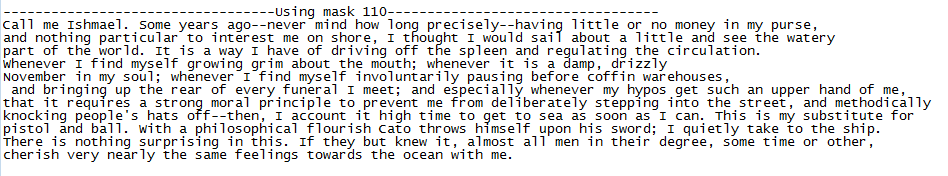
Timothy Indrieri

Brandon Artner

Edward Morkunas

Assignment 6

The given ciphertext was given as ASCII characters encoded as an eight-byte representation that had a “8 bit one-time pad key ” applied to them. This means that each 8-bit section was xored with the same eight bit key. Since the key is eight bits, there are 28 number of keys possible. Using brute force we went through every 256 combination with an eight bit representation and xored them to the encrypted text. Knowing the properties of xor we realized that one of the xored bits would give us the decrypted text. Since there was more than just the regular ascii characters such as a new line character being encoded as a bit we went through a masking of all the bits into ascii until we saw something that made sense and that happened to be the mask of 110 as shown below and is included in the raw\_plain.txt shown below. To further lower the amount of raw output from the brute-force algorithm we calculated the rolling average of the ascii character values. The algorithm stopped if this ever went outside the range of 32 to 126 (the alpha-numeric values in ascii). 

If the encoding was xored with different 8 bit keys instead of using the same key the encryption would be very difficult if not impossible to break. The methods that we used are extremely reliant on the fact that the masking is only 8-bits, if it was even 16-bits we would have had to combinations through 65536 different results which would have been unfeasible to do. Also the rolling averaging technique we used only works due to some important assumptions, namely that the plaintext starts with an alpha-numeric. However, this assumption turned out to be useful in reducing the amount of output and could have been removed if no valid output had been found.