DAT 510: Assignment 3

Submission Deadline: 23:59, Friday, Nov. 11, 2022

P2P Secure File Sharing Implementation

In this assignment, you will implement a P2P file sharing system. You must implement a secure file sharing system using P2P network. Encryption and decryption techniques must be used to secure the communication between the users.

Task 1 Create P2P network, where each node can communicate directly to other nodes without any medium, no central server required for communication and file storage. The figure 1 show a P2P network for four nodes.

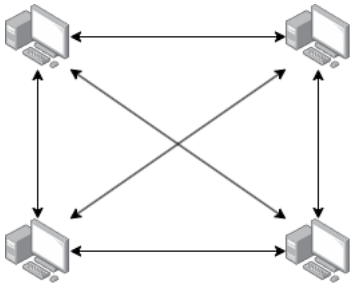

Figure 1: P2P Network

Task 2 Create Distributed Hash Table (DHT) to store address of each node to encourage direct communication among nodes. DHT should allow each node to join and leave the network anytime, update its routing table. Each client will have a public and private key, address itself, can work as a public key. Maintain separate hash tables for each node.

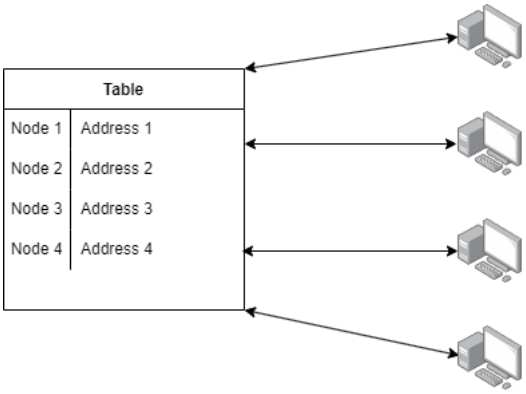

Figure 2: Distributed Hash Table (DHT)

Task 3 *Add a new node*

Make sure once the new node gets added, DHT is synced to this node as well.

Task 4 *Down a node*

Just simulate that, a user system is down. Make sure DHT is synced to this node as well.

Task 5 *Add the names of all the files and hash values in to DHT.*

Make sure, this database (DHT) is synced with all the users in the network.

Even for the user who are added at a later point.

Task 6 *Request access to a specific file from a user.*

You can make use of your implementation from Assignment 2 for this part. (Make sure this communication is secure).

Your program **should** have user interface which:

- Show your P2P network at any point (active nodes and down nodes).
- Show current DHT.
- Show you are able to add new nodes to P2P network.
- Show the P2P network , when a node is down.
- Show all the files whether they are accessible or not.
- Show files can be shared based on the request.

Assignment Approval (by TA and SA)

Assignment approval will have a weight on your grade for the assignment. If you are not going to get the approval before the deadline, your assignment will not be evaluated and you will fail the assignment.

What needs to be done to get the approval of the assignment:

1. Show all parts of assignment are working i.e, show the code with proper comments, results.
2. The code should have a proper README file that describes the contents of the directory and any special instructions needed to run your programs (i.e. if it requires and packages, commands to install the package. describe any command line arguments with the required parameters).
3. Source code submitted for the assignment should be your own code. If you have used sources from the internet everything should be added to the references. If you used someone's code without reference, that will also be treated as plagiarism.
4. Provide the references in the code and Report, show these parts for TA's and Student Assistants.
5. You should **NOT** use available libraries/packages/classes for implementing the core functionality of the assignment.

You may use any "reasonable" programming language for part one of the assignment. Reasonable languages include: Java, C, C++, Python, MatLab, R and others with permission of Jayachander Surbiryala (Email: `jayachander.surbiryala@uis.no`) or Chunming Rong (Email: `chunming.rong@uis.no`).

Assignment Submission

Deadline: 23:59, Friday, Nov. 11, 2022 (submit your assignment through canvas)

Final submission:

1. Source Code
   - Source code submitted for the assignment should be your own code. If you have used sources from the internet everything should be added to the references. If you used someone's code without reference, that will also be treated as plagarism.
   - Source code should be single, compressed directory in .tar.gz or .zip format.
   - Directory should contain a file called README that describes the contents of the directory and any special instructions needed to run your programs (i.e. if it requires and packages, commands to install the package. describe any command-line arguments with the required parameters).
   - You should **NOT** use available libraries/packages/classes for implementing the core functionality of the assignment.
2. A **separate** report with PDF format
   - Texts in the report should be readable by human, and recognizable by machine;
   - Other formats will **NOT** be opened, read, and will be considered missing;
   - Report should follow the formal report style guide in next page.
   - Each student should write an individual report. Each report will be checked for plagiarism. If it is copied from some where else, you will fail the assignment.

NOTE: Please upload the archive file in *.zip, *.tar only and report in *.pdf format only to the website https://stavanger.instructure.com/.

Note: The assignment is individual and can **NOT** be solved in groups.

Project Title

Abstract

A one-paragraph summary of the entire assignment - your choices of cryptographic primitives a and their parameters, procedure, test results, and analysis.

1. Introduction

A description of the scientific background for your project, including previous work that your project builds on. (Remember to cite your sources!) The final sentence (analogous to the thesis statement in a term paper) is the objective of your experiment.

2. Design and Implementation

A detailed description (in paragraph format) of the design, procedure, and implementation of your project. This should be the main part of the report.

3. Test Results

Results of testing the software, as you observed/recorded them. Note that this section is only for observations you make during testing. Your analysis belongs in the Discussion section.

4. Discussion

Your analysis of what your testing results mean, and your analysis.

5. Conclusion

A short paragraph that restates the objective from your introduction and relates it to your results and discussion, and describes any future improvements that you would recommend.

References

A bibliography of all of the sources you got information from in your report.