Assignment #1. Cryptography

컴퓨터소프트웨어학부 2018008722 유수영

1. 사용한 프로그래밍 언어 : 파이썬(3.7.0 버전)

2. 컴파일 환경 및 방법 : Windows에서 코드 작성 및 실행

   1) Window CMD창에서 실행

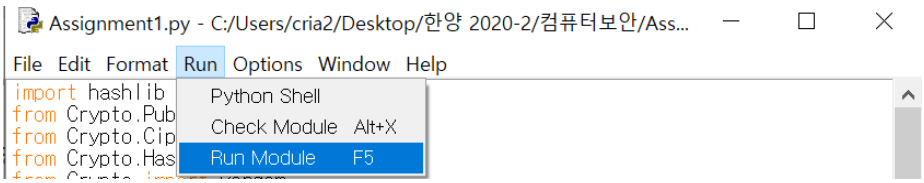
```
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:₩Users₩cria2>cd C:₩Users₩cria2₩Desktop₩한양 2020-2₩컴퓨터보안

C:₩Users₩cria2₩Desktop₩한양 2020-2₩컴퓨터보안>python Assignment1.py
```

   ➔ cmd창에서 코드가 있는 path로 이동한 다음, python [코드파일 이름].py로 컴파일 및 실행

   2) python shell에서 실행



   ➔ idle 코드 파일에서 F5(Run module)을 통해 컴파일 및 바로 실행

3. 사용 API : Crypto와 hashlib

```
C:₩Users₩cria2>cd C:₩Users₩cria2₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Python 3.7

C:₩Users₩cria2₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Python 3.7>pip install pycryptodome
Requirement already satisfied: pycryptodome in c:₩users₩cria2₩appdata₩local₩programs₩python₩python37₩lib₩site-packages
3.9.8)
```

   ➔ Crypto API 사용을 위해 pycryptodome을 cmd창에서 설치

```
import hashlib
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_v1_5
from Crypto.Hash import SHA
from Crypto import Random
from Crypto.Cipher import AES
from Crypto.Cipher import DES
from Crypto.Cipher import DES3
```

4. 구현한 암호화 방법

   1) 대칭키 암호화 : DES, 3DES, AES 구현

   2) Hash 함수 : SHA1, SHA256, SHA384, SHA512 구현

3) 비대칭키 암호화 : RSA 구현

5. 함수 설명

1) hashFunction

```
def hashFunction(plain) :
    print("₩n---------- Hash Function ----------")
2   while True :
3       hashtype = input("hash type(SHA1 / SHA256 / SHA384 / SHA512) : ")
        if hashtype == "SHA256" :
4           m = hashlib.sha256()
            # use 'encode()' to make plaintext(string) to bytes
            m.update(plain.encode())
            print(m.hexdigest())
5           return
        elif hashtype == "SHA384" :
            m = hashlib.sha384()
            m.update(plain.encode())
            print(m.hexdigest())
            return
        elif hashtype == "SHA512" :
            m = hashlib.sha512()
            m.update(plain.encode())
            print(m.hexdigest())
            return
        elif hashtype == "SHA1" :
            m = hashlib.sha1()
            m.update(plain.encode())
            print(m.hexdigest())
            return
3       else :
            print("Please enter only one of SHA1, SHA256, SHA384 or SHA512
```

1- hash function을 위한 함수로, 사용자가 입력한 plain text를 plain이라는 이름으로 받는다. 이 함수 안에서 SHA1, SHA256, SHA384, SHA512를 수행하며, 각 hash 함수들은 hashlib에 있는 함수들을 이용하여 실행된다.

2- 사용자가 잘못된 입력을 했을 경우 안내 후 다시 입력을 할 수 있도록 하기 위해 무한루프를 사용한다.

3- 사용자에게 hash type 4가지 중 하나를 입력 받고, 4가지 중 하나를 입력하지 않으면 다시 입력하도록 한다. (올바른 값을 입력할 때까지 계속해서 입력받는다.)

