A beginner's guide to codebreaking

Written by Prof. Graham A. Niblo
Edited by Dr. Claire Swabey

School of Mathematical Sciences
University of Southampton

Version 1.1d
19th September 2018

About these notes

These notes form a brief introduction to using and cracking substitution ciphers and transposition ciphers, to accompany the teaching materials provided with the University of Southampton National Cipher Challenge. The website for the competition can be found at www.cipherchallenge.org.

Substitution ciphers

Caesar shift ciphers

The easiest method of enciphering a text message is to replace each letter by another, using a fixed rule, so for example every letter a may be replaced by D, and every letter b by the letter E and so on.
Applying this rule to the previous paragraph produces the text

```
WKH HDVLHVW PHWKRG RI HQFLSKHULQJ D WHAW PHVVDJH LV
WR UHSODFH HDFK OHWWHU EB DQRWKHU XVLQJ D ILAHG UXOH,
VR IRU HADPSOH HYHUB OHWWHU D PDB EH UHSODFHG EB G,
DQG HYHUB OHWWHU E EB WKH OHWWHU H DQG VR RQ.
```

Note the convention in these notes that cipher-text is written in capital letters, while plaintext is usually lowercase.

Such a cipher is known as a shift cipher since the letters of the alphabet are shifted round by a fixed amount, and as a Caesar shift since such ciphers were used by Julius Caesar.

There are only 26 Caesar shift ciphers (and one of them does nothing to the text) so it is not too hard to decipher the text by brute force. We can try each of the shifts in turn on the first word of the cipher text until we discover the correct shift. This process can be simplified by using a cipher wheel, a simple mechanical device that allows us to generate each of the Caesar shift ciphers, and to encode or decode messages using it. At the back of this leaflet you will find a sheet which can be photocopied onto thin card in order to make a cipher wheel. Cut out the two discs, and fasten through their centres with a paper fastener to make the wheel. Use the convention that you read cipher-text on the rim of the inner, smaller, wheel and plaintext from the outer, larger wheel, reading inwards to encode and outwards to decode.

Just because we can use brute force to solve the cipher doesn't mean we have to. If that was all there was to codebreaking it would be entirely the province of computer scientists and engineers who are very smart at speeding up that sort of computation. At the cutting edge of cryptography it is the interaction of those disciplines with mathematics which enables governments (and criminal hackers) to read poorly encrypted communications, and we can begin to see where mathematics comes into the picture even when considering a simple cipher like the Caesar shift.

Notice that in order to know which shift cipher has been used it is enough to work out where one of the letters has been shifted. That tells us the amount of shift and therefore the entire cipher. This can be done, for example, by discovering which character has replaced the plaintext letter e. The letter e has been chosen here for a reason, it is the single most common letter to be found in English text (curiously, largely because the word the is one of the most common words - we will come back to that point in a minute). It is an interesting exercise to choose some text and analyse the letter frequencies to confirm that for yourself, and you can find a useful online text analyser at

http://www.dcode.fr/frequency-analysis

to speed that up.

Running the cipher text above through the analyser we see that the letter H appears more than twice as frequently than any other letter, more than 20% of the time. We can compare the frequencies with those for the lead story from the BBC news site this morning which has been run through the same checker. The frequencies are not quite the same partly because the first paragraph of this guide was carefully written to ensure it had a lot of the letter e in it to make a point. Nonetheless this is often a good way to identify which letter has been used to encrypt the letter e and for a Caesar shift cipher that is all we need.

For a more sophisticated cipher like the affine shift cipher or the keyword cipher we will need to know more than just one letter to break it, so let's look again at our first example. The spaces in the text imply that the encryption has left the word structure intact. So we might guess that the three-letters starting the sentence form a 3 letter word, and as remarked above the most common 3 letter word in English is "the". This fits with our frequency count which suggests (correctly) that e has been replaced by H, and a quick check shows that the Caesar shift by 3 does indeed encode the word the as WKH, and it is easy to complete the decryption.

Keyword substitution ciphers

To see just how powerful frequency analysis can be we will next consider how to tackle a keyword substitution cipher. These were introduced by security services as a highly secure, reliable and easy to use field cipher for agents. Of course security depends on the ability of the enemy to crack the cipher and they would be hopelessly inadequate now, but they are still many times harder to break than the Caesar shift cipher as we will see.

To build a keyword substitution cipher we design an encryption table by choosing a keyword or phrase which is used to jumble the alphabet as follows:

First write down the phrase, with no spaces between the letters, and omitting any repeated character then continue round the alphabet in order until every letter appears exactly once, and write the list under the standard alphabet:

a	b	c	d	e	f	g	h	i	j	k	l	m
S	I	M	P	O	N	Q	R	T	U	V	W	X
n	o	p	q	r	s	t	u	v	w	x	y	z
Y	Z	A	B	C	D	E	F	G	H	J	K	L

Here we have chosen the key word SIMPSONS so we continue the alphabet from the first unused letter after the last used letter, N, which is Q. Of course if the key phrase is carefully chosen (for example "The quick brown fox jumps over the lazy dog") there might not be any letters left to use up but such a choice is not necessary. If instead of using a genuine word or phrase we allow us to use any ordering of the letters in our cipher-text alphabet then the number of such ciphers

is 26!, or approximately 10^{27}, and brute force cannot be used to attack the problem. In practice a fully random encryption table would be impossible for an agent to reliably memorise (they work under conditions of extreme stress after all) so a genuine word or phrase will have been used which reduces the number of ciphers considerably. According to the Oxford English Dictionary authorities "there are, at the very least, a quarter of a million distinct English words", which would still make a brute force attack impossible without the aid of a computer, but frequency analysis still works, especially if we can see the word shapes.

Consider the text

```
VEP HYXHLVHTP MO AWFJYFLT H RFNEPS HJNEHAPV FL VEFU
ZHC FU VEHV FV FU PHUC VM KPKMSFUP VEP IPCZMSY MS
IPCNESHUP, HLY EPLRP VEP RFNEPS HJNEHAPV. VEFU FU
FKNMSVHLV, APRHWUP FO VEP UPLYPS EHU VM IPPN VEP
RFNEPS HJNEHAPV ML H NFPRP MO NHNPS, VEP PLPKC RHL
RHNVWSP VEP NHNPS, YFURMXPS VEP IPC, HLY SPHY HLC
RMKKWLFRHVFMLU VEHV EHXP APPL PLRSCNVPY ZFVE FV.
EMZPXPS FO VEP IPC RHL AP RMKKFVVPY VM KPKMSC FV FU
JPUU JFIPJC VM OHJJ FLVM PLPKC EHLYU.
```

