Ain Shams University
Faculty of Engineering

University of East London

Course Assessment Specification (CAS)

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%**

Maximum mark available:

- 25 on software project

Lecturer : Prof. Ayman M. Bahaa-Eldin

Contact : If you have any issues with this coursework you may contact your lecturer.

Contact details are: Email: ayman.bahaa@eng.asu.edu.eg

Hand-out Date : As shown in submission matrix
Hand-in Date : As shown in submission matrix
Hand-in Method : Submission through LMS
Feedback Date : Your work will be marked and returned within two weeks.

Introduction

This coursework is itemized into several parts to get the 60 marks associated to it.
You must use the templates provided by the instructor to prepare your work.
All assignments and projects will be handed-in electronically, while quizzes and exams are written
.

Learning Outcome to be assessed

5. Analyse different problems that may arise during data communication and the impact of different security breaches on computer security?
6. Select suitable ciphers for different applications
7. Implement different ciphers and cryptanalysis techniques
9. Work and communicate effectively in team by effective collaboration and task management, working in a constrained stressful environment, and leading and motivating individuals.

Detail of the task

Attached separate file for each task.

89% and above:

Your work must be of outstanding quality and fully meet the requirements of the coursework specification and learning outcomes stated. You must show independent thinking and apply this to your work showing originality and consideration of key issues. There must be evidence of wider reading on the subject. In addition, your proposed solution should:

- illustrate a professional ability of drafting construction details,
- express a deep understanding of the in-hand problem definition,
- and applying, masterly, the learned knowledge in the proposed solution.

76% - 89%:

Your work must be of good quality and meet the requirements of the coursework specification and learning outcomes stated. You must demonstrate some originality in your work and show this by applying new learning to the key issues of the coursework. There must be evidence of wider reading on the subject. In addition, your proposed solution should:

- illustrate a Good ability of drafting construction details,
- express a very Good understanding of the in-hand problem definition,
- and applying most of the learned knowledge, correctly, in the proposed solution.

67% - 76%:

Your work must be comprehensive and meet all of the requirements stated by the coursework specification and learning outcomes. You must show a good understanding of the key concepts and be able to apply them to solve the problem set by the coursework. There must be enough depth to your work to provide evidence of wider reading. In addition, your proposed solution should:

- illustrate a moderate ability of drafting construction details,
- express a good understanding of the in-hand problem definition,
- and applying most of the learned knowledge, correctly, in the proposed solution.

60% - 67%:

Your work must be of a standard that meets the requirements stated by the coursework specification and learning outcomes. You must show a reasonable level of understanding of the key concepts and principles and you must have applied this knowledge to the coursework problem. There should be some evidence of wider reading. In addition, your proposed solution should:

- illustrate a fair ability of drafting construction details,
- express a fair understanding of the in-hand problem definition,
- and applying some of the learned knowledge, correctly, in the proposed solution.

Below 60%:

Your work is of poor quality and does not meet the requirements stated by the coursework specification and learning outcomes. There is a lack of understanding of key concepts and knowledge and no evidence of wider reading. In addition, your proposed solution would be:

- Illustrate an inability of drafting construction details,
- Failed to define the parameters, limitations, and offerings of the in-hand problem,
- Failed to apply correctly the learned knowledge for proposing a valid solution.

Academic Misconduct

The University defines Academic Misconduct as 'any case of deliberate, premeditated cheating, collusion, plagiarism or falsification of information, in an attempt to deceive and gain an unfair advantage in assessment'. This includes attempting to gain marks as part of a team without making a contribution. The department takes Academic Misconduct very seriously and any suspected cases will be investigated through the University's standard policy. If you are found guilty, you may be expelled from the University with no award.

