Secure-Shared-File-Storage-Using-Hybrid-Cryptography and FTP

Project Team:

Ahmed Khaled Saad Ali 1809799

Omar Yehia 18p7177

Mahmoud Abdalla Mohasseb 20p2787

Requirements:

1.First, we need to split file into N blocks.

2.We need to encrypt all parts of file by two encryption algorithm DES, AES & RSA

3.We will generate 3 symmetric keys randomly to be used as one key for one algorithm.

4.Algorithm used for encryption is changed with every part in round robin fashion.

5.Grouping keys used for encryption in one file and encrypt that file with different algorithm.

6.Key used for encryption of key file is generated randomly and it is called master key

7.The encrypted parts of file and key file should be uploaded to FTP server.

8.We should store master key locally in a file its name is encrypted file uploaded to FTP server.

9.User can download encrypted file and key file from FTP server.

10.To decrypt these files user should provide its public key to owner.

11.Owner will encrypt master key with user's public key and send it to user.

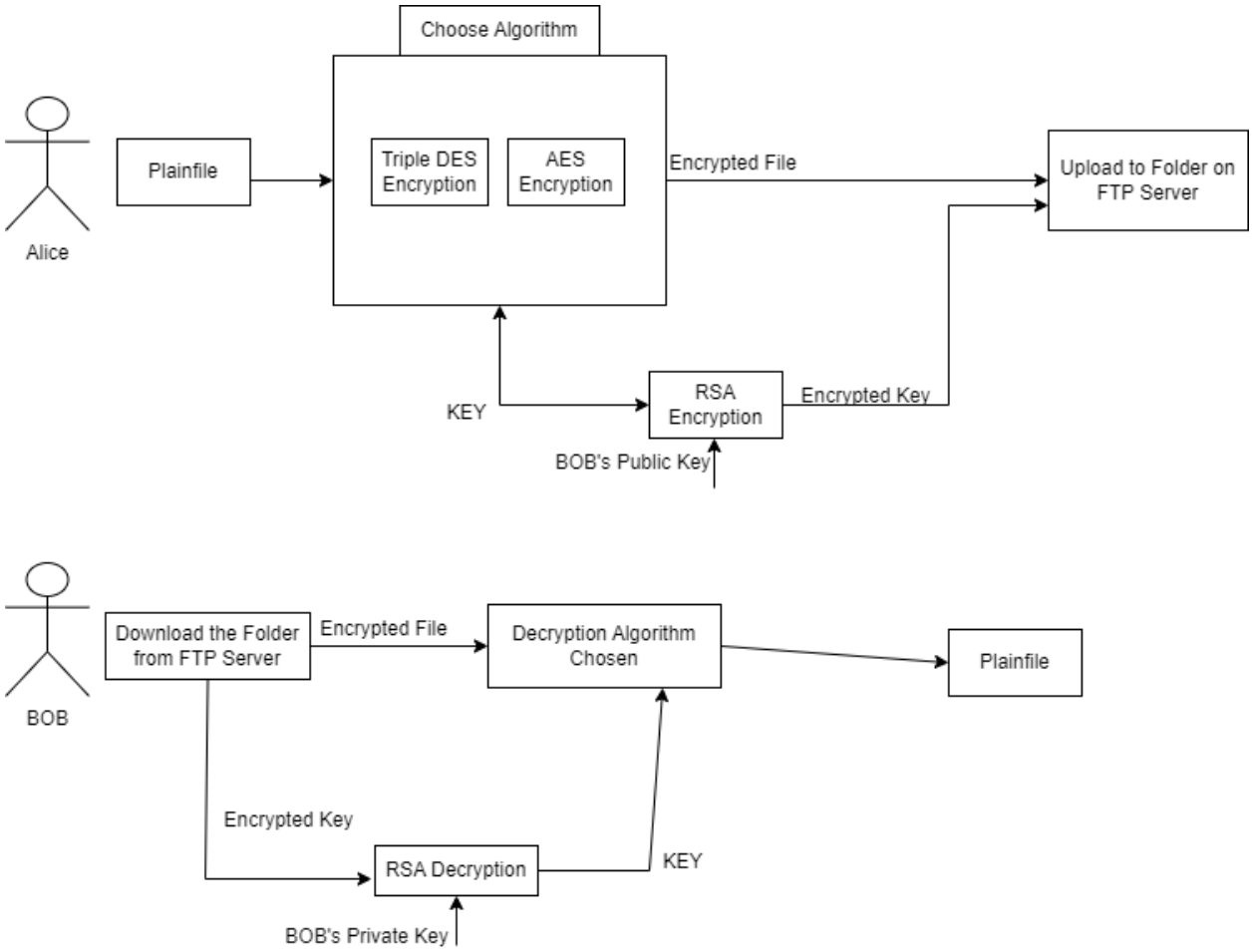
12.The user will decrypt master key with its private key and then use master key for decryption of key file.

13.User now have keys used for encryption of file ,so user will use these keys for decrypting file

Achieved Requirements:

1. Used whole file didn’t split it
2. Encryption algorithms implemented AES & DES & RSA for Encryption of Key
3. Generated 2 random keys for algorithms (AES & DES)
4. Encryption done by choice not round robin
5. At sender,Key of algorithm encrypts file and and Key gets encrypted by RSA algorithm using public key of receiver to a KeyFile
6. The encrypted file and key file uploaded to FTP server.
7. At receiver, we download key file and encrypted file
8. Reciever decrypts key file with his private key which will be used to decrypt encrypted file

Design:


Choose Algorithm
Alice
Plainfile
Triple DES Encryption
AES Encryption
Encrypted File
Upload to Folder on FTP Server
KEY
RSA Encryption
Encrypted Key
BOB's Public Key
BOB
Download the Folder from FTP Server
Encrypted File
Decryption Algorithm Chosen
Plainfile
Encrypted Key
RSA Decryption
KEY
BOB's Private Key

Implementation

Project Code Drive Link:

https://drive.google.com/drive/folders/1MPFjruE1GSMLf5bTG2vHn9s3GExpaNah?usp=share_link

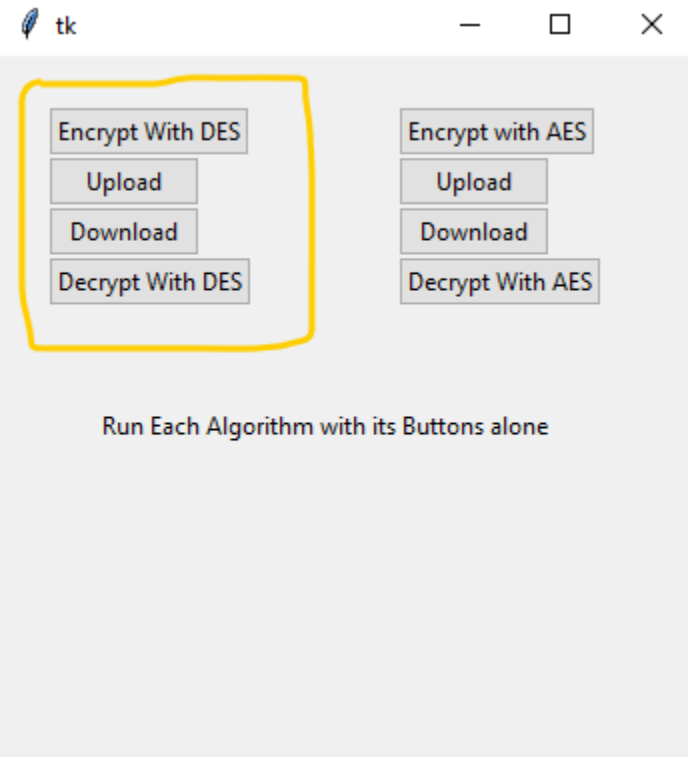
Test Cases:

You can Encrypt & Decrypt with both Algorithms each one after another, but once you commit to one algorithm you must press all its buttons

You should run this command to have FTP Server running:

```
python -m python_ftp_server -u "username" -p "P@ssw0rd" --ip 0.0.0.0 --port 6060 -d "c:\ftptemp"
```

1. DES Encryption Algorithm: We will use DES buttons only



Plain file: data.txt

data.txt - Notepad

File Edit Format View Help

```
    key = "key.txt"

    # Encode given key to 16 byte ascii key with md5 operation
    key_hash = md5(key.encode('ascii')).digest()

    # Adjust key parity of generated Hash Key for Final Triple DES Key
    tdes_key = DES3.adjust_key_parity(key_hash)

    #  Cipher with integration of Triple DES key, MODE_EAX for
Confidentiality & Authentication
    #  and nonce for generating random / pseudo random number which is used
for authentication protocol
    cipher = DES3.new(tdes_key, DES3.MODE_EAX, nonce=b'0')

    # Open & read file from given path
    with open(file_path, 'rb') as input_file:
        file_bytes = input_file.read()

        if operation == '1':
            # Perform Encryption operation
            new_file_bytes = cipher.encrypt(file_bytes)
        else:
            # Perform Decryption operation
            new_file_bytes = cipher.decrypt(file_bytes)
```