As before we notice that the first word has three letters and, since it occurs several times, may well be the word "the". This gives a strong hint that the letter e is enciphered as the letter P in the cipher. Of course other three letter words are possible, e.g., "and" or "but". Nonetheless a quick check shows us that the letter P is the most common letter in the enciphered text, so it is reasonable to assume that the correct decryption translates P to e. This also suggests that V stands for t and E for h, allowing us to begin to decipher the text. We will use the convention that UPPER CASE LETTERS denote enciphered letters and lowercase letters denote plaintext characters:

```
VEP HYXHLVHTP MO AWFJYFLT H RFNEPS HJNEHAPV FL VEFU
the      t e                 he    h   et  th

ZHC FU VEHV FV FU PHUC VM KPKMSFUP VEP IPCZMSY MS
      th t  t    e    t   e     e the  e

IPCNESHUP, HLY EPLRP VEP RFNEPS HJNEHAPV. VEFU FU
 e  h   e,     he  e the    he     h  et. th

FKNMSVHLV, APRHWUP FO VEP UPLYPS EHU VM IPPN VEP
     t  t,  e    e    the  e  e  h   t   ee  the

RFNEPS HJNEHAPV ML H NFPRP MO NHNPS, VEP PLPKC RHL
   he     h  et       e e       e , the e e

RHNVWSP VEP NHNPS, YFURMXPS VEP IPC, HLY SPHY HLC
   t  e the    e ,       e  the  e ,      e

RMKKWLFRHVFMLU VEHV EHXP APPL PLRSCNVPY ZFVE FV.
        t    th t h  e  ee  e    te   th  t.

EMZPXPS FO VEP IPC RHL AP RMKKFVVPY VM KPKMSC FV FU
h  e e     the  e       e      tte  t   e      t

JPUU JFIPJC VM OHJJ FLVM PLPKC EHLYU.
 e      e   t         t  e e   h    .
```

Reading carefully we see the single letter word H, and the four letter word th_t circled above, and guess that H is a vowel, almost certainly the letter a. Making that replacement we get the following, where we are using our upper/lowercase convention to save space:

```
the aYXaLtaTe MO AWFJYFLT a RFNheS aJNhaAet FL thFU
ZaC FU that Ft FU eaUC tM KeKMSFUe the IeCZMSY MS
IeCNhSaUe, aLY heLRe the RFNheS aJNhaAet. thFU FU
FKNMStaLt, AeRaWUe FO the UeLYeS haU tM IeeN the
RFNheS aJNhaAet ML a NFeRe MO NaNeS, the eLeKC RaL
RaNtWSe the NaNeS, YFURMXeS the IeC, aLY SeaY aLC
RMKKWLFRatFMLU that haXe AeeL eLRSCNteY ZFth Ft.
hMZeXeS FO the IeC RaL Ae RMKKFtteY tM KeKMSC Ft FU
JeUU JFIeJC tM OaJJ FLtM eLeKC haLYU.
```

Now the two 2 letter words ending with “t” are “at” and “it” so the word Ft circled above is one of these, and since it is followed by another two letter word, FU, beginning with the same letter we probably have “it is” here, meaning that F enciphers i and U enciphers s.

Hence we get:

```
the aYXaLtaTe MO AWiJYiLT a RiNheS aJNhaAet iL this
ZaC is that it is easC tM KeKMSise the IeCZMSY MS
IeCNhSase, aLY heLRe the RiNheS aJNhaAet. this is
iKNMStaLt, AeRaWse iO the seLYeS has tM IeeN the
RiNheS aJNhaAet ML a NieRe MO NaNeS, the eLeKC RaL
RaNtWSe the NaNeS, YisRMXeS the IeC, aLY SeaY aLC
RMKKWLiRatFMLs that haXe AeeL eLRSCNteY Zith it.
hMZeXeS iO the IeC RaL Ae RMKKitteY tM KeKMSC it is
Jess JiIeJC tM OaJJ iLtM eLeKC haLYs.
```

Before reading on it is worth looking at this to see if you can spot any other likely substitutions of your own.

Appropriate guesses would be:

```
tM    = to,   so M = o
haXe  = have, so X = v
easC  = easy, so C = y
```

As we identify more letters it gets easier to guess even more and we can decipher the text to get the following extract from Simon Singh's excellent history of codes and ciphers, The Code Book:

"The advantage of building a cipher alphabet in this way is that it is easy to memorise the keyword or key-phrase, and hence the cipher alphabet. This is important, because if the sender has to keep the cipher alphabet on a piece of paper, the enemy can capture the paper, discover the key, and read any communications that have been encrypted with it. However if the key can be committed to memory it is less likely to fall into enemy hands."

Can you identify the keyword for this cipher?

Frequency analysis

We have already seen how frequency analysis can help us to identify common letters and common words. We can go further with this analysis, comparing the number of occurrences of each character in the cipher text with an expected frequency for the standard English alphabet. In the plain text above a character count gives us the following table of occurrences.

A	B	C	D	E	F	G	H	I		K	L	M	N	O	P		R	S	T	U	V	W		Y	
32	7	14	11	55	5	2	26	27		6	9	11	20	18	16		17	17	35	4	4	4		12	

The consonants h,s,t are relatively common in English plaintext as are the vowels a,e,i and o. The vowel u is much less common and any occurrence of q is

almost guaranteed to be followed by a u. It is also possible to analyse common letter pairs and triples. as we have seen the triple "the" is the most common in english. (Cryptographers refer to triples of letters as trigrams and pairs of letters as digraphs or bigrams, and you can look up standard, bigram and trigram frequency tables on the web, for example at: http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/

Disguising the word structure

A chink in the armour of our ciphers so far has been the preservation of word structure. This allows us to spot common words. In order to avoid such weakness cryptographers usually remove punctuation and block the characters together in groups of four or five, so our previous cipher text looks like

```
VEPHY XHLVH TPMOA WFJYF LTHRF NEPSH JNEHA PVFLV EFUZH
CFUVE HVFVF UPHUC VMKPK MSFUP VEPIP CZMSY MSIPC NESHU
PHLYE PLRPV EPRFN EPSHJ NEHAP VVEFU FUFKN MSVHL VAPRH
WUPFO VEPUP LYPSE HUVMI PPNVE PRFNE PSHJN EHAPV MLHNF
PRPMO NHNPS VEPPL PKCRH LRHNV WSPVE PNHNP SYFUR MXPSV
EPIPC HLYSP HYHLC RMKKW LFRHV FMLUV EHVEH XPAPP LPLRS
CNVPY ZFVEF VEMZP XPSFO VEPIP CRHLA PRMKK FVVPY VMKPK
MSCFV FUJPU UJFIP JCVMO HJJFL VMPLP KCEHL YU
```

Usually the length of the text groups doesn't matter, however, in analysing a Vigenère cipher (see below) a carelessly chosen block length may make the length of the keyword more apparent, since it can reveal the repetitions more easily.

To attack cipher text that has been grouped in this way we have to work with letters not words. To do so we use the frequency analysis described above, together with a little judgement (or luck!). The process can be hard, but wars have been won or lost on the back of it, and so have fortunes. As remarked by Jericho, the lead character in Robert Harris's novel "Enigma",

> “It was hard going, but Jericho didn’t mind. He was taking action, that was the point. It was the same as code-breaking. However hopeless the situation, the rule was always to do something. No cryptogram, Alan Turing used to say, was ever solved by simply staring at it.”

Affine shift ciphers

Despite the advantages for an agent in using keyword substitution most modern ciphers are automated and rely on a mathematical encryption algorithm. Indeed the Caesar shift cipher can be viewed in this way.

