CS641A: Bonus Assignment

Rijndael
Abhishek Jain, Arham Chopra, Ayush Tulsyan
13027, 14130, 14807167

1 Convert hex to base64

Converted the hex string to binary and then to base64 format.

2 Fixed XOR

Converted both hex string to binary did the xor and converted the resultant xor back to hex string

3 Single-byte XOR cipher

1. For each character possible among the 256 ASCII character set
   (a) Converted the character to 4 bit length binary and repeated sufficient no of times to be as long as the cipher.
   (b) Xored the new repeated key with the bitstring of the cipher.
   (c) Used a scoring mechanism to decide the score for each output text from the previous xor.
   (d) Select and return the character with the highest score as the acutal output string and the key.

4 Detect single-character XOR

Made use of the code in the Part 3 to do the same thing here.

5 Implement repeating-key XOR

Converted the key string to binary and repeated it to be atleast as long as the cipher bitstring. Then xored both of them together and converted the output to a hexstring.

6 Break repeating-key XOR

The key as well as key length is unknown. The challenge page on Cryptopals' website describes an elegant way of finding the key length. It makes use of Hamming distances and the fact that the hamming distance between small alphabets is small.

For all key lengths (from 2 to 40), We computed the average hamming distance between first five consecutive blocks, each having key-length characters. The lengths corresponding to smallest 5 average values were chosen. *These average values can be printed by setting the debug flag in script*

For all possible key lengths, the following procedure is followed. Arrange the input text in the form of a matrix. Each row having `key-length` characters. And when we concatenate the rows in their order, we get the original string. Now take the transpose of this matrix. Now each row has been XORed with a single character. This single character XOR problem has been solved in previous challenges.

Collecting the character key from each row of the transposed matrix gives us the required key. This key in now XORed iteratively with the cipher to get the message.

The key-length obtained was 29 and the key is: `Terminator X: Bring the noise`

7 AES in ECB mode

Made use of PyCrypto module in python to decrypt the AES Cipher in ECB Mode.

8 Detect AES in ECB mode

1. For each cipher(converted to bitstring) in the file
   (a) Broke into chunks of size 128 bits each.
   (b) Checked if there exist some blocks occur more than once in a cipher.
   (c) If such multiple occurences of a block are found then this can be valid cipher encrypted in ECB mode.

Only one string was found with such duplicate blocks so that must be the one encrypted in ECB mode.