Name:- Piyush Chaudhari

Roll no:- 5007

Batch:- MBA Tech (Data Science)

Q1

Ans Diffie-Hellman Key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers ~~retat~~ raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted.

ex:- Credit Card transaction email

Q2

Ans $n = 17$

$a = 5$

Private Key Alice = 4

Private Key Bob = 6

Public Key Alice = $5^4 \bmod 17$

$= 13$

Public Key of Bob = $5^6 \bmod 17$

$= 2$

secret Key of Alice = $2^4 \bmod 17$

$= 16$

secret Key of Bob = $13^6 \bmod 17$

$= 16$

The value of common secret Key 16

Q3

Ans Encryption

$E_i = (P_i + K_i) \bmod 26$

Decryption

$D_i = (E_i - K_i + 26) \bmod 26$

Q4

Ans
```
x = lambda a,b : a*b
print (x(5,6))
```
output : 30

Q5

Ans To implement Diffie-Hellman, two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive numbers p & q.
such that p is prime & q is generator of p. The q is no that, when raised to positive whole no. power less than p, never produces same result for any two such whole no. The p may be large but the value of q is small

Alice	Bob
Public Keys = P, G	Public Keys = P, G
Private Key = a	Private Key = b
Key generated = $X = G^a \bmod P$	Key generated = $Y = G^b \bmod P$

Exchange of the Keys

Keys recieved = Y	Keys recieved = X
Secret Key = $Y^a \bmod P$	Secret Key = $X^b \bmod P$

Q6

Ans Vignere cipher is method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of original text is done using Vignere square or Vignere table

A table consists of alphabets written out 26 times in different rows. At different points in the encryption process, the cipher uses a different alphabets from one of the rows

ex :-

Plain text :- G E E K S F O R G E E K S

Keyword :- A Y U S H A Y U S H A Y U S

Cipher text :- G C Y C Z F M L Y L E I M

Encryption :- $E_i = (P_i + K_i) \bmod 26$

Decryption :- $D_i = (E_i - K_i + 26) \bmod 26$

Q7

string = "GEEKSFORGEEKS"
Key word = "Sharan"

```
def generateKey(string, Key):
    Key = list(Key)
    if len(string) == len(Key):
        return(Key)
    else:
        for i in range(len(string) - len(Key)):
            Key.append(Key[i % len(Key)])
        return("".join(Key))
def encrypt_cipherText(string, Key):
    cipth cipher_text = []
    for i in range(len(string)):
        x = ((ord(string[i]) + ord(Key[i])) %26) + ord('A')
        cipher_text.append(chr(x))
    return("".join(cipher_text))
Key = generateKey(string, Keyword)
print("Original Message", string)
print("Keyword", Keyword)
cipher_text = encrypt_cipherText(string, Key)
print("Ciphertext:", cipher_text)
```

output
original : GEEKSFORGEEKS
Keyword : SHARAN
Ciphertext: YLEBSSGYGVEXK