Encoding each letter by its numerical position in the alphabet a = 1, b=2 and so on, the shift cipher is obtained by addition modulo 26. So a shift by 3 sends 7 to 7+3=10, which corresponds to mapping g to J. At the end of the alphabet we have x mapping to A, y mapping to B and z mapping to C which correspond to the modular arithmetic 24+3=1 mod 26, 25+3=2 mod 26 and 26+3=3 mod 26.

There is a convenient shorthand for the Caesar shift by n, given by x → x+n. It is confusing since here we are using x to stand for the position of a letter, and n to stand for the shift amount, i.e., x and n are each one of the values 1 … 26 rather than letters in the English alphabet. It is clear that since the shift is defined by the integer n this gives all 26 Caesar shift ciphers.

There is a bigger class of shift ciphers which can be written in these terms known as the affine shift ciphers, and they exploit the fact that we can multiply as well as add integers in modular arithmetic. It is slightly complicated to set up formally but rather easy to do in practice so we will work through an example.

The affine shift x→3x+5

We start as before with the position table, but this time instead of replacing a position x with the number x+3 we will replace it by the number 3x+5, where this number is interpreted appropriately. So for example 2 → 3x2+5 = 11, while 8 → 3x8+5 = 29 which is interpreted as 3, since 29=26+3. As with the Caesar shift cipher whenever the result of the computation is larger than 26 we keep subtracting 26 until it becomes smaller. More formally we compute 3x+5 and then take the remainder after division by 26. This gives us the following encryption/decryption table:

a	b	c	d	e	f	g	h	i	j	k	l	m
H	K	N	Q	T	W	Z	C	F	I	L	O	R
n	o	p	q	r	s	t	u	v	w	x	y	z
U	X	A	D	G	J	M	P	S	V	Y	B	E

The affine shift ciphers can also be written in shorthand form x → ax+b and the Caesar shift ciphers are special cases of the affine shift ciphers with a=1. We think of the pair of numbers (a,b) as the key to the cipher. It is an interesting question, that we will consider later, how many possible keys there are!
Notice that in both the Caesar shift x → x+3 and the affine shift x → 3x+5 the letter y is enciphered as B, since 25+3 = 28 = 26+2, and 3x25+5 = 80 = 3x26+2. It follows that two different affine shift ciphers can encrypt a letter in the same way, so it is no longer sufficient to discover the letter substituting for e in order to decipher the message. Mathematicians would say that there are "two degrees of freedom" in our choice of cipher so we might hope that deciphering two letters is sufficient. Luckily this is true, since if we know two values of the expression ax+b we can solve the two corresponding simultaneous equations to find the integers a and b.
We may be more familiar with this exercise when solving pairs of equations over the real numbers, but the same method works for modular arithmetic, with one important caveat.

We cannot always divide in modular arithmetic.

We will look more carefully at when division is allowed in a minute. For now it is worth noting that this caveat has an interpretation in cryptography. In order for the rule x → ax+b to define a cipher it had better be the case that each of the numbers 1 … 26 appears exactly once in the list of numbers ax+b as x ranges from 1 to 26. It doesn't matter which value we choose for the addition term b, but if we choose the multiplication factor a carelessly (so that we can't divide by a mod 26) this might not be the case.

For example the rule x →2x tries to encipher both m and z as Z, since 2.13=26 and 2.26 = 52 both of which are equal to 26 mod 26. Such an encryption cannot easily be deciphered since the recipient of the message is unable to determine whether the sender intended Z to be read as m or z.
From a mathematician's point of view the enciphering rule defines a function from the alphabet to itself, and this encryption function needs an inverse (the decryption function) if the cipher is to be decipherable in a deterministic way. In other words the number theory function x → ax+b needs to have an inverse in mod 26 arithmetic. It is a fact from elementary number theory that it will have such an inverse if and only if the multiplication factor a is coprime to 26, that is, the only common divisor for 26 and a is 1.
There are 12 numbers less than 26 and coprime to it (those odd numbers not divisible by 13) so we have 12 possible choices of the number a, and 26 choices for the number b, yielding 312 affine shift ciphers. This makes a brute force attack without frequency analysis less practical than the much simpler situation for Caesar shift ciphers, but certainly a small team could carry out such an attack quickly, and frequency analysis can help us to speed things up. In order to pin down the affine shift used we now need to identify two of the letters, say t and e, and then solve the corresponding equations to find the encryption key pair a and b.

Let's try an example. Suppose that we have been given a cipher text which we believe to be encrypted with an affine shift cipher, and that the two most common letters in the cipher text are S and L, appearing, respectively, roughly 12% and 9% of the time. We guess that this means e is encrypted as S and t as L. In terms of the modular arithmetic this tells us that

$$5a + b = 19 \bmod 26$$
$$20a + b = 12 \bmod 26.$$

As with ordinary simultaneous equations we can take the difference to deduce that

$$15a = -7 \bmod 26, \text{ or}$$
$$15a = 19 \bmod 26,$$

since counting back 7 from 26 gives 19.

It is tempting to solve this by dividing both sides by 15, to get a=19/15, but this won't work as a has to be an integer. What we really have to do is find the multiplicative inverse for 15 mod 26. This fancy phrase just means we need to find a number a' so that 15a' = 1 mod 26, or in other words find a' so that 15a' is 1 plus a multiple of 26. This would then allow us to deduce that a=(15a')a=(15a)a'=19a', so multiplying 19 by a' will give us a.

We could do this by trial and error. For each a the number a' will have to be one of the twelve odd numbers other than 13, and there are clever ways to try to solve for a', but to speed things up, here is a table of multiplicative inverses mod 26 for the twelve numbers that have them:

a	1	3	5	7	9	11	15	17	19	21	23	25
a'	1	9	21	15	3	19	7	23	11	5	17	25

So the a' we need for 15 is 7, and we get a=19x7=133=3 (mod 26).
Substituting the value of a=3 into the first equation 5a+b=19 gives us 5x3+b=19, which gives b=4, so our affine shift cipher is $x \to 3x+4$.

Polyalphabetic ciphers

The main weakness allowing us to tackle a substitution cipher is the irregularity in the distribution of letters in English text. Other languages demonstrate similar (though language specific) irregularities and you can find frequency tables for them on the web.

In order to remove this weakness from a cipher it is necessary to disguise the frequencies of letters in the plaintext and the easiest way to do this is by using a polyalphabetic cipher. In such a cipher each plaintext letter may be encoded in more than one way so that, for example, the letter e may be enciphered as both X and G within the cipher-text. One problem with this approach is that if X and G both encode for e we don't have enough letters left to encode the other 25 letters. One elegant solution to this problem is the famous French cipher known as the Vigenère cipher.

In a Vigenère cipher ANY letter might be encoded by any other; a given Vigenère cipher uses a subset of the 26 possible Caesar shift ciphers. Of course for a genuine recipient to have any hope of deciphering the message there has to be a way to determine for each cipher character which of the shifts has been used. The answer to this tricky problem is to choose a sequence of them known to both parties but to no-one else.

