Design Document
Assignment 1

Chayan Dhaddha Anuraag Kansara

1 User Data Structure

The user data structure will contain the following fields:

- string Username
- [ ]byte Uuid
- [ ]byte Key
- userlib.PrivateKey PrivateKey
- map FileRecord
- [ ]byte HashKey

2 InitUser

- This function takes username and password as function arguments and returns a User Data Structure.
- A uid is created using Argon2Key which takes key as password and salt as username.
- The Uuid(32 bytes), Key(16 bytes) and HashKey(16 bytes) fields of the User Data Structure are extracted from uid.
- PrivateKey and PublicKey are generated using GenerateRSAKey() function and PublicKey is uploaded to Public key store using KeystoreSet().
- The User data Structure is encrypted using CFBEncrypter using Key.
- Then HMAC is computed using HashKey and Computed Hash value is appended to the encrypted data and the whole thing is uploaded to Data server using uuid as key.

3 GetUser

- It takes username and password as function arguments and returns user data structure created.
- Then it computes uid, uuid, key, Hashkey using the same method as in InitUser.
- It fetches the encrypted User Data Structure using uuid as key with DatastoreGet() function.
- Then the Appended MