**NAME :** Mansi Dwivedi

**BRANCH :** IT

**UID :** 2019140016

**BATCH :** A

**COURSE :** CSS LAB

**EXPERIMENT :** 1 and 2

**AIM :** To execute and understand the working of the various traditional crypto methods for experiment 1 and implementing the Diffie Hellman Key Exchange Algorithm for the experiment 2.

**PROBLEM STATEMENT :**

I) Write a single program which fits all algorithms. YOU should generate output in the following manner:

1. Select the Cryptography Method Provide Choice 1…5 for subject crypto methods

a. Substitution

i. Your choice

ii. Enter Plain text to be encrypted

iii. Enter the no. of Position shift

iv. Encrypted Message

v. Decrypted Message

b. ROT 13

i. Your choice

ii. Enter Plain text to be encrypted

iii. Encrypted Message

iv. Decrypted Message

c. Transpose

i. Your choice

ii. Enter Plain text to be encrypted

iii. Encrypted Message

iv. Decrypted Message

d. Double Transposition

i. Your choice

ii. Enter Plain text to be encrypted

iii. Encrypted Message

iv. Decrypted Message

e. Vernam Cipher

i. Your choice

ii. Enter Plain text to be encrypted

iii. Input Key

iv. Encrypted Message

v. Decrypted Message

II) Write a program which executes the Diffie-Hellman algorithm. YOU should generate output in the following manner:

a. Diffie Hellman

i. Enter the Prime Number

ii. Enter second Prime Number

iii. Enter the Secret x:

iv. Enter the Secret y

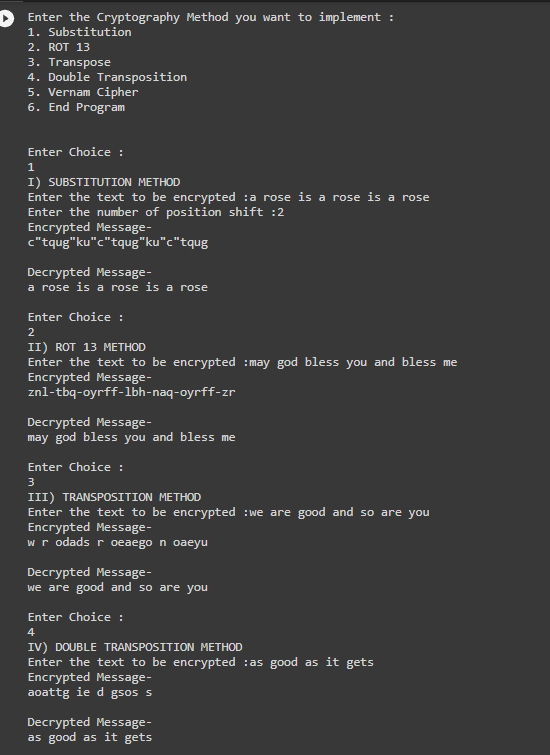
v. K1:

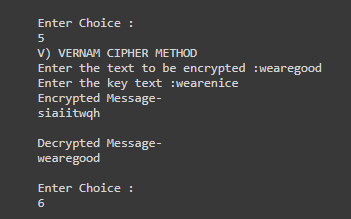
vi. K2:

**CODE :** <https://github.com/mansidw/CSS_Lab>

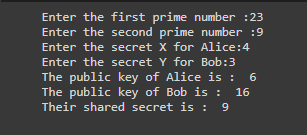
**OUTPUT :**

1. **Experiment 1**

****

****

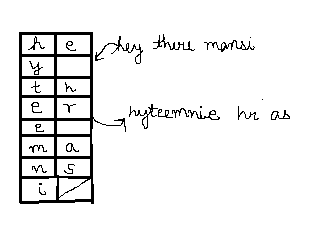
1. **Experiment 2**

****

**EXPLANATION :**

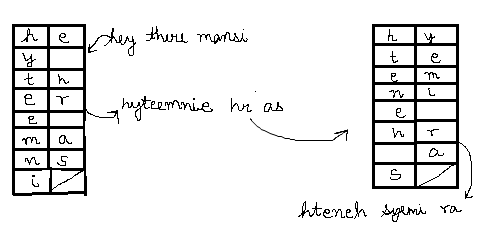
1. In the first case of Substitution Cipher I have used the ascii value method and substituted the characters by shifting them by place provided by the user and similarly decrypted the message similarly by left shifting the character’s position by the same value as inputted by the user.
2. In the second case of ROT13 I have replaced the particular character with the one having an ascii value after adding 13 to the previous one and similarly for decrypting moving left by 13 places.
3. In the third case of Transposition Cipher I have used a matrix of 2\*length(text)/2 to store the plain text and then move column wise while encrypting the text, similarly for decryption move in multiples of length(text)/2 for the next character.

e.g.)



1. In the fourth case of double transposition cipher I have used the same initial method as above then transposed the text again after step 1 similarly performed 2 steps backwards to decrypt the text.

e.g.)



1. In the fifth case of Vernam Cipher I have used the ascii values of both the text and key and added them together to obtain the encrypted text and similarly subtracted the decrypted text’s ascii value from the key’s and thus obtained the plain text.

**CONCLUSION :**

After learning about these encryption techniques and also implementing them I was able to conclude that :

* In the case of substitution cipher the key space is only 26 for which the attacker might take only a short amount of time to break or to try all the possible combinations of the shift variable. Hence it is not at all a reliable method/algorithm to transfer and protect data.
* The case of ROT13 is also similar to the above substitution cipher just that the shift variable takes the value of 13 and letters of the alphabet are offset 13 places. Hence this type of algorithm though a standard one yet is not secure at all.
* The transposition here implemented by me is the columnar one where the message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order. Both the length of the rows and the permutation of the columns are usually defined by a keyword variable. The columnar transposition technique of encryption is easy to understand and implement but still complex to break by brutal force attack or cryptanalysis. The hacker or the intermediate person cannot break this code unless otherwise he knows the method. Hence this algorithm is definitely better than the above two but also has some disadvantages that it is prone to slowness of encryption and also sensitive to lossy data or increased characters in the plain text.
* Similar to the above columnar transposition the double transposition has also been implemented just that the same steps have been repeated twice for more security and hence protection. This algorithm is way better to break than the above ones.
* Lastly, in the Vernam Cipher method since its key-exchange procedure uses real random number generation and secure key distribution, the Vernam Cipher with a one-time pad is said to be an unbreakable symmetric encryption technique. Each plaintext character from a message is mixed with one character from a key stream, according to the Vernam Cipher. If we utilise a completely random key stream, we'll get a truly random ciphertext that has no relation to the plaintext. Hence it proves to be a good encryption technique.
* Diffie Hellman is a key exchange method that allows 2 parties to jointly establish a mutual secret key over a public channel without it being transmitted over the Internet, hence it is a fundamental part of securely exchanging data online. I found that as long as it is implemented alongside an appropriate authentication method and the numbers have been selected properly, it is not considered vulnerable to attack. Hence if the value of ‘the first prime or p’ is taken to be a safe prime and the value of ‘the second prime or g’ is taken to be a primitive root mod p then the chances of the leaking of message is close to impossible hence the most security. Hence I infer that while it is really tough for someone snooping the network to decrypt the data and get the keys, it is still possible if the numbers generated are not entirely random.