4- 각 경우에 hsahlib의 sha1, sha256, sha384, sha512함수를 사용하여 각 hash 객체들을 만들어 m이라는 변수에 넣어주고, 사용자에게 입력 받은 plaintext를 m에 바이트 객체로 공급해준다. (바이트 객체로 공급해야 하므로 encode함수 – utf8사용 - 를 이용하며, 공급은 update 함수로 이루어진다) hash 함수를 이용해서 encrypt된 결과는 hexdigest함수를 이용해서 확인할 수 있다.

5- Hash 함수를 이용한 encrypt가 끝나면 return을 하여 다음 단계로 넘어간다.

2) RSAFunction

```
def RSAFunction(plain):
    print("₩n---------- Asymmetric Encryption(RSA) ----------")
2   while True :
        length = input("key length(Choose above 1024) : ")
        if length.isdigit() :
            if int(length) >= 1024 :
                # variable 'rsakey' is a class RsaKey
3               rsakey = RSA.generate(int(length))

4               cipher = PKCS1_v1_5.new(rsakey)

                if cipher.can_encrypt() :
5                   cipherText = cipher.encrypt(plain.encode())
                    print("encrypted :",cipherText)

6                   sentinel = b"DECRYPT ERROR"
                    dePlain = cipher.decrypt(cipherText,sentinel)
                    # use 'decode()' to make bytes to plaintext(string)
                    print("decrypted :",dePlain.decode())

                else :
                    print("This plain text can not encrypt with RSA")
                return

            else :
                print("Please enter ar least 1024")
        else :
            print("Plesae enter number")
```

1- 비대칭 암호화를 하기 위한 RSA암호화 기법을 수행하는 함수이다. 사용자가 입력한 plain text의 내용을 plain이라는 변수로 받는다.

2- 사용자에게 RSA에서 사용할 key의 길이(1024bit 이상)를 입력 받고, 숫자가 아니거나 1024 미만을 입력하면 다시 입력할 수 있도록 무한루프와 if-else구문을 사용한다. (적절한 값을 입력할 때까지 계속해서 입력 받도록 한다.)

3- RSA의 generate 함수를 이용해서 사용자가 입력한 길이의 key를 생성하고 rsakey 변수에 저장한다.

4- Encryption과 decryption을 수행하기 위한 cipher를 PKCS1_v1_5의 new함수를 이용하여 만든다. Key는 RSA로 만든 값을 사용한다.

5- Encryption이 가능하면 encryption을 수행한다. Plain text를 byte의 형태로 바꾸어 진행하며, 이 때 plain text의 길이는 RSA modulus - 11보다는 짧아야 한다. Encrypt한 결과는 cipherText 변수에 저장한다.

6- decrypt에서 에러가 일어났을 때 return하게 될 sentinel을 byte형 문자열로 설정해준 뒤, decrypt를 실행한다. encrypt했던 결과를 decrypt하기 위해 parameter로

넘겨주며, decrypt한 결과를 보여주기 위해서 decode하여 출력한다.

3) Symmetric

```
def Symmetric(plain) :
    print("---------- Symmetric Encryption ----------")
    while True :
        ciphertype = input("cipher type(DES / DES3 / AES) : ")
        if ciphertype == "DES" :
            DESFunction(plain)
            return
        elif ciphertype == "DES3" :
            DES3Function(plain)
            return
        elif ciphertype == "AES" :
            AESFunction(plain)
            return
        else :
            print("Please enter only one of DES, DES3 or AES")
```

1- 대칭키 암호화를 위한 함수이다. 사용자가 입력한 plain text를 plain이라는 이름으로 받으며, 사용자가 DES, DES3, AES 중 선택하는 암호화 방법에 따라 각각 DESFunction, DES3Function, AESFunction을 실행한다.

2- 사용자가 3가지 방법 이외의 것을 입력하면 다시 입력할 수 있도록 무한 루프를 돈다.(적절한 입력을 할 때까지 계속 입력을 받는다.)