So the two parties might agree to use shifts of 22, 9, 7, 5, 14, 5 18, and 5 in that order and to continue repeating the pattern for the entire text: 22, 9, 7, 5, 14, 5 18, 5, 22, 9, 7, 5, 14, 5 18, 5, 22, 9 etc..

In order to decode the cipher text the recipient shifts the first cipher character back by 22, the second back by 9 and so on to recover the cipher text. Of course the question remains how one can memorise the correct sequence, but here we borrow an idea from the keyword cipher. The shift numbers 1, …, 26 are taken to stand for the alphabet a, …, z, and then the pattern 22, 9, 7, 5, 14, 5 18, 5 spells the word vigenere.

To set up a Vigenère cipher the two parties agree in advance to use the shift pattern encoded by some agreed keyword or phrase; in our previous Golden Jubilee Cipher challenge we used a Vigenère cipher based on the keyword GOLD, so characters were shifted in turn by 7, 15, 12, 4. Such a cipher is very hard to crack.

There are two very effective methods for tackling this cipher, one based on the analysis of repeated strings in the text, was discovered (independently) by the mathematicians Babbage and Kasiski. An analysis of repeated strings of letters is used to try to determine the length of the keyword, and once this is done a standard frequency analysis is applied to each part of the cipher-text encoded by a single cipher. A very good account of Babbage-Kasiski deciphering can be read in chapter 2 of Simon Singh's The Code Book.

The method we will use is based on the index of coincidence (ioc). This is, like bigram frequency analysis, based on the idea that there are hidden patterns in English which make certain letter combinations more likely than others. In this case we study the likelihood that if we pick two letters at random in the ciphertext then they will be the same. This probability is given by summing the numbers $(F_i^2-F_{i)}/(n^2-n)$, where n is the number of characters in the text we are analysing and the numbers F_i represent the number of occurrences of the letters of the alphabet in the text.

If we consider a standard long English text then the index of coincidence will be around 0.0686. On the other hand, if we compute it for a genuinely random sequence then each letter will appear roughly 1/26 times so the index of coincidence will be about 0.038466. A typical ciphertext will therefore have an ioc of something between these values. If the value is close to 0.0686 then it is likely that we are considering a simple substitution cipher like the shift or keyword ciphers we considered above, or perhaps a transposition cipher as considered later on in these notes.

If the ioc is lower then something more subtle is going on and we may well be looking at a polyalphabetic cipher. The weakness in the Vigenère cipher is that for some number k every kth letter in the cipher-text has been encrypted with the same Caesar shift, so if we knew k we could carry out a standard frequency analysis on those letters to figure out the shift. The index of coincidence can help us find k in the following way.

For each k from 1 to around 9, we split the text into blocks of size k. Next we read down the columns to extract the sequences of letters k apart in the text and compute the index of coincidence for each column. If our k is not the k used by the Vigenère cipher then the letters we are considering have been encrypted by

different shifts, so we would expect the index of coincidence to be low, whereas if we have found the correct k then it should be close to 0.0686 for each of the columns. The process is a little laborious, but also somewhat miraculous. The Vigenère cipher was the main diplomatic cipher across Europe for a long time, and widely considered intractable. The ioc attack makes it almost routine to decipher messages using it.

The Enigma Machine

The most famous polyalphabetic cipher in the world must surely be the Enigma cipher, implemented by the engineer Arthur Scherbius as the infamous Enigma machine. The machine was first designed around 1918 and was sold in large numbers to banks and commercial enterprises, before becoming the new standard for secure communications in the German military. Polish cryptographers, no doubt worried about the expansion of the powerful military next door, studied the machine in depth throughout the 1930s and as war broke out they risked their lives to share what they new with their colleagues in the UK, the fore-runners of the staff at GCHQ.

The machine is an electro mechanical device which implements a polyalphabetic cipher which in practice encrypts each individual letter of a message with its own custom cipher. Each of these is a cipher which switches the letters of the alphabet in pairs, so if a particular letter a is encrypted as G then if g had been at that position in the message it would have been encrypted as A. To know how a letter would be encrypted you need to know not just what letter it was, but whereabouts it is in the message and how the machine was configured when you started. If you set up two Enigma machines in exactly the same way and ask them to encrypt a message they will both encrypt it the same. Furthermore if one of them is given the encrypted message from the first one and reset to the original settings, then rather than encrypt it further it will decrypt it instead. That is because letters had been switched in pairs as we remarked above.

The “key” to an Enigma cipher is the way the machine is set up, and there are an incredibly large number of these. The military Enigma had over 158,962,555,217,826,360,000 different settings so brute force was never going to be enough to crack this code even with huge teams of codebreakers. (Compare this to the 250,000 keyword ciphers we discussed above).

Even getting hold of a military Enigma machine was impossible, but luckily the Polish cryptographer Rejewski had worked on reconstructing it from the little information the Polish cipher bureau could glean from intelligence reports and cipher analysis. Just nine days before the outbreak of war he and his colleagues handed everything they knew about the machine, its workings and its weaknesses to Commander Denniston, head of the British Government Codes and Cypher School and to Dilly Knox, the British chief cryptographer. Together with the large team of brilliant experts at Bletchley they completed the work that Rejewski had started and cracked the Enigma machine. The story has been told many times now, though for decades it remained the most closely guarded secret, not least because rotor machines like the Enigma continued to be used by governments around the word until the end of the 1970s.

A genuine Enigma machine would cost a lot of money now, but you can download several emulators for your computer, tablet or phone. Just search for EnigmaEnigma in your app store. You can even find a fully featured emulator on the web at

http://summersidemakerspace.ca/projects/enigma-machine/

These programmes are beautiful, and it can be great fun playing with the settings on them, but they don't necessarily help to understand the workings of the machine. Back in 2005, as part of the story for the National Cipher Challenge we invented the Pringle Can Enigma, which I think illustrates much better how it worked. Our design was based on the paper slips used by Turing and others in their original work on the cipher, but I am sure that if Pringles had existed they would have used the can!

You can make one yourself for the cost of a can of Pringles (and who doesn't like them). Since we introduced the Pringle Can Enigma to tackle the Fialka cipher in the National Cipher Challenge in 2005 a number of other people have produced their own variations on the theme and you will find them across the web. You can find out more on the resources page at

www.cipherchallenge.org/resources/

Even 70 years after the war this obsolete cipher machine continues to fascinate and enchant fans of spy-craft, codes and ciphers and steampunk engineering.

Transposition ciphers

Sometimes when you carry out a frequency analysis you will find that each letter occurs with about the same frequency as you would expect in natural English text (or whichever language you are studying). This is a broad hint that the text is not enciphered using a substitution cipher, but rather by an anagram or transposition cipher. In such a cipher the letters of the message are not replaced by substitutes, but rather jumbled using some rule which allows them to be untangled again to decipher the message. As with substitution ciphers we will need a key and this will be given by choosing a keyword, preferably with no repeated letters in it.
As an example we will encipher the text `The quick brown fox jumps over the lazy dog` using the keyword "BAD".

