# Security Coursework

## Part A

To brute force the key the algorithm must generate all possible permutations of the key, and use the generated key to attempt to decrypt the ciphertext. As the plaintext is valid English the algorithm uses a regular expression to check if the decrypted plaintext only contains alphanumeric, punctuation, and space characters. If the regular expression matches we have found the correct key.

As the key only uses lower case letters (a – z), the key space can be calculated using:

Where *l* is the length of the key.

When I run the algorithm on my computer I get an average of 8250 attempts per second. Knowing this, we can calculate the time taken to attempt every key.

|  |  |  |
| --- | --- | --- |
| Key Length | Key Space | Time Taken |
| 4 |  | ~ 55 seconds |
| 5 |  | ~ 24 minutes |
| 6 |  | > 10 hours |

As the key length increases, the key space and time taken to attempt all possible keys grows exponentially.

## Part B

I expect this attack to work as we can retrieve the keystream which was used to encrypt the plaintext.

If we have as byte of the ciphertext, as byte of the plaintext, and as byte of the keystream generated by the key, , where are the start and end bytes of the ciphertext to replace and then:

Thus, if we know both the plaintext and ciphertext we can recover the keystream by:

We can then use to the replacement plaintext, , to generate the replacement ciphertext, :

We then replace the with .

Without the original plaintext, we wouldn’t be able to recover the keystream, and thus would be unable to attack in this way.

## Part C

Similarly to part B, we use the plaintext and ciphertext to recover the keystream used to encrypt the plaintext:

We are now able to XOR the recovered keystream with the second ciphertext , to recover the second plaintext :

However, we are unable to recover the key, , that was used to generate the keystream.

If is equal to the size in bytes of , and is equal to the size in bytes of , and the case were , we would only be able to recover the first bytes of .