25$ are

$$\begin{gathered} .0372, .0267, .0395, .0624, .04741, .0279, .0319, .0504, .0378, \\ .0351, .0367, .0395, .0264, .0415, .0427, .0362, .0322, .0457, \\ .0526, .0397, .0322, .0299, .0364, .0372, .0352, .0406 \end{gathered}$$

The largest of these values is the fourth, namely .0624, which equals $\mathbf{W} \cdot \mathbf{A}_3$. Therefore, the best guess is that the first shift is 3, which corresponds to the key letter d. The other three elements of the key can be found similarly, again yielding $c, \ o, \ d, \ e, \ s$ as the key.

Notice that the largest dot product was significantly larger than the others in both cases, so we didn't have to make several guesses to find the correct one. In this way, the present method is superior to the first method presented; however, the first method is much easier to do by hand.

Why is the present method more accurate than the first one? To obtain the largest dot product, several of the larger values in $\mathbf{W}$ had to match with the larger values in an $\mathbf{A}_i$. In the earlier method, we tried to match only the e, then looked at whether the choices for other letters were reasonable. The present method does this all in one step.

To summarize, here is the method for finding the key. Assume we already have determined that the key length is n.

For $i = 1$ to n, do the following:

1. Compute the frequencies of the letters in positions i mod n, and form the vector $\mathbf{W}$.

2. For $j = 0$ to 25, compute $\mathbf{W} \cdot \mathbf{A}_j$.

3. Let $k_i = j_0$ give the maximum value of $\mathbf{W} \cdot \mathbf{A}_j$.

The key is probably $\{k_1, \ \ldots, \ k_n\}$.

2.4 Substitution Ciphers

One of the more popular cryptosystems is the substitution cipher. It is commonly used in the puzzle section of the weekend newspapers, for example. The principle is simple: Each letter in the alphabet is replaced by another (or possibly the same) letter. More precisely, a permutation of the alphabet is chosen and applied to the plaintext. In the puzzle pages, the spaces between the words are usually preserved, which is a big advantage to the solver, since knowledge of word structure becomes very useful. However, to increase security it is better to omit the spaces.

The shift and affine ciphers are examples of substitution ciphers. The Vigenère cipher (see Section 2.3) is not, since it permutes blocks of letters rather than one letter at a time.

Everyone "knows" that substitution ciphers can be broken by frequency counts. However, the process is more complicated than one might expect.

Consider the following example. Thomas Jefferson has a potentially treasonous message that he wants to send to Ben Franklin. Clearly he does not want the British to read the text if they intercept it, so he encrypts it using a substitution cipher. Fortunately, Ben Franklin knows the permutation being used, so he can simply reverse the permutation to obtain the original message (of course, Franklin was quite clever, so perhaps he could have decrypted it without previously knowing the key).

Now suppose we are working for the Government Code and Cypher School in England back in 1776 and are given the following intercepted message to decrypt.

LWNSOZBNWVWBAYBNVBSQWVWOHWDIZWRBBN
PBPOOUWRPAWXAW

PBWZWMYPOBNPBBNWJPAWWRZSLWZQJBNWIAX
AWPBSALIBNXWA

BPIRYRPOIWRPQOWAIENBVBNPBPUSREBNWVWPA
WOIHWOIQWAB

JPRZBNWFYAVYIBSHNPFFIRWVVBNPBBSVWXYAW
BNWVWAIENBV

ESDWARUWRBVPAWIRVBIBYBWZPUSREUWRZWAI
DIREBNWIATYV

BFSLWAVHASUBNWXSRVWRBSHBNWESDWARWZB
NPBLNWRWDWAPR

JHSAUSHESDWARUWRBQWXSUWVZWVBAYXBIDW
SHBNWVWWRZVIB

IVBNWAIENBSHBNWFWSFOWBSPOBWASABSPQSOI
VNIBPRZBSIR

VBIBYBWRWLESDWARUWRBOPJIREIBVHSYRZPBIS
RSRVYXNFAI

RXIFOWVPRZSAEPRIKIREIBVFSLWAVIRVYXNHSAU
PVBSVWWUU

SVBOICWOJBSWHHWXBBNWIAVPHWBJPRZNPFFI
RWVV

A frequency count yields the following (there are 520 letters in the text):

W	B	R	S	I	V	A	P	N	O	...
76	64	39	36	36	35	34	32	30	16	...

The approximate frequencies of letters in English were given in Section 2.3. We repeat some of the data here in Table 2.2. This allows us to guess with reasonable confidence that W represents e (though B is another possibility). But what about the other letters? We can guess that B, R, S, I, V, A, P, N, with maybe an exception or two, are probably the same as t, a, o, i, n, s, h, r in some order. But a simple frequency count is not enough to decide which is which. What we need to do now is look at digrams, or pairs of letters. We organize our results in Table 2.3 (we only use the most frequent letters here, though it would be better to include all).

Table 2.2 Frequencies of Most Common Letters in English

e	t	a	o	i	n	s	h	r
.127	.091	.082	.075	.070	.067	.063	.061	.060

Table 2.2 Full Alternative Text

Table 2.3 Counting Digrams

	W	B	R	S	I	V	A	P	N
W	3	4	12	2	4	10	14	3	1
B	4	4	0	11	5	5	2	4	20
R	5	5	0	1	1	5	0	3	0
S	1	0	5	0	1	3	5	2	0
I	1	8	10	1	0	2	3	0	0
V	8	10	0	0	2	2	0	3	1
A	7	3	4	2	5	4	0	1	0
P	0	8	6	0	1	1	4	0	0
N	14	3	0	1	1	1	0	7	0

Table 2.3 Full Alternative Text

The entry 1 in the W row and N column means that the combination WN appears 1 time in the text. The entry 14 in the N row and W column means that NW appears 14 times.

We have already decided that $W = e$, but if we had extended the table to include low-frequency letters, we would see that W contacts many of these letters, too, which is another characteristic of e. This helps to confirm our guess.

The vowels a, i, o tend to avoid each other. If we look at the R row, we see that R does not precede S, I, A, N very often. But a look at the R column shows that R follows S, I, A fairly often. So we suspect that R is not one of a, i, o. V and N are out because they would require a, i, or o to precede $W = e$ quite often, which is unlikely. Continuing, we see that the most likely possibilities for a, i, o are S, I, P in some order.

The letter n has the property that around 80% of the letters that precede it are vowels. Since we already have identified W, S, I, P as vowels, we see that R and A are the most likely candidates. We'll have to wait to see which is correct.

The letter h often appears before e and rarely after it. This tells us that $N = h$.

The most common digram is th. Therefore, $B = t$.

Among the frequent letters, r and s remain, and they should equal V and one of A, R. Since r pairs more with vowels and s pairs more with consonants, we see that V must be s and r is represented by either A or R.