4) DESFunction

```
def DESFunction(plain) :
2   while True :
        key = input("key(Type 8 but use 7 only) : ")
        if len(key) == 8 :
3           DESkey= key.encode()
            cipher = DES.new(DESkey, DES.MODE_ECB)

            # make plaintext 8 bytes through padding because DES has a fixed data block size of 8 bytes
            # use underbar for padding char
            # DES block size = 8
4           plain = plain + ((DES.block_size - (len(plain) % DES.block_size)) * '_')

5           cipherText = cipher.encrypt(plain.encode())
            print("encrypted: ", cipherText)

6           dePlain = cipher.decrypt(cipherText)
            print("decrypted : ", dePlain.decode())
            return
        else :
            print("Please enter a key that is 8 in length")
```

1- 대칭키 암호화를 할 때 사용자가 DES 방법을 선택했을 때 Symmetric함수에서 실행이 되는 함수이다. 사용자가 입력했던 plain text를 Symmetric함수에서 그대로 plain이라는 이름의 변수로 넘겨받는다.

2- 사용자에게 DES의 key로 사용할 문자열을 입력받는다. DES는 56bit 짜리 key를 사용하기 때문에 8글자를 입력받고 그 중 7글자를 사용한다. 길이가 8인 문자열

을 입력하지 않으면 다시 입력을 받으며, 길이가 8인 key를 입력할 때까지 계속해서 문자열을 받도록 무한 루프를 돈다.

3- 사용자가 길이가 8인 key를 입력하면, 변수 key에 그 문자열을 저장한다음, 바이트로 만들기 위해 encoding한다. Encoding된 key값을 이용하여 DES cipher(class EcbMode)를 만들어 cipher라는 변수에 저장한다. (만들어진 이 cipher는 DES encrypt, decrypt를 실행할 수 있는 함수가 있다.)

4- DES는 64bit의 plaintext block을 사용하는 block cipher기법이기 때문에, 사용자가 입력했던 plain text를 64bit의 배수의 길이로 맞춰주는 padding을 실행하도록 한다. (plain text의 길이가 64bit의 배수가 아니라면 _를 plain text에 추가해주어 길이를 64bit의 배수로 가지도록 만들어준다.)

5- Encrypt를 하고, 이를 cipherText 변수에 저장한 뒤 출력한다(plain은 문자열에서 byte로 바꿔주는 encode를 실행한 뒤 encrypt를 위해 넘겨진다)

6- Decrypt를 하고, 이를 dePlain 변수에 저장한 뒤 출력해준다. 출력은 byte를 다시 문자열로 고쳐주는 decode를 실행한 뒤 출력한다.

5) DES3Function

```
def DES3Function(plain) :
2   while True :
        key = input("key(16 / 24) : ")
        if (len(key) == 16 and key[:7] != key[8:15]) or (len(key) == 24 and key[:7] != key[8:15] and key[8:15] != key[16:23]) :
3           DES3key = DES3.adjust_key_parity(key.encode())
            cipher = DES3.new(DES3key, DES3.MODE_ECB)

            # make plaintext 8 bytes through padding because DES3 has a fixed data block size of 8 bytes
            # use underbar for padding char
            # DES3.block_size = 8
4           plain = plain + ((DES3.block_size - (len(plain) % DES3.block_size)) * '_')

5           cipherText = cipher.encrypt(plain.encode())
            print("encrypted : ", cipherText)

            dePlain = cipher.decrypt(cipherText)
            print("decrypted : ", dePlain.decode())
            return
        else :
            print("Please enter a key that is 16 or 24 in length.")
            print("If you enter a 16-digit key, the first eight parts and the last eight parts must be different. ")
            print("If you enter a 24-digit key, the eight parts in the middle mush be different form the first eight and the last eight")
```

1- 대칭키 암호화를 할 때 사용자가 3DES 방법을 선택했을 때 Symmetric함수에서 실행이 되는 함수이다. 사용자가 입력했던 plain text를 Symmetric함수에서 그대로 plain이라는 이름의 변수로 넘겨받는다.