B	A	D
t	h	e
q	u	i
c	k	b
r	o	w
n	f	o
x	j	u
m	p	s
o	v	e
r	t	h
e	l	a
z	y	d
o	g	x

We write the keyword at the head of a table with three columns, then enter the plain-text in the boxes below. The last, empty, box is padded with an X (usually - there is no fixed rule for which character is used) so that all the boxes are full. Next we rearrange the columns so that the letters in the keyword are now in alphabetic order, ABD, and read off the rows grouping the letters in blocks of 5 for easy and accurate transmission:

```
HTEUQ IKCBO RWFNO JXUPM SVOET RHLEA YZDGO X
```

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

If the keyword contains repeated letters then we delete them as we would if it were the keyword for a substitution cipher before constructing the grid. Hence if the keyword was TOFFEE we would use a grid of width 4 with header TOFE and we would rearrange the grid so that its header appeared as EFOT to encipher the message.

How do we tackle such a cipher?

Clearly the length of the keyword is quite crucial. You should be able to guess this from the length of the cipher-text, which will be a multiple of it. So in our example the cipher-text has length 36 which has factors 2,3,4,6,9, 12 and 18. So we could try laying out the text in grids of these widths and examining the rows.

The best hope for a quick solution is to find a crib. If there is a word you think ought to appear in the cipher text then you could try looking for anagrams of that word. This is made difficult by the fact that the table splits the text into blocks (blocks of three in the example), and if your crib word does not take up an entire block then even the characters from the crib that do appear will be jumbled with other nearby characters, so you need a reasonably long crib. On the other hand if it is too long only part of the word will appear in that block so you are looking for anagrams of parts of the crib.

In our example if we knew, for some reason, that the text was likely to contain the word "jumps" we could look for anagrams of "JUM", "UMP" or "MPS". Looking carefully you should see the anagram PMS in the text and we might guess that the first and second columns have been transposed while the third has remained fixed. Checking this we find have cracked the cipher.

Things are harder with longer keywords but the principle remains the same. Things get tougher if the plaintext is not in our own language, since it is harder to say what makes sense. Of course even in this case it may be that part of the message is in your language and the rest in another. In this case you might hope to crack the cipher-text corresponding to your native language, and apply the knowledge that gives you about the cipher to write down a decrypt of the entire message, even when the text is unfamiliar.

Other (subtle) cribs: In English the letters q and u occur together so if they are separated either you are not looking at English text or they should be brought back together by undoing the anagram.

Numbers often represent dates, so for example the letters/numbers 2, 1, S, T in proximity might represent 21st, while 2 , 1 , T , H might represent 12th.

Toughening the transposition cipher

The transposition cipher described above can be made much more secure by reading the cipher-text off by columns rather than rows. So our message will read

```
HUKOF JPVTL YGTQC RNXMO REZOE IBWOU SEHAD X
```

Now the three letters P, M, S are nowhere near one another so you might think that anagramming won't help, but it can, once more, help us to work out the length of the keyword. This is the key to solving the cipher as it allows us to lay out the text in the appropriate grid. To see this in action look again for the letters P, M and S. They appear in the 7th, 19th and 31st position in the table, so they are 12 apart. Thinking about how the cipher works suggests that the encryption table could have 12 rows which is enough for us to get started. Even without a crib like the word JUMPS we could use the numerous cribs provided by the English language. The word "the" for example, or the fact that "q" is almost always accompanied by the letter "u".

Here is an example to try

```
SIEID ATTPW ADIVL SOLWO IYMRD AOSTT TDUHM AGTTT HSEOO
TAEST EOGNU AEDLN HNRDH KIWOA MENEE INEAS NPAIT SLIAI
AOJDN TCAET SOKEE EIULD HRAUE WSYSA IRBCT WNNSN TARHH
SUHAS MNOAG SVEPI AGINE IOAIS EBGRS TTWYO GTLNO EVMRT
WGTOI SAHHI ECAWP HTRAO TCRTS YRBYG
```

The cipher-text has 210 characters and 210=2x3x5x7 so possible key lengths are 2,3,5,6,7,10,14,15,21,30,35,42,70,105,210. We will first try the simple crib "the" tabulating the positions of, and then the distances between, the letters "T" and "H" in the text.

These are tabulated below. The first row of the table gives the positions of the letter H in the text, the first column gives the position of T and the entries in the

table are the absolute values of the difference of the positions, telling us how far apart each T is from each H.

	34	41	61	65	111	134	135	138	188	189	196
7	27	34	54	58	104	127	128	131	181	182	189
8	26	33	53	57	103	126	127	130	180	181	188
29	5	12	32	36	82	105	106	109	159	160	167
30	4	11	31	35	81	104	105	108	158	159	166
31	3	10	30	34	80	103	104	107	157	158	165
38	4	3	23	27	73	96	97	100	150	151	158
39	5	2	22	26	72	95	96	99	149	150	157
40	6	1	21	25	71	94	95	98	148	149	156
46	12	5	15	19	65	88	89	92	142	143	150
50	16	9	11	15	61	84	85	88	138	139	146
85	51	44	24	20	26	49	50	53	103	104	111
96	62	55	35	31	15	38	39	42	92	93	100
100	66	59	39	35	11	34	35	38	88	89	96
125	91	84	64	60	14	9	10	13	63	64	71
131	97	90	70	66	20	3	4	7	57	58	65
166	132	125	105	101	55	32	31	28	22	23	30
167	133	126	106	102	56	33	32	29	21	22	29
172	138	131	111	107	61	38	37	34	16	17	24
180	146	139	119	115	69	46	45	42	8	9	16
183	149	142	122	118	72	49	48	45	5	6	13
197	163	156	136	132	86	63	62	59	9	8	1
201	167	160	140	136	90	67	66	63	13	12	5
204	170	163	143	139	93	70	69	66	16	15	8

We are looking for patterns here thrown up by the fact that T and H would often have been adjacent in a row, and so after permuting the columns the distance between them will be a multiple of the column height. Also the column height will be a divisor of 210, so we are looking for common occurrences of a multiple of a divisor of 210. To do that we tabulate the greatest common divisors of each of the distances with 210.

gcd	34	41	61	65	111	134	135	138	188	189	196
7	3	2	6	2	2	1	2	1	1	14	21
8	2	3	1	3	1	42	1	10	30	1	2
29	5	6	2	6	2	105	2	1	3	10	1
30	2	1	1	35	3	2	105	6	2	3	2
30	3	10	30	2	10	1	2	1	1	2	15
38	2	3	1	3	1	6	1	10	30	1	2
39	5	2	2	2	6	5	6	3	1	30	1
40	6	1	21	5	1	2	5	14	2	1	6
46	6	5	15	1	5	2	1	2	2	1	30
50	2	3	1	15	1	42	5	2	6	1	2
85	3	2	6	10	2	7	10	1	1	2	3
96	2	5	35	1	15	2	3	42	2	3	10
100	6	1	3	35	1	2	35	2	2	1	6
125	7	42	2	30	14	3	10	1	21	2	1
131	1	30	70	6	10	3	2	7	3	2	5
166	6	5	105	1	5	2	1	14	2	1	30
167	7	42	2	6	14	3	2	1	21	2	1
172	6	1	3	1	1	2	1	2	2	1	6
180	2	1	7	5	3	2	15	42	2	3	2
183	1	2	2	2	6	7	6	15	5	6	1
197	1	6	2	6	2	21	2	1	3	2	1
201	1	10	70	2	30	1	6	21	1	6	5
204	10	1	1	1	3	70	3	6	2	15	2

