Final Project due Dec 12 11:59pm on Canvas

In this project, you are going to design and (partially) implement a secure communication system between two parties.

Requirements:
The requirements of the system include:

1.) The two parties have each other's RSA public key. Each of them holds his/her own RSA private key.
2.) Each party's message (from a .txt file) is encrypted using AES before sending it to another party.
3.) The AES key used in 2) is encrypted using the receiver's RSA public key. The encrypted AES key is sent together with the encrypted message obtained from 2).
4.) Message authentication code should be appended to data transmitted. You are free to choose the specific protocol of MAC.
5.) The receiver should be able to successfully authenticate, decrypt the message, and read the original message.

You need to implement a program for each role (i.e., sender and receiver). You don't need to include actual socket programming in your code. You can just use local files as the channel to simulate the communication in the network. For example, to implement requirement 1 above, we let each party locally generate a key pair and save each key in a corresponding file. The other party will be able to know the public key by accessing the file. You can create a file called "Transmitted_Data", which can include all data transmitted between sender and receiver, i.e., encrypted message, encrypted AES key, and the MAC. This file is written by the sender and read by the receiver.

Programming language and library
You can choose either OpenSSL Crypto Library, Java Cryptography Architecture or Python Crypto for your project. For Java based implementation, more helpful information can be found in the following links.
https://docs.oracle.com/en/java/javase/19/security/java-security-overview1.html
https://howtodoinjava.com/java/java-security/java-aes-encryption-example/
https://www.tutorialspoint.com/java_cryptography/java_cryptography_creating_mac.htm

Deliverables

1. A report which includes following components.

Explanations of your system design, particular algorithms used, key lengths used, etc, and a brief explanation on how to use your programs. All the requirements above should be met.

2. Well-commented source codes.