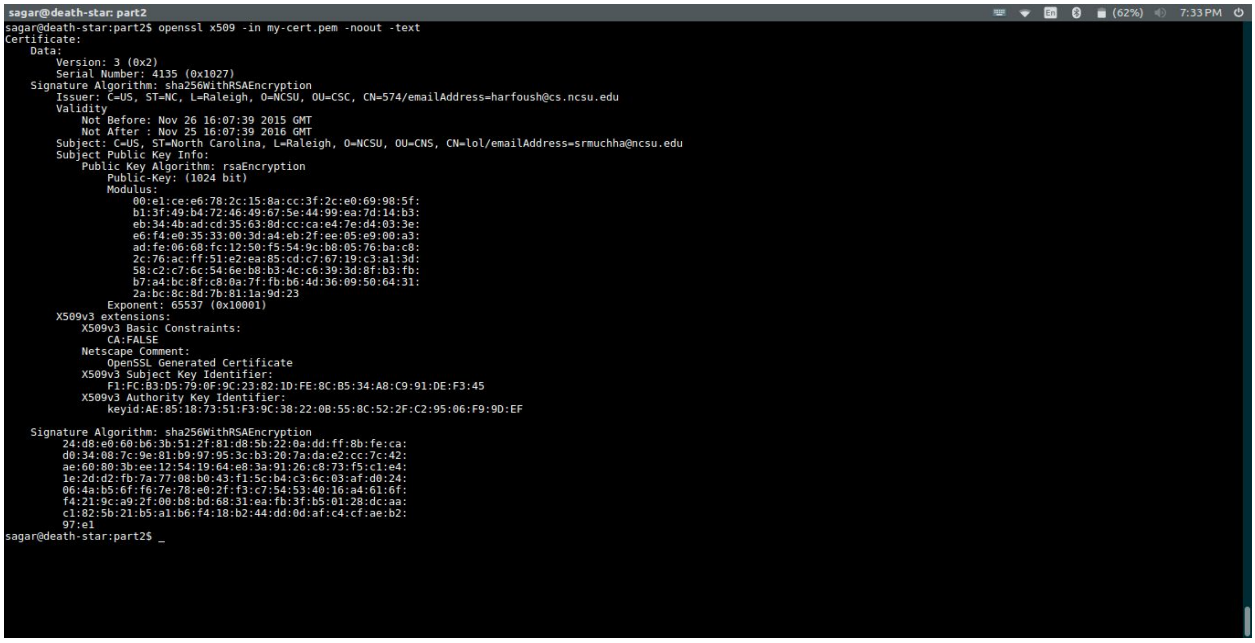
Project 2 - Email Security

Sagar Muchhal (srmuchha)
Rajashree Mandaogane (rsmandao)

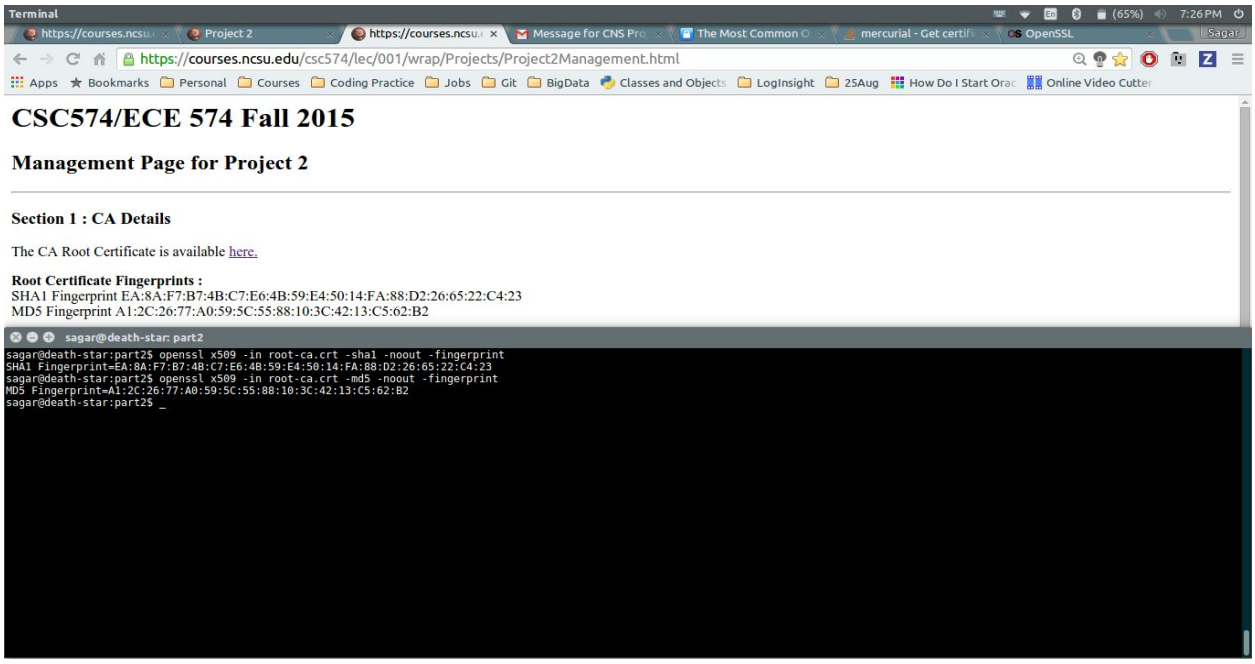
The motivation of the project was to understand the working of OpenSSL and implement an email security system which encrypts and decrypts the mails using a mock Certification Authority provided in class.
The project was divided into three parts:

1. **OpenSSL library:**
   The goal of this part was to get familiar with the OpenSSL library. The OpenSSL library used was **1.0.1f 6 Jan 2014**. It also required us to write a script which uses brute-force approach to break the key used for encrypting a given output file.
   In this case, the key used for decryption was **box** and the decrypted message was:
   A four-pound rock that left Mars 16 million years ago may hold the clues to ancient life on the planet. The rock is one of 13 Mars meteorites found on Earth. These rocks may contain clues to the ancient history of planet Mars, believed to have been a warmer, wetter place over 3 billion years ago. NASA's planetary scientists tell us why they think the Mars rock contains evidence of ancient life.
   Below are the implementation details of the script:
   - The script generates a list of probable keys beginning from ‘aaa’ to ‘zzz’.
   - The following command is used to decrypt the file:
     openssl enc -des-cbc -d -base64 -in outfile.txt -out <input_key.txt> -k <key>
     where, key: key used for this iteration
     input_key.txt: output file generated if successful
   - Separate sub-processes containing the above command and one value of the key are queued in a thread pool.
   - The thread pool executes the processes and upon successful execution and a non-zero return code an output file containing the decrypted message gets created.

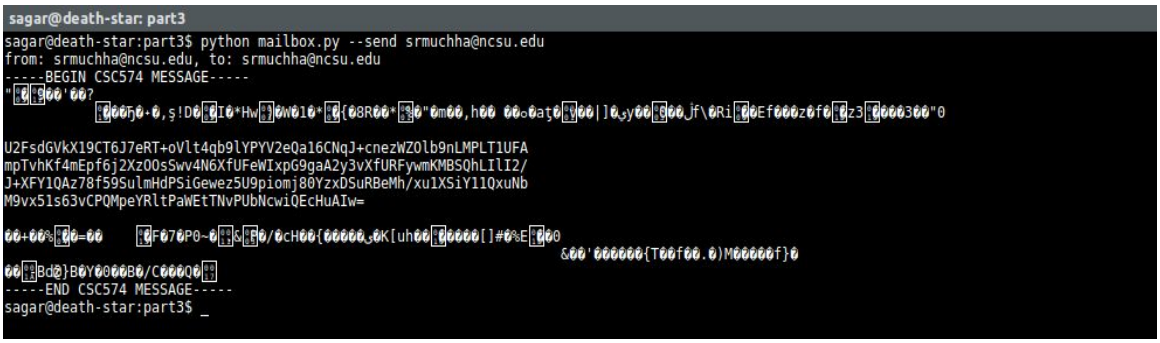
2. **Certification Authority:**
   The goal of this part was to understand the working of a Certification Authority. For this part, a 1024 bit RSA key pair and a corresponding Certificate was generated to be signed by the CA. The CSR was submitted to the CA whose output was a .pem file which was uploaded to a central repository which was accessible to everyone.
   Also, to demonstrate our understanding, we were supposed to generate the textual format of the certificate which is a below:



   To demonstrate that the correct version of the Root CA is being used, the fingerprints of the root CA were supposed to be compared with the fingerprint values posted on the course website.



3. **Sending and receiving emails:**
   This part required us to develop an email encryption and decryption system using the certificate repository provided in class. It also required us to maintain a cache of certificates available locally. If a certain certificate is not present in the database, then a copy of the certificate should be pulled from the central repository and cached locally.
   - Database of certificates:
     To display the database of certificates cached locally, use the following command:
     python mailbox.py --list
     The file contains the output of the above command:
     Link to database file
   - Send email
     To send an email, use the following command:
     python mailbox.py --send <unity_id@ncsu.edu>



   - Receive email
     To decrypt an email, use the following command:
     python mailbox.py --receive <email_message_file>