There are a number of different entries in this table corresponding to different possible column heights, and the number of times each appears is given in the following table:

Height	1	2	3	5	6	7	10	14	15	21	30	35	42	70	105
Count	60	64	26	15	27	6	12	5	7	6	9	4	6	3	3

S	T	H	A	I	A	W
I	D	N	O	R	G	G
E	U	R	J	B	I	T
I	H	D	D	C	N	O
D	M	H	N	T	E	I
A	A	K	T	W	I	S
T	G	I	C	N	O	A
T	T	W	A	N	A	H
P	T	O	E	S	I	H
W	T	A	T	N	S	I
A	H	M	S	T	E	E
D	S	E	O	A	B	C
I	E	N	K	R	G	A
V	O	E	E	H	R	W
L	O	E	E	H	S	P
S	T	I	E	S	T	H
O	A	N	I	U	T	T
L	E	E	U	H	W	R
W	S	A	L	A	Y	A
O	T	S	D	S	O	O
I	E	N	H	M	G	T
Y	O	P	R	N	T	C
M	G	A	A	O	L	R
R	N	I	U	A	N	T
D	U	T	E	G	O	S
A	A	S	W	S	E	Y
O	E	L	S	V	V	R
S	D	I	Y	E	M	B
T	L	A	S	P	R	Y
T	N	I	A	I	T	G

Column heights of 1,2,3,5,6,7,10 all seem unlikely given that the keyword or phrase would then have to have at least 21 letters in it. On the other hand a column height of 30 would correspond to a keyword of length 7, which is quite feasible, and gives rise to a good number (9) of TH adjacencies, as marked in green in the corresponding 30x7 grid.

Notice that in three cases, rows 8, 9 and 16 the T and H appear in columns 2 and 7 respectively. This suggests that whatever order the columns should be in we should end up with column 2 next to (and to the left of) column 7. In two of the rows, 9 and 16, there is an E in the fourth entry so we are led to try putting these three columns together in the order 2,7,4.

Assuming this is not an Olde English text, ruling out “Twas” as a word, these three columns are not likely to be the first three, so we need something to the left and the possibilities for that put S,H, I or A to the left of the string TWA in the first row. Trying each in turn we get STWA, HTWA, ITWA or ATWA and the first two seem unlikely. Playing the odds and considering the possibilities for arranging the remaining three letters on the top row we are led to consider ITWAS.

One possibility for the remaining columns reads AHITWAS but then the next row reads GNRDGOI which is clearly wrong. There is one other way to rearrange the columns to get this first row, but that is also unlikely as it gives the second row ONRDGGI. On the other hand these same letters might suggest the word GOING in row 2 and a rearrangement and further experimentation gives the final arrangement, which you might recognise from earlier:

I	T	W	A	S	H	A
R	D	G	O	I	N	G
B	U	T	J	E	R	I
C	H	O	D	I	D	N
T	M	I	N	D	H	E
W	A	S	T	A	K	I
N	G	A	C	T	I	O
N	T	H	A	T	W	A
S	T	H	E	P	O	I
N	T	I	T	W	A	S
T	H	E	S	A	M	E
A	S	C	O	D	E	B
R	E	A	K	I	N	G
H	O	W	E	V	E	R
H	O	P	E	L	E	S
S	T	H	E	S	I	T
U	A	T	I	O	N	T
H	E	R	U	L	E	W
A	S	A	L	W	A	Y
S	T	O	D	O	S	O
M	E	T	H	I	N	G
N	O	C	R	Y	P	T
O	G	R	A	M	A	L
A	N	T	U	R	I	N
G	U	S	E	D	T	O
S	A	Y	W	A	S	E
V	E	R	S	O	L	V
E	D	B	Y	S	I	M
P	L	Y	S	T	A	R
I	N	G	A	T	I	T

“It was hard going, but Jericho didn’t mind. He was taking action, that was the point. It was the same as code-breaking. However hopeless the situation, the rule was always to do something. No cryptogram, Alan Turing used to say, was ever solved by simply staring at it.”

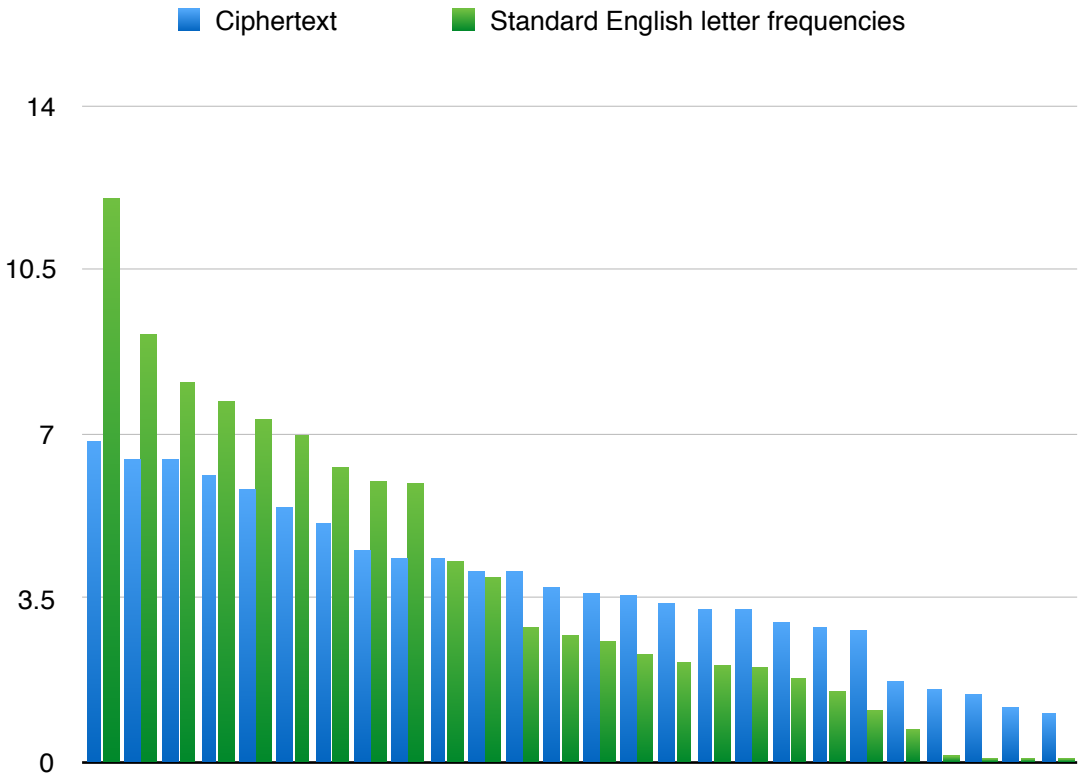
We stared pretty hard at this, but there was nothing simple about breaking it. I think Jericho, and maybe even Alan Turing, would approve.