2- 3DES는 DES를 3번 실행하는 암호화 기법인데, DES의 키를 2개나 3개를 사용하기 때문에 key의 길이가 16(byte) 또는 24여야 한다. 따라서 사용자에게 16 또는 24의 길이를 가지는 key를 입력 받고, 그 외 길이를 입력하면 다시 입력할 수 있도록 무한 루프를 돈다.(올바른 길이의 key를 입력할 때까지 계속해서 입력을 받는

다) 이 때, 파이썬이 이 실행에서는 3DES의 key에 대한 조건이 존재한다. 24를 세 부분으로 나누어 모든 sub key가 다른 값을 가지거나, 첫 번째 subkey와 세번째 sub key는 같더라도 이들은 두 번째 sub key와는 달라야 한다. 또한, 각 subkey들이 한 문자가 반복되는 값일 때, 그 문자들이 연속적이면 안된다(EX) 111111112222222233333333)

3- 사용자가 입력한 key를 adjust_key_parity를 통해 key를 adjust해준 뒤, 그 key를 이용해서 3DES cipher를 만들어 cipher변수에 넣어준다. (DES와 마찬가지로 cipher는 encrypt, decrypt할 수 있는 함수가 존재)

4- DES와 똑같이, 3DES도 plaintext block size가 64bit이기 때문에 _를 사용해서 padding을 해준다. padding해준 plaint text를 다시 plain변수에 저장해준다.

5- Encrypt와 decrypt를 실행한다. Encrypt를 실행할 때는 plain을 encode하여 byte로 바꿔주고, 다시 decrypt할 때는 byte를 문자열로 바꾸는 decode를 한 다음 출력해준다.

6) AESFunction

```
def AESFunction(plain) :
2   while True :
        # AES uses 128/192/256 bits key -> my python bit is 64bits(128/8 = 16)
        key = input("key(16 / 24 / 32) : ")
        if len(key) == 16 or len(key) == 24 or len(key) == 32 :
            AESkey = key.encode()
            cipher = AES.new(AESkey, AES.MODE_EAX)

            # make plaintext 16 bytes through padding because AES has a fixed data block size of 16  bytes
            # use underbar for padding char
            # cipher.block_size = 16
            plain = plain + ((cipher.block_size - (len(plain) % cipher.block_size)) * '_')
            cipherText = cipher.encrypt(plain.encode())

            print("encrypted : ", cipherText)

            de = AES.new(AESkey, AES.MODE_EAX, cipher.nonce)
            dePlain = de.decrypt(cipherText)
            print("decrypted : ", dePlain.decode())
            return
        else :
            print("Please enter a key that is 16 or 24 or 32 in length")
```

1- 대칭키 암호화를 할 때 사용자가 3DES 방법을 선택했을 때 Symmetric함수에서 실행이 되는 함수이다. 사용자가 입력했던 plain text를 Symmetric함수에서 그대로 plain이라는 이름의 변수로 넘겨받는다.

2- AES는 key size가 128, 192, 256 중 하나이다. 따라서 사용자에게 16, 24 또는 32 길이의 key를 입력 받고, 입력 받은 key의 길이가 조건을 충족하지 못하면, 올바른 값을 입력할 때까지 계속해서 다시 key의 값을 입력받도록 한다.

3- DES나 3DES와 거의 비슷한 흐름을 가지는데, 다만 AES의 plaintext의 block size는 64bit가 아닌 128bit이기 때문에, 128bit에 맞추어 plain text를 padding해준다.

그 외 encryption이나 decryption은 동일하다.

7) Main

```
def main() :
    plain = input("Plain text : ")
    print("")
    plain.rstrip("₩n")
    Symmetric(plain)
    hashFunction(plain)
    RSAFunction(plain)


if __name__ == "__main__":
    main()
```

1- main함수에서는 사용자에게 가장 먼저 암호화와 복호화를 진행할 plain text를 입력받는다.(DES나 DES3를 사용할 경우에는 한글을 plain text로 입력하면 안된다) 그리고, 대칭키 암호화, hash function 암호화, RSA(비대칭키)암호화 함수들을 차례로 실행해준다.

6. 실행 화면

1) DES / SHA1 / RSA - 1024의 길이

2) DES3 – key의 길이가 16 / SHA256 / RSA – 2048의 길이

```
---------- Symmetric Encryption ----------
cipher type(DES / DES3 / AES) : DES3
key(16 / 24) : 12345678aaaaaaaa
encrypted :  b'₩x05₩xff₩x9b₩xae@₩x92:tF₩t₩x8a>₩xea₩xb6₩xfft₩xa4₩x8f₩xd0₩xefm₩xfb₩xc
5₩xa4'
decrypted :  Computer Science________

---------- Hash Function ----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA256
445ddaa59df684ea81e04e1b2aad2ec113a4a8a36ce8c5e802c1542266b8921d

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 2048
encrypted : b"₩xa6C₩x04₩x0e 6₩x8c₩xf5^₩x1e₩xba-Z<A₩xaee₩xc1_₩xf8₩x17₩xe4₩x83₩xb1₩xb
2₩xe3₩x93₩xb8₩xaeA5V₩xd8'₩xc15₩x1b₩xbd₩x03₩xc3₩xe7₩x8c₩x98₩x88₩xb0z,R₩xca₩x9a₩xf8₩x
06₩xca₩xb82y₩xc85₩xbc₩xb6D:₩xb0₩x1f₩xb5t₩xff4₩xe2₩x0f₩n₩xabV₩xe9K~4v₩xb1₩xaci₩xdbD₩
x13₩xf6₩xd7₩x16₩x15₩x1a'2₩xdcxZ₩x83₩xa2₩xd2₩x9e0₩x08₩xc2-₩xf8dj₩xedjs60₩xa4₩x87₩xfc
₩x83.b₩x10₩xef₩xa42#;₩x89₩x9e₩xa7₩x9bQL1₩xce₩xa6g₩xc6j₩xda₩xdc₩xcd₩xdc₩xb1u4[₩x9b₩x
01₩xe0₩xf9I₩xebE7:@R₩xbbDt₩xfc₩x97₩x18₩xe3₩xf2z₩x0f₩xe5₩xea₩x84z&₩xaa₩xe6₩x9a₩xbb₩x
d9₩xb3₩x04;₩xae?₩xef₩x81₩xf0₩xf9₩x1cnJ₩x1f₩r)/₩x8c₩x9b₩xa0₩x10₩xdf₩xdc₩xd8#G>₩x82₩x
b6&₩xc8₩xca₩x84₩xac₩xa3₩xd33NhY`=₩xd4/₩xca₩xed₩x1dT6₩xd7P₩x03₩xa3}₩xd1f₩r₩xf7b?₩x0f
₩x8c₩xd7₩x0eu₩x07₩x16₩x93₩xf6₩xd4₩xe2₩xd0E4₩x1e₩x9c₩xc3₩x1b₩r₩xbdt₩xb2₩xb8₩x16"
decrypted : Computer Science
>>>
```

3) DES3 – key의 길이가 24 / SHA384 / RSA – 1025의 길이

```
======= RESTART: C:/Users/cria2/Desktop/한양 2020-2/컴퓨터보안/Assignment1.py =====
==
Plain text : Cryptographic Tools~!~!

---------- Symmetric Encryption ----------
cipher type(DES / DES3 / AES) : DES3
key(16 / 24) : 12341234aaaaaaaa98769876
encrypted :  b'₩xb3₩x1c₩x82j₩x86₩x17₩xda2₩x9b₩xddn₩x8a₩xd8₩xd3₩xff₩xa7₩xac+Y₩x7fI₩x
bf₩x8b₩xfc'
decrypted :  Cryptographic Tools~!~!_

---------- Hash Function ----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA384
86c5d2865aaaf248c71142991436823156b3c120c61a8a265e66ed379bbe8010e632a575eee2fba2f2c
78d1e67475c84

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 1025
encrypted : b'₩x00fD₩xc8₩n₩x84₩xff₩xd7;₩xb8c₩x9d₩xc8₩x84S{5₩xf0I₩xe5(E₩x03₩x8f₩x93₩
x81AI₩xbd₩xcd₩x1btu₩x97₩xb87C₩xd2₩xc4W₩x85bF₩xfe₩x1c₩xe4₩xc8₩xae-yihDIS₩x08₩xfep₩xd
7₩xea₩xf3₩xd9₩x96₩x8e₩x14₩xe5₩x86₩xd9₩₩₩x04₩xacb₩x1eak=₩x8cs=p₩xda₩t₩xb7[₩xbarJ₩xbf
V₩xa3₩x19₩x9b9.₩x97₩xed₩x9b<₩xca₩xefT₩xcfBn₩x9d₩x85kIR*₩xdf₩xa7g₩xae96₩xfc<fTx₩x81>
I+CD₩xc1y'
decrypted : Cryptographic Tools~!~!
>>>
```

