CSL707 - Assignment 4

By Gaurav Mittal, Kaushal Yagnik, Deepak Chawla

README

Problem Statement

To develop an application which may store and retrieve files from a local directory to AWS S3 cloud storage with properly protected so that the cloud vendor may not view the contents or other information about the files being stored.

Description of the Solution

The solution we have proposed is a Java based application having a Java Swing GUI interface to interact with the user which provide a convenient and efficient way to connect to the Amazon S3 Cloud and provide functionality for storing and retrieve data between the local machine and the cloud.

More importantly, apart from this functionality, the application provides an effective means to encrypt the data before sending it to the cloud and decrypt the data on receiving it from the cloud so as to ensure that the contents or other information are protected from any intrusion by the cloud vendor or otherwise.

This encryption/decryption mechanism is provided the random generating an AES-256 session key to perform symmetric encryption over the data in the file and then encrypting this session key using a public-private key based RSA encryption, after which the session key is stored along with the data on the cloud for later downloading. This mechanism is chosen over directly using RSA encryption since RSA is very computationally intensive causing it to require a really large amount of time to even encrypt and decrypt files over mere few megabytes.

The application, in addition to this protective mechanism, provides the following features:

- Ability to work on different bucket by giving functionality to choose.
- Ability to select an encryption key pair to be used
- Ability to select a file or even a directory to be stored on cloud
- List the files stored in the cloud
- Download files from the cloud

- Perform upload and download operations asynchronously (in a non blocking way) enabling multiple upload and download operations to be performed simultaneously and also resulting in a highly responsive GUI.
- Show progress to the user in terms of a user friendly progress bar and also shows log messages to track what is happening.

Contents

- **src** - folder containing the java source project (can be opened using Eclipse)
- **AmazonS4.jar -** runnable jar file with all dependencies being taken care of. Can be easily run using the command: $java -jar AmazonS4.jar
- **README -** readme file describing the solution, instruction to compile and run and other details
- **DesignDocument -** document explaining the design of the solution
- **extras -** folder containing sample public private keys

How to Compile and Run

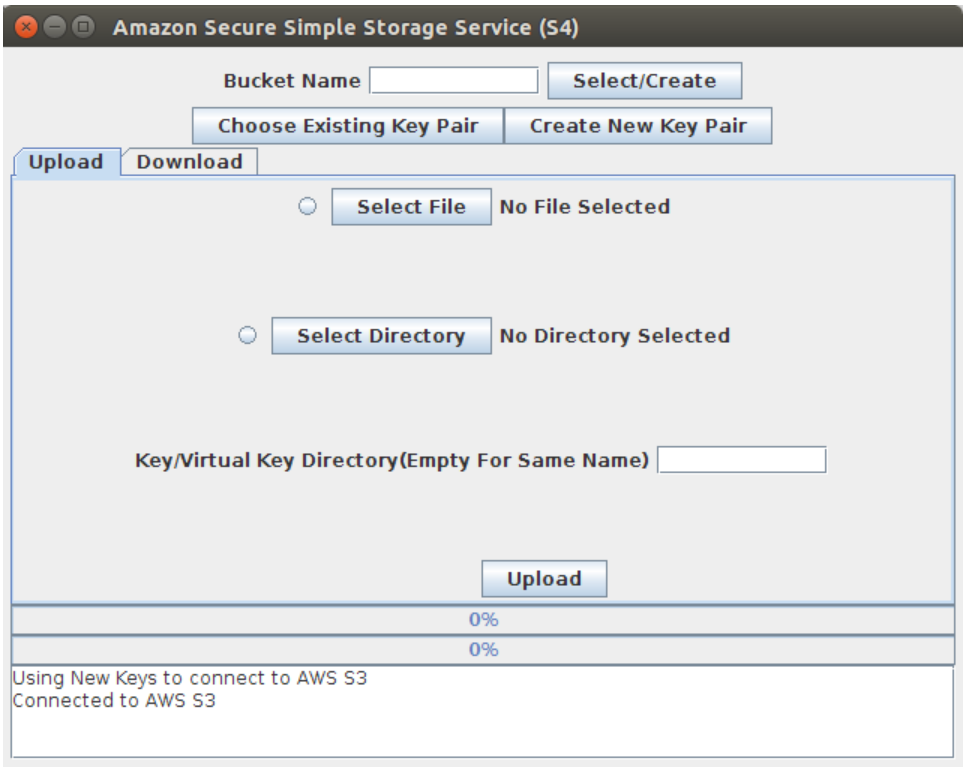
- To run the application, go into the project directory and run the runnable jar AmazonS4.jar using the following command:

  $ java -jar AmazonS4.jar

Sample Run (Workflow)

- On running the above command, a Java GUI application should appear on the screen giving two options to the user:
    - Access Using Default Keys : On clicking this option, the application will try to access using the access keys present in the config.properties file of the application. If valid keys are found, the user will proceed further or will be asked to access by providing new access keys.
    - Access Using New Access Keys: On clicking this option, the access keys filled by the user in the specified text boxes will be used to access the AWS. The user is also given an option to set these keys to default by storing them in config.properties so that next time the user can use the first option.

      Note: If the user wants to erase the default keys from the machine, the config.properties file of the application can be deleted. You can manually modify config.properties by opening the jar file using Archive Manage, then open config.properties, modify the file, save it and update the .jar package.



- On successfully accessing the s3 service, the user will be taken to a panel where following details need to be specified by the user to perform any upload or download:
    - Bucket Name : The user can specify the name of an already existing bucket on S3 which it has access to or specify a valid new name to create a bucket on S3.
    - RSA Encryption Key Pair: The user has to specify a public-private key pair to carry out RSA encryption with AES on the files. There are choice present:
        - Use existing key pair from file - The user can choose valid public and private key files to load the keys to use for encryption. (A sample pair of keys is provided in the extras folder to use)
        - Create New Pair - The user can create a new pair of public - private keys and store them in an appropriate destination.
- After meeting both the above requirements which can be altered anytime during the running of the application, the user can perform two key operations:

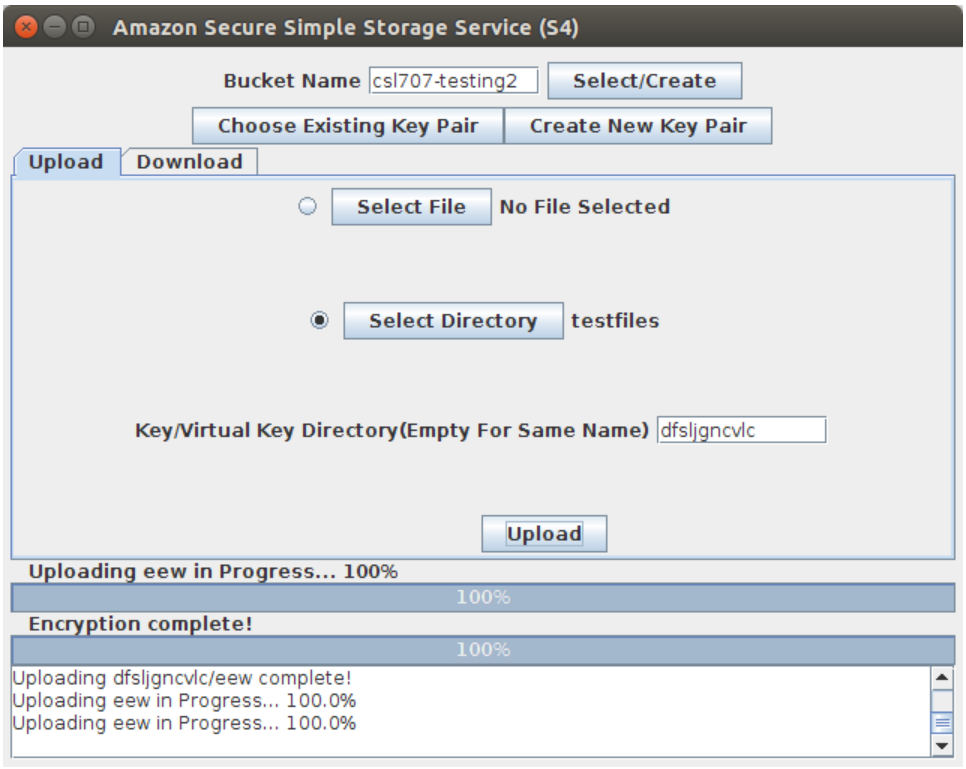
    - Upload - The user has the option to upload a single file or an entire directory by selecting the appropriate radio button followed by selecting the desired file/directory.

      The user can also optionally provide a virtual key where the resource will be stored in the bucket.(By default, the name of the file/directory is used as the

key). On clicking the upload button, first the file/directory(recursively) is encrypted followed by uploading the resource on the cloud.

- Download - The user can view the list of keys representing resources on the cloud in the download tab. Particular keys can be searched by providing an appropriate prefix and clicking on the search button.

  To download a file, select the key for that file from the list and click on the download button. The file first is downloaded and then decrypted using the provided pair of keys before getting stored on the local machine.



- The user throughout the running of the application can see the various log message appearing at the bottom keeping the user aware of the progress and what all is happening in the application. In case of an error, these log message may help the user to troubleshoot and take appropriate action.
- In addition to the log message, a couple of progress bars help in displaying the progress of upload/download and encryption/decryption in a very user friendly manner.