A challenge

Now let's try to put together everything we have learned to crack the following cipher-text which is from a BBC news article about encryption in the modern world.

```
XSFJD JMNRF RUDJV LMYFT GWWHP TUDIA HWRMS XXAHJ DNBRH
QTOFF NWFGH GLDJJ ATQWH UEQEM DMHRH LMCGL ZAYBT HUWIC
MHDJI CGFVZ TJHWR FYBXB HTTLX AHFLY MHDKM ZKTPS SUMRH
FHLRU WATHU JVLTQ LZSGS NAFWL WUGXD UYCHS WZJWH SIAIY
GYLSQ CMDDF IMXHX JNNRY REFEX NWHTM LNEDJ CYDRM HIGXL
VJLXQ HUYLH SLUYL TSVSH NBTQK FHWTQ DNHXU DQRYG YVSQF
MMRKJ QHZOV SIMGH HTMLN EDJQB YKGZN XSFJD JMNRF XUBIG
JRUKP PSSOE NVSXY GNRJQ YVYXJ JLBSF JDJMT JJFJA DDLYB
XZQAA YKXLL DIYXX JWYRF WAYML NPHQY LYHFH LRUWA THBXD
DQUUT XLYLT SVXTL FNQYN HMJOD NABGO WSOFG HJXIK YHPYM
HZQVX UGILE FAXXL FYITX WJJUF TIFTH LJQKJ NAJUW FLXRD
FDGTS BOFSL YRHJL YTUEY BTYWJ FHLKR JRUMN RFXIF JVLWU
BLKLK IKBDJ IUGIV GRYOJ UQHIF UOWCG HXWAS PHQYW XQTUS
ASAEJ WLJLL KRJSO FGHJX UGIXK JGTYK KYIWT WZJNK FQKKI
KRDLN IGMRO JPXWQ GRUMY HJBBB HKEJN ATGAX OLJGL MYKJV
MQNBS JKHLT REDJX WFWSX NKJDE XBHZO VLCOJ QGMCG YVSGI
NYKGB CMBDK JHVWB HYYWI XJNHZ BRJQX PFUAN NAJDD QCXXV
UTLXI VGRYG TWSGF XALUY IKNHK FATNQ KYNAJ JWWGT SVTJW
TZVWY BXNUW SWKDS LNIGX BKYYF XGAIH HYVMK ZBHLW SNEDV
UWUFG OWRYL XDYJM KNJGW INXPS YBXRD LNWTQ DFFFR XLKGS
TQOAJ XVTGW HLTHN WWMEF LVGUK JSSYN XWQKM CWIHF BCMML
FYBXR HKXUZ JVSSX NXHVY BXRWG WYVWH SYYMM HEFWA NQWZM
XIWGJ HVWBH YNAJP LMILJ FGIYL WHNTF OJGSW INSGL MYNXH
GKMXH UWYEX DVLMU MBHJJ MAFUW IUFTQ YYBHX HOMIG JHVJX
MTFGR GNSLU FNXXH UZLXQ BLMYL JDJJE GTZFF MLDPE JNKNF
WSWKD SLNIG XBKYY FXDFI BTAHS BYTPQ WXMBS WZFNX AHJDI
GJLFA IEAHV MULYR HTMLJ VKYBX XDEJM XYRXX YVWHL PYRX
```

First we carry out frequency analysis of the whole text and compare it to the standard distribution in English. The following chart allows us to compare the



frequency distributions, and we see that the cipher-text distribution, while not uniform is much flatter and lacks the distinctive spike at the left, suggesting that the frequency distribution of letters is not a good match to the standard English language. From this we conclude that the text is not encrypted with a transposition or a mono alphabetic substitution cipher like one of the shifts, or the keyword substitution we studied above.

So we guess that the text has been encrypted with a polyalphabetic cipher, and since we only know about the Vigenère cipher we will assume that is what we have here.

The first step is to try to find the likely keyword length, which we will denote k, which is at least 2 since we are not considering a mono-alphabetic substitution. To do this we will compute the index of coincidence for sequences of letters spaced k apart in the cipher-text. Start by taking k=2. We consider the sequence of every

other letter XFDMRRDVMFW… HPR, starting at the first. This sequence has index of coincidence 0.04695494261123 which is not close to the ioc of standard English text.
Next we try k=3 and examine the sequence XJMF… of every third letter. This has ioc 0.054357657988021 which is closer to the standard but still not good. Taking k=4,5,6,7 in turn we get the following table of iocs.

k	2	3	4	5	6	7	8	9
ioc	0.04695	0.05435	0.04616	0.047614	0.069209	0.046907	0.047228	0.04809

Notice that for k=6 we obtain an ioc of 0.069209, which is very close to the expected value of 0.0668 for English text, whereas the other values of k give a much lower value, which suggests that key length is 6.

The next step is to split the text into blocks of 6 and to carry out frequency analysis on each of the 6 columns this gives us. Here is the first of the six sequences that gives us

XMDFPHXBFHAEHGTMGHXXMKMRULAGHHGMXRXLYGXHTBWXGMHMLBXMBKEGV
BMAXKYRLLRBUTFMBFKHGXTFLAXTLYTLMFBKGOFHHTELFGTWKKGXMBALKB
TWKHOGNMVWZPAXXGXKTATTXKGFHBEFLKNXTRTVTEKXWMXZNXVMAXVALLO
NNXXMAFHGMNXXLGLKKGFTTBXGELLXXVR

The most common letter by far is X, so we deduce that e has been encrypted by X in this sequence, and since the Vigenère cipher uses Caesar shift ciphers this gives a decrypt of

etkmwoeimohlonatnoeetrtybshnoonteyesfneoaidentotsietirlnc
itherfyssyibamtimroneamsheasfastmirnvmooalsmnadrrnetihsri
adrovnutcdgwheenerahaaernmoilmsrueayacalredteguecthechssv
uueethmontueesnsrrnmaaienlsseecy

If this looks like nonsense, don't worry, it is. This just gives us the decrypt of the first, seventh, thirteenth, twentieth letters of the plain-text and so on. We have to decipher the other five columns and inter-splice them to get the decrypt.
The next sequence, which begins SNJT is a little tougher. The frequency analysis shows that J is the most common letter at 11.33% and then X at 10.84%, so either of the shifts e to J or e to X is possible. We will leave it there for the moment and move on to the third column beginning FRV … Here H is clearly the most common letter so we assume that the shift cipher mapping e to H has been used and see if we can use our knowledge that the is a very common triple to settle the ambiguity over column 2.

