1900319030
M. Vasavi

CACD - LAB - 1.

Pre - lab.

1) Define Cryptography & write any two applications of Cryptography.

Ans Cryptography is method of transfering data securly via Codes. So that the only decided person can receive the data securly. where the data will encrypted with a secret key stored to decided person. So that they can de-crypt the data with the stored-key.

Application of Crypto-graphy

1. In transactions.
2. In storages
3. Authentication systems
4. E-signatures, sign, Certificates, Cash.

(2) Types of Cryptographic - algo's

Ans
1. Public key Crypto-graphy.
2. Secreck key Crypto graphy.
3. DES - Data encryption standered
4. Symmetric key, Asymmetric key.

(3) Cryptography behind Block-chain, E-mail.

Ans In Blockchain the data is stored in Blocks. where each block has the info abt the last block hash & Preset block had details like string data like transaction amount wallet address etc. to it is a encrypted block chain. If we change any data of block. it causes error. It will check of data over decrypted data.

Symmetric system are often found in e-mail encryption - which has (DES) (AES) for encrypting mail data. using the same algo on receivers end we can decrypt the data in e-mail.

4. what is need of encryption?

Ans * Encryption helps in transfering the data securly without any breach.

* It's keeps ISP from accessing your data.
* with this we have privacy.
* Encryption prevent from hacking.

5) Symmetric & Assymmetric Cryptography.

Symmetric:-

The encryption & decryption uses same key. it also called as secret key encryption. which uses DES

* It is simple & fast.
* The two parties exchange the key in secure way.

Drawback:-

The only drawback is if key is leaked. Then there is a risk of data breach.

assymetric key:-

It is public key cryptography. It works in terms of symmetric key. It requires 2 keys. one for encrypt & one for decrypt. The public key is used for encrypting private key.

Drawback:- * low encryption speed.
* key management crucial.

It uses RSA.

In lab

① Caesar Cipher

Pycode:-

```
def encrypt(string, shift):
    cipher = ''
    for char in string:
        if char == ' ':
            cipher = cipher + char
        elif char.isupper():
            cipher = cipher + chr((ord(char) + shift - 65) % 26 + 65)
        else:
            cipher = cipher + chr((ord(char) + shift - 97) % 26 + 97)
```

Getar Cipher

```
text = input("Enter string: ")
s = int(input("enter shift number: "))
print("original string: ", text)
print("after encryption: ", encrypt(text, s))
```

Output

enter string: vaavn

enter string shift number : 4

original: Vaavn

encrypt : Zeeyr.

② Vigenere Cipher

Code

```
def generate key (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 encryption (string, key):
    encrypt_text = []
    for i in range(len(string)):
        x = (ord(string [i]) + ord (key [i])) % 26
        x += ord('A')
        encrypt_text.append (chr(x))
    return ("".join (encrypt_text))

def decryption (encrypt_text, key):
    orig_text = []
    for i in range (len(encrypt_text)):
        x = (ord(encrypt_text[i]) - ord (key [i]) + 26) % 26
        x += ord('A')
        orig_text.append (chr(x))
    return ("".join (orig_text))

if __name__ == "__main__":
    string = input("enter msg:")
    keyword = input("enter key:")
    key = generatekey (string, keyword)
    encrypt_txt = encryption (string, key)
    print ("encrypt text:", encrypt_txt)
    print ("decrypt text:", decrypt_txt)
```

output

```
Enter msg: VARUN
Enter key: VENOM
encrypt msg: QEEIZ
decrypt msg: VARUN
```

Postlab

① Pseudo code for encryption & decryption of Caesar, Vigenere Cipher

Vigenere :-

```
def encrypt(plain text : str, key : str) -> str:
    cipher text = ""
    m = len(key)
    for i, letter in enumerate(plaintext.lower()):
    cipher text += chr((char_2_num(letter) + char_2_num
                        (key[i % m])) % 26 + ord('a'))
    return cipher text.upper()

def char_2_num(character : str) -> int:
    return ord(character.lower()) - ord('a')
```

decryption:

```
    m = length of key
for index, character in ciphertext:
    Plaintext[i] <-- (character - key[index % m]) % 26
return Plain text
```

Caesar:-

```
def caesar_encrypt():
    word = input("enter plain text:")
    c=""
    for i in word:
        if (i == ' '):
            c+=' '
        else:
            c+ = (chr(ord(i)+3))
        return c

def caesar_decrypt():
    word = input("enter the ciple text:")
    c=""
    for i in word:
        if (i == ' '):
            c+=' '
        else:
            c+ = ((chr(ord(i)-3))
        return c

plain = hello
cipher = caesar_encrypt(plain)
decipher = caesar_decrypt(cipher)
```