Team 3 Progress Report

17 April, 2019

Progress Report

April 3rd 2019: Proposal Recap

After meeting with Professor Mohan and discussing our original ideas, we expanded on our project to take on a more advanced and realistic approach. To recap, our idea started with creating an algorithm to encrypt and decrypt messages. Once we had presented this idea to Professor Mohan, he then proposed a new idea that would allow access to encryption/decryption of emails through the Allegheny network. This new idea was intriguing to us as a group and we decided to indulge into it. The group was excited to hear that Professor Mohan wanted to join our team and help us through some of the connection part of the code to allow java to cross with the Allegheny email access. Since then, we have worked several aspects of the project to allow us with more access and more expansion opportunities with this project.

April 4th 2019: Prof. Mohan Meeting

This meeting was the first of meetings that we have had with Professor Mohan. Present at the meeting were Nathan, Professor Mohan, and Ryan. In this meeting Prof Mohan presented ground breaking code implementation, mailTester, that would spark the beginning of our project. The code ran through emails through the users Allegheny email (Professor Mohan), and stored all of the information about that email into different arrays. For example, if the user requested to read through ten emails, the email subject, body, sent from, time/date, and attachments were all broken down and separated into different arrays. To run the code there are two “jar” files and they were built by an external source much like the scanner method in java. Jar files are from external code that are not related to java, but help us bring in the outside information, emails, that we would like to use. By having these we are granted access to the Allegheny user email account. The code itself uses many concepts that we have covered in class such as breaking down files, and storing strings into an array.

To run this code the user has to have an app password. An app password is a created to allow the user to grant access to their account to an outside project. In our case, the app password is generated for gmail, on either mac or microsoft (this is dependent on user). Once this is completed, the password is hard coded into the java code, which then allows java access to the email. After this is applied to the code, it can simply be run with the commands “./compile.sh” and “./run.sh mailTester”. The output that is produced starts as the code pulling in all of the emails, and breaking them down into there more basics elements. After this is finished the emails are displayed in the terminal.

April 5th 2019: Secure Email Connect App

Professor Mohan introduces a new idea through the team 3 slack channel. The new idea that professor mohan introduces is an app called the “Secure Email Connect App”. In his words, the app will transform the original text message file into an encrypted message using a repository of encrypt / decrypt algorithms. How it would work is, an email would be sent to an individual like a normal email. After the email is sent it would be sent through the app and then encrypted so no one can tell the message. The new message is then sent to the individual still encrypted.

The individual will still use gmail to find said message, but it will be encrypted until the user unlocks the message by logging into the “Secure Email Connect App”. Once the user is logged in the email would be decrypted, allowing for the user to read and comprehend the original message. Professor mohan found this to be interesting as we are using gmail to communicate daily on the campus of allegheny, as well as the encryption and decryption of outside email messages. With several of the group members favoring this idea, it is now a new goal to reach by the end of the semester. Professor Mohan also challenged the team to think of other practical, real world ideas that we could use with the software we are developing.

April 9th 2019: Nate’s Question Inquiry

The basis of this meeting was about Nate’s new source code with the user interactive messaging. Nate stated, he created a new repository, separate from the original team 3 repository with all of the mailTester java/jar files. He created this to separate the sample code that Professor Mohan provided and the actual code used for the final application of the project. There are two methods inside this code implementation. The main method, Email in.java, takes very similar form to the original code that Professor Mohan provided with pulling in the gmail messages. To actual pull in the messages, the file calls the other file, Email Data.java, to read in the emails. When the code is breaking down the email into its smaller parts, an array is created internally by using a getter and setter approach. There are several arrays that store different “data” from the email, for example, name, time, date, message, attachments, etc. There are few with no array creation, but instead, they call other set methods to store the information.

Although all of this might seem like a positive, the main reason for this meeting initially was because Nate was struggling to output the different types of email content. A multipart email is difficult to print out, like an email with an attachment. Even comparing his code to the original code, Nate was struggling to find the output he desired. He feels his problem is that the correct part of the code is not implemented correctly into his code, allowing these errors to occur.

To answer this question Professor Mohan simply multipart emails will come inside an else if statement and print out as a multipart, if there are any attachments this will be done internally, based on the if else, it will go into each message and capture the content in each message. For the purpose of this project, we are going so simply focus on text emails, without any attachments or complexity. Text messages will only be used from this point further. These feature could be extended later on if we would want for another class or other kind of group project.

Professor Mohan continued by saying from this point moving forward, we need to identify some use cases for Secure mailConnect, trying to use this aspect of encryption to hide the gmail messages that we want to send and receive. Also, how will this connect to the Allegheny community? Professor Mohan encourages us to think about some uses to our Allegheny community, and really give a reason why the audience should care about this idea, and why it should be important to them.

For a final conclusion of the meeting, the group discussed on what was the next step for the project and which direction to move in. The group decided that the next direction to move in would be working on how to send an email from the terminal using the code that has already been created (expanding Email Main/Data). Using an IMAP protocol would be a good source to find this information. It was finalized by Thursday April 11th, we should have found and present a solution for this problem. Professor Mohan explained, if by then the group had not found a solution, Professor Mohan would step in with his solution. One final point would be to start implementing different algorithms to encrypt/decrypt strings that we can later implement into the code.

April 10th 2019: Lab Session Meeting

During this session, Nate was able to propose a solution to this problem. His code simply took the message in as a text converted to a string, then put into an array where it would be stored until called later. Then by logging into his own account with user oriented programs for the user to log into the server on terminal. The information is compared through the “jar” files, and then if the password matches the username, the user is granted access to see or send emails. If a user picks to see emails, then the program that was already implemented will run. If a user selects send, they will have to enter a recipient, subject, and body to their email. This information was then stored into an array and sent through the “jar” files to match google format and send as an email. Professor Mohan was impressed with the work that Nathan has done with this part of the code.

During these team session meeting both Ryan and Adam presented algorithms to encrypt and decrypt emails. Ryan implemented a Caesar Cipher method to complete the task. Both of these were excellent steps to bring, as now we were able to begin working on applying both of the algorithms to our Email in.java. Once implemented the user simply was asked what method they wanted to use for encryption, depending on the users choice it sent the value to an if statement where it was compared to the value and ran the correct call method to use the corresponding algorithm of encryption. By the end of the class period, as a group, we were able to implement both Adam and Ryan's algorithms to the code, allowing us to now send encrypted emails through our java code. Wow...

April 11th 2019: Cory's Research

While bouncing around ideas for the project and contributing to algorithms, Cory was able to find a third way to implement an encryption method. The method was labeled as "AES", or Secret Key. Simply the code creates a secret key and then, sends the key to an array storing it and allowing for it to be manipulated by changing the secret key. Cory also looked into and continues to find other types and kinds of encryption methods that can be used to compare to other algorithms that we are using in the main method. Along with furthering his research and Cory is researching other algorithms to see if we would need a fourth option