The combination rn should appear more than nr, and AR is more frequent than RA, so our guess is that $A = r$ and $R = n$.

We can continue the analysis and determine that $S = o$ (note that to is much more common than ot), $I = i$, and $P = a$ are the most likely choices. We have therefore determined reasonable guesses for 382 of the 520 characters in the text:

L	W	N	S	O	Z	B	N	W	V	W	B	A	Y	B	N	V	B	S	
	e	h	o			t	h	e	s	e	t	r		t	h	s	t	o	
Q	W	V	W	O	H	W	D	I	Z	W	R	B	B	N	P	B	P	$\cdots$	
	e	s	e			e		i		e	n	t	t	h	a	t	a	$\cdots$	

At this point, knowledge of the language, middle-level frequencies (l, d, $\ldots$), and educated guesses can be used to fill in the remaining letters. For example, in the first line a good guess is that $Y = u$ since then the word $truths$ appears. Of course, there is a lot of guesswork, and various hypotheses need to be tested until one works.

Since the preceding should give the spirit of the method, we skip the remaining details. The decrypted message, with spaces (but not punctuation) added, is as follows (the text is from the middle of the Declaration of Independence):

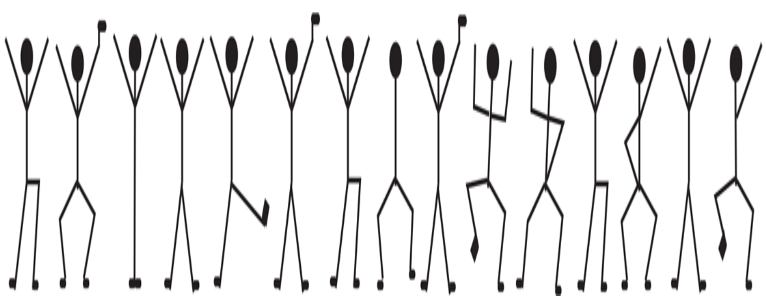
we hold these truths to be self evident that all men are created equal that they are endowed by their creator with certain unalienable rights that among these are life liberty and the pursuit of happiness that to secure these rights governments are instituted among men deriving their just powers from the consent of the governed that whenever any form of government becomes destructive of these ends it is the right of the people to alter or to abolish it and to institute new government laying its foundation on such principles and organizing its powers in such form as to seem most likely to effect their safety and happiness

2.5 Sherlock Holmes

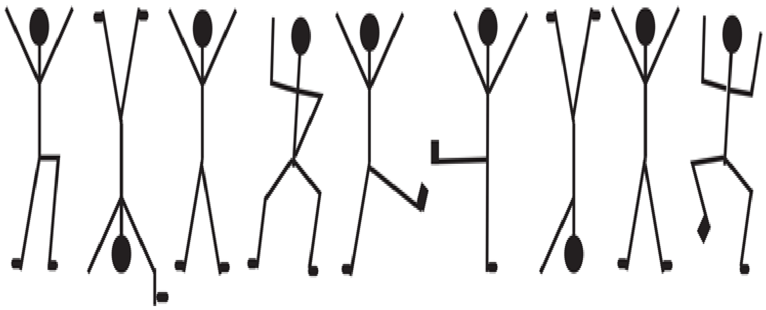
Cryptography has appeared in many places in literature, for example, in the works of Edgar Allen Poe (The Gold Bug), William Thackeray (The History of Henry Esmond), Jules Verne (Voyage to the Center of the Earth), and Agatha Christie (The Four Suspects).

Here we give a summary of an enjoyable tale by Arthur Conan Doyle, in which Sherlock Holmes displays his usual cleverness, this time by breaking a cipher system. We cannot do the story justice here, so we urge the reader to read The Adventure of the Dancing Men in its entirety. The following is a cryptic, and cryptographic, summary of the plot.

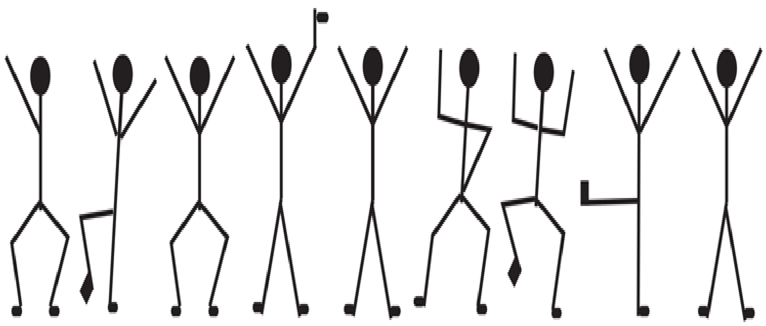
Mr. Hilton Cubitt, who has recently married the former Elsie Patrick, mails Sherlock Holmes a letter. In it is a piece of paper with dancing stick figures that he found in his garden at Riding Thorpe Manor:



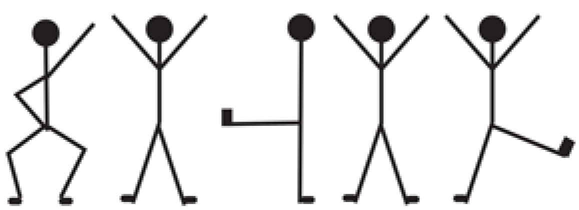
Two weeks later, Cubitt finds another series of figures written in chalk on his toolhouse door:



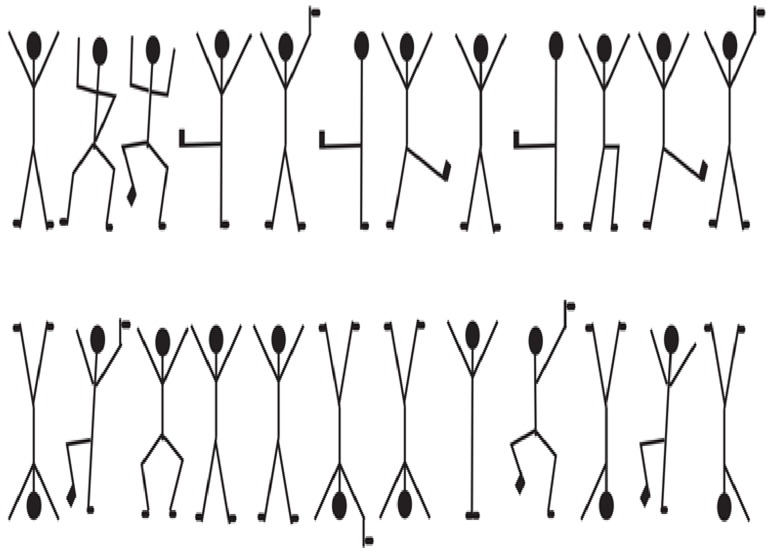
Two mornings later another sequence appears:



