Assignment 1a

Explanation of encryption algorithm:

- The program logic flow of the encryption script in Encrypt.py is based on differential xoring of bit blocks.
- Here, initially, plaintext file is scanned into several blocks of 64 bits and the encrypted output produced for each block is a function of the corresponding encryption of the previous blocks.
- The encryption script requires a key and a passphrase to generate the ciphertexts.
- The plaintext file is scanned into several blocks, with each block being of size BLOCKSIZE.
- BLOCKSIZE is defined as 64 bits.
- Since the size of plaintext in bits may not be an integral multiple of BLOCKSIZE, an appropriate number of zero bytes are appended to the last block to make its size same as BLOCKSIZE. Hence it guarantees same length of plaintext and ciphertext.
- Passphrase is used as initialization vector (IV) and is used in order to encrypt the 1st block and begin the encryption process of entire plaintext.
- For the initial block(first 8 characters of the message summing up to 64 bits) the encrypted message is obtained by XORing the plaintext with Passphrase and the key.
- For the remaining blocks, the encrypted message is obtained by XORing the plaintext with previous output [encrypted block] and the key.

Let ‘pi’ represents the blocks of plaintext [Plaintext=p1p2p3p4….] and Ci represents corresponding blocks of ciphertext. Key is requested from the user.

Here xor is represented by ‘$\oplus$’.
So the ciphertext blocks will be obtained from the following operations-

$C1 = p1 \oplus key \oplus IV$
$C2 = p2 \oplus key \oplus C1 = p2 \oplus p1 \oplus key \oplus IV$
$C3 = p3 \oplus key \oplus C2 = p3 \oplus p2 \oplus p1 \oplus key \oplus IV$
$C4 = p4 \oplus key \oplus C3 = p4 \oplus p3 \oplus p2 \oplus p1 \oplus key \oplus IV$
$C5 = p5 \oplus key \oplus C4 = p5 \oplus p4 \oplus p3 \oplus p2 \oplus p1 \oplus key \oplus IV$
...