Programming Assignment 2, Part 2
CS 478/513 Computer Security (Spring 2020)
Total Points: 100

Due: April 30^{th} 2020, 11:59 pm

This project is divided into some smaller related sub-projects for your convenience. You need to complete each sub-project individually at its own designated due date. Other than submitting the required results, given at the end of each part, you may need to keep a copy for yourself since you need them as the input of the future steps. Please keep in mind that without completing the previous parts, you can not work on the rest of the project since they are highly related. If you miss a due date, you will miss the point of that sub-project, but you still need to finish it to be able to finish future parts.

Project Overview

The long term goal of this project is involving you with real world implementation of the digital certificate, RSA and AES cryptography. You need to start with requesting a digital certificate from a certificate authority, which provides you the public/private key set bounded to your email address. Then, you can use this key pair for asymmetric key cryptography algorithm, RSA, for encryption/decryption and digital signature. Eventually, you need to use the session key, securely transmitted by your asymmetric key set, for symmetric key cryptography such as AES or Triple-DES. Throughout this project you need to read more about the C++ OpenSSL cryptographic library and use the built in functions and structures.

Second Deliverable

In the second part of this project, you need to use your public and private keys, extracted from your digital certificate in the first part to perform RSA cryptography. Hence, we provide a third party public key, named as pubkey.pem, and a message encrypted with the third party private key. You need to write a program that can perform the following steps:

1. Takes the encrypted message, the third party public key and your private key as the command line parameters in the mentioned order.

2. The program then, should use the third party public key to decrypt the encrypted message and store the message in a text file as symmetric.txt . This would be your symmetric key.

3. Then, it needs to use the decrypted message as a symmetric key to encrypt a text file with one of the symmetric key cryptography algorithms (AES or 3DES). You either can write the code for this part or use the openssl command with system function. In the text file, you should mention your name and your banner ID. You also should explain which symmetric key algorithm you used in addition to the project evaluation.

4. After that, you have to sign the file content, already encrypted in step 3, with your private key, and store the result in a file.

You need to name your program as the project2_encryption.cpp.
Then, you should write the second program which helps me on decryption. The second program should work as follow:

1. Takes the file that is generated in the last step of previous program, the appropriate public key, for signature verification, and the symmetric key file, symmetric.txt, in the mentioned order.

2. Verify the signed file with the public key.

3. After this verification, it needs to do the final decryption step by using the content of symmetric.txt file and return the text file, containing your information and evaluation, in plaintext. Again, you can write the code for this part or you can use the openssl command with system function.

Name this file as project2_decryption.cpp.
For submission, you need to submit all the following items in one tar folder (you can use $tar - cvf foldername.tar foldername$)

- Both programs
- The provided third party public key, publickey.pem
- The symmetric key file which you named it as symmetric.txt
- Your pub.pem and priv.pem keys that you get in first part of the project and used it here.
- The encrypted file which you encrypt it in step 4 of the first program (This would be one of the input file for second program).
- You also have to provide a readme file which explain the compilation and running command with necessary information. If you do not submit readme file or we can not compile and run your programs with provided command, you will get zero for this project.

Please follow the instruction for each step, follow the naming scheme that is given.
Remember that your programs should take three command line parameters explained in step 1 of each program.
Provide meaningful prompt for the user.
You will lose some point for missing any of the aforementioned items.