Levi Amen

Quentin Covert

Mark Hernandez

Cameron Johnson

Collin Victor



CSCE 361

Dr. Cohen

**Implemented Features**



Completed Features

Successfully implemented requirements are listed below. These particular requirements were chosen in effort to deliver a minimal viable product. Many of the original features were cut from this list in an effort to finish a functional program. These functions ended up being much more complicated than originally anticipated, leading to difficulties during coding.

1. GUI
   1. The program shall present the user with a GUI once the user runs the program executable:
      * Upon startup, a GUI will be generated.
   2. The GUI shall provide means for the user to exit:
      * Clicking the exit button on the top right corner of the GUI will close the program.
   3. The GUI shall provide means for the user to specify a target remote file system for the program to sync:
      * On first time startup, the system will prompt the user to specify a target file sync location.
   4. The GUI shall display to the user relevant information of a selected file contained within a watched root directory:
      * By clicking on a file in the GUI during normal operation, the user can see information about the file displayed in the GUI.
   5. The GUI shall provide means for auto-encryption to be enabled for files or directories under watched root directories:
      * By default, any files in the watched root directory will be automatically synced to the remote location, including encryption.
   6. The program shall provide a record of each file watched by the program:
      * The GUI displays this information.
2. Watchman
   1. The program shall update local files if the corresponding files on the remote filesystem has a more recent modification time:
      * This is performed every two seconds using the Watchman interface.
3. Crypto
   1. The crypto module shall automatically produce a 32-bit symmetric, fernet format, key for the user if none is provided:
      * The program automatically generates a fernet key upon startup.
   2. The crypto module shall provide a method to encrypt target files:
      * Files will be encrypted when synced to the remote system. Should the file be too large to sync efficiently, it will be split into multiple encryption pieces and put into a single file.
   3. The crypto module shall provide a method to decrypt target files:
      * The inverse of the encryption algorithm was created and tested alongside the encryption and is functional.
4. General
   1. The program shall allow the user to specify a remote file space for files to be synced to:
      * Upon startup, the user is prompted to specify a remote file space for files to be synced.
   2. The program shall automatically sync changed files under root directories to the remote filesystem:
      * Watchman detects changes in files and automatically runs our syncing algorithms upon detection.
5. Not yet completed requirements
   1. General Requirements
      1. The program shall provide the user the ability to choose how files are synced to the remote filesystem:
         1. The program shall encrypt files and sync to the remote, or
         2. The program shall not encrypt files but still sync to the remote.
      2. The program shall update local files if the corresponding files on the remote filesystem has a more recent modification time.

## GUI Requirements

* + 1. The GUI shall provide means for auto-encryption to be disabled for files or directories under watched root directories.

Next Phase Plans

The following requirements are those that could not be completed within the deadline for the current phase.

1. Proposed Requirements
   1. General Requirements
      1. The program shall watch more than one root directory trees.
      2. The program shall allow the user to specify files and directories within root directories to be ignored by the system.
      3. The program will automatically delete files on startup that have been deleted on another system

## Crypto Requirements

* + 1. The crypto module shall provide a method to encrypt target files.
       1. This method can be manually executed by the user.
    2. The crypto module shall provide a method to decrypt target files.
       1. This method can be manually executed by the user.
    3. The crypto module shall automatically encrypt synced files below a specified size.
       1. The default size for the crypto module shall be 2GB.
       2. The crypto module shall provide a method for the user to specify a custom maximum file size.

## GUI Requirements

* + 1. The GUI shall provide means for the maximum auto-encryption files size to be adjusted by the user.
    2. The GUI shall provide means for local files to be manually encrypted. These files do not need to be under root directories watched by the program.
    3. The GUI shall provide means for local files to be manually unencrypted. These files do not need to be under root directories watched by the program.
    4. The GUI shall provide means for files or directories, under watched root directories, to be ignored and not automatically synced.