In this programming assignment, you will implement a simple image sharing system with several security features. You are allowed to work in groups of 2 members.

Prerequisite: It is assumed that you have main knowledge about socket programming.

The required features of the image sharing system are described as follows.

1. **Registration and Public Key Certification**. Each user should generate a public-private key pair once and register the server with her username and public key. Server would sign this key together with the username (it creates a certificate), stores the certificate and also sends a copy to the user. When user receives the certificate, she verifies that the certificate is correct and the public key is correctly received by the server. You may assume that public key of the server is known by every user (the server is considered as a trusted certificate authority).
2. **Image posting**. Any user can post an image to the system. Owner of an image first generates an AES key and encrypts the image with the AES key in CBC mode (Initialization Vector can be generated randomly). She also generates a digital signature of the image using her private key and SHA256 hash function. She then encrypts the AES key with the public key of the server. She sends a POST_IMAGE message along with the image_name, encrypted image, digital signature, encrypted AES key and Initialization Vector (IV) to the server. Server stores these together with the name of the owner.
3. **Notification.** When an image is posted to the server, server sends a "NEW_IMAGE image_name owner_name" message (where image_name is the name of the image and owner_name is the owner of the image) to all online users.
4. **Image download.** After receiving the NEW_IMAGE message, a user can download the image by sending a "DOWNLOAD image_name" message to the server. When server receives this message, it sends the encrypted image, digital signature, certificated public key of the owner, and the AES key encrypted with the public key of requesting user.
5. **Decryption and verification.** After receiving these, user first extracts the AES key. Next, she decrypts the image. Then, she checks the integrity and authentication of the image by verifying the digital signature. If everything is OK, she displays or stores the image.

Your system should avoid posting and downloading images without registration.

Print all the messages sent and received by any user or the server to a log file (except the large image files).

BONUS (up to 30 points): Provide an enhancement so that not all online users, but only intended users would be able to download and decrypt the images (even the server would not be able to decrypt). These users may be offline at the time of image posting. Describe your design choices and implementations in detail.

Security Holes

Although you will implement many security features described above, there could still exist some security holes. Try to identify security holes as much as possible and offer some solutions (You may implement additional features and get more bonus points).

You can use any programming language of your choice. Basic user interface would be OK.