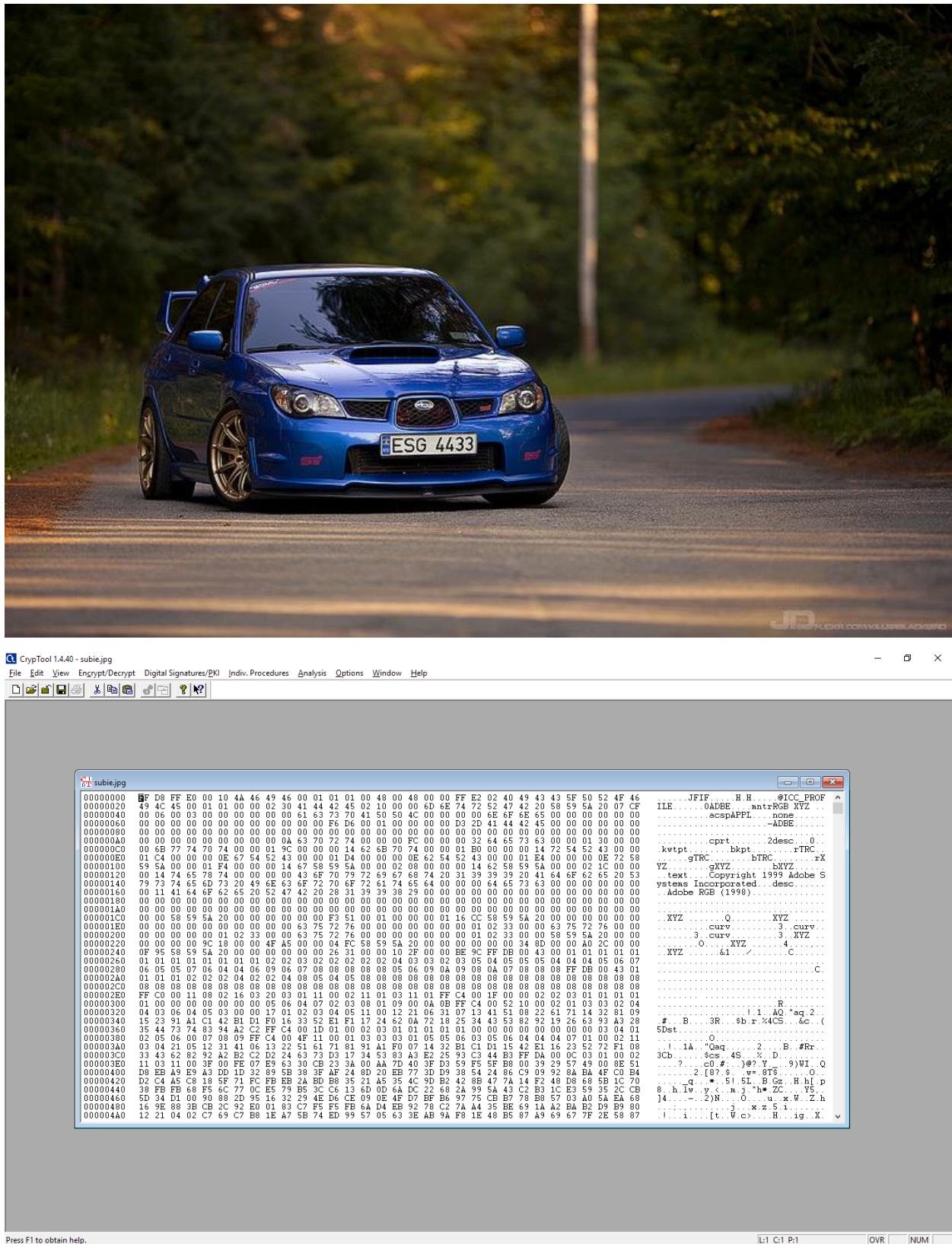
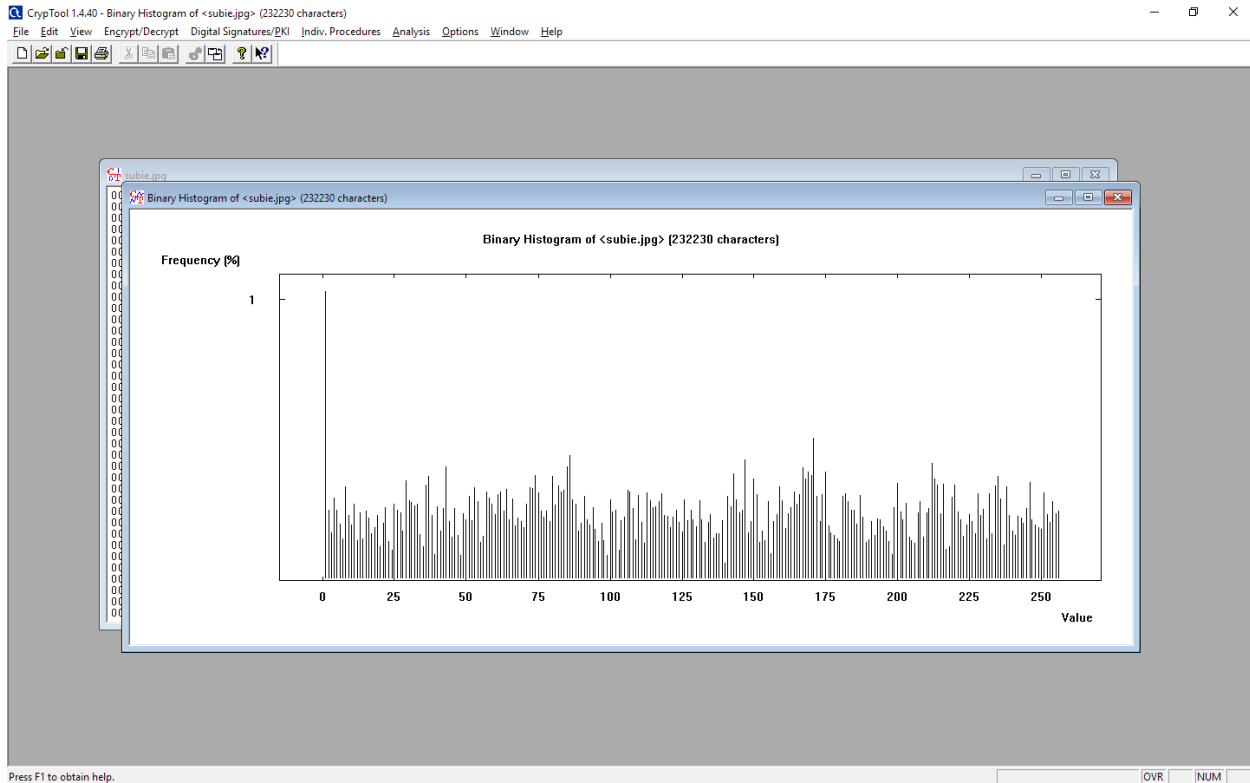


