ASSIGNMENT -2

HYBRID_RSA

SUBMITTED BY:-

Subrat kumar,2021QIZ8247

Kashish jain, 2021JCS2240

SUBMITTED TO:-

Professor A.K Bhateja

Hybrid_block_rsa.py

We have implemented Vignere cipher and RSA algorithm in order to built Hybrid_RSA .

Keys:-

Vignere key will be given by the user, in a file: ./keys/vkey.txt.

And we have used strong primes for generating public and private keys for sender and receiver.

Sender private key: SKA, Sender public key: PKA

Receiver private key: SKB, Receiver public key: PKB

Working:-

Initially user's key is crafted according to the length of user's plain text as per the algorithm and then user's plain text is given to Vignere encryption function along with that crafted key, which produces encrypted text.

After this, Vignere encrypted text and key are divided into the blocks of 5 characters after which each block is converted into a integer using make_block function and then both given to RSA encrypt function, which encrypts them using sender's private key : SKA.

Encryption function:(message)^e mod n .

Then produced encrypted text and key are given as input again to RSA encrypt function which then produces another encrypted text and key, using public key of receiver, PKB.

Then this encrypted text and key are given as input to RSA decrypt function , with key as Private key of receiver SKB.

Decryption function:(encrypted_text)^d mod n, where d is calculated using multiplicative modular inverse of e:e^-1 mod phi(n), where phi(n) :(p-1)*(q-1) where p and q are strong primes, and e is coprime to phi(n).

Then the produced decrypted text and key are given again to RSA decrypt function with key as public key of sender, PKA.

Finally we get our encrypted text back which was given by Vignere encryption function We then give this text to Vignere decryption function which decrypts the text, and thus produces the original plaintext, given by user.

LIBRARY USED: We have used python's GMP library: gmpy2 library, for working with long integers in our code.(installed using !pip install gmpy2).

FILES TO BE EDIT:

We have provided a zip folder which is having following files in it:

1.Keys folder

2.texts folder

3.hybrid_block_rsa.py(main file)

4.hybrid_rsa.py

User can give plain text in: ./texts/message.txt

User can give key in: ./keys/vkey.txt

Final decrypted text at receiver side will be generated in: ./texts/plain_text.txt