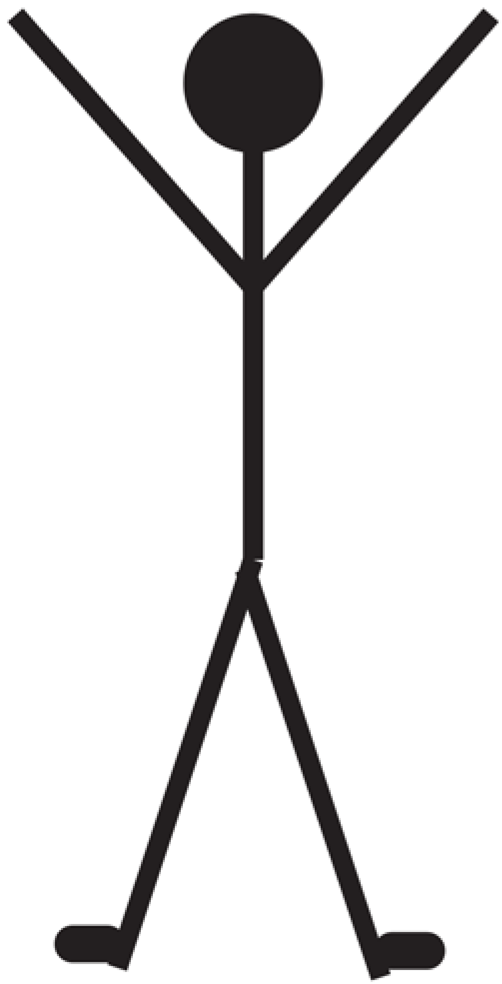
Three days later, another message appears:



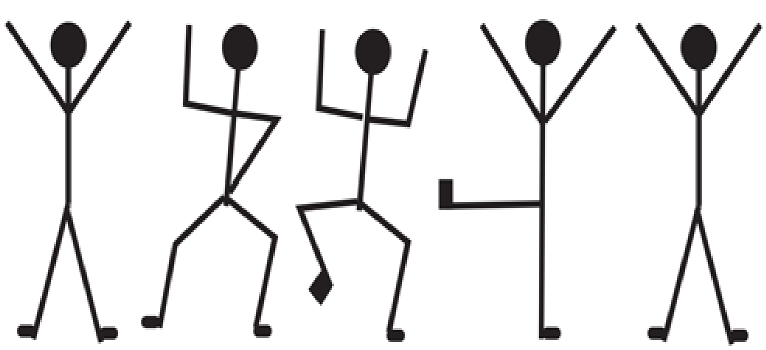
Cubitt gives copies of all of these to Holmes, who spends the next two days making many calculations. Suddenly, Holmes jumps from his chair, clearly having made a breakthrough. He quickly sends a long telegram to someone and then waits, telling Watson that they will probably be going to visit Cubitt the next day. But two days pass with no reply to the telegram, and then a letter arrives from Cubitt with yet another message:



Holmes studies it and says they need to travel to Riding Thorpe Manor as soon as possible. A short time later, a reply to Holmes's telegram arrives, and Holmes indicates that the matter has become even more urgent. When Holmes and Watson arrive at Cubitt's house the next day, they find the police already there. Cubitt has been shot dead. His wife, Elsie, has also been shot and is in critical condition (although she survives). Holmes asks several questions and then has someone deliver a note to a Mr. Abe Slaney at nearby Elrige's Farm. Holmes then explains to Watson and the police how he decrypted the messages. First, he guessed that the flags on some of the figures indicated the ends of words. He then noticed that the most common figure was

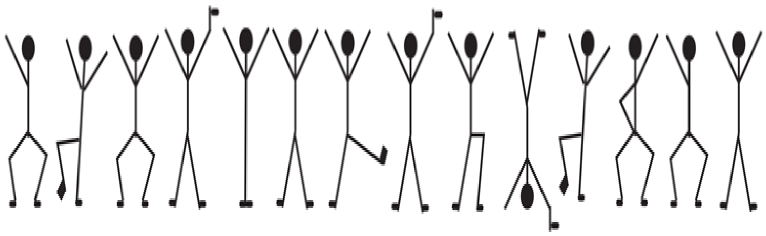


so it was likely *E*. This gave the fourth message as –*E*–*E*–. The possibilities *LEVER, NEVER, SEVER* came to mind, but since the message was probably a one word reply to a previous message, Holmes guessed it was *NEVER*. Next, Holmes observed that



had the form *E*– – –*E*, which could be *ELSIE*. The third message was therefore – – – *E* *ELSIE*. Holmes tried several combinations, finally settling on *COME ELSIE* as the only viable possibility. The first message therefore was $-M\ \ -ERE\ \ --E\ \ SL-NE-$. Holmes guessed that the first letter was A and the third letter as H, which gave the message as *AM HERE A–E SLANE–*. It was reasonable to complete this to *AM HERE ABE SLANEY*. The second message then was $A-\ \ ELRI-ES$. Of course, Holmes correctly guessed that this must be stating where Slaney was staying. The only letters that seemed reasonable completed the phrase to *AT ELRIGES*. It was after decrypting these two messages that Holmes sent a telegram to a friend at the New York Police Bureau, who sent back the reply that Abe Slaney was “the most dangerous crook in Chicago.” When the final message arrived, Holmes decrypted it to *ELSIE –RE–ARE TO MEET THY GO–*. Since he recognized the missing letters as P, P, D, respectively, Holmes became very concerned and that’s why he decided to make the trip to Riding Thorpe Manor.

When Holmes finishes this explanation, the police urge that they go to Elrige’s and arrest Slaney immediately. However, Holmes suggests that is unnecessary and that Slaney will arrive shortly. Sure enough, Slaney soon appears and is handcuffed by the police. While waiting to be taken away, he confesses to the shooting (it was somewhat in self-defense, he claims) and says that the writing was invented by Elsie Patrick’s father for use by his gang, the Joint, in Chicago. Slaney was engaged to be married to Elsie, but she escaped from the world of gangsters and fled to London. Slaney finally traced her location and sent the secret messages. But why did Slaney walk into the trap that Holmes set? Holmes shows the message he wrote:



From the letters already deduced, we see that this says *COME HERE AT ONCE*. Slaney was sure this message must have been from Elsie since he was certain no one outside of the Joint could write such messages. Therefore, he made the visit that led to his capture.

Comments

What Holmes did was solve a simple substitution cipher, though he did this with very little data. As with most such ciphers, both frequency analysis and a knowledge of the language are very useful. A little luck is nice, too, both in the form of lucky guesses and in the distribution of letters. Note how overwhelmingly E was the most common letter. In fact, it appeared 11 times among the 38 characters in the first four messages. This gave Holmes a good start. If Elsie had been Carol and Abe Slaney had been John Smith, the decryption would probably have been more difficult.