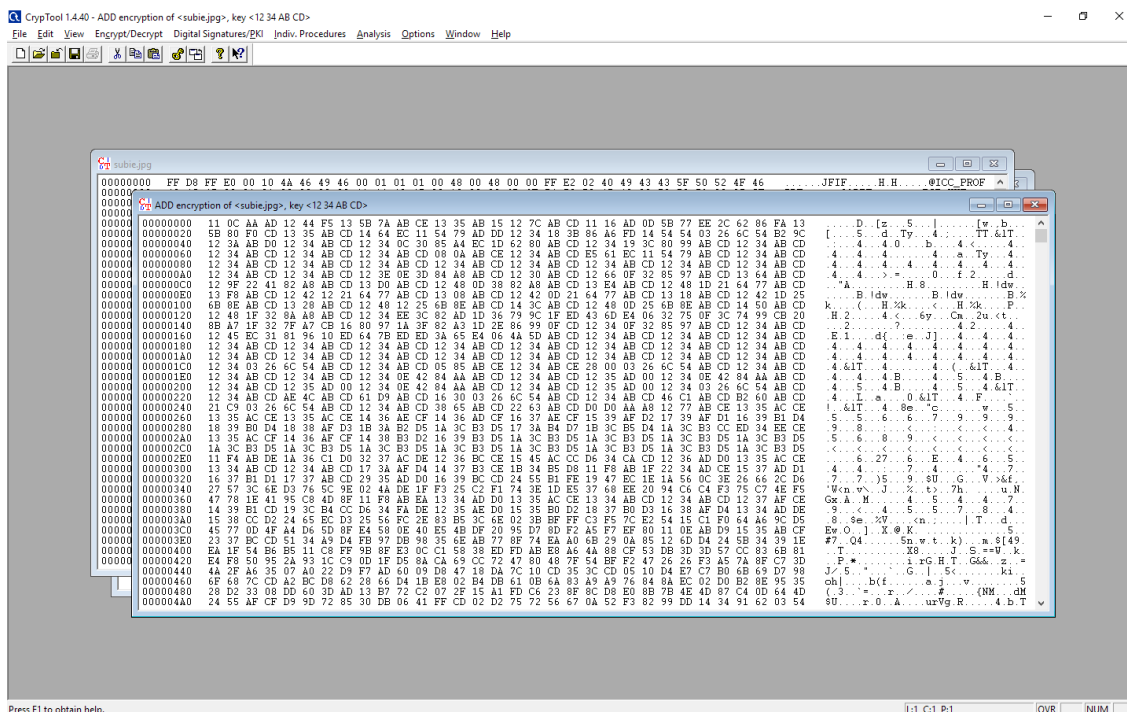
I opened the image below in CrypTool and was able to see the hexadecimal data of the file.