To do so we look for the pattern t_e across the first three columns after decrypting columns 1 and 3. We find this pattern in rows 23, 48,109,164 and 176 where the encryption string is MRH, MGH, MYH, MMH and MBH, so if any of these are an encryption of the then h must be encrypted as R, G, Y, M or B. These correspond to the affine shifts mapping e to O, D, V, J or Y. We have already seen from our frequency analysis that the most likely encryption of e is either to map it to J or to X, and putting this together with the list we just produced that makes the mapping to J more likely so we assume that our second column is enciphered using a shift mapping e to J and make that substitution.

enc_ _ _tio_ _ _kes_ _ _mod_ _ _wor_ _ _oro_ _ _eve_ _
ime _ _mak_ _ _obi_ _ _hon_ _ _llb_ _ _ome_ _ _ngw_ _
acr _ _tca_ _ _nas_ _ _oro_ _ _ewe_ _ _eve_ _ _tca_ _
rom _ _tme_ _ _ypt_ _ _bes_ _ _sup_ _ _hat_ _ _nsa_ _
ont _ _onf_ _ _nti_ _ _tya_ _ _ecu_ _ _yto_ _ _eit_ _
sib _ _fyo_ _ _nsi_ _ _ele_ _ _oni_ _ _ans_ _ _ion_ _
don _ _epa_ _ _nts_ _ _tho_ _ _oul_ _ _tbe_ _ _sib_ _
ith _ _enc_ _ _tio_ _ _idd_ _ _rkm_ _ _lis_ _ _nio_ _
ctu _ _inc_ _ _tog_ _ _hya_ _ _eun_ _ _rsi_ _ _fsu_ _
yat _ _sim_ _ _ste_ _ _ypt_ _ _isa_ _ _bou_ _ _ans_ _
min _ _tel_ _ _ibl_ _ _mbe_ _ _rte_ _ _oun_ _ _ndi_ _
esi _ _ast_ _ _mof_ _ _sen_ _ _her_ _ _ema_ _ _any_ _
sto _ _for_ _ _att_ _ _sfo_ _ _tio_ _ _mes_ _ _igh_ _
rwa _ _nds_ _ _ver_ _ _mpl_ _ _ost_ _ _olv_ _ _app_ _
let _ _sfo_ _ _mbe_ _ _ndu_ _ _ath_ _ _dot_ _ _ran_ _
rma _ _nho_ _ _ern_ _ _tte_ _ _ich_ _ _hod_ _ _sed_ _
res _ _ing_ _ _amb_ _ _dat_ _ _rea_ _ _oul_ _ _ven_ _
nts _ _uth_ _ _twa_ _ _cry_ _ _ddu_ _ _gwo_ _ _war_ _
hea _ _ess_ _ _eds_ _ _not_ _ _evi_ _ _rie_ _ _ain_ _
heg _ _ans_ _ _aus_ _ _eir_ _ _ryp_ _ _nsy_ _ _msd_ _
ots _ _ici_ _ _lys_ _ _mbl_ _ _ssa_ _ _rig_ _ _usm_ _
ema _ _ala_ _ _ysi_ _ _all_ _ _cod_ _ _ack_ _ _lai_ _
rep _ _ern_ _ _dde_ _ _thi_ _ _eme_ _ _ges_ _ _use_ _
emt _ _cre_ _ _the_ _ _hin_ _ _edt_ _ _cry_ _ _hem_ _
sec _ _sre_ _ _ved_ _ _und_ _ _use_ _ _ecr_ _ _eys_ _
twe _ _har_ _ _mon_ _ _ose_ _ _nee_ _ _toc_ _ _uni_ _
ese _ _ely_ _ _sea_ _ _now_ _ _sym_ _ _ric_ _ _ryp_ _
nsy _ _msa_ _ _ave_ _ _akn_ _ _int_ _ _eve_ _ _nei_ _
lve _ _sto_ _ _ses_ _ _esa_ _ _eto_ _ _cre_ _ _ys

Now we are getting somewhere. We know this is an article about encryption, and right at the start of the text we see the pattern `enc_ _ _tio_`. This corresponds to the cipher-text `XSFJD JMNRF` suggesting that `ryp` in positions 4,5,6 have been enciphered as `JDJ` in turn using shifts mapping `r` to `J`, `y` to `D` and `p` to `J`. Trying the `J` to `r` shift as the decrypt on the fourth column, the `D` to `y` to the fifth and the `J` to `p` on the sixth gives us the following

```
encryptionmakesthemodernworldgoroundeverytimeyoumakeamobi
lephonecallbuysomethingwithacreditcardinashoporonthewebor
evengetcashfromanatmencryptionbestowsuponthattransactiont
heconfidentialityandsecuritytomakeitpossibleifyouconsider
electronictransactionsandonlinepaymentsallthosewouldnotbe
possiblewithoutencryptionsaiddrmarkmanulisaseniorlecturer
incryptographyattheuniversityofsurreyatitssimplestencrypt
ionisallabouttransformingintelligiblenumbersortextsoundsa
ndimagesintoastreamofnonsenseherearemanymanywaystoperfor
mthattransformationsomestraightforwardandsomeverycomplexm
ostinvolveswappinglettersfornumbersandusemathstodothetran
sformationhowevernomatterwhichmethodisusedtheresultingscr
ambleddatastreamshouldgivenohintsabouthowitwasencrypteddu
ringworldwariitheallliesscoredsomenotablevictoriesagainstt
hegermansbecausetheirencryptionsystemsdidnotsufficientlys
cramblemessagesrigorousmathematicalanalysisbyalliedcodecr
ackerslaidbarepatternshiddenwithinthemessagesandusedthemt
orecreatethemachineusedtoencryptthemthosecodesrevolvedaro
undtheuseofsecretkeysthatwereresharedamongthosewhoneededtoc
ommunicatesecurelythesareknownassymmetricencryptionsyste
msandhaveaweaknessinthateveryoneinvolvedhastopossessthesa
mesetofsecretkeys
```

The shift ciphers used have therefore been shifts by 19,5,3,18, 5, 20 respectively. How would the spies have remembered this sequence? It might have been chosen

as the lottery numbers one week, but actually it spells out the word SECRET, with our usual convention a=1, b=2, c=3 and so on.

It may feel like we cheated a bit using the crib, but that is how real world cipher cracking works. Modern ciphers are highly sophisticated algorithms designed, as far as possible, to conceal the patterns and rhythms of language so that simple frequency analysis is at best unreliable, and on its own hopelessly inadequate. Sometimes a crib is what you need, and since there is always a context to any communication a crib is often available.
In the war much was made of the fact that the naval enigma was used to transmit weather reports. Comparing these with reports from Allied vessels in the same area was sometimes all it took to crack open the key to that day's transmissions. Careless use of call signs or mission codewords can also fatally weaken the security of a cipher.

This was a far from easy exercise, and it used everything we know about letter frequencies, common patterns, cribs and the index of coincidence. Combining them has allowed us to decipher a message that would have defeated all but the best cryptographers in the past.

Deciphering a secure message is a combination of hard work, luck, knowledge and skill. But above all it takes perseverance. When one tool lets you down you need to try another, and another. And another.

As Turing said,

> “No cryptogram was ever solved by simply staring at it.”

To apply a Caesar shift turning a into D, (the shift x+4)rotate the inner wheel until the “D” on its outer ring lines up with “a” then read your message one letter at a time from the outer wheel to the inner.

To apply the affine shift corresponding to the function 3x instead, just use the inner wheel. Read your plain text letters on its outer rim and replace them with the corresponding letters on the second ring. The third ring corresponds to the affine shift 5x.

The middle ring is for you to use to design your own cipher. Fill it in with the letters of the alphabet in any order you like. A good way to do that is to use a key word or phrase as described above.