ASHI SAINI
1121025

BCA-6-A
Information Security & Cyberlaws

MCQ

1.) a.) Symmatric key encryption with reciever public key

2.) c.) Spyware

3.) c.) An authentication of an electronic record

4.) d.) None

5.) a.) Only on alphanumeric

6.) c.) All

7.) a.) Hash Values

8.) d.) option a and c are right

9.) b.) to make even number of letters

10.) c.) Possibility of replacement

5) Caeser Cipher

pt = plaintext ⇒ "Attack from North"

code ⇒ Encryption ⇒.

```
def encrypt (pt, key):
    res = ""
    for i in range (len (pt)):
        ch = pt[i]
        if (ch.isupper ()):
            res = res + chr ((ord (ch) + 3 - 65) % 26 + 65)
        else:
            res = res + chr ((ord(ch) + 3 - 97) % 26 + 97)
    return res

pt = "Attack from North"
key = 3

print ("plain text:", pt)
print ("cipher text", encrypt (pt, key))
```

Decryption ⇒.

ct = "HIIM"

Cletter = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

Sletter = "abcdefghijklmnopqrstuvwxyz"

~~if (ch.isupper()):~~

```
for i in range (len (pt(ct))):
    ch = ct[i]
    if (ch.isupper ()):
```

```
        for k in range (len(cletter)):
            resl = ""
            for s in ct:
                if s in cletter:
                    n = cletter.find(s)
                    n = n-k
# for uppercase       if n<0:
decryption                  n = n+ len(cletter)
                    resl = resl + cletter[n]
                else:
                    resl = resl + s

    else:
        for k in range (len(sletter)):
            resl = ""
            for s in ct:
                if s in sletter:
                    n = sletter.find(s)
# for lowercase       n = n-k
decryption          if n<0:
                        n = n+ len(sletter)
                    resl = resl + sletter[n]
                else:
                    resl = resl + s

    print("Decrypted message is:" resl)
```

4.) OTP of length 6 suppose:

```
import math , random

def generateOTP( ):
    x = "0123456789"
    OTP = " "
    for i in range (6):
        OTP = OTP + x[ math.floor(random.random()
                                  * 10 ]

    return OTP

if __name__ == "__main__":
    print ("OTP is :", generateOTP())
```

3) Vigenere Cipher

pt = "CRYPTOGRAPHY"
key = "MONARCHY"
Code ⇒

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

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

```
if __name__ == "__main__":

    string = "CRYPTOGRAPHY"
    key1 = "MONARCHY"
    key = generatekey ( string , key1)

    etext = encryption (string , key)

    print ("Encrypted text:", etext )
    print ("Decrypted text:", decryption( etext, key))
```

Thank you...