COL759
Assignment – 2

FILES :-

1. ***ass2.py***

Functions in it :-

1. **getBlockSize :-** This function calculates the block size to be used in the RSA.
2. **getChunks :-** This function calculates and stores the chunks of the block size in a list and returns it.
3. **crt :-** This
4. **rsaencryptblock :-** It returns the cipher text of a single block which we get after encrypting the block by scaling the charaters appropriately by charSize according to their position.
5. **rsadecryptlock :-** It returns the plain text of a single block which we get after decrypting the block by de-scaling the charaters appropriately by charSize according to their position.
6. **encryptRSA :-** It concatenates all the blocks which are encrypted using rsaencryptblock and then returns the complete cipher text.
7. **decryptRSA :-** It concatenates all the blocks which are decrypted using rsadecryptblock and then returns the complete plain text.
8. **encryptVig :-** This function encrypts the given text using the Vigenere encryption technique, i.e.,

$$c_i = p_i + k_i \pmod{charSize}$$

9. **decryptVig :-** This function decrypts the given text using the Vigenere decryption technique, i.e.,

c_i = p_i - k_i (mod) charSize

10. run :- First this function calls the genKeys function to generate keys. Then it gets the block size by the function getBlockSize. Then it uses the encryption and decryption techniques for both the users as given in the assignment pdf.

The technique used in the final run is :-

Encryption by user A, works as:

$c_s \leftarrow \mathrm{E}(m, k),\ (c, k') \leftarrow \mathrm{E}(\mathrm{D}(c_s, k, sk_A), pk_B)$

Decryption by user B, works as:

$(c_s, k) \leftarrow \mathrm{E}(\mathrm{D}(c, k', sk_B), pk_A)$, if $\mathrm{D}(c_s, k) = m$ then output (m), otherwise reject

2. ***genKeys.py***

Functions in it :-

1. genST :- This function is used to generate random primes.

2. genStrongP :- This function is used to generate a strong prime.

3. genKeys :- This function generates the key for the RSA algorithm.

To generate strong primes to use we used the "GORDON METHOD" for generating it.

The method is as follows :-

Theory

In RSA, we create a modulus of $n = pq$, and where p and q are random prime numbers. A strong prime number (p) is when r, s and t satisfy the following:

- $(p - 1)$ has a large prime factor of r.
- $(p + 1)$ has a large prime factor of s.
- $(r - 1)$ has a large prime factor of t.

The following defines the Gordon method of generating strong primes. First we generate two random prime numbers (s and t). Next we select an integer (i_0) and find the first prime number for:

$r = 2it + 1$

and where $i = i_0, i_0 + 1$, etc, until we find a prime number for r. Next we calculate:

$p_0 = 2(s^{r-2} \pmod{r})s - 1$

We then select j_0 and find the prime:

$p = p_0 + 2jrs$

and where $j = j_0, j_0 + 1$, etc, until we find a prime number for p. The results in a strong prime (p).

Submitted By :-

Shruti Kumari , 2018CS50420

Nikita Bhamu , 2018CS50413