Authentication is an important issue in cryptography. If Eve breaks Alice's cryptosystem, then Eve can often masquerade as Alice in communications with Bob. Safeguards against this are important. The judges gave Abe Slaney many years to think about this issue.

The alert reader might have noticed that we cheated a little when decrypting the messages. The same symbol represents the V in *NEVER* and the *P*s in *PREPARE*. This is presumably due to a misprint and has occurred in every printed version of the work, starting with the story's first publication back in 1903. In the original text, the R in *NEVER* is written as the B in ABE, but this is corrected in later editions (however, in some later editions, the first C in the message Holmes wrote is given an extra arm and therefore looks like the M). If these mistakes had been in the text that Holmes was working with, he would have had a very difficult time decrypting and would have rightly concluded that the Joint needed to use error correction techniques in their transmissions. In fact, some type of error correction should be used in conjunction with almost every cryptographic protocol.

2.6 The Playfair and ADFGX Ciphers

The Playfair and ADFGX ciphers were used in World War I by the British and the Germans, respectively. By modern standards, they are fairly weak systems, but they took real effort to break at the time.

The Playfair system was invented around 1854 by Sir Charles Wheatstone, who named it after his friend, the Baron Playfair of St. Andrews, who worked to convince the government to use it. In addition to being used in World War I, it was used by the British forces in the Boer War.

The key is a word, for example, *playfair*. The repeated letters are removed, to obtain *playfir*, and the remaining letters are used to start a 5×5 matrix. The remaining spaces in the matrix are filled in with the remaining letters in the alphabet, with i and j being treated as one letter:

$$\begin{array}{ccccc} p & l & a & y & f \\ i & r & b & c & d \\ e & g & h & k & m \\ n & o & q & s & t \\ u & v & w & x & z \end{array}$$

Suppose the plaintext is *meet at the schoolhouse*. Remove spaces and divide the text into groups of two letters. If there is a doubled letter appearing as a group, insert an x and regroup. Add an extra x at the end to complete the last group, if necessary. Our plaintext becomes

$$me\,et\,at\,th\,es\,ch\,ox\,ol\,ho\,us\,ex.$$

Now use the matrix to encrypt each two-letter group by the following scheme:

- If the two letters are not in the same row or column, replace each letter by the letter that is in its row and is in the column of the other letter. For example, *et* becomes MN, since M is in the same row as e and the same column as t, and N is in the same row as t and the same column as e.
- If the two letters are in the same row, replace each letter with the letter immediately to its right, with the matrix wrapping around from the last column to the first. For example, *me* becomes EG.
- If the two letters are in the same column, replace each letter with the letter immediately below it, with the matrix wrapping around from the last row to the first. For example, ol becomes VR.

The ciphertext in our example is

EG MN FQ QM KN BK SV VR GQ XN KU.

To decrypt, reverse the procedure.

The system succumbs to a frequency attack since the frequencies of the various digrams (two-letter combinations) in English have been tabulated. Of course, we only have to look for the most common digrams; they should correspond to the most common digrams in English: th, he, an, in, re, es, $\dots$. Moreover, a slight modification yields results more quickly. For example, both of the digrams re and er are very common. If the pairs IG and GI are common in the ciphertext, then a good guess is that e, i, r, g form the corners of a rectangle in the matrix. Another weakness is that each plaintext letter has only five possible corresponding ciphertext letters. Also, unless the keyword is long, the last few rows of the matrix are predictable. Observations such as these allow the system to be broken with a ciphertext-only attack. For more on its cryptanalysis, see [Gaines].

The ADFGX cipher proceeds as follows. Put the letters of the alphabet into a 5×5 matrix. The letters i and j are treated as one, and the columns of the matrix are labeled with the letters A, D, F, G, X. For example, the matrix could be

$$\begin{array}{c|ccccc} & A & D & F & G & X \\ \hline A & p & g & c & e & n \\ D & b & q & o & z & r \\ F & s & l & a & f & t \\ G & m & d & v & i & w \\ X & k & u & y & x & h \end{array}$$

2.6-12 Full Alternative Text

Each plaintext letter is replaced by the label of its row and column. For example, s becomes FA, and z becomes DG. Suppose the plaintext is

$$Kaiser\,Wilhelm.$$

The result of this initial step is

$$XA\,FF\,GG\,FA\,AG\,DX\,GX\,GG\,FD\,XX\,AG\,FD\,GA.$$

So far, this is a disguised substitution cipher. The next step increases the complexity significantly. Choose a keyword, for example, *Rhein*. Label the columns of a matrix by the letters of the keyword and put the result of the initial step into another matrix:

$$\begin{array}{ccccc} R & H & E & I & N \\ \hline X & A & F & F & G \\ G & F & A & A & G \\ D & X & G & X & G \\ G & F & D & X & X \\ A & G & F & D & G \\ A & & & & \end{array}$$

2.6-13 Full Alternative Text

Now reorder the columns so that the column labels are in alphabetic order:

$$\begin{array}{ccccc} E & H & I & N & R \\ \hline F & A & F & G & X \\ A & F & A & G & G \\ G & X & X & G & D \\ D & F & X & X & G \\ F & G & D & G & A \\ & & & & A \end{array}$$

2.6-14 Full Alternative Text

Finally, the ciphertext is obtained by reading down the columns (omitting the labels) in order:

$$FAGDFAFXFGFAXXDGGGXGXGDGAA.$$

Decryption is easy, as long as you know the keyword. From the length of the keyword and the length of the ciphertext, the length of each column is determined. The letters are placed into columns, which are reordered to match the keyword. The original matrix is then used to recover the plaintext.

The initial matrix and the keyword were changed frequently, making cryptanalysis more difficult, since there was only a limited amount of ciphertext available for any combination. However, the system was successfully attacked by the French cryptanalyst Georges Painvin and the Bureau du Chiffre, who were able to decrypt a substantial number of messages.

Here is one technique that was used. Suppose two different ciphertexts intercepted at approximately the same time agree for the first several characters. A reasonable guess is that the two plaintexts agree for several words. That means that the top few entries of the columns for one are the same as for the other. Search through the ciphertexts and find other places where they agree. These possibly represent the beginnings of the columns. If this is correct, we know the column lengths. Divide the ciphertexts into columns using these lengths. For the first ciphertext, some columns will have one length and others will be one longer. The longer ones represent columns that should be near the beginning; the other columns should be near the end. Repeat for the second ciphertext. If a column is long for both ciphertexts, it is very near the beginning. If it is long for one ciphertext and not for the other, it goes in the middle. If it is short for both, it is near the end. At this point, try the various orderings of the columns, subject to these restrictions. Each ordering corresponds to a potential substitution cipher. Use frequency analysis to try to solve these. One should yield the plaintext, and the initial encryption matrix.