Pressing “Encrypt with DES”

Generated data_enc.txt

data_enc.txt - Notepad

File Edit Format View Help


```
å>T3îŠn▯▯▯NƒMÑ▯>▯'ÏÇýÞ▯$›?Žk¢'P®'ÌÇß•å▯*▯▯Ìƒ˜ £pÛAý°▯¦ž=p“♠ ▯▯±óüÄ[\▯▯›[▯<ˆüi©Ù
[„écØ9Z¾îØªÓ!îQŸ# ▯,ÿRm¥PñÃº“úˆ™vW¾Ü`WD(paºU▯EVÛ0‡±^à2õ@Áñž▯▯7kÿ¹kÂ•š•ch&Qi▯Ê
£å¤8éÅ£ÿKâ<KhnDÂO´n"û▯»EYumLÅ>§¤▯åêâ¢S.&í™ID&EKðü@Ÿ`-
ŸhMÓ▯*ê‡)uºž®▯¿¹▯Äó÷6wã®«ú¼ijn&Â▯+½½Š▯
ßƒz6z¨Ÿ¤▯SdN£Ú1Àm▯6É b§HÃíðð›s@íçt*Ÿðxzªß▯»▯k ¿>Æ±/ÄÓ¼À▯§¸çáÇŠ▯ž4õ#Ò▯3,|
áÊØcGBœ▯èSxx;)O™µ!º$hÊÍaöìâ±xAÂC÷ëÓe
‰▯3Åå]E¹*ã ¬Ž›÷çÀ¹:ÅFˆà*±à,{*h¾ÌÝ▯Ü▯Ûáç!è>TƒÒY÷â¨v©vcE ¹3}6öØp▯+"D
▯íÕ«+è▯óäŽVò£= ºdŠ ▯»x▯X¶2/¶¤▯]▯uíED<*£«+.i™^Z▯“¦fxçÀ$'&«
Õ¬*¿¦¼TäËljp¶bWB▯0dPÀØK˜¬▯ôjkèÎŒ;àŸè3Y÷%ë▯▯ÉëË
’▯: 6Ç▯ÿZä}OÈ?.Òâ0▯•▯D!€,Ô▯Ø|*▯gz¢a▯œ9CX¤;ä^€ŠUšÑÙƒƒV:Î▯ßù▯,eã-nˆ¡Áúv3N›„UäœC>Ÿ
´ƒÚó▯"JJä«µy:«▯Å¼¬˜@ŽwŒðw½<ò,}▯;iGÙ~JñiAßô/M’Áï™£Ë▯½6·^?¬▯Ä´ú…▯¦¹H▯Q´w▯òŽYì–˜XÔ
±z”*r¤
1±ÒCð«lakÂœ-▯¾©õé▯%8Æa’ Yœ˜%_
¬ÖŸ~ 8q▯F:û •Ô-▯ ù«ßÆ|*ãêáE;zbL’ñj♠▯U=ó›Òð†▯‰¬·F±¾ÿ▯♠L▯ÌB‘²RÌ>í
±ÒiÑ}▯Ó7•Þ÷»ÒßÕŒÓ0¢▯9}▯Ie▯}Tv
L„”´$ ¸çVç²O
▯Ô¨ºÃ¢ã»Œ«▯ÂÍêsÆƒ♠ñ▯˜▯·RÃw
```


Wrote generated DES key in key.des then append key.des in Keyfile.txt (2nd line)

key.des - Notepad

File Edit Format View Help


```
FKKlbw4XP6AZjYi_YYyEH4lVRRtU7ZHSNVQUzreJhtA=
```


keyfile.txt - Notepad

File Edit Format View Help


```
MD8SAuGFkEsB2kUMLTkZywDdZrA_bmQ-3seOhPPg0II=
FKKlbw4XP6AZjYi_YYyEH4lVRRtU7ZHSNVQUzreJhtA=
```


Encrypted Keyfile.txt by Public_Key.pem with RSA Algorithm in encrypted_keys.txt

public_key.pem - Notepad

File Edit Format View Help


```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtm0z79Yh9kPoBUMVzevf
7fvKOH1JVhGQO7iCqBl4vpNyzjqPTi/clav/gIv2CTo1MLHVDq461PocuRS9FV1H
xiIWb32DPiDJhgrVnTHcZoretc3htpCUv92IxaSE6Yz3IBhCTw22KYQ5zMeYT+vD
NOMER9wM8KFQkoDOaNi3pk5hnRtvg0V6rHxcu0lfQMZfixxdmpYzmoMfPDUu6rR4
S+sIUJAYEpsr5XYs5IWeq+A2aJpxuZqnnLaNLthszgKiI4LePWW7Xaz/g2SXVXZU
qUD9KrtOsuFy6dyr/hzVu+wQd+ZMZvw/rJqwtySBx10lgink0cUGRZF20M7aQgTt
3QIDAQAB
-----END PUBLIC KEY-----
```