4) AES / SHA512 / RSA – 2000의 길이

```
>>>
======= RESTART: C:/Users/cria2/Desktop/한양 2020-2/컴퓨터보안/Assignment1.py =====
==
Plain text : Hello World Hello World Hello World

---------- Symmetric Encryption -----------
cipher type(DES / DES3 / AES) : AES
key(16 / 24 / 32) : 1234123412341234
encrypted :  b'₩x1dh₩x15+₩x83₩xe58₩x0fi₩xdb₩xa0₩xe5₩xc5 ₩xc4,u₩xba₩x81₩xdb₩xf6₩x8cg
₩x93₩x96<₩xa2₩x90F₩xa6₩x85₩xa9₩x0f ₩xcb₩xe5₩xc6m<s₩xa3₩xf2₩x90₩xfa₩xa5E₩xc00'
decrypted :  Hello World Hello World Hello World_____________

---------- Hash Function -----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA512
b28f367c08ca48618fc86cfe5b55bacf551d6ea29fea888430299d9ae42e3974da72bd4dfae8dade9c8
a4b22333a8fbec7c55c0806f3e9714253624a3cf585bc

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 2000
encrypted : b'P₩xaf₩xfe₩xd4k₩x04f₩x9c₩xaf₩xafL₩xf5a₩xe6₩xb200₩xc0<₩x03₩xf75₩x05~₩x8
8₩xe91₩xecG{₩xb8Gw₩xa7₩tp₩xd1₩xd6₩xe1₩xa0₩xf1gA₩x9f₩xbc₩xa4₩xf3₩x93₩x8c₩xce:₩xd1₩xb
f₩x0f₩xd2₩xcf₩xfc₩xc5q₩x1eCX₩x88@₩xe6₩xf4₩x06g₩xa4₩xb3₩x9e₩x80QH₩xcai~F₩x00₩xbb₩'₩x
00₩₩₩x7f₩x85₩x97₩x07jJ₩xa3₩x1b₩x1f₩xabY₩x87₩x8a4w₩xb8₩x00₩x97₩xd4₩xf4=₩x12]₩xe2₩x93
!₩xaa₩x8c₩x03₩xa8.₩t₩x86d₩xa6₩xf9!₩xc5₩xf7₩xdc₩xdc₩x8a₩x1b]₩x7fP₩xaa₩xf8W₩xbd₩xf3₩x
92 ₩x1b₩xa6₩x06S₩x82₩x00`@₩x13₩xc2r-)₩xbdl₩xfb₩x81C₩x1f₩xb6₩x89₩x94₩x16₩x8f₩x86₩x1b
₩x01_u₩x13#₩xd5₩t₩xbfW₩x0co~c₩₩₩xa1₩xb6₩'₩x96₩x8e7₩x16₩x84₩xceb₩x98₩xdf₩x83₩x83₩x03
₩x03₩x1b₩x82₩xe86₩x9e₩xbf₩xed-₩xa9Y₩xbd₩x947M)₩x8cp₩x9d₩xf0₩xd4₩xa5₩xa0b₩xc2₩x18₩xb
d₩xf7a₩x88₩x1d"₩xe8cS₩x13₩xf7₩xba₩xc3G₩xcb₩x15*₩x05W₩xe3₩x15Y₩x08₩x18₩n₩xad₩xf6#₩xb
0₩x87,₩rH'
decrypted : Hello World Hello World Hello World
>>>
```