The letters $ADFGX$ were chosen because their symbols in Morse code ($\cdot-, - \cdot \cdot, \cdot \cdot - \cdot, - - \cdot, - \cdot \cdot -$) were not easily confused. This was to avoid transmission errors, and represents one of the early attempts to combine error correction with cryptography. Eventually, the $ADFGX$ cipher was replaced by the $ADFGVX$ cipher, which used a 6×6 initial matrix. This allowed all 26 letters plus 10 digits to be used.

For more on the cryptanalysis of the ADFGX cipher, see [Kahn].

2.7 Enigma

Mechanical encryption devices known as rotor machines were developed in the 1920s by several people. The best known was designed by Arthur Scherbius and became the famous Enigma machine used by the Germans in World War II.

It was believed to be very secure and several attempts at breaking the system ended in failure. However, a group of three Polish cryptologists, Marian Rejewski, Henryk Zygalski, and Jerzy Różycki, succeeded in breaking early versions of Enigma during the 1930s. Their techniques were passed to the British in 1939, two months before Germany invaded Poland. The British extended the Polish techniques and successfully decrypted German messages throughout World War II.

The fact that Enigma had been broken remained a secret for almost 30 years after the end of the war, partly because the British had sold captured Enigma machines to former colonies and didn't want them to know that the system had been broken.

In the following, we give a brief description of Enigma and then describe an attack developed by Rejewski. For more details, see for example [Kozaczuk], which contains appendices by Rejeweski giving details of attacks on Enigma.

We give a basic schematic diagram of the machine in Figure 2.1. For more details, we urge the reader to visit some of the many websites that can be found on the Internet that give pictures of actual Enigma machines and extensive diagrams of the internal workings of these machines. There are also several online Enigma simulators. Try one of them to get a better understanding of how Enigma works.

Figure 2.1 A Schematic Diagram of the Enigma Machine

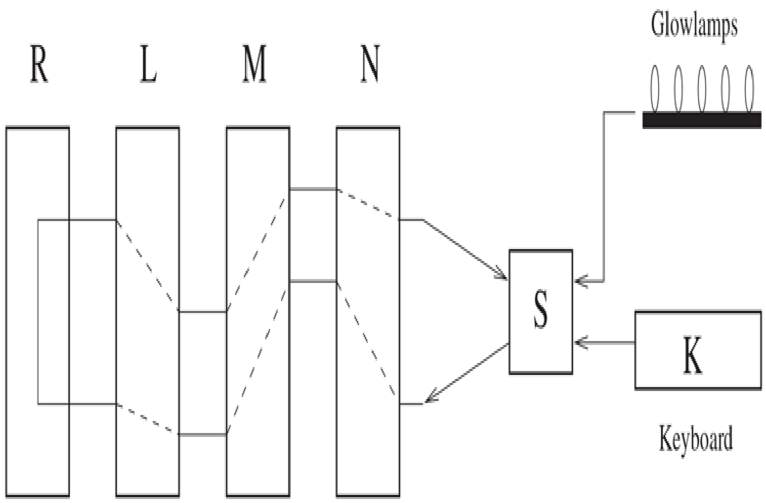


Figure 2.1 Full Alternative Text

L, M, N are the rotors. On one side of each rotor are 26 fixed electrical contacts, arranged in a circle. On the other side are 26 spring-loaded contacts, again arranged in a circle so as to touch the fixed contacts of the adjacent rotor. Inside each rotor, the fixed contacts are connected to the spring-loaded contacts in a somewhat random manner. These connections are different in each rotor. Each rotor has 26 possible initial settings.

R is the reversing drum. It has 26 spring-loaded contacts, connected in pairs.

K is the keyboard and is the same as a typewriter keyboard.

S is the plugboard. It has approximately six pairs of plugs that can be used to interchange six pairs of letters.

When a key is pressed, the first rotor N turns 1/26 of a turn. Then, starting from the key, electricity passes through S, then through the rotors N, M, L. When it reaches the reversing drum R, it is sent back along a different path through L, M, N, then through S. At this point, the electricity lights a bulb corresponding to a letter on the keyboard, which is the letter of the ciphertext.

Since the rotor N rotates before each encryption, this is much more complicated than a substitution cipher. Moreover, the rotors L and M also rotate, but much less often, just like the wheels on a mechanical odometer.

Decryption uses exactly the same method. Suppose a sender and receiver have identical machines, both set to the same initial positions. The sender encrypts the message by typing it on the keyboard and recording the sequence of letters indicated by the lamps. This ciphertext is then sent to the receiver, who types the ciphertext into the machine. The sequence of letters appearing in the lamps is the original message. This can be seen as follows. Lamp “a” and key “a” are attached to a wire coming out of the plugboard. Lamp “h” and key “h” are attached to another wire coming out of the plugboard. If the key “a” is pressed and the lamp “h” lights up, then the electrical path through the machine is also connecting lamp “a” to key “h”. Therefore, if the “h” key were pressed instead, then the “a” key would light.

Similar reasoning shows that no letter is ever encrypted as itself. This might appear to be a good idea, but actually it is a weakness since it allows a cryptanalyst to discard many possibilities at the start. See Chapter 14.

The security of the system rests on the keeping secret the initial settings of the rotors, the setting of the plugs on the plugboard, and the internal wiring of the rotors and reversing drum. The settings of the rotors and the plugboard are changed periodically (for example, daily).

We'll assume the internal wiring of the rotors is known. This would be the case if a machine were captured, for example. However, there are ways to deduce this information, given enough ciphertext, and this is what was actually done in some cases.

How many combinations of settings are there? There are 26 initial settings for each of the three rotors. This gives $26^3 = 17576$ possibilities. There are six possible orderings of the three rotors. This yields $6 \times 17576 = 105456$ possible ways to initialize the rotors. In later versions of Enigma, there were five rotors available, and each day three were chosen. This made 60 possible orderings of the rotors and therefore 1054560 ways to initialize the rotors.

On the plugboard, there are 100391791500 ways of interchanging six pairs of letters.

In all, there seem to be too many possible initializations of the machine to have any hope of breaking the system. Techniques such as frequency analysis fail since the rotations of the rotors change the substitution for each character of the message.

So, how was Enigma attacked? We don't give the whole attack here, but rather show how the initial settings of the rotors were determined in the years around 1937. This attack depended on a weakness in the protocol being used at that time, but it gives the general flavor of how the attacks proceeded in other situations.

Each Enigma operator was given a codebook containing the daily settings to be used for the next month. However, if these settings had been used without modification, then each message sent during a given day would have had its first letter encrypted by the same substitution cipher. The rotor would then have turned and the second letter of each text would have corresponded to another substitution cipher, and this substitution would have been the same for all messages for that day. A frequency analysis on the first letter of each intercepted message during a day would probably allow a decryption of the first letter of each text. A second frequency analysis would decrypt the second letters. Similarly, the remaining letters of the ciphertexts (except for the ends of the longest few ciphertexts) could be decrypted.