It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules. If you are unclear about what is required, please ask

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

Introduction

Objective: To Achieve a secure shared file access using Hybrid Cryptography and FTP
FTP is an old Internet protocol where files can be shared via upload/download mechanism.
The purpose of this project is to build a secure platform of file sharing using FTP and multiple ciphers encryption. The requirements are that a file owner can upload an encrypted file to an FTP server. Other users can download the file from the server and then request the encryption keys of the file using their public keys, hence only the users granted this key can decrypt the downloaded file
A simple FTP server and clients using python are shown in APPENDIX (A)
You can use any python crypto library like the pycryptodome(https://pypi.org/project/pycryptodome/) for different ciphers implementation

Methodology

To achieve the above goal, the following methodology needs to be followed:

File Storage:

1. Dividing the file to upload into N parts. (N depends on the file size)
2. Generate m keys randomly, where m is the number of symmetric ciphers used (at least 3 ciphers including DES and AES, and you may choose a third one or even your own cipher)
3. Encrypting all the parts of the file using one of the selected algorithms (Algorithm is changed with every part in round robin fashion). And the parts are put together in a single file as ordered.
4. The keys for cryptography algorithms are then grouped in a key file and encrypted using a different algorithm and the key for this algorithm is also generated randomly and is called the file master key.
5. The data file and the key file are than uploaded to the FTP server
6. A copy of the master key is kept in a local file with the file name to be shared.

This has to be done through a GUI APP with the entire process encapsulated in a single use-case that is "Secure Upload".

File Retrieval:

1. A user requesting the master key must provide his public key to the owner
2. The owner then encrypts the master key of the requested file with the requesting user public key and sends it to him
3. The user can then download the data file and the key file, decrypts the master key with his private key and then decrypts the data file

Requesting the master key can be done outside your app, but the encrypted master key must be imported to the application of the file retriever and used to decrypt the file.

APPENDIX (A), Simple FTP with python

First you need to install the python ftp server as follows:

```
python -m pip install python-ftp-server
```

Create a folder somewhere on your local storage

For example c:\ftptemp on windows or /home/ftptemp on Linux

Then run the ftp server, indicating the port number to listen on and the home folder for the FTP files. You also specify the user name and password for the users who would like to access the FTP server.

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

Make sure you run the server using the same user that you created the folder with (preferred to have admin privileges)

You should have something like this:
Local address: ftp://0.0.0.0:6060
User: username
Password: P@ssw0rd

Note that you have identified the port number as 6060 for your server, you asked to bind with all the Ips that the machine have (0.0.0.0 means any IP that the machine have) and you defined a username and a password

You can test your server using any FTP Client program such as WinCSP free FTP client program https://winscp.net/eng/download.php

Now create a folder, add a text file to it and name it "`sometextfile.txt`", Use the FTPUploader to upload it

```python
import ftplib
# FTP server credentials
FTP_HOST = "127.0.0.1"
FTP_PORT = 6060
FTP_USER = "username"
FTP_PASS = "P@ssw0rd"
# connect to the FTP server
ftp = ftplib.FTP()
ftp.connect(FTP_HOST,FTP_PORT)
ftp.login(FTP_USER,FTP_PASS)
# force UTF-8 encoding
ftp.encoding = "utf-8"
# local file name you want to upload
filename = "sometextfile.txt"
with open(filename, "rb") as file:
    # use FTP's STOR command to upload the file
    ftp.storbinary(f"STOR {filename}", file)
# quit and close the connection
ftp.quit()
```

Using your browser, check that the file is uploaded to the FTP home folder you used.

Add another text file and name it ”anothertextfile.txt” in the FTP server home directory . Use the downloader code to download it to another folder

```
import ftplib
FTP_HOST = "127.0.0.1"
FTP_PORT = 6060
FTP_USER = "username"
FTP_PASS = "P@ssw0rd"
# connect to the FTP server
ftp = ftplib.FTP()
ftp.connect(FTP_HOST,FTP_PORT)
ftp.login(FTP_USER,FTP_PASS)
# force UTF-8 encoding
ftp.encoding = "utf-8"
# the name of file you want to download from the FTP server
filename = "anothertextfile.txt"
with open(filename, "wb") as file:
    # use FTP's RETR command to download the file
    ftp.retrbinary(f"RETR {filename}", file.write)
# quit and close the connection
ftp.quit()
```