Explanation of the encryption algorithm

The algorithm first reduces the passphrase into a BitVector which is a memory efficient way to store bit vectors. It takes the passphrase in blocks of size 64. For the first block, it performs XOR with a BitVector containing all 0s but for subsequent blocks, an XOR is taken with the output obtained from the previous block.

The key is obtained from the user and the key is converted into a BitVector using a similar method as was done for the passphrase.

After this, we create a BitVector of size 0, to store the encrypted message.

Then we read from the plaintext in blocks of size 64. We XOR the first block with the BitVectors of the key and the passphrase. All subsequent blocks are XORed with the output of the previous block and the BitVector of key. At the end of each iteration of the loop, we add the encrypted block to the BitVector of size that we had originally created.

We are essentially using XORing that relies not only on the key but also on the previous block. This method essentially removes repetitive patterns in the plaintext and makes the encryption secure from analytical attacks.

Once the loop finishes, we convert the obtained BitVector into Hexadecimal representation and write it to the output file.