To avoid this problem, for each message the operator chose a message key consisting of a sequence of three letters, for example, $r,\ f,\ u$. He then used the daily setting from the codebook to encrypt this message key. But since radio communications were prone to error, he typed in rfu twice, therefore encrypting $rfurfu$ to obtain a string of six letters. The rotors were then set to positions $r,\ f$, and u and the encryption of the actual message began. So the first six letters of the transmitted message were the encrypted message key, and the remainder was the ciphertext. Since each message used a different key, frequency analysis didn’t work.

The receiver simply used the daily settings from the codebook to decrypt the first six letters of the message. He then reset the rotors to the positions indicated by the decrypted message key and proceeded to decrypt the message.

The duplication of the key was a great aid to the cryptanalysts. Suppose that on some day you intercept several messages, and among them are three that have the following initial six letters:

dmqvbn

vonpuy

pucfmq

All of these were encrypted with the same daily settings from the codebook. The first encryption corresponds to a permutation of the 26 letters; let's call this permutation A. Before the second letter is encrypted, a rotor turns, so the second letter uses another permutation; call it B. Similarly, there are permutations C, D, E, F for the remaining four letters. The strategy is to look at the products AD, BE, and CF.

We need a few conventions and facts about permutations. When we write AD for two permutations A and D, we mean that we apply the permutation A then D (some books use the reverse ordering). The permutation that maps a to b, b to c, and c to a will be denoted as the 3-cycle (abc). A similar notation will be used for cycles of other lengths. For example, (ab) is the permutation that switches a and b. A permutation can be written as a product of cycles. For example, the permutation

$$(dvpfkxgzyo)(eijmunqlht)(bc)(rw)(a)(s)$$

is the permutation that maps d to v, v to p, t to e, r to w, etc., and fixes a and s. If the cycles are disjoint (meaning that no two cycles have letters in common), then this decomposition into cycles is unique.

Let's look back at the intercepted texts. We don't know the letters of any of the three message keys, but let's call the first message key xyz. Therefore, $xyzxyz$ encrypts to $dmqvbn$. We know that permutation A sends x to d. Also, the fourth permutation D sends x to v. But we know more. Because of the internal wiring of the machine, A actually interchanges x and d and D interchanges x and v. Therefore, the product of the permutations, AD, sends d to v (namely, A sends d to x and then D sends x to v). The unknown x has been eliminated. Similarly, the second intercepted text tells us that AD sends v to p, and the third tells us that AD sends p to f. We have therefore determined that

$$AD = (dvpf\cdots)\cdots.$$

In the same way, the second and fifth letters of the three messages tell us that

$$BE = (oumb\cdots)\cdots$$

and the third and sixth letters tell us that

$$CF = (cqny\cdots)\cdots.$$

With enough data, we can deduce the decompositions of AD, BE, and CF into products of cycles. For example, we might have

$$\begin{aligned} AD &= (dvpfkxgzyo)(eijmunqlht)(bc)(rw)(a)(s) \\ BE &= (blfqveoum)(hjpswizrn)(axt)(cgy)(d)(k) \\ CF &= (abviktjgfcqny)(duzrehlxwpsmo). \end{aligned}$$

This information depends only on the daily settings of the plugboard and the rotors, not on the message key. Therefore, it relates to every machine used on a given day.

Let's look at the effect of the plugboard. It introduces a permutation S at the beginning of the process and then adds the inverse permutation S^{-1} at the end. We need another fact about permutations: Suppose we take a permutation P and another permutation of the form SPS^{-1} for some permutation S (where S^{-1} denotes the inverse permutation of S; in our case, $S = S^{-1}$) and decompose each into cycles. They will usually not have the same cycles, but the lengths of the cycles in the decompositions will be the same. For example, AD has cycles of length 10, 10, 2, 2, 1, 1. If we decompose $SADS^{-1}$ into cycles for any permutation S, we will again get cycles of lengths 10, 10, 2, 2, 1, 1. Therefore, if the plugboard settings are changed, but the initial positions of the rotors remain the same, then the cycle lengths remain unchanged.

You might have noticed that in the decomposition of AD, BE, and CF into cycles, each cycle length appears an even number of times. This is a general phenomenon. For an explanation, see Appendix E of the aforementioned book by Kozaczuk.

Rejewski and his colleagues compiled a catalog of all 105456 initial settings of the rotors along with the set of cycle lengths for the corresponding three permutations AD, BE, CF. In this way, they could take the ciphertexts for a given day, deduce the cycle lengths, and find the small number of corresponding initial settings for the rotors. Each of these substitutions could be tried individually. The effect of the plugboard (when the correct setting was used) was then merely a substitution cipher, which was easily broken. This method worked until September 1938, when a modified method of transmitting message keys was adopted. Modifications of the above technique were again used to decrypt the messages. The process was also mechanized, using machines called “bombes” to find daily keys, each in around two hours.

These techniques were extended by the British at Bletchley Park during World War II and included building more sophisticated “bombes.” These machines, designed by Alan Turing, are often considered to have been the first electronic computers.

2.8 Exercises

1. Caesar wants to arrange a secret meeting with Marc Antony, either at the Tiber (the river) or at the Coliseum (the arena). He sends the ciphertext $EVIRE$. However, Antony does not know the key, so he tries all possibilities. Where will he meet Caesar? (Hint: This is a trick question.)

2. Show that each of the ciphertexts $ZOMCIH$ and $ZKNGZR$, which were obtained by shift ciphers from one-word plaintexts, has two different decryptions.

3. The ciphertext UCR was encrypted using the affine function $9x + 2$ mod 26. Find the plaintext.

4. The ciphertext JLH was obtained by affine encryption with the function $9x + 1$ mod 26. Find the plaintext.

5. Encrypt *howareyou* using the affine function $5x + 7 \pmod{26}$. What is the decryption function? Check that it works.

6. You encrypt messages using the affine function $9x + 2$ mod 26. Decrypt the ciphertext GM.

7. A child has learned about affine ciphers. The parent says *NONONO*. The child responds with *hahaha*, and quickly claims that this is a decryption of the parent's message. The parent asks for the encryption function. What answer should the child give?

8. You try to encrypt messages using the affine cipher $4x + 1$ mod 26. Find two letters that encrypt to the same ciphertext letter.

9. The following ciphertext was encrypted by an affine cipher mod 26:

   $$CRWWZ.$$

   The plaintext starts *ha*. Decrypt the message.

10. Alice encrypts a message using the affine function $x \mapsto ax \pmod{26}$ for some a. The ciphertext is *FAP*. The third letter of the plaintext is T. Find the plaintext.

11. Suppose you encrypt using an affine cipher, then encrypt the encryption using another affine cipher (both are working mod 26). Is there any advantage to doing this, rather than using a single affine cipher? Why or why not?

12. Find all affine ciphers mod 26 for which the decryption function equals the encryption function. (There are 28 of them.)

13. Suppose we work mod 27 instead of mod 26 for affine ciphers. How many keys are possible? What if we work mod 29?

14. The ciphertext *XVASDW* was encrypted using an affine function $ax + 1$ mod 26. Determine a and decrypt the message.

15. Suppose that you want to encrypt a message using an affine cipher. You let $a = 0, b = 1, \ldots, z = 25$, but you also include $? = 26, ; = 27, \text{"} = 28, ! = 29$. Therefore, you use $x \mapsto \alpha x + \beta \pmod{30}$ for your encryption function, for some integers α and β.

    1. Show that there are exactly eight possible choices for the integer α (that is, there are only eight choices of α (with $0 < \alpha < 30$) that allow you to decrypt).

    2. Suppose you try to use $\alpha = 10, \ \beta = 0$. Find two plaintext letters that encrypt to the same ciphertext letter.

16. You are trying to encrypt using the affine function $13x + 22$ mod 26.

    1. Encrypt *HATE* and *LOVE*. Why is decryption impossible?

    2. Find two different three-letter words that encrypt to *WWW*.

    3. Challenge: Find a word (that is legal in various word games) that encrypts to *JJJ*. (There are four such words.)

17. You want to carry out an affine encryption using the function $\alpha x + \beta$, but you have $\gcd(\alpha, 26) = d > 1$. Show that if $x_1 = x_2 + (26/d)$, then $\alpha x_1 + \beta \equiv \alpha x_2 + \beta \pmod{26}$. This shows that you will not be able to decrypt uniquely in this case.

18. You encrypt the message *zzzzzzzzzz* (there are 10 z's) using the following cryptosystems:

    1. affine cipher

    2. Vigenère cipher with key length 7

       Eve intercepts the ciphertexts. She knows the encryption methods (including key size) and knows what your plaintext is (she can hear you snoring). For each of the two cryptosystems, determine whether or not Eve can use this information to determine the key. Explain your answer.

19. Suppose there is a language that has only the letters a and b. The frequency of the letter a is .1 and the frequency of b is .9. A

message is encrypted using a Vigenère cipher (working mod 2 instead of mod 26). The ciphertext is BABABAAABA. The key length is 1, 2, or 3.

1. Show that the key length is probably 2.
2. Using the information on the frequencies of the letters, determine the key and decrypt the message.

20. Suppose you have a language with only the three letters a, b, c, and they occur with frequencies $.9$, $.09$, and $.01$, respectively. The ciphertext *BCCCBCBCBC* was encrypted by the Vigenère method (shifts are mod 3, not mod 26). Find the plaintext (Note: The plaintext is not a meaningful English message.)

21. Suppose you have a language with only the three letters a, b, c, and they occur with frequencies .7, .2, .1, respectively. The following ciphertext was encrypted by the Vigenère method (shifts are mod 3 instead of mod 26, of course):

    $ABCBABBBAC.$

    Suppose you are told that the key length is 1, 2, or 3. Show that the key length is probably 2, and determine the most probable key.

22. Victor designs a cryptosystem (called "Vector") as follows: He writes the letters in the plaintext as numbers mod 26 (with $a = 0,\ b = 1$, etc.) and groups them five at a time into five-dimensional vectors. His key is a five-dimensional vector. The encryption is adding the key vector mod 26 to each plaintext vector (so this is a shift cipher with vectors in place of individual letters).

    1. Describe a chosen plaintext attack on this system. Give the *explicit* plaintext used and how you get the key from the information you obtain.
    2. Victor's system is not new. It is the same as what well-known system?

23. If **v** and **w** are two vectors in n-dimensional space, $\mathbf{v} \cdot \mathbf{w} = |\mathbf{v}||\mathbf{w}| \cos\ \theta,$, where θ is the angle between the two vectors (measured in the two-dimensional plane spanned by the two vectors), and $|\mathbf{v}|$ denotes the length of **v**. Use this fact to show that, in the notation of Section 2.3, the dot product $\mathbf{A}_0 \cdot \mathbf{A}_i$ is largest when $i = 0$.

24. Alice uses an improvement of the Vigenère cipher. She chooses five affine functions

    $$a_1x + b_1,\ a_2x + b_2,\ \ldots,\ a_5x + b_5 \pmod{26}$$

    and she uses these to encrypt in the style of Vigenère. Namely, she encrypts the first plaintext letter using $a_1x + b_1$, the second letter using $a_2x + b_2$, etc.

1. What condition do $a_1, a_2, \ldots, a_5$ need to satisfy for Bob (who knows the key) to able to decrypt the message?

2. Describe how to do a *chosen plaintext* attack to find the key. Give the plaintext *explicitly* and explain how it yields the key. (Note: the solution has nothing to do with frequencies of letters.)

25. Alice is sending a message to Bob using one of the following cryptosystems. In fact, Alice is bored and her plaintext consists of the letter a repeated a few hundred times. Eve knows what system is being used, but not the key, and intercepts the ciphertext. For systems (a), (b), and (c), state how Eve will recognize that the plaintext is one repeated letter and decide whether or not Eve can deduce the letter and the key.

    1. Shift cipher

    2. Affine cipher

    3. Vigenère cipher

26. The operator of a Vigenère encryption machine is bored and encrypts a plaintext consisting of the same letter of the alphabet repeated several hundred times. The key is a seven-letter English word. Eve knows that the key is a word but does not yet know its length.

    1. What property of the ciphertext will make Eve suspect that the plaintext is one repeated letter and will allow her to guess that the key length is seven?

    2. Once Eve guesses that the plaintext is one repeated letter, how can she determine the key? (Hint: You need the fact that no English word of length seven is a shift of another English word.)

    3. Suppose Eve doesn't notice the property needed in part (a), and therefore uses the method of displacing then counting matches for finding the length of the key. What will the number of matches be for the various displacements? In other words, why will the length of the key become very obvious by this method?

27. Use the Playfair cipher with the keyword *Cryptography* to encrypt

$$Did\ he\ play\ fair\ at\ St\ Andrews\ golf\ course.$$

28. The ciphertext

$$BP \quad EG \quad FC \quad AI \quad MA \quad MG \quad PO \quad KB \quad HU$$

was encrypted using the Playfair cipher with keyword *Archimedes*. Find the plaintext.

29. Encrypt the plaintext *secret* using the ADFGX cipher with the 5×5 matrix in Section 2.6 and the keyword *spy*.

30. The ciphertext *AAAAFXGGFAFFGGFGXAFGADGGAXXXFX* was encrypted using the ADFGX cipher with the 5×5 matrix in Section 2.6 and the keyword *broken*. Find the plaintext.

31. Suppose Alice and Bob are using a cryptosystem with a 128-bit key, so there are 2^{128} possible keys. Eve is trying a brute-force attack on the system.

