**Initial Project Iteration**

Austin – Networking Leader

Wei Jin – Cryptography Leader

Phillip – Traffic Control and Team Leader

Scope

This project aims to be a pure peer-to-peer network file sharing and privacy application that does not store any sort of user data. This network will be able to securely and efficiently search, download, and upload files without providing sufficient information to deduce the file sources or destination to any peer on the network. To provide encryption and security to users as they download and upload files without storing the entire file in a single place while providing the network speeds associated with downloading the file from a single source. Users will be able to remain anonymous without having any determinable link between the download, searches, and uploads of users.

Vision

Create a peer-to-peer file sharing system that is efficient and secure. Ensure that when using this network to upload or download files, that the locations of those files maintain anonymity and are distributed to the point of obfuscation. This project aims to utilize two proxy peers in the transfer of data to ensure that the download peer is kept completely in the dark when it comes to where the data is coming from. Maintain a network that provides backups to data as well as the locations of data to ensure that if any peer were to disconnect, it would not detrimentally effect the integrity of the network. Ensure that any and all data that could lead to the locations of the data are as obfuscated and distributed across the network as is humanly and efficiently possible. Ensure that data can remain accessible in an effective way and that the network maintains high data transfer speeds. Make sure that even those who hold pieces of the data, or directions to get there, do not know what data they are storing and that they are so encrypted and obfuscated that even trying to get a single piece without authorization would be near impossible.

Initial Glossary

* Peer-to-Peer Networking – A type of network that distributes the storage of files
* Traffic Control Administrator – Phillip, in charge of programming and maintaining how traffic will be handled throughout the network.
* Cryptography Administrator – Wei Jin, in charge of programming and maintaining the cryptographic elements of the system.
* Networking Administrator - Austin, in charge of programming and maintaining how multithreading, uploading, and downloading files across peers within the network.
* General User – Anyone who will utilize our software for the use of data storage or download
* Cryptography – The act of encrypting and obfuscating data to the point that it cannot be read in plaintext
* Multithreading – Utilizing multiple threads on the CPU to run aspects of a program in parallel
* UPnP – Universal Plug and Play allows for connections between the application and the world outside of a LAN
* Data Obfuscation – Changing parts of the data and the way that it is stored in such a way that it is secure
* Proxy Peer – A peer type that allows a downloader to connect and stream data through it as to not show the downloader where the data is coming from
* Middle Peer – A peer type between the proxy peer and the storage peer that obfuscates the location of the data from the proxy peer – thinks that it itself is a proxy peer
* Storage Peer – A peer type that is the actual end location for a chunk of data that is encrypted, not given a whole file
* Upload peer – A peer that is uploading a file
* Download Peer – A peer that is downloading a file
* Confirmation Peer – A peer that confirms that a downloader is who they say they are and directs them to the location to get to the encryption keys on the Key Peer
* Directory Peer – A peer type that stores the location of the Confirmation Peer as well as the location for the first proxy peer in a chain in order to retrieve a file
* File Name Peer – A peer type that stores the location of a Directory Peer and also houses the file name for a file that is searchable
* Key Peer – A peer type that holds the encryption keys for a file
* Load Balancing – The act of balancing network traffic in such a way that no single node is receiving too much traffic
* Block Encryption – A type of file encryption that requires a key to encrypt and decrypt and only those with that key can do so. Encrypts the file in blocks of data rather than individual bits.

Initial Risk Assessment

1. Miscommunication and poor time management can result in badly configured code that will not work with other modules written by the other teammates. Poor time management can result in requirements not being interpreted or implemented correctly and can also result in code that plainly does not get the job done correctly.
   1. Solution: Weekly meetings with the team to go over problems that have been encountered since the previous meeting can help to mitigate miscommunication with having to change the requirements of a class or a piece of code that may change the overall layout of the inputs and outputs. These meetings are also a good time to ensure that each team member is not falling behind on the schedule and ensure that time management is being practiced.
2. Poorly documented code is a great way to hinder the advancement of a project. When a developer other than the original person that wrote the code tries to refactor a class or function, the non-existence or poor documentation of code can lead to confusion in what the function or class really do which can result in broken, non-function or inefficient code.
   1. Solution: Ensure that each class has a description of its own inputs and outputs as well as ensuring that each method provides an overview of what is done within it. UML diagrams can provide a good graphical way to show team members how a class and method function and the use cases of it.
3. Poorly managed network infrastructure and no thought given to the processes entailed within the scope of connecting to other users can result in slow moving downloads, files sent to the wrong user, as well as the complete inability to send proper files across the network
   1. Solution: Create a robust system that can handle traffic properly and log the direction that data needs to be sent. Ensuring that the locations of pieces of data is known in some sense is crucial and good programming and testing techniques are required to ensure that the software is working properly
4. Version control can be a big issue in terms of team members fixing and uploading code while other members are still working on outdated versions. This can result in miscommunication and non-working code.
   1. Solution: Use software such as GitHub to manage the version control and simply check the code base before making updated to ensure that you are working on the most current version. It is also important to update team members on whether a requirement has been met and whether the code base has changed.
5. Project complexity can provide an immense challenge if the project is too difficult to complete in the allotted time. This can result in hastily put together software with code bases that have not been tested or do not meet the requirements.
   1. Solution: If the project is complex, ensure that proper time management is being upheld and determine whether requirements need to be changed in order to get a functional system within the allotted amount of time or request more time if that is an option. If the complexity is simply too much then there needs to be some change to the methods used to complete the project, or a change needs to be made to the project itself.
6. Poor information transfer between teammates can lead to miscommunications and no information on current versions or where the whole team is at versus where they should be. This leads to a lot of problems in the development cycle in regards to ensuring that code is compatible and can work.
   1. Solution: Ensure that team members are providing updates that are accurate and truly do show where the project is at. Sugar coating will always result in making false promises so even if the information is bad news, it is still important to convey it so that other members of the team know where everyone is at.

Initial Project Plan

* Networking
  + Peer infrastructure setup
  + Auto-connect/disconnect from peers
  + Multithreading
* Traffic
  + File Transfers
  + File Integrity
  + Obfuscation of network
  + Management of peers/peer roles
  + Manage peer interaction and hierarchy
  + Load balancing
* Cryptography
  + Attach hashes to files
  + Encrypt/decrypt with block cypher
  + Manage private/public key ecosystem
  + Ensure proper encryption
* Assembly of components

Requirements Analysis

* Maintain complete anonymity in process of sharing/downloading files
* Create peers that are able to communicate with one another without network load balancing issues and without inconsistent communication
* Ensure that the multithreaded parts of the system are correctly generating and terminating threads and that each thread can maintain a connection
* Implement an obfuscation algorithm for correctly cutting, hashing, and distributing file chunks
* Implement an encryption algorithm that doubly encrypts data, the first time from the plain text, and the second time from chunks of the encrypted text
* Implement AES as the block encryption standard
* Implement file streaming so that data can correctly be passed from the source node, through a proxy node, and to the destination node
* Create a system for correctly terminating the application that involves uploading all data contained on the peer as a shutdown process
* Implement a seeding process for uploading new files
* Implement security measures to ensure that sudden loss of a peer initiates a copy protocol from a peer containing the same data to another random peer
* Implement the process of putting together downloaded data chunks
* Implement the classes for the File Name Peer, Directory Peer, Key Peer, Confirmation Peer, Proxy Peer, Middle Peer, Uploading, Downloading, and Storage Peer
* Create a system for ensuring that peers are fluidly connected and disconnected from the network including the initial connection process for peers
* Maintain robust network
* Fluid experience with little to no network lag