345 (City of Lancaster) Squadron

Guide to Cryptography and Cryptanalysis

v2.2

Shift Cipher (a.k.a. **Caesar Cipher**)

This is a very basic monoalphabetic substitution cipher and is quite easy to break even without cryptanalysis skills.

Each letter in the cipher text is a fixed number of letters away from the letter in plain English.

To encipher a message, you just need to select a number by which to shift and then replace each letter with the one that many positions away from it in the alphabet. For example, with a shift of 4, A would become E; and Y would become C (as you loop back to A after you hit Z). As long as both the sender and receiver know the shift number, all that you have to do to decipher the message is shift them back in the other direction.

If you do not know the shift number, then you will have to work it out to be able to decipher the message. The best way to do that is to pick one word from the cipher text, and then write out each combination of letters that it could be (knowing that all of the plain text letters are the same distance away from the cipher text letters). I can then see which of makes sense as a word, which will reveal the shift number, allowing me to decipher the rest of the message.

For example, here is some ciphered text:

DOO WKDW LV JROG GRHV QRW JOLWWHU

I don’t want to pick a word that is too short (as it could have more than one possibility), so I will go with JOLWWHU. I will then just write out the alphabet after each one, continuing from Z-A if I reach the end, in order to reveal the word, and therefore the shift:

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

As you can see, there is only really one possibility that makes a word (GLITTER), and the answer is that all of the plain text letters are 3 to the left of the cipher text letters (which is easier to work out than 23 to the right…) letters (e.g. J = G, because G is 3 letters before J in the alphabet).

Using this information, we can decipher out the whole message as normal with a -3 shift (D - 3 = A etc…):

DOO WKDW LV JROG GRHV QRW JOLWWHU
ALL THAT IS GOLD DOES NOT GLITTER

Breaking a Shift Cipher

It is relatively straightforward to break a shift cipher manually using a *brute force* approach where you write the cipher vertically, then wrote out the rest of the alphabet (restarting at A when you reach Z). The column number that reads as plain English is the shift number to decipher.

Note that, if you get a number >13, you can make life easier for yourself by subtracting 26 to get an easier shift. For example, a shift of 23 is easier to implement as a shift of -3.

See below for an example, where the shift revealed to be -3 (23):

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	**A**	B	C
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	**L**	M	N
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	**L**	M	N
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	**T**	U	V
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	**H**	I	J
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	**A**	B	C
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	**T**	U	V
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	**I**	J	K
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	**S**	T	U
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	**G**	H	I
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	**O**	P	Q
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	**L**	M	N
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	**D**	E	F
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	**D**	E	F
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	**O**	P	Q
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	**E**	F	G
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	**S**	T	U
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	**N**	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	**O**	P	Q
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	**T**	U	V
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	**G**	H	I
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	**L**	M	N
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	**I**	J	K
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	**T**	U	V
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	**T**	U	V
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	**E**	F	G
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	**R**	S	T

Keyword Cipher

This is another monoalphabetic substitution cipher. It is harder to break than the shift cipher, but can usually be achieved using basic cryptanalysis.

Each letter is translated by creating a tool using a **keyword**. This achieved by the following steps:

1. Write out your **keyword**, omitting any repeated letters
2. Follow it with all of the rest of the letters in the alphabet that do not appear in the keyword, in alphabetical order
3. Underneath, write the alphabet as normal

For example, if the keyword was **Lancaster**, then the tool would look like this:

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

You can then decipher messages by replacing letters from the top line (cipher text) with the corresponding letter from the bottom line (plain text). For example, imagine that the keyword was **LANCASTER** and you wanted to decipher the message below:

```
TOJH QRS LPRSP L TBOS PRLGG AS WJFSI
```

Using the above tool, T would become F (as shown in yellow), O would become R, and so on, until you have deciphered the message. The deciphered message is therefore:

```
TOJH QRS LPRSP L TBOS PRLGG AS WJFSI
FROM THE ASHES A FIRE SHALL BE WOKEN
```

You can encipher the message in exactly the same way, by replacing letters from the bottom line (plain text) with those from the top line (cipher text):

```
FROM THE ASHES A FIRE SHALL BE WOKEN
TOJH QRS LPRSP L TBOS PRLGG AS WJFSI
```

Breaking a Keyword Cipher

If the **keyword** is not known, then the message can only be deciphered using **cryptanalysis**. Cryptanalysis is extremely complex subject that cannot be summarised in a short guide. However, when you are dealing with relatively simple ciphers (such as the keyword cipher), then there are some good tips that can help you to get started.

The first of these is **frequency analysis**. This is based on the idea that different **words** and **letters** typically appear more and less frequently in the English language, and with a little understanding of this it can be possible to identify them. This is something that works better with longer pieces of text: the shorter that the text is, the less likely it is to follow the expected patterns of letter and word frequency.

For example, here is some enciphered text:

```
LGG QRLQ BP EJGC CJSP IJQ EGBQQSO
IJQ LGG QRJPS WRJ WLICSO LOS GJPQ
QRS JGC QRLQ BP PQOJIE CJSP IJQ WBQRSO
CSSK OJJQP LOS IJQ OSLNRSC AY QRS TOJPQ
TOJH QRS LPRSP L TBOS PRLGG AS WJFSI
L GBERQ TOJH QRS PRLCJWP PRLGG PKOBIE
OSISWSC PRLGG AS AGLCS QRLQ WLP AOJFSI
QRS NOJWIGSPP LELBI PRLGG AS FBIE
```

If I wanted to decipher this and did not know anything about the type of cipher that had been applied to it (or I did know, but didn’t have the **keyword**), then I could try some **word frequency analysis** and **letter frequency analysis** using the following rules of thumb:

1. The most common letters in the English language are (in order): **E T A**, followed by **O I N H S R**; the rarest are **Q**, **Z** and **J**.
2. The most common pair of letters is **TH**, and the most common triplet is **THE**
3. A pair of identical letters is likely to be one of: **EE FF LL MM OO SS TT**
4. A single letter is likely to be either **A** or **I**
5. A letter following an apostrophe (‘) is likely to be **S**. Two identical letters are likely to be **LL**.

Using this information, you can count how many times each letter appears:

S	29
Q	24
J	21
L	21
P	21
G	19
R	19
O	16
I	13
C	10
B	9
W	8
E	7
A	6
T	4
F	3
H	2
K	2
N	2

Y 1

S and Q appear the most often, and the most frequent word is QRS, so we can be reasonably sure that this is THE. We can therefore go through the whole message and swap Q for T, R for H and S for E.

L appears on its own, so is likely A or I. It also appears at the start of LGG, which could be ALL (not many words start with I and then two identical letters, unless they have an apostrophe). Let’s therefore change L to A and G to L *(remember, you can always change it back if you are wrong!)*.

In the case of keyword ciphers, it is also worth remembering that: all of the letters after the latest letter in the alphabet that is used in the **keyword** will be correct. For example, where the keyword is **LANCASTER**, the latest letter in the alphabet is **T**, and so all letters after T are unchanged in the cipher:

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

W (8 appearances) and Y (1 appearance) are therefore quite likely to be correct, so we can fill them in. This gives us a new word that we have nearly solved: **WRJ.** If we now think that W is correct, and we already think that R is H, then J is probably O (as the only word that it is likely to be is WHO).

Of the letters that were found more than twenty times in the frequency analysis, we now only need P (found 21 times), and it is very likely that this is one of O I N S R. Because it often appears at the end of words (e.g. `CJSP`), including two letter words (e.g. `BP`), I think that this would be a good candidate for S. In this case, B is probably I, as the two-letter word ending with S is probably IS; let’s swap those out too.

Already we are making good progress here:

```
ALL THAT IS  OL   OES  OT  LITTE
LGG QRLQ BP EJGC CJSP IJQ EGBQQSO

 OT ALL THOSE WHO WAN E  A E LOST
IJQ LGG QRJPS WRJ WLICSO LOS GJPQ

THE OL  THAT IS ST ON   OES  OT WITHE
QRS JGC QRLQ BP PQOJIE CJSP IJQ WBQRSO

 EE   OOTS ARE  OT  EA HE  BY THE   OST
CSSK OJJQP LOS IJQ OSLNRSC AY QRS TOJPQ

  O  THE ASHES A  IRE SHALL BE WOKE
TOJH QRS LPRSP L TBOS PRLGG AS WJFSI

A LIGHT   O  THE SHA OWS SHALL S  ING
L GBERQ TOJH QRS PRLCJWP PRLGG PKOBIE
```

```
 E EWED SHALL  E  LA E THAT WAS   O E .
OSISWSC PRLGG AS AGLCS QRLQ WLP AOJFSI

THE   OW LESS AGAI  SHALL BE  I  .
QRS NOJWIGSPP LELBI PRLGG AS FBIE
```

Now we can start looking for patterns and making some educated guesses. For example, JGC is OL*, which is probably OLD, which means that and EJGC is probably GOLD. So now we have O and G!

IJQ is *OT and is followed by ALL…, which is probably NOT ALL…; and so now we have N.

At this point, the first line is clearly ALL THAT IS GOLD DOES NOT GLITTER, which means that we now have D and R, and by keeping going on like that, it isn't long before we have the whole lot!

ALL THAT IS GOLD DOES NOT GLITTER
LGG QRLQ BP EJGC CJSP IJQ EGBQQSO

NOT ALL THOSE WHO WANDER ARE LOST
IJQ LGG QRJPS WRJ WLICSO LOS GJPQ

THE OLD THAT IS STRONG DOES NOT WITHER
QRS JGC QRLQ BP PQOJIE CJSP IJQ WBQRSO

DEEP ROOTS ARE NOT REACHED BY THE FROST
CSSK OJJQP LOS IJQ OSLNRSC AY QRS TOJPQ

FROM THE ASHES A FIRE SHALL BE WOKEN
TOJH QRS LPRSP L TBOS PRLGG AS WJFSI

A LIGHT FROM THE SHADOWS SHALL SPRING
L GBERQ TOJH QRS PRLCJWP PRLGG PKOBIE

RENEWED SHALL BE BLADE THAT WAS BROKEN
OSISWSC PRLGG AS AGLCS QRLQ WLP AOJFSI

THE CROWNLESS AGAIN SHALL BE KING
QRS NOJWIGSPP LELBI PRLGG AS FBIE

Remember – this is about trial and error – don't be afraid to take a guess and then undo it later – you will get there in the end!

Alphabet Cipher (simplified **Vigenère Cipher**)

This is a polyalphabetic cipher that is not easy to break using simple cryptanalysis.

Each letter is enciphered and deciphered using the below grid and a **keyword**:

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

For example, imagine that the keyword was **`LANCASTER`** and you wanted to decipher the message below:

```
EHR QLV MLRE IF UTJHRX OORU NGM AZEHRT
```

The first thing to do is to write your keyword across the top of the letters, repeating as many times as is necessary.

```
LAN CAS TERL AN CASTER LANC AST ERLANC
EHR QLV MLRE IF UTJHRX OORU NGM AZEHRT
```

You then decipher the message by finding the letter from your top line (`L`) on the left-hand side of the grid, then following it along the row until you find the first letter in your bottom line (`E`). Then from there, go up to the top, and it tells you which letter it should be (`T`: see the yellow line in the diagram above). If you repeat this process with the rest of the letters, you will decipher the message:

```
EHR QLV MLRE IF UTJHRX OORU NGM AZEHRT
THE OLD THAT IS STRONG DOES NOT WITHER
```

To encipher a message using this approach, you simply use the same grid and keyword. Once again, write your keyword above your message, repeating it as many times as is necessary. For example:

```
LAN CAS TERL AN CASTER LANC AST ERLANC
THE OLD THAT IS STRONG DOES NOT WITHER
```

Now, you simply find your new letters by finding the point of intersection between the letter on the top line, and the one on the bottom. For example, the first letter in this cipher is E, as this is the intersection between L and T, as can also be seen in the yellow lines in the grid above. The enciphered message is therefore (as we already know…):

```
EHR QLV MLRE IF UTJHRX OORU NGM AZEHRT
```

Breaking a Simplified Vigenère Cipher

Polyalphabetic ciphers such as these are unlikely to be broken by hand unless you have some information to start with (sometimes called a crib). Though the Simplified Vigenère is theoretically less secure than the “full” Vigenère cipher, it is also a little more unusual than the “full” version, and I do not know of any tools designed specifically to attack it (though I would expect that a decent cryptanalyst would be able to break it relatively easily using general-purpose cryptanalysis software).

Vigenère Cipher

This is the classic polyalphabetic cipher – it is much more challenging to break with cryptanalysis, but can be broken using a brute-force attack with a computer.

A ‘proper’ Vigenère Cipher is similar to the alphabet cipher above, but is based on a keyword, making it harder to reconstruct the matrix to decrypt it.

Whereas the alphabet cipher uses a different alphabet to encipher each letter of the alphabet, the Vigenère cipher uses a different alphabet for each letter *in the message*.

For example, if the keyword was **CADET**, then you would construct a matrix like this:

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

Now, say that you are going to encipher the word **LANCASTER**, then you would simply take the L column, and then get the corresponding letter from the first row (beginning C), which is **N**. You then get the A from the second row (which is also A), the N from the third row (**Q**), and so on. When you get to the last row, you simply return to the top row and keep repeating until the message is enciphered, in this case resulting in:

LANCASTER
NAQGTUTHV

To decipher an enciphered message, simply find the column that each letter is in, working through the rows one by one (then back to the start) in the same way as before:

NAQGTUTHV
LANCASTER

This is a simple example just to explain the operation, but it is important to note that the encryption becomes more secure the longer the keyword is.

Using the same grid above, we can encipher and decipher a longer text to demonstrate:

A LIGHT FROM THE SHADOWS SHALL SPRING
C LLKAV FUSF VHH WACDRAL UHDPE UPUMGI
A LIGHT FROM THE SHADOWS SHALL SPRING

Breaking a Vigenère Cipher

Polyalphabetic ciphers such as these are unlikely to be broken by hand unless you have some information to start with (sometimes called a crib). However, they are vulnerable to computer-based attacks.

A good example of an online tool for attacking a Vigenère cipher is available here: https://www.dcode.fr/vigenere-cipher. To decipher the above, you would need to attack it in two ways: first to determine the length of the keyword, and then using that knowledge you can undertake a brute force attack.

Telegraph Cipher

This is very similar to a Vigenère cipher, but it is a little easier to use as you don't need to create a matrix. It is also a little more elegant because the method for enciphering and deciphering is identical (just like the Enigma).

Each letter in the cipher text is derived from the below tool and a **keyword**. The tool shown below should be in two pieces, each containing a copy of the alphabet:

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

For example, imagine that the keyword was **LANCASTER** and you wanted to decipher the message below:

```
IWJN JEFLZ LJJ PMZ CARJTJZ ZU AXN GJZKH
```

The first thing to do is to write your keyword across the top of the letters, repeating as many times as is necessary:

```
LANC ASTER LAN CAS TERLANC AS TER LANCA
IWJN JEFLZ LJJ PMZ CARJTJZ ZU AXN GJZKH
```

You then decipher the message by sliding the top line over the bottom line of your tool until the current letter from the keyword (L) on the top line is aligned with the letter that you want to decode (I) on the bottom line. The deciphered letter is then the letter on the top line that is aligned with the letter A (D in this case) on the bottom line, as is shown below:

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

If you repeat this for every pair of letters (AW, NJ, CN etc…) then you will decipher the message:

DEEP ROOTS ARE NOT REACHED BY THE FROST

Enciphering text using this method is exactly the same, by sliding the top line over the bottom line of your tool until the current letter from the keyword (L) on the top line is aligned with the letter that you want to encode (D) on the bottom line. The deciphered letter is then the letter on the top line that is aligned with the letter A on the bottom line (I in this case)

```
LANC ASTER LAN CAS TERLANC AS TER LANCA
DEEP ROOTS ARE NOT REACHED BY THE FROST
```

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

The enciphered message is therefore (as we already know…):

IWJN JEFLZ LJJ PMZ CARJTJZ ZU AXN GJZKH

Breaking a Telegraph Cipher

Polyalphabetic ciphers such as these are unlikely to be broken by hand unless you have some information to start with (sometimes called a crib). However, unlike the Vigenère cipher, the Telegraph cipher is quite unusual, and I do not know of any tools specifically designed to attack it (though I would expect that a decent cryptanalyst would be able to break it using general-purpose cryptanalysis software).

Complete Columnar Transposition Cipher

Transposition ciphers are those that consist of jumbling up the existing letters, rather than substituting them. This is a simple but effective example.

The trick to a transposition cipher is that it should be sufficiently complicated that you can't simply re-shuffle the letters and work out what it says. The first step is to pick a keyword, let's use **FLIGHT** – this is a 6-letter keyword, meaning that we can encipher a message of up to $6 \times 6 = \mathbf{36}$ characters, and the number must divide equally into 6 (i.e., 6, 12, 18, 24, 30 or 36). Here is a message:

```
RENEWED SHALL BE BLADE THAT WAS BROKEN
```

This is 32 characters long (excluding spaces), so we need to add 4 other characters for padding to make it equal 36:

`RENEWEDSHALLBEBLADETHATWASBROKEN`**`KSYC`**

Now, we simply plot our into a table, starting a new row every 6 characters (the length of our keyword):

```
FLIGHT
RENEWE
DSHALL
BEBLAD
ETHATW
ASBROK
ENKSYC
```

Now, reorganise the columns of your table so that the letters of the keyword are in alphabetical order (so **FLIGHT** becomes **FGHILT**):

```
FGHILT
REWNEE
DALHSL
BLABED
EATHTW
AROBSK
ESYKNC
```

Now simply re-assemble the sentence by joining the rows back together, and you get the ciphertext:

`RENEWEDSHALLBEBLADETHATWASBROKENKSYC`

`REWNEEDALHSLBLABEDEATHTWAROBSKESYKNC`

To reverse the process, it is simply a matter of assembling the table with the keyword in the alphabetical order, and then re-ordering the columns so that the keyword is re-assembled again:

```
FLIGHT
RENEWE
DSHALL
BEBLAD
ETHATW
ASBROK
ENKSYC
```

It is then down to the analyst to locate the spaces and reject the padding (which will just be random text).

RENEWED SHALL BE BLADE THAT WAS BROKEN ~~KSYC~~

As you can see, the above cipher text still resembles the plain text a little, in part because the word **FLIGHT** doesn't need much shuffling to be put into alphabetical order. A word with a better mix of letters will result in a more complex transposition, but even a simple one like this is still very effective on longer texts.

Note that you can actually skip the keyword step, and just make a cipher out of the un-shuffled table, but that is much less secure (see below).

Breaking a Complete Columnar Transposition Cipher

Though transposition ciphers have a reputation for being tricky to decipher, the simplest versions (a complete columnar transposition as described above) can be broken relatively easily.

For example, consider our cipher above:

```
REWNEEDALHSLBLABEDEATHTWAROBSKESYKNC
```

This is 36 letters long. Because we are dealing with a **complete** columnar transposition (i.e., all of the slots in the table are full), then we know that the length of the keyword (and hence the width of the table table) must have been a number that divides evenly into 36: either 2, 3, 4, 6, 9, 12, or 18. This seems like a lot of options, but it has narrowed things down quite nicely for us.

Now we just need to test each one using the following method. It doesn't matter too much what order you test them in, but it is probably a little less likely that the table will be either very tall or very wide as a very short keyword would be insecure, whereas a very long one would be quite impractical. It will therefore most likely be closer to a square, so we will start with the numbers that are closest to a square and work up from there: let's test in the order: 6 (6x6=36), 4(4x9), 9(9x4), 3(3x12), 12(12x3), 2(2x18), 18(18x2).

To test a 6x6 table, let's lay our ciphertext into a table, starting a new row every 6 letters:

```
REWNEE
DALHSL
BLABED
EATHTW
AROBSK
ESYKNC
```

If the cipher does NOT use a keyword, then we should be able to find a word by searching the rows and columns for some recognisable text (none is visible in the example above). If we draw out the table into all of the forms listed above (6x6, 4x9 etc…), then one of the tables will reveal some plain text when reading either up or down the columns, or left or right on the rows.

If this is not the case (as in this example), then the cipher might be using a keyword. To solve this, we do the same process as above to lay out the cipher into the most likely table size. We then need to look at **vowel frequencies** – in a typical block of English, about 40% of the letters will be vowels and 60% consonants. We can use this information to look for any deviations in the rows or columns, which might give us a clue to the direction. For example, below I have identified a column that is all consonants – there is therefore almost no chance that the cipher should be read up or down, and so we know that the text must be read left or right.

REW**N**EE
DAL**H**SL
BLA**B**ED
EAT**H**TW
ARO**B**SK
ESY**K**NC

Because this is a transposition cipher (i.e., just shuffled letters) and we know that we need to read the rows (not the columns), we can simply search for anagrams in the rows. The trick here is that the letters in each row all need to be shuffled in the same way (e.g., if the first and last letter need to be swapped in one row, that must be the case in all of the others too, as we are swapping the columns in the tables).

Because we have used a poor keyword here (as its letters were already close to alphabetical order) – this is quite easy in the above example – I can see possible words in the first three lines: RENEW, SALAD and BLADE.

I can now take these and see if I can rearrange the columns to make those words (this can be made easier by writing the text onto a set of vertical strips of paper so that you can easily rearrange the columns).

When I try this, it turns out that I can't make all of these words appear (as we know, salad is wrong), but by rearranging the columns in different ways I can quickly work out that see that RENEW works, SALAD was wrong (it is SHALL), and BLADE is correct, but is across two lines:

RENEWE
D**SHALL**
BE**BLAD**
ETHATW
ASBROK
ENKSYC

In this form, I can see that there are lots of other words visible now – the cipher is broken!

RENEWEDSHALLBEBLADETHATWASBROKENKSYC

It is then a trivial matter to separate the text into individual words and remove the padding from the end.

RENEWED SHALL BE BLADE THAT WAS BROKEN ~~KSYC~~

As luck would have it, we were right first time with our guess of length 6, but if we couldn’t find any plaintext, then we would just move down the list and try a 4(4x9) table, and so on, until we (hopefully) find an answer.

If we don’t manage to find an answer after trying all of the possible sizes, then it might be a more complex transposition cipher (e.g., an incomplete transposition cipher), and we could try a computational attack, like the one available in the tool here: https://www.dcode.fr/columnar-transposition-cipher. Note that this is a brute force attack, and so the longer the keyword, the less likely it is to find the answer (in this case, it can only test up to a keyword length of 6 letters).

Detecting Ciphers

It is often possible to work out what type of cipher you are dealing with using a combination of **frequency analysis** (see next page) and calculating the **index of coincidence**.

Frequency Analysis is simply a matter of counting the number of times that each letter appears in a cipher text. You can calculate it manually or using online tools such as this one: https://www.dcode.fr/frequency-analysis .

In plain English, most common letters should be **approximately**:

E:12.7%
T: 9.1%
A: 8.2%
O 7.5%
etc…

It is not unusual for several letters (e.g., Q, Z or J) to be missing entirely from a text.

Index of Coincidence is simply the probability that if you selected two random characters from the cipher text they would be the same (e.g. both 'e'). You can calculate it using online tools such as this one: https://www.dcode.fr/index-coincidence.

The index of coincidence for plain English should be **approximately**:

6.8% (0.068)

The index of coincidence for random text, on the other hand, **approximately**:

3.8% (0.038)

These simple statistical metrics that help cryptanalysis to establish (amongst other things) what type of cipher might have been used to encrypt a given ciphertext. For example, as a rule of thumb:

Monoalphabetic Substitution Ciphers are simply swapping some letters for other letters, so you would expect:

- The frequency analysis to roughly match with plain English, but with different letters attached to each frequency (i.e., one letter will be something like 12-13%, but it will not be 'e').
- The index of coincidence to roughly match with plain English

Polyalphabetic Substitution Ciphers draw letters from multiple alphabets, and so generally you would expect:

- The frequency analysis to show less variation, with the most common letters being less frequent (more like ~7-8% rather than ~12-13%), and the least common letters being more frequent (with few, if any at 0).
- The index of coincidence will be closer to random text (~3.8%) than plain English (~6.8%)

Transposition Ciphers simply mix up the plain English letters, so you would expect:

- The frequency analysis to roughly match with plain English, both in terms of the frequencies and the letters themselves
- The index of coincidence to roughly match with plain English (~6.8%)

Appendix 1: Frequency Analysis Crib Sheet

Rules of Thumb:

The most common letters in the English language are (in order): **E T A**, followed by **O I N H S R**; the rarest are **Q, Z, X** and **J**.

The most common pair of letters is **TH**, other common pairs are **ER**, **ON** and **AN**.

The most common triplet is **THE**

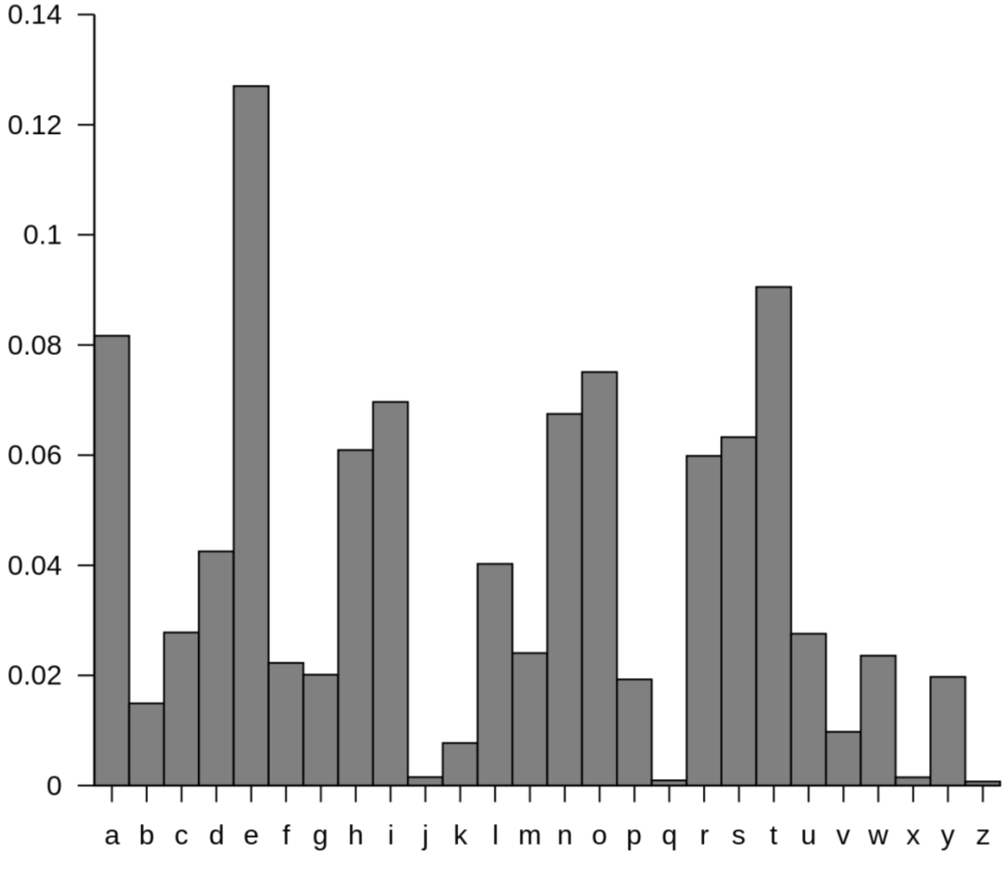
A pair of identical letters is likely to be one of: **EE FF SS TT,** others include: **LL MM OO, NN** and **PP**

A single letter is likely to be either **A** or **I**

A letter following an apostrophe (‘) is likely to be **S**.

The ratio of vowels to consonants is about 40% / 60%

Letter Frequencies in English


0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z