5) 입력해야 하는 key의 길이와 맞지 않는 key를 입력하거나, 명시되어 있지 않은 암호화 방법을 입력했을 경우 예외 처리

```
>>>
======= RESTART: C:/Users/cria2/Desktop/한양 2020-2/컴퓨터보안/Assignment1.py =====
==
Plain text : This is the case of incorrect input

---------- Symmetric Encryption -----------
cipher type(DES / DES3 / AES) : ARC4
Please enter only one of DES, DES3 or AES
cipher type(DES / DES3 / AES) : DES
key(Type 8 but use 7 only) : 1234
Please enter a key that is 8 in length
key(Type 8 but use 7 only) : 12341234
encrypted:  b'o₩x90₩xc7₩xdcXw₩x90i₩xc2₩xf5₩xf2₩xdc₩x97₩x08₩xcf₩xd8P₩xd06₩xa2₩xb3₩₩₩
x16₩x18l₩x95₩xa4₩xa8~₩x19₩xa2!₩xfd₩x0e₩x9b₩xc4₩xac₩x85`W'
decrypted :  This is the case of incorrect input_____

---------- Hash Function -----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA
Please enter only one of SHA1, SHA256, SHA384 or SHA512
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA1
5db6db7941c921453f84e7f96c6f4699fa52fdad

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 1000
Please enter ar least 1024
key length(Choose above 1024) : 1024
encrypted : b'₩x07₩xa7₩xe1N₩x925A96X₩xd0₩xe1₩x1d₩xffS₩xa3B₩x14₩xc8₩x17₩x1e₩xe0}₩xd2
₩xc8₩x10₩xcd<₩x1bB₩xb8₩xad{₩xfch₩xb0#₩xae0₩x11₩x83₩x91]₩xecX9$X₩x95c[]Xr₩x9b₩xdaA₩x
91_₩xb9V₩xee~₩xb1₩xf8₩xf2₩xdc₩x9c.₩x18₩xbb,₩x95ah₩x1f₩x96₩xb3₩xdf₩x01w@₩xe0₩x16e₩xe
f₩xe84<₩xe9₩xee@₩x14₩xd8₩x13₩x99s₩xc8₩x1f₩xe3₩x0c₩x88₩x92!₩xe6g-F)₩xb4YF|₩xe9₩xc6^₩
x8a₩xa2L₩xbd8₩xc8₩x83<₩x07<₩xdc₩x1e'
decrypted : This is the case of incorrect input
>>>
```

6) 문자열의 길이가 100일 경우 잘 실행

```
======= RESTART: C:/Users/cria2/Desktop/한양 2020-2/컴퓨터보안/Assignment1.py =====
==
Plain text : aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

---------- Symmetric Encryption -----------
cipher type(DES / DES3 / AES) : DES
key(Type 8 but use 7 only) : 12341234
encrypted:  b'₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf
4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9
₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩
xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xad₩xdf₩x854K₩xf4₩xf9₩xae₩xb1₩x
1e₩xe9₩x877₩x83₩xd8₩xb8'
decrypted :  aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa____

---------- Hash Function -----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA256
2816597888e4a0d3a36b82b83316ab32680eb8f00f8cd3b904d681246d285a0e

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 1024
encrypted : b"₩xbcl₩xab₩xbdGW7%₩xf2L₩x07₩xf2*₩xccX₩x8f₩x8c₩x1b)₩xc1₩x9e:D₩x98₩xfcU%
₩x96₩xe8₩xc3*₩xccG₩xfb ₩xaa@₩xc9₩x10₩xaep₩x90₩x93₩x80₩xe9p₩xea₩xc1₩x9e₩xc8r₩xc9₩x8a
#₩xa4₩xb0.₩xdb`?₩xf6₩xf3q₩x16₩xf9₩xee₩x05o'₩x9fT₩xe0₩xcf₩x00₩xd4₩xd8₩x1e₩xe0w₩x80₩x
fa₩x1f₩xb7S=t₩x07₩x9dl₩xc1₩x9f₩xbe₩xc8g₩xbf₩xeca.5₩x0e₩xc9h₩xe7₩x8c₩xe2₩xf9₩x85₩xb3
₩xd2jkg₩x08₩xf2mD₩x96₩xf7j2₩xf7₩xcd₩xf9A+WjP"
decrypted : aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
>>> |
```

