COL759 Assignment 2
Ashwini Kumar
2018MT60778

In this assignment we have implemented RSA Cryptosystem.
INPUT: Plaintext, a prime number 'r' and a number 'g' belonging to Zp* where p=2*r+1
OUTPUT:
User1(Server): Output for user 1 is the cipher text generated from the plaintext taken as input.
User2(Client): Output is the decrypted text received from user 1.

We find common session key(e) as following:

(1) User 1 chooses a random key k1 in Zp* and computes x1 = (g**k1) (mod p)
(2) User 2 chooses a random key k2 in Z p * and computes x2 = (g**k2) (mod p)
(3) User 1 and User 2 exchange x1 and x2 .
(4) User 1 computes x1,2 = (x2**k1) (mod p)
(5) User 2 computes x2,1 = (x1**k2) (mod p)
(6) e=x1,2 if x1,2 is odd and e=x1,2-1 if x1,2 is even

We find e=x1,2 such that it has a modular inverse w.r.t mod (p-1).

Then we calculate the private key as d= (e**-1) mod (p-1).

User 1(Server):
Optimized Block size(b) is found by finding a value of b such that (29**b)<p<(29**b+1). This block size is used to break the plaintext into blocks each of size b. Then we calculate the value of M as:
Let b=2, first block is IN then M= 13+ 8*29
Then to find cipher text we calculate C=(M**e) mod p = 26 and then we write
C=26+0*29 and since C<29**b ,this results in C=.AA

User 2(Client):
User 2 receives the cipher text from User 1 and calculates value of C as done previously for calculating M. Then we calculate M= (C**d) mod p. Then we find plaintext from M as we did previously for C.

This way plaintext is divided into blocks by user 1 and then it sends the cipher text to user 2. Further user 2 receives the cipher text and then decrypts it to find the plaintext

HOW TO RUN:
Type
python assignment2.py
It will ask for taking r, g and plaintext as input so input
r
g
Plaintext
Output:
Two files User1.txt and User2.txt depicting some information on the values calculated