ENCRYPTION ALGORITHM:

The given Encrypt.py is a Block cipher which has the following Components:

(1) Plaintext (to be encrypted)
(2) Key
(3) Passphrase

The algorithm Begins by encrypting passphrase into a 64-Bit vector. This is achieved using python library Bit Vector which first converts the passphrase into bits and if length is less than BLOCKSIZE , appends zeros at the End.

The same process is followed for key i.e. first conversion to Bit Vector and then appending zeros if required.

Now the plaintext is also converted into bit Vector and is divided into blocks of size BLOCKSIZE. Every such block is differentially XORed with the key block and previous encrypted block. If in the last iteration of Algorithm, plaintext block has less than BLOCKSIZE bits , zeros are appended to complete the block.

This previous block is initially initialized to the passphrase block for the first iteration.

After encryption the data is stored in the hexadecimal form.

The basic encryption step can be summarized as:

C[n] = C[n-1] ^ key ^ P[n] where

C[n] → Ciphertext (Bit Vector) for nth Block

P[n] → Plaintext (Bit Vector) for nth Block

Key → key Bit Vector

^ → Bitwise XOR operator

Example:

	Iteration 1	Iteration 2	Iteration 3
Plaintext	10101010	11110000	10101000
Key	00011100	00011100	00011100
Previous Block	00001111	**10111001**	01010101
First XOR	10110111	11101100	………………
Second XOR/ Ciphertext	**10111001**	01010101	………………..

Note: (1) Size of every iteration cell is equal to BLOCKSIZE

(2) Previous Block of Iteration 1 is generated from Passphrase
(3) Ciphertext of previous block(C[n-1]) is essentially the previous block for next iteration

Submitted By

Sanjeev Singla (2017A7PS0152P)