7) 문자열이 한글일 경우

```
>>>
======= RESTART: C:/Users/cria2/Desktop/한양 2020-2/컴퓨터보안/Assignment1.py =====
==
Plain text : 안녕하세요 저는 유수영입니다

---------- Symmetric Encryption -----------
cipher type(DES / DES3 / AES) : AES
key(16 / 24 / 32) : 1234123412341234
encrypted :  b'x₩x90₩x93₩xb0₩x0cy₩xd3?₩xc9mm₩x8d₩x8d₩xcaH₩xf8₩x8e₩xa7₩xf9₩xba₩x89#₩
xf1₩xc9₩x1d₩x9bT~₩xf2₩xff₩xfb₩x89₩x9c₩x17f%₩x08₩xdf₩x10₩xc4+₩x9e'
decrypted :  안녕하세요 저는 유수영입니다_

---------- Hash Function -----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA1
34d2861806bba89a26ad10ecedc80a574051a973

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 1024
encrypted : b'L₩xb9^₩xb2₩x81₩xc9₩x82₩xe6₩x0c₩xec₩xd6₩xff₩x8e₩x1f₩xc4₩xf3j₩xc0₩xbc₩x
d0u₩xb4₩x8c₩xe1₩xe53}₩x08Y₩xf2₩x18₩xd4₩xe7₩xe2₩x88₩x04₩'₩xe9₩xaeK₩'₩xa7₩xdd₩xb8₩x9e
₩xa4#₩xe3₩xdcH₩xfc₩x17₩'₩x04yM₩x18₩x94₩xcb3^₩xb1X₩xfb₩xd9₩xba[₩x1b~9M>₩x93`oD.n₩x84
D₩x87₩xf1₩xaf₩x9907V₩x1e₩xae₩xa1₩xda₩xb5₩x8e`₩x86₩x01₩xfa₩xcb0₩xa74;₩x9dfV₩x8f₩xd5₩
xb5₩xaa₩xaaj₩xe1c0₩xd8₩xc0₩x1a₩xdf₩xab₩x19"p₩x99₩xfa₩x82₩x7f₩xfd₩x86'
decrypted : 안녕하세요 저는 유수영입니다
>>>
```

8) CMD 창에서 실행

```
Plain text : This is cmd

---------- Symmetric Encryption -----------
cipher type(DES / DES3 / AES) : ARC4
Please enter only one of DES, DES3 or AES
cipher type(DES / DES3 / AES) : DES
key(Type 8 but use 7 only) : 1234
Please enter a key that is 8 in length
key(Type 8 but use 7 only) : 12341234
encrypted:  b'o₩x90₩xc7₩xdcXw₩x90i?₩x11₩xc8₩x946₩x97₩xd7h'
decrypted :  This is cmd_____

---------- Hash Function -----------
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA
Please enter only one of SHA1, SHA256, SHA384 or SHA512
hash type(SHA1 / SHA256 / SHA384 / SHA512) : SHA1
3a312183fe9eb31279e62e6dbbc38f2609916960

---------- Asymmetric Encryption(RSA) ----------
key length(Choose above 1024) : 10
Please enter ar least 1024
key length(Choose above 1024) : 1024
encrypted : b"1P₩x0b₩x8b₩x13be[₩xf70Qj₩xc8₩xcfz₩xb6?₩x9a₩x9f₩xfbw₩x14p!R₩x89₩xc5₩x0
6₩xc3/₩xcb₩x9c₩xb4₩x18₩x03₩xabS₩xfd₩xbd[₩xd7|.₩xab₩xed=₩x0cNco=₩xe2_₩xb9₩x7f₩xfe₩₩₩
xaf₩xb8t₩xd6₩xaeCh₩x02₩x18₩x02₩o₩₩₩x15s₩x8f₩xedd₩xa6₩xef:₩xee₩xd8₩xf1₩x0e₩xdc₩xac$₩
xba₩xf7&₩xcb`₩x81₩x1bGIdx₩xd7e₩xb6Q₩x16₩x7ff₩x92o₩xb3₩x06₩x1a₩xdb₩xa6₩xc8₩x83₩xce₩x
97₩x13hEo₩x98₩xea₩x9c₩xc3₩x02₩xb1<O₩x8cs"
decrypted : This is cmd

C:₩Users₩cria2₩Desktop₩한양 2020-2₩컴퓨터보안>
```