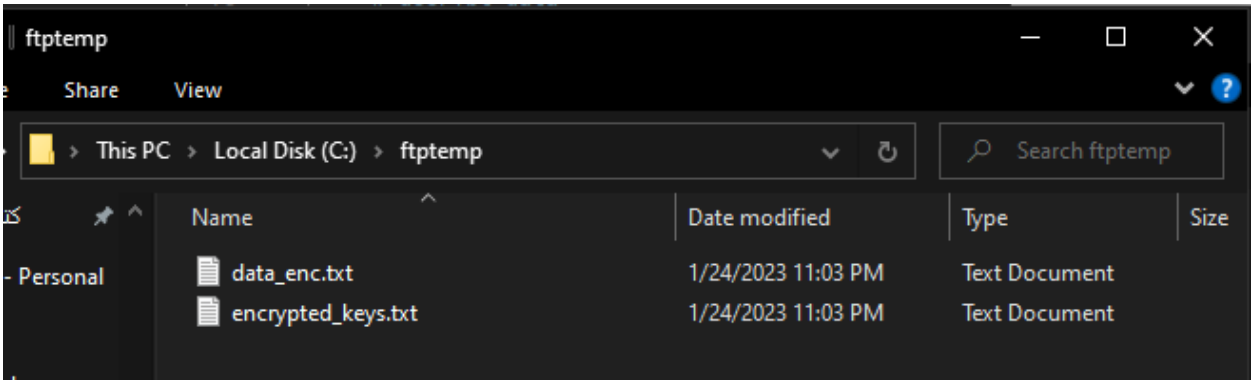

encrypted_keys.txt - Notepad

File Edit Format View Help


```
瀧줍回ᄼ톼ㅋ餡ᄒ췰痢ᄒ獸ᄒ吕ᄒᄒᄒ檻赴啡ᄒᄒᄒᄂ煨랍ᄒᄒᄒ袜넬ᄒᄒᄒ詳댈娠ᄒ晴ᄒ繆룝살ᄀ鍔ᄇ뤽땐깃山爍虹ᄒᄒ萌
廬혤꼍눴錢ᄂ揀乏ᄋ堡ᄇᄒ廳췰揞嘗ᄒ隅ᄒ疮ᄒ霰ᄒ澜坐ᄒᄒ翊ᄀ髏酔唇凈ᄅ蛞蟾梃뷕诮ㅎ糔ㅂᄀ 洉蟋췰哺ᄒᄀ
愛珗ᄒᄋ毅湲룦ᄌ졜쑨ᄂᄉ籺ᄉ췰癡壩뮴뷉揆ㅂ쾽
```

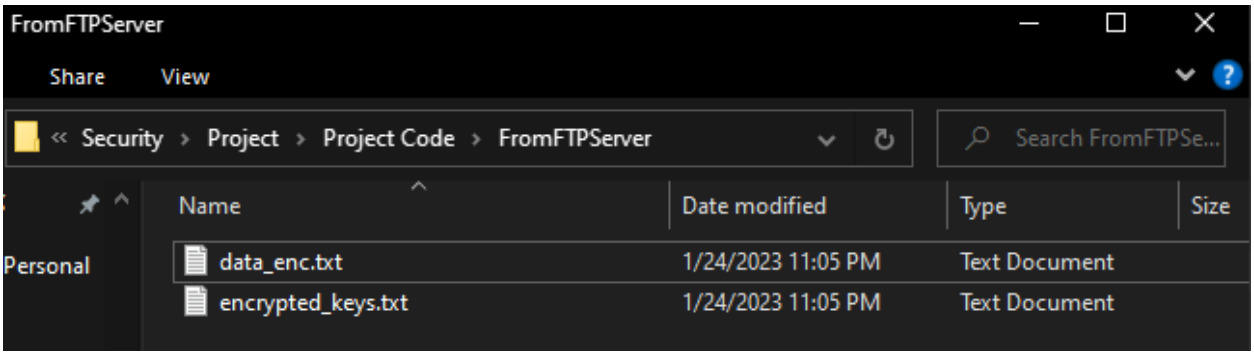

Pressing “Upload”

Uploaded data_enc (encrypted file) & encrypted_key.txt to FTP Server



Pressing “Download”

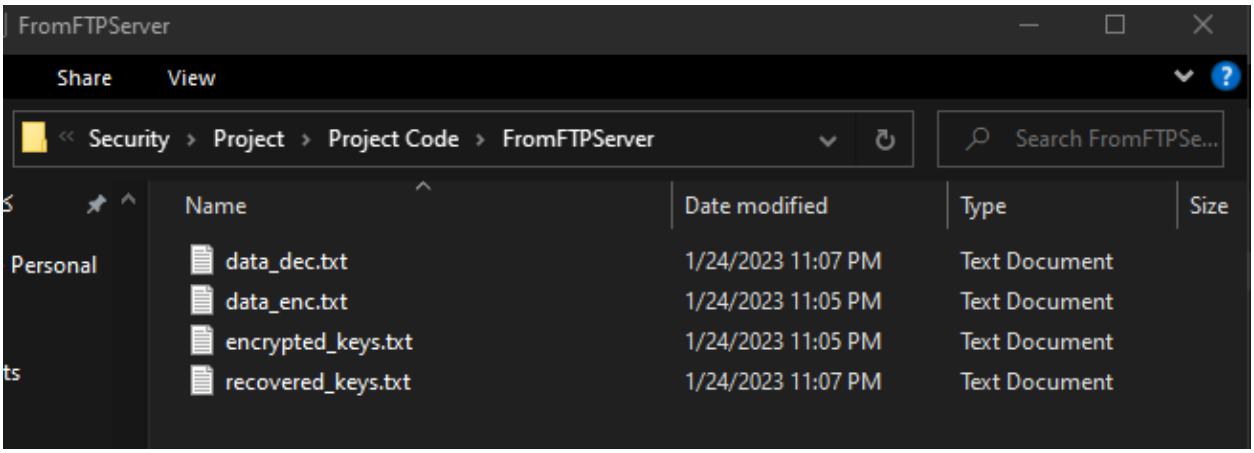
Downloaded files from FTP server to a local folder “FromFTPServer” for Reciever



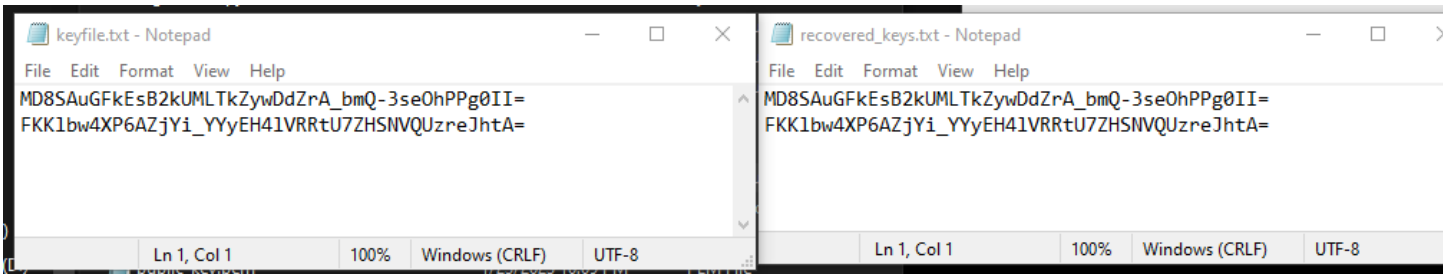
Pressing “Decrypt With DES”

Reciever now decrypted files and have **recovered_keys** which he got from decrypting encrypted_key with the **reciever’s private key**

Then used **recovered_keys** to decrypt **data_enc** to have the Plain File back in **data_dec**



See that **keyfile** that was generated from sender is same as **recovered keys** from reciever



See that **data.txt** (Original Plain file) at sender is same as **data_dec.txt** at receiver

So, Operation successful.

data.txt - Notepad

File Edit Format View Help

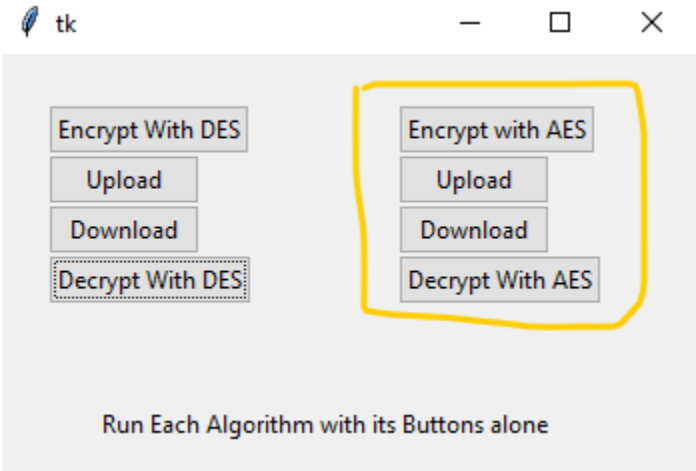
```
    key = "key.txt"

    # Encode given key to 16 byte ascii key with md5 operation
    key_hash = md5(key.encode('ascii')).digest()

    # Adjust key parity of generated Hash Key for Final Triple DES Key
    tdes_key = DES3.adjust_key_parity(key_hash)

    #  Cipher with integration of Triple DES key, MODE_EAX for Confidentiality &
Authentication
    #  and nonce for generating random / pseudo random number which is used for
authentication protocol
    cipher = DES3.new(tdes_key, DES3.MODE_EAX, nonce=b'0')

    # Open & read file from given path
    with open(file_path, 'rb') as input_file:
        file_bytes = input_file.read()

        if operation == '1':
            # Perform Encryption operation
            new_file_bytes = cipher.encrypt(file_bytes)
        else:
            # Perform Decryption operation
            new_file_bytes = cipher.decrypt(file_bytes)
```

data_dec.txt - Notepad

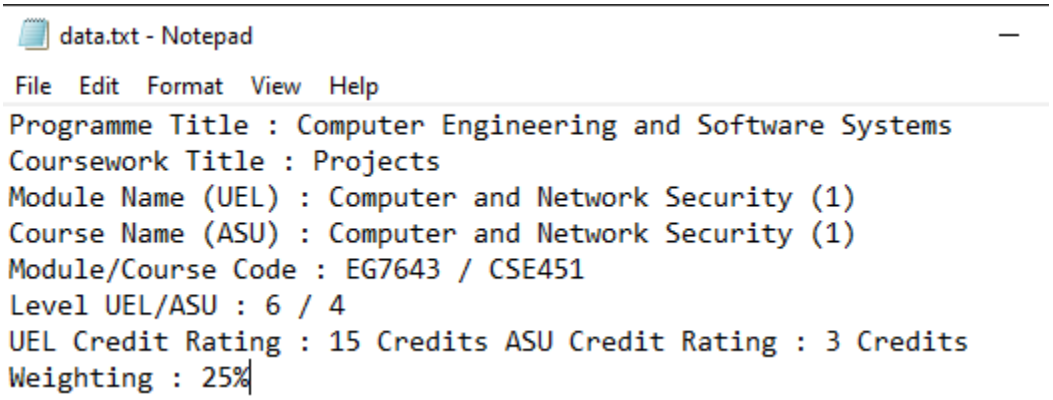
File Edit Format View Help

```
    key = "key.txt"

    # Encode given key to 16 byte ascii key with md5 operation
    key_hash = md5(key.encode('ascii')).digest()

    # Adjust key parity of generated Hash Key for Final Triple DES Key
    tdes_key = DES3.adjust_key_parity(key_hash)

    #  Cipher with integration of Triple DES key, MODE_EAX for Confidentiality &
Authentication
    #  and nonce for generating random / pseudo random number which is used for
authentication protocol
    cipher = DES3.new(tdes_key, DES3.MODE_EAX, nonce=b'0')

    # Open & read file from given path
    with open(file_path, 'rb') as input_file:
        file_bytes = input_file.read()

        if operation == '1':
            # Perform Encryption operation
            new_file_bytes = cipher.encrypt(file_bytes)
        else:
            # Perform Decryption operation
            new_file_bytes = cipher.decrypt(file_bytes)
```

2.AES Encryption Algorithm: Use Only AES Buttons



Plain file: **data.txt**



Pressing "Encrypt with AES"

Data.txt encrtpted to data_enc by key.key (AES generated key)

data_enc.txt - Notepad

File Edit Format View Help


```
gAAAAABj0ExO6ILEnUHrGsEoyeHph0NG-
xznfgyOooiQ8_UyyuLEUhq6vaDt_yjKdxnblK0Osr9DzZxWX42v1i2sDMTjsnTy6AuiKjaUV--
8DyfWM40AbMH4vqD2W5XXf6VwCFLh3sahUBxH-FVIPp26wpzN1zp6mN7-
RUnCNcNWBfCYv9c4yRVEmwqFhVt6dYqwBnjB3-0fK5Zz4iVknEx53SfV-
AskAJJpcdy7dkEtxBP1c9n3p8Kvuj1jZZQJdXqZI3VJ57PJKWKMixGuruF90p0ZSCtpNbQ6Wyn
vTsLtBX1PC4GtvPvDXmf7C4TwAuWgWNl05bqmFfdArNcPyzwGyWr9bmc7eWsbxzvXO1kVBO7pT
7-
R7RlrSt2ijS_cc0ATOV71YWBLS3sMIYkRzjC5fzHovlcLr1rkZyPuisLKY3tH2Dtlpo44iyRO3
acQOSIDUhUvUSxuSaDEsvF3p5iRipCBg5ZvQ_DjPT91dUb3WV91uYZDkjPsgmAI_3KBiDK8yT-
RHaeBD9ckRPsX2cf5zoa5LuByUg==
```


key.key - Notepad

File Edit Format View Help


```
JxJ0xaFIOHBPZ23U7m1Rlp06RcQoRxkjYeY86xC91kw=
```


Saved AES key in **key file** (1st line)

keyfile.txt - Notepad

File Edit Format View Help


```
JxJ0xaFIOHBPZ23U7m1Rlp06RcQoRxkjYeY86xC91kw=
FKKlbw4XP6AZjYi_YYyEH4lVRRtU7ZHSNVQUzreJhtA=
```


Encrypted **keyfile** to **encrypted keys.txt** using public key with RSA algorithm

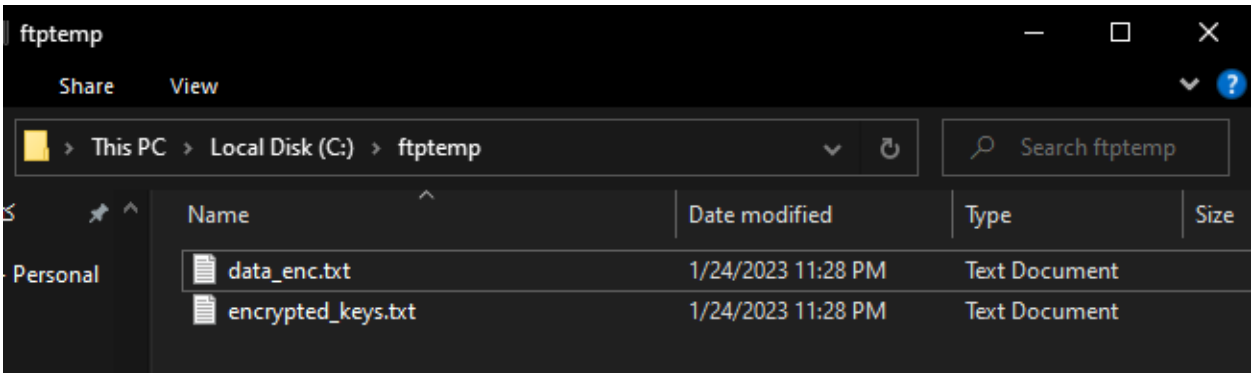
encrypted_keys.txt - Notepad

File Edit Format View Help


```
[illegible]
```

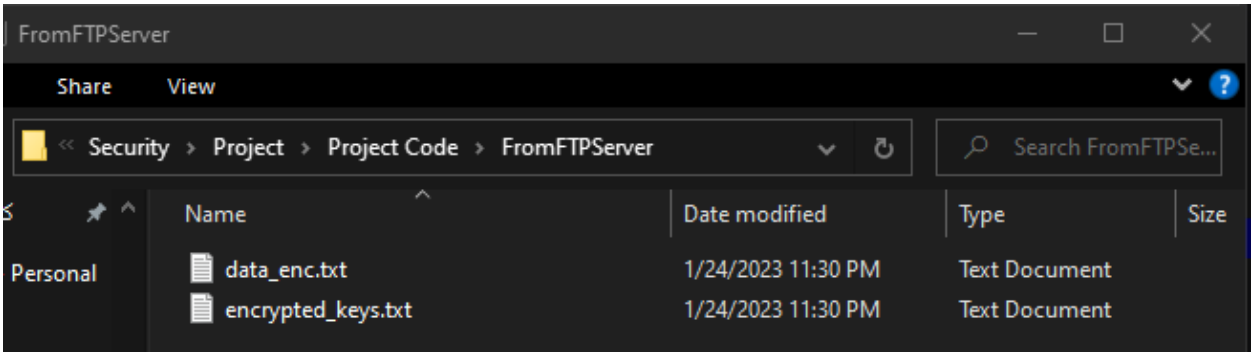

<u>Pressing"Upload"</u>

Uploaded **data_enc** (encrypted file) & **encrypted_key** to FTP Server



Pressing"Download"

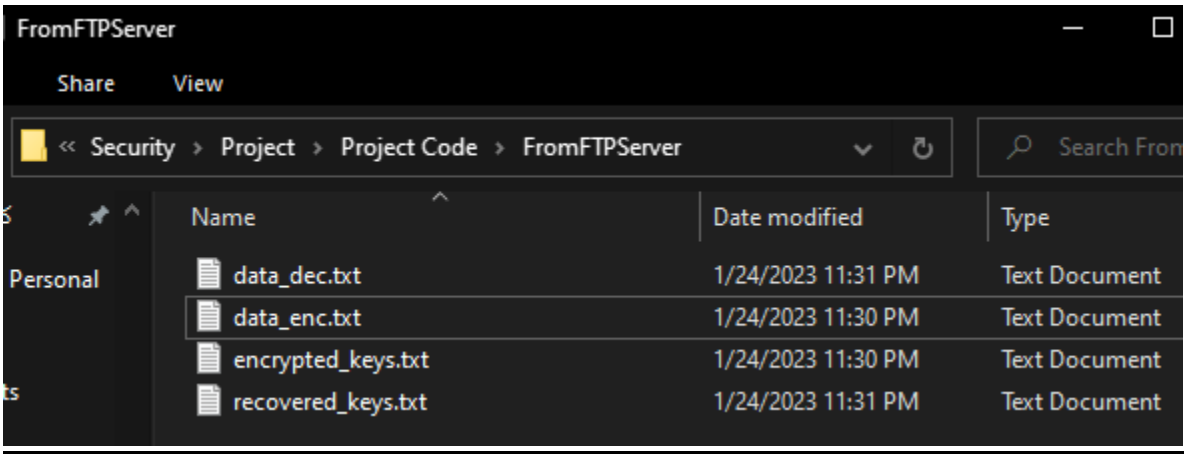
Downloaded files from FTP server to a local folder "FromFTPServer" for Reciever



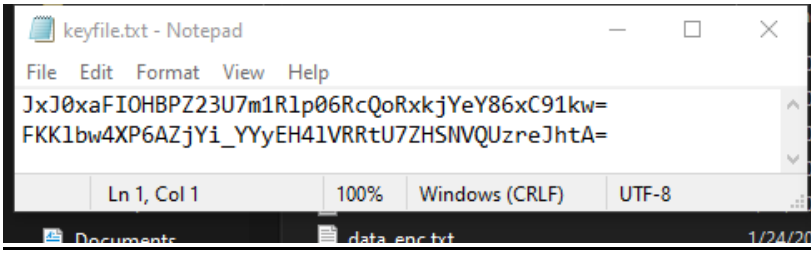
Pressing "Decrypt with AES"

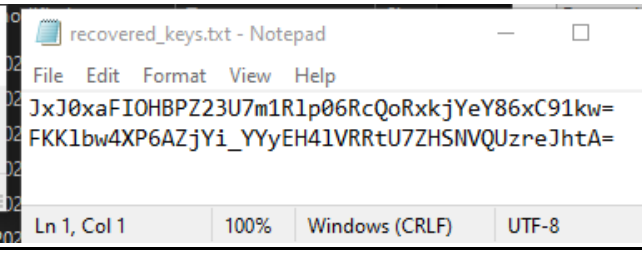
Reciever now decrypted files and have **recovered_keys** which he got from decrypting **encrypted_key** with the **reciever's private key**

Then used **recovered_keys** to decrypt **data_enc** to have the Plain File back in **data_dec**



See that **keyfile** that was generated from sender is same as **recovered keys.txt** from reciever





See that data.txt (Original Plain file) at sender is same as data_dec.txt at receiver

So, Operation successful.

data.txt - Notepad

File Edit Format View Help

```
Programme Title : Computer Engineering and Software Systems
Coursework Title : Projects
Module Name (UEL) : Computer and Network Security (1)
Course Name (ASU) : Computer and Network Security (1)
Module/Course Code : EG7643 / CSE451
Level UEL/ASU : 6 / 4
UEL Credit Rating : 15 Credits ASU Credit Rating : 3 Credits
Weighting : 25%
```

---

data_dec.txt - Notepad

File Edit Format View Help

```
Programme Title : Computer Engineering and Software Systems
Coursework Title : Projects
Module Name (UEL) : Computer and Network Security (1)
Course Name (ASU) : Computer and Network Security (1)
Module/Course Code : EG7643 / CSE451
Level UEL/ASU : 6 / 4
UEL Credit Rating : 15 Credits ASU Credit Rating : 3 Credits
Weighting : 25%
```
