SUB: Information Security

AY 2023-24 (Semester-V)

Experiments No: 3

Name: Divyesh Khunt **Sapid:60009210116** **Batch:D12**

Aim: Design and implement Encryption and Decryption Algorithm for Caesar Cipher, Shift Cipher. Also Perform Brute Force Attack on Ciphers.

Theory:

1. **Caesar Cipher.**

The Caesar cipher is the simplest and oldest method of cryptography. The Caesar cipher method is based on a mono-alphabetic cipher and is also called a shift cipher or additive cipher. Julius Caesar used the shift cipher (additive cipher) technique to communicate with his officers. For this reason, the shift cipher technique is called the Caesar cipher. The Caesar cipher is a kind of replacement (substitution) cipher, where all letters of plain text is replaced by another letter. Caesar ciphers is a weak method of cryptography. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.

The formula of encryption is:

$$E_n(x) = (x + n) \bmod 26$$

The formula of decryption is:

$$D_n(x) = (x_i - n) \bmod 26$$

2. **Shift Cipher.**

SUB: Information Security

The shift Cipher is similar to Casear Cipher. Here only the key value is fixed i.e. 3. So the formula changes to

The formula of encryption is:

$$E_n(x) = (x + 3) \bmod 26$$

The formula of decryption is:

$$D_n(x) = (x_i - 3) \bmod 26$$

3. Brute Force Attack on Ciphers

A brute force attack against a cipher consists of breaking a cipher by trying all possible keys. Statistically, if the keys were originally chosen randomly, the plaintext will become available after about half of the possible keys are tried. During the attack, the intruder tries all possible keys or passwords until the correct one is found. The amount of time necessary to break a cipher is proportional to the size of the secret key . A brute force attack is a simple yet reliable tactic for gaining unauthorized access to individual accounts and organizations' systems and networks

Example:

1) ATTACK K=3
2) ACADEMY K=25

SUB: Information Security

1.

CODE:

Caesar Cipher

```
def encryption(s, key):
    encrypt= ''

    for char in s:
        if char in alow:
            index = (ord(char) - ord('a') + key) % 26
            encrypt+=alow[index]
        elif char in ahigh:
            index = (ord(char) - ord('A') + key) % 26
            encrypt+=ahigh[index]
        else:
            rencrypt+=char
    return encrypt

text=input("Enter the string: ")
key=3
result=encryption(text, key)
print("Encrypted value is",result)
```

OUTPUT:

```
Plain text: ATTACK
Key: 3
Cipher text DWWDFN
```

SUB: Information Security

```
def decrypt(s, key):
    print("Cipher text ",s)
    key=-key
    result = ''

    for char in s:
        if char in alow:
            index = (ord(char) - ord('a') + key) % 26
            result+=alow[index]
        elif char in ahigh:
            index = (ord(char) - ord('A') + key) % 26
            result+=ahigh[index]
        else:
            result+=char
    print("Decrypted key is",result)
    result

decrypt(result, key)
```

```
Cipher text  DWWDFN
Decrypted key is ATTACK
```

SUB: Information Security

2.

CODE:

Shift cipher

```
alow=[]
for i in range(26):
  alow.append(chr(97 + i))
ahigh=[]
for i in range(26):
  ahigh.append(chr(65 + i))
print(alow)
print(ahigh)
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
```

```
def encryption(s, key):
    encrypt= ''

    for char in s:
        if char in alow:
            index = (ord(char) - ord('a') + key) % 26
            encrypt+=alow[index]
        elif char in ahigh:
            index = (ord(char) - ord('A') + key) % 26
            encrypt+=ahigh[index]
        else:
            rencrypt+=char
    return encrypt

text=input("PLain Text: ")
key=int(input('Enter the key: '))
result=encryption(text, key)
print("Cipher Text",result)
```

```
PLain Text: ACADEMY
Enter the key: 25
Cipher Text ZBZCDLX
```

SUB: Information Security

```
#Decrypting
def decrypt(s, key):
    print("Chiper text:",s)
    key=-key
    result = ''

    for char in s:
        if char in alow:
            index = (ord(char) - ord('a') + key) % 26
            result+=alow[index]
        elif char in ahigh:
            index = (ord(char) - ord('A') + key) % 26
            result+=ahigh[index]
        else:
            result+=char
    print("Decrypted key is",result)

decrypt(result, key)
```

```
Chiper text: ZBZCDLX
Decrypted key is ACADEMY
```

SUB: Information Security

3.

CODE:

Brute force

```
possible=[]
def brute(s):
    print("Encrypted string is ",s)
    for i in range(1,26):
        result = ''
        key=-i
        for char in s:
            if char in alow:
                index = (ord(char) - ord('a') + key) % 26
                result+=alow[index]

            elif char in ahigh:
                index = (ord(char) - ord('A') + key) % 26
                result+=ahigh[index]
            else:
                result+=char
        possible.append(result)
    return possible

possible = brute("lipps")
print("All the possibilities are\n")
print("(key, 'Posiibilities')")
for count, ele in enumerate(possible, 1):
    print (count, ele)
```

SUB: Information Security

OUTPUT:

```
Cipher Text:DWWDFN
Encrypted string is  DWWDF
All the possibilities are

(key, 'Posiibilities')
1 CVVCEM
2 BUUBDL
3 ATTACK
4 ZSSZBJ
5 YRRYAI
6 XQQXZH
7 WPPWYG
8 VOOVXF
9 UNNUWE
10 TMMTVD
11 SLLSUC
12 RKKRTB
13 QJJQSA
14 PIIPRZ
15 OHHOQY
16 NGGNPX
17 MFFMOW
18 LEELNV
19 KDDKMU
20 JCCJLT
21 IBBIKS
22 HAAHJR
23 GZZGIQ
24 FYYFHP
25 EXXEGO
```

```
Cipher Text:ZBZCDLX
Encrypted string is  ZBZCDLX
All the possibilities are

(key, 'Posiibilities')
1 YAYBCKW
2 XZXABJV
3 WYWZAIU
4 VXVYZHT
5 UWUXYGS
6 TVTWXFR
7 SUSVWEQ
8 RTRUVDP
9 QSQTUCO
10 PRPSTBN
11 OQORSAM
12 NPNQRZL
13 MOMPQYK
14 LNLOPXJ
15 KMKNOWI
16 JLJMNVH
17 IKILMUG
18 HJHKLTF
19 GIGJKSE
20 FHFIJRD
21 EGEHIQC
22 DFDGHPB
23 CECFGOA
24 BDBEFNZ
25 ACADEMY
```

SUB: Information Security

Conclusion:

In conclusion, we successfully designed and implemented encryption and decryption algorithms for both the Caesar Cipher and the Shift Cipher. These algorithms allow for the secure encryption and decryption of messages with a specified shift value. However, both ciphers are vulnerable to brute force attacks due to their limited key space (26 possible shifts). The brute force attack demonstrated the importance of using strong encryption methods with larger key spaces for better security.