

Assignment1

OpenSSL & RSA

1. Show your setup environment and you have installed openssl(on any environment like WSL,Mac OS, Linux, etc) (2 Points)

My system is Mac os and openssl version is 1.1.1. k.

```
(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % openssl version
OpenSSL 1.1.1k  25 Mar 2021
```

2. Generate private and public key (2 Points)

```
(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % openssl genrsa -out rsa.private 1024
```

```
Generating RSA private key, 1024 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

```
(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % openssl rsa -in rsa.private -out rsa.public -pubout -outform PEM
writing RSA key
```

3. Pick a file, encrypt it with the public key (3 Points)
[Your response goes below.]

```
(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % openssl rsautl -encrypt -inkey rsa.public -pubin -in test1.txt -out test1.enc
(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % ls
Applications      README.md
Desktop           WeChatProjects
Documents         firsttestingencryption.txt
Downloads         hs_err_pid3073.log
Library           nltk_data
Movies            opt
Music             rsa.private
Pictures          rsa.public
Postman           test1.enc
Public            test1.txt
PycharmProjects

(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % hexdump -C ./test1.enc
00000000  3b 0a 93 5f bf 1d 6a 2a  fe 26 70 82 ce ab d8 44  |;....j*.&p....D|
00000010  83 0b c3 6c 9d cb 99 5f  6e d9 56 20 24 a2 0f a3  |...l....n.V $...|
00000020  46 2f 3d 20 03 06 99 11  b5 14 7b 45 fe 43 4d 17  |F/= .....{E.CM.|
00000030  1a b5 6f d3 1f 3e 0a 23  6a a3 7c a6 84 31 77 85  |..o.>.#j...1w.|
00000040  6e 22 5b 3f b5 c7 bb ef  2d 1a 53 8e b4 a5 20 c4  |n"[?....-S... .|
00000050  c9 6c 16 be 7a 07 58 7d  da 18 58 76 ab bd d1 a3  |.l..z.X)..Xv....|
```

4. Decrypt the encrypted file with the private key (3 Points)

[Your response goes below.]

```
(base) jinchengbaby@jinchengbabys-MacBook-Air ~ % openssl rsautl -decrypt -inkey rsa.private -in test1.enc > test2.txt
```

II. Caesar Encryption (10 Points)

1. Suppose we have Shift Key = 13, and text = "SECURITY IS IMPORTANT"

What would be the encrypted message? (2 Points)

Note, the shift is right shift, that is key = 1, 'a' -> 'b'

[Your response goes below.]

The answer would be "FRPHEVGL VF VZCBEGNAG"

2. Now, given the text "SECURITY IS IMPORTANT" and the encrypted message you got from Step 2. Can you write a brute force function that returns the key?

You need to write the function code and show that it passes the test case "SECURITY IS IMPORTANT". You will also need to provide a screenshot for this step.(4 Points)

[Your response goes below.]

```

message = 'SECURITYISIMPORTANT' #encrypted message
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

encrypt_message = "FRPHEVGLVVFVZCBEGNAG"

for key in range(len(LETTERS)):
    translated = ''
    for symbol in message:
        if symbol in LETTERS:
            num = LETTERS.find(symbol)
            num = num - key
            if num < 0:
                num = num + len(LETTERS)
            translated = translated + LETTERS[num]
        else:
            translated = translated + symbol

    if translated == encrypt_message:
        print(key)
    #print('Hacking key %s: %s' % (key, translated))

```

13

3. What is the time and space complexity of the hacking (2 Points)
[Your response goes below.]

Set the length of message to be N , the letter length would always be 26, the time complexity is $O(26 * N) \Rightarrow$ The time complexity would be $O(N)$

The space complexity would also be $O(N)$

4. If the text size is large, could you find a better way to do the hacking or approach the problem? Explanation is enough. No need to post the code screenshot. (2 Points)
[Your response goes below.]

Using the dictionary to store the 26 shift of alphabet which would increase the loop up time complexity to be $O(1)$, it might increase the overall speed.

III. Diy: make your own cipher (Open Design Problem) 10 Points

1. Completeness of your codes that contains at least encrypt and decrypt functionalities **(3 Points)**

```
monoalpha_cipher = {
```

```
    'a': 'm',
```

```
    'b': 'n',
```

```
    'c': 'b',
```

```
    'd': 'v',
```

```
    'e': 'c',
```

```
    'f': 'x',
```

```
    'g': 'z',
```

```
    'h': 'a',
```

```
    'i': 's',
```

```
    'j': 'd',
```

```
    'k': 'f',
```

```
    'l': 'g',
```

```
    'm': 'h',
```

```
    'n': 'j',
```

```
    'o': 'k',
```

```
    'p': 'l',
```

```
    'q': 'p',
```

```
'r': 'o',  
's': 'i',  
't': 'u',  
'u': 'y',  
'v': 't',  
'w': 'r',  
'x': 'e',  
'y': 'w',  
'z': 'q',  
    ':',  
}
```

```
inverse_monoalpha_cipher = {}
```

```
for key, value in monoalpha_cipher.items():
```

```
    inverse_monoalpha_cipher[value] = key
```

```
def encrypt(message):
```

```
    encrypted_message = []
```

```
    for letter in message:
```

```
        encrypted_message.append(monoalpha_cipher.get(letter, letter))
```

```
    return "".join(encrypted_message)
```

```
def decrypt(encrypted_message):

    decrypted_message = []

    for letter in encrypted_message:

        decrypted_message.append( inverse_monoalpha_cipher.get(letter, letter))

    return ".join( decrypted_message )
```

2. Your explanation of your design how to approach the problem **(4 Points)**

[Your response goes below.]

A monoalphabetic cipher uses fixed substitution over the entire message.

We can build a monoalphabetic cipher using a Python dictionary, from the above dic, is the original encrypt key and we need to reverse the key to have the one dictionary used for decryption. Then we can use the two dictionary to encrypt and decrypt using simple iteration through the message.

3. Write and pass some test cases. **(3 Points)**

```
message = "security is important"
```

```
message = "security is important"
```

```
def encrypt(message):
    encrypted_message = []
    for letter in message:
        encrypted_message.append(monoalpha_cipher.get(letter, letter))

    return ''.join(encrypted_message)
```

```
encrypt_message = encrypt(message)
encrypt_message
```

```
'icbyosuw si shlkoumju'
```

```
def decrypt(encrypted_message):
    decrypted_message = []
    for letter in encrypted_message:
        decrypted_message.append( inverse_monoalpha_cipher.get(letter, letter))

    return ''.join( decrypted_message )
```

```
decrypt(encrypt_message)
```

```
'security is important'
```

```
message = "I love security"
```

```
def encrypt(message):

    encrypted_message = []
    for letter in message:
        encrypted_message.append(monoalpha_cipher.get(letter, letter))

    return ''.join(encrypted_message)
```

```
encrypt_message = encrypt(message)
encrypt_message
```

```
'I gktc icbyosuw'
```

```
def decrypt(encrypted_message):
    decrypted_message = []
    for letter in encrypted_message:
        decrypted_message.append( inverse_monoalpha_cipher.get(letter, letter))

    return ''.join( decrypted_message )
```

```
decrypt(encrypt_message)
```

```
'I love security'
```