RSA.java

David Robertson

CS 1632 - FINAL DELIVERABLE

RSA File Encryption With Tests

Summary

I decided to create an RSA file encryption program. It can create random RSA public and private keys. You specify a file to encrypt or decrypt, and a new file will be created. I created tests as I was working on this program, because the methods I was building often required the use of features from other methods. For example, the encrypt and decrypt methods required the success of the keyGen function, and the decrypt function required the success of the encrypt function. It wasn’t exactly test driven design, because I was thinking of additional cases as I was writing the RSA. I was however, writing tests to ensure that the needed elements from previous methods would be available for newer methods.

After I finished the RSA program, there were a few other things that I felt still needed to be examined. I had written an automated test to ensure that the encrypted file differed from the original file, but I felt that a manual test was needed. I wrote a manual test to ensure that the encrypted file was unrecognizable in comparison to the original file. This test would be incredibly difficult for an automated test to have determined.

Issues

A problem that I noted when working on this project is that there is an exponential slow down in performance as the size of the keys is increased. I believe this is unavoidable, but needs to be noted as an issue. If a user picks a large enough number, the program will not run very quickly. It does however, run well enough for an input to create sufficiently large keys.

I would like to run more tests on the math behind the RSA key generation. I do not have a good enough math background to write tests that ensure that the file will be sufficiently encrypted, or that these would even make practical automated tests. When writing the encryption, I knew how to make it work correctly, but not the specifics on how these numbers can be sufficiently tested, or if further automated testing could be done in a reasonable time.

Assessment

No tests failed, but as mentioned before, caution needs to be taken with the choice of key size. I believe that the program works well, and will be reliable. I believe that this program is ready for release because the tests cover all of the features and came up green. The user simply needs to be warned that larger key sizes will cause increased time to run. The user also needs to be aware that though the encrypted and decrypted files are appended, they can still be opened in a text editor.

Code can be found at:

<https://github.com/RobertsonDavid/RSAFileEncryption-WithTests>

Screenshot can be found at:

<https://github.com/RobertsonDavid/RSAFileEncryption-WithTests/tree/master/Automated%20Test%20Results>

Manual Test

Test Cases

**IDENTIFIER: 1**

TEST CASE: Ensure that the encrypted file is unrecognizable from the original file

PRECONDITIONS: Run the Tests from RSATests.java. Assuming that they run successfully, they will set up the correct environment for this test.

INPUT VALUES:

EXECUTION STEPS:

1. Open the file test.txt
2. Open the file test.txt.enc. This can be done using a text editor, which you may need to specify manually
3. Compare the files and ensure that they are not similar.

OUTPUT VALUES:

POSTCONDITIONS: