CSCE 44303/54203 Homework 3 (Programming)

Release date: September 6, 2024

Due date: September 24, 2024

Full Grade: 120 pts

I. Task Description

In this assignment, you will implement encrypted communications between two parties, Alice and Bob, and evaluate the performance of AES and RSA under different parameters. For simplicity, Alice and Bob will be simulated by two programs running on the same computer. When Alice sends a message to Bob, she writes the message to a file; Bob receives the message through reading from the file. (If you know socket/network programming, you can also directly implement socket/network communications between the two.)

Part 1: Implement encryption and decryption using AES with 128-bit key. Assume that Alice and Bob already have a shared secret key *k* (e.g., they can read the key from the same file). Alice encrypts an 18-byte message *m* (the message is manually input from command line), and writes the ciphetext into a file named *ctext*. Bob reads the ciphertext from the file, decrypts it, and prints the message *m*. The encryption should use the CBC mode.

Part 2: Implement encryption and decryption using RSA with 2048-bit key. Assume that Alice already has got Bob's public key (you need to figure out a way to do this). Alice encrypts an 18-byte message *m* (the message is manually input from command line) using Bob's public key, and writes the ciphetext into a file named *ctext*. Bob reads the ciphertext from the file, decrypts it, and prints the message *m*.

Part 3: Measure the performance of AES and RSA under different parameters. This is to explore how the key size affects the computation cost of AES and RSA. Take a 7-byte message manually input from command line. Implement AES with 128-bit, 192-bit, and 256-bit keys. For each key size, run the encryption over the 7-byte message and decryption of its ciphertext for one hundred times, measure the average time needed for one encryption, and measure the average time needed for one decryption. Implement RSA with 1024-bit, 2048-bit, and 4096-bit keys. For each key size, run the encryption over the 7-byte message and decryption of its ciphertext for one hundred times, measure the average time needed for one encryption, and measure the average time needed for one decryption. Print the average time of encryption and the average time of decryption for each key size for AES and RSA.

II. Tests

You need to demo your program to the grader. A demo sign-up sheet will be distributed later. During the demo, your program will be tested in the following ways.

Test of Part 1: For the encryption function, your program needs to take a manually input plaintext message from the command line, and print the derived ciphertext.
For the decryption function, your program needs to print the received ciphertext and the deciphered plaintext.

Test of Part 2: Same as Part 1.

Test of Part 3: Your program needs to take a manually input plaintext message from the command line, print the average time of encryption and the average time of decryption for each key size for AES, and print the average time of encryption and the average time of decryption for each key size for RSA.

III. Other Instructions

Programming language requirement: C/C++, Java, or Python. Any other programming language needs instructor approval.

Messages must be manually input from command line (not from a file). Results must be printed to the command line (not to a file).

Submit your code and necessary support files (including an empty *ctext* file and, if any, the file(s) used to exchange shared secret key and public key) to Blackboard as a .zip file named in this format: HW3.YourLastName.YourFirstName.zip.