    1. Suppose it takes 1 day for Eve to try 2^{64} possible keys. At this rate, how long will it take for Eve to try all 2^{128} keys? (Hint: The answer is not 2 days.)

    2. Suppose Alice waits 10 years and then buys a computer that is 100 times faster than the one she now owns (so it takes only 1/100 of a day, which is 864 seconds, to try 2^{64} keys). Will she finish trying all 2^{128} keys before or after what she does in part (a)? (Note: This is a case where Aesop's Fable about the Tortoise and the Hare has a different ending.)

32. In the mid-1980s, a recruiting advertisement for NSA had 1 followed by one hundred 0s at the top. The text began "You're looking at a 'googol.' Ten raised to the 100th power. One followed by 100 zeroes. Counting 24 hours a day, you would need 120 years to reach a googol. Two lifetimes. It's a number that's impossible to grasp. A number beyond our imagination."

    How many numbers would you have to count each second in order to reach a googol in 120 years? (This problem is not related to the cryptosystems in this chapter. It is included to show how big 100-digit numbers are from a computational viewpoint. Regarding the ad, one guess is that the advertising firm assumed that the time it took to factor a 100-digit number back then was the same as the time it took to count to a googol.)

2.9 Computer Problems

1. The following ciphertext was encrypted by a shift cipher:

   ```
   ycvejqwvhqtdtwvwu
   ```

   Decrypt. (The ciphertext is stored in the downloadable computer files (`bit.ly/2JbcS6p`) under the name *ycve*.)

2. The following ciphertext was the output of a shift cipher:

   ```
   lcllewljazlnnzmvyiylhrmhza
   ```

   By performing a frequency count, guess the key used in the cipher. Use the computer to test your hypothesis. What is the decrypted plaintext? (The ciphertext is stored in the downloadable computer files (`bit.ly/2JbcS6p`) under the name *lcll*.)

3. The following was encrypted by an affine cipher: `jidfbidzzteztxjsichfoihuszzsfsaichbipahsibdhu hzsichjujgfabbczggjsvzubehhgjsv`. Decrypt it. (This quote (NYTimes, 12/7/2014) is by Mark Wahlberg from when he was observing college classes in order to play a professor in "The Gambler." The ciphertext is stored in the downloadable computer files (`bit.ly/2JbcS6p`) under the name *jidf*.) (Hint: The command "frequency" could be useful. The plaintext has 9 e's, 3 d's, and 3 w's.)

4. The following ciphertext was encrypted by an affine cipher:

   ```
   edsgickxhuklzveqzvkxwkzukcvuh
   ```

   The first two letters of the plaintext are *if*. Decrypt. (The ciphertext is stored in the downloadable computer files (`bit.ly/2JbcS6p`) under the name *edsg*.)

5. The following ciphertext was encrypted by an affine cipher using the function $3x + b$ for some b:

```
tcabtiqmfheqqmrmvmtmaq
```

Decrypt. (The ciphertext is stored in the downloadable computer files (`bit.ly/2JbcS6p`) under the name *tcab*.)

6. Experiment with the affine cipher $y \equiv mx + n \pmod{26}$ for values of $m > 26$. In particular, determine whether or not these encryptions are the same as ones obtained with $m < 26$.

7. In this problem you are to get your hands dirty doing some programming. Write some code that creates a new alphabet $\{A, C, G, T\}$. For example, this alphabet could correspond to the four nucleotides adenine, cytosine, guanine, and thymine, which are the basic building blocks of DNA and RNA codes. Associate the letters A, C, G, T with the numbers $0, 1, 2, 3$, respectively.

    1. Using the shift cipher with a shift of 1, encrypt the following sequence of nucleotides, which is taken from the beginning of the thirteenth human chromosome:

       GAATTCGCGGCCGCAATTAACCCTCACTAAAGGGATCT

       CTAGAACT.

    2. Write a program that performs affine ciphers on the nucleotide alphabet. What restrictions are there on the affine cipher?

8. The following was encrypted using by the Vigenère method using a key of length at most 6. Decrypt it and decide what is unusual about the plaintext. How did this affect the results?

```
hdsfgvmkoowafweetcmfthskucaqbilgjofmaqlgsp
vatvxqbiryscpcfr
mvswrvnqlszdmgaoqsakmlupsqforvtwvdfcjzvgso
aoqsacjkbrsevbel
vbksarlscdcaarmnvrysywxqgvellcyluwwveoafgc
lazowafojdlhssfi
ksepsoywxafowlbfcsocylngqsyzxgjbmlvgrggokg
fgmhlmejabsjvgml
nrvqzcrggcrghgeupcyfgtydycjkhqluhgxgzovqsw
pdvbwsffsenbxapa
sgazmyuhgsfhmftayjxmwznrsofrsoaopgauaaarmf
tqsmahvqecev
```

(The ciphertext is stored under the name *hdsf* in the downloadable computer files (`bit.ly/2JbcS6p`). The plaintext is from Gadsby by Ernest Vincent Wright.)

9. The following was encrypted by the Vigenère method. Find the plaintext.

```
ocwyikoooniwugpmxwktzdwgtssayjzwyemdlbnqaa
avsuwdvbrflauplo
oubfgqhgcscmgzlatoedcsdeidpbhtmuovpiekifpi
mfnoamvlpqfxejsm
xmpgkccaykwfzpyuavtelwhrhmwkbbvgtguvtefjlo
dfefkvpxsgrsorvg
tajbsauhzrzalkwuowhgedefnswmrciwcpaaavogpd
nfpktdbalsisurln
psjyeatcuceesohhdarkhwotikbroqrdfmzghguceb
vgwcdqxgpbgqwlpb
daylooqdmuhbdqgmyweuik
```

(The ciphertext is stored under the name *ocwy* in the downloadable computer files (`bit.ly/2JbcS6p`). The plaintext is from The Adventure of the Dancing Men by Sir Arthur Conan Doyle.)

10. The following was encrypted by the Vigenère method. Decrypt it. (The ciphertext is stored under the name *xkju* in the downloadable computer files (`bit.ly/2JbcS6p`).)

```
xkjurowmllpxwznpimbvbqjcnowxpcchhvvfvsllfv
xhazityxohulxqoj
axelxzxmyjaqfstsrulhhucdskbxknjqidallpqsll
uhiaqfpbpcidsvci
hwhwewthbtxrljnrsncihuvffuxvoukjljswmaqfvj
wjsdyljogjxdboxa
jultucpzmpliwmlubzxvoodybafdskxgqfadshxnxe
hsaruojaqfpfkndh
saafvulluwtaqfrupwjrszxgpfutjqiynrxnyntwmh
cukjfbirzsmehhsj
shyonddzzntzmplilrwnmwmlvuryonthuhabwnvw
```