Shift (Caesar) Ciphers

If you have a message you want to transmit securely, you can encrypt it (translate it into a secret code). One of the simplest ways to do this is with a *shift cipher*. Famously, Julius Caesar used this type of cipher when sending messages to his military commanders.

A shift cipher involves replacing each letter in the message by a letter that is some fixed number of positions further along in the alphabet. We'll call this number the *encryption key*. It is just the length of the shift we are using. For example, upon encrypting the message "cookie" using a shift cipher with encryption key 3, we obtain the encoded message (or *ciphertext*): FRRNLH.

To make all of this more mathematical, consider the following conversion table for the English alphabet:

0	1	2	3	4	5	6	7	8	9	10	11	12
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*

13	14	15	16	17	18	19	20	21	22	23	24	25
N	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*

i. Using the table, we can represent the letters in our message "cookie" with their corresponding numbers: 2 14 14 10 8 4.
ii. Now add 3 (the encryption key) to each number to get: 5 17 17 13 11 7.
iii. Now use the table to replace these numbers with their corresponding letters: FRRNLH.

There is a small complication when we want to encrypt a message that contains a letter near the end of the alphabet. For example, if we consider the new message "pizza," then what letter should we use to replace the "z" when we encrypt?

After performing a shift cipher encryption with encryption key 3, the message "pizza" becomes SLCCD. The letter "z" was replaced with the letter "C," which we can view as being 3 places further along than "z" if, after we reach "z," we cycle the alphabet around to the beginning again.

In terms of the numerical representations of our letters, the encryption of the message "pizza" looks this way:

$$15\ 8\ 25\ 25\ 0 \quad \longrightarrow \quad 18\ 11\ 2\ 2\ 3.$$

What have we done mathematically? There is a handy mathematical concept that describes this very nicely. Define the following notation for integers a and b and integer $m > 1$:

$$a \equiv b \pmod{m} \quad \text{means} \quad m \text{ is a divisor of } a - b.$$

In our situation, we take the number m (the *modulus*), to be equal to the size of our character set, so $m = 26$. Now take each number x from the representation of the message and perform the following arithmetic: add 3 to x, and if the result is between 0 and 25, stop; otherwise, replace $x + 3$ with the integer y between 0 and 25 that satisfies $y \equiv x + 3 \pmod{26}$.

In summary, our encryption of the message "pizza" using a shift cipher with encryption key 3 looks like this:

$$\begin{array}{lllllll}
\text{p} & \longrightarrow & 15 & \longrightarrow & 15 + 3 \equiv 18 \pmod{26} & \longrightarrow & \text{S} \\
\text{i} & \longrightarrow & 8 & \longrightarrow & 8 + 3 \equiv 11 \pmod{26} & \longrightarrow & \text{L} \\
\text{z} & \longrightarrow & 25 & \longrightarrow & 25 + 3 \equiv 2 \pmod{26} & \longrightarrow & \text{C} \\
\text{z} & \longrightarrow & 25 & \longrightarrow & 25 + 3 \equiv 2 \pmod{26} & \longrightarrow & \text{C} \\
\text{a} & \longrightarrow & 0 & \longrightarrow & 0 + 3 \equiv 3 \pmod{26} & \longrightarrow & \text{D}
\end{array}$$

How is the original (*plaintext*) message recovered from the ciphertext if the encryption key is known? The following ciphertext was produced using a shift cipher with encryption key 9: LQXLXUJCN. To decrypt it (i.e., to recover the plaintext message), we need to add 17 (... or subtract 9 ... why is that the same?) to each of the numbers representing the ciphertext letters. Here 17 is the *decryption key* for the shift cipher with encryption key 9. Again, we must sometimes replace the result of this addition with the appropriate number between 0 and 25:

$$
\begin{array}{ccccccc}
\text{L} & \longrightarrow & 11 & \longrightarrow & 11+17 \equiv 2 \pmod{26} & \longrightarrow & \text{c} \\
\text{Q} & \longrightarrow & 16 & \longrightarrow & 16+17 \equiv 7 \pmod{26} & \longrightarrow & \text{h} \\
\text{X} & \longrightarrow & 23 & \longrightarrow & 23+17 \equiv 14 \pmod{26} & \longrightarrow & \text{o} \\
\text{L} & \longrightarrow & 11 & \longrightarrow & 11+17 \equiv 2 \pmod{26} & \longrightarrow & \text{c} \\
\text{X} & \longrightarrow & 23 & \longrightarrow & 23+17 \equiv 14 \pmod{26} & \longrightarrow & \text{o} \\
\text{U} & \longrightarrow & 20 & \longrightarrow & 20+17 \equiv 11 \pmod{26} & \longrightarrow & \text{l} \\
\text{J} & \longrightarrow & 9 & \longrightarrow & 9+17 \equiv 0 \pmod{26} & \longrightarrow & \text{a} \\
\text{C} & \longrightarrow & 2 & \longrightarrow & 2+17 \equiv 19 \pmod{26} & \longrightarrow & \text{t} \\
\text{N} & \longrightarrow & 13 & \longrightarrow & 13+17 \equiv 4 \pmod{26} & \longrightarrow & \text{e}
\end{array}
$$

Let's think about security now. Suppose you intercept a transmission of an encrypted message, and you know that the sender has used a shift cipher on the English alphabet, but you do not know the encryption key. How difficult would it be for you to break the code?

If we exclude the encryption key 0, there are only 25 distinct shifts that might have been used. It probably wouldn't take very long (especially with computer help) to test each of these shifts in turn (an *exhaustive search*). An incorrect shift length is likely to produce gibberish, while the correct shift length will produce a sensible message. Note however, that spaces between words, punctuation, etc., will not be included in the plaintext that is recovered. For example, the message "fish 'n' chips" would appear as "fishnchips" in its plaintext form.

Can you find the plaintext that produced the ciphertext below? A shift cipher with undisclosed encryption key was used.

UEUFXGZOTFUYQKQF

Probably you answered the above question by performing an exhaustive search. However, if the message had been longer, possibly you could have taken a shortcut. Suppose you have a somewhat lengthy passage of ciphertext that you want to decrypt, but you do not have the key. If you know that a shift cipher was used on a plaintext message written in standard English, then you can employ *frequency analysis*: determine which letter occurs most often in the ciphertext. It is very likely that this letter represents the plaintext letter "e." Assuming that it does, the decryption key can now be calculated. Then try decrypting the entire message using this probable key.

Chances are good that the above procedure will find the correct plaintext on the first try. If it doesn't, i.e., if the outcome is gibberish, then find the next most likely representative of the plaintext letter "e" and try again. Below is ciphertext produced by a shift cipher with undisclosed encryption key. Decrypt it using frequency analysis. (Note that the line breaks do not necessarily occur between words!)

WIVHLVETPRERCPJZJTRESVLJVUFEFKYVIKPGVJFWTZGYVIJKFFSLKZKZJEFKLJLR
CCPRJJZDGCVRJZKZJNZKYJYZWKTZGYVIJDREPFWKYVJVFKYVITZGYVIJNZCCRGG
VRIYVIVZEKYVEVOKWVNDFEKYJ