This is the histogram generated from analyzing the image. Most of the data is zero, the rest seems to be the percentage of each value for the colors of the image.



I used the key “1234ABCD” to encrypt the file and it produced the following data file. After saving the encrypted file, the computer was unable to open the image again.



Running the Byte-Addition analysis on the encrypted file with a length of 4 and the prefilled Expected common character of 00 derived the correct key.

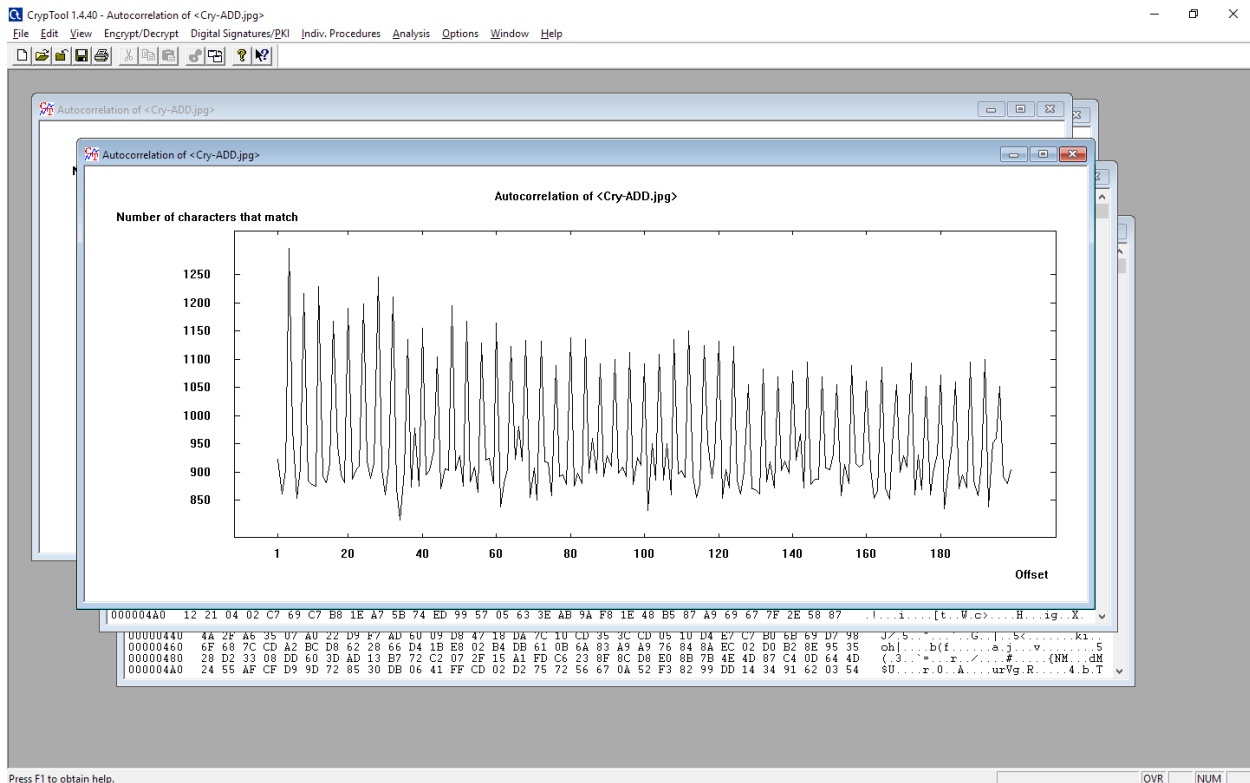
Byte-by-byte Addition Analysis [X]

Derived key length:

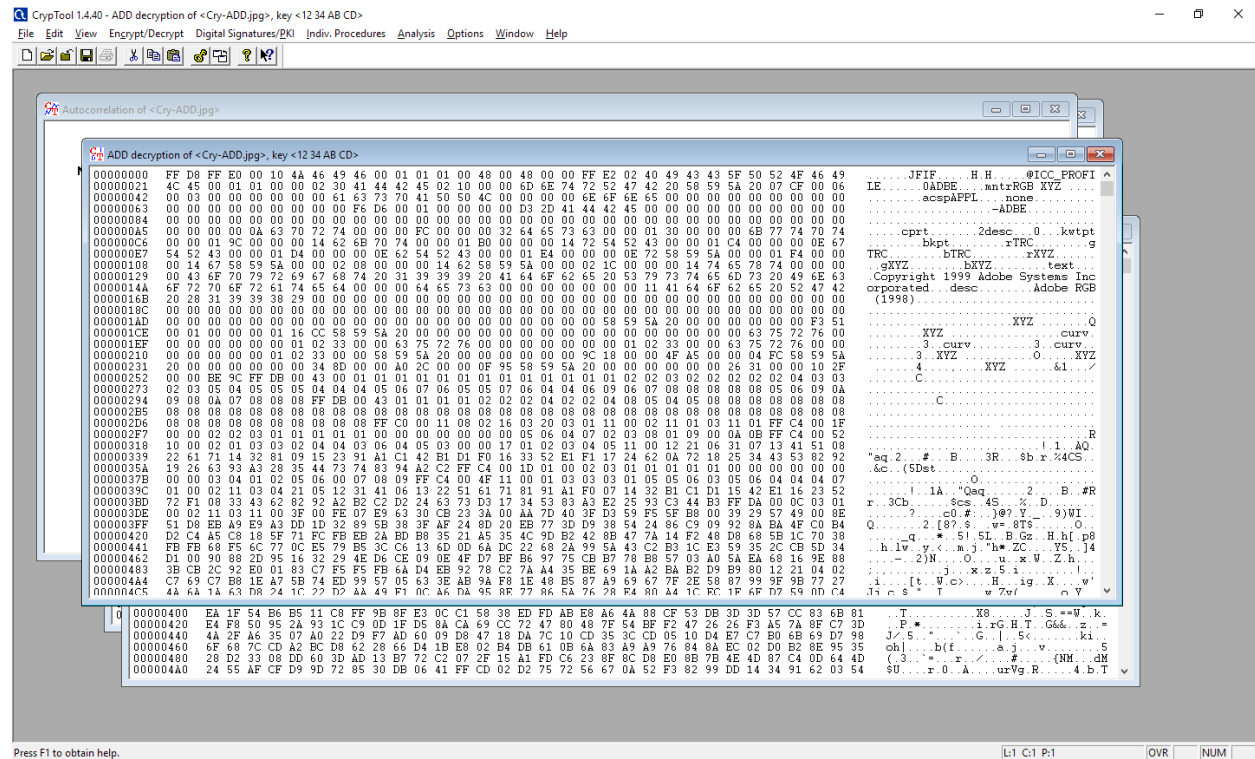
Expected most common character (hex):

Byte-by-byte Addition Analysis [X]

Derived key:



Decrypting the file produced the same file as the original. I was able to save the decrypted file and open it as the correct image.



File compression works by identifying patterns and abstracting them. It indexes the removed pattern and specifies where it can be found. This works well for text files, because text is repeated a lot and is easy to index. It is harder to compress images, because there is so much more unique detail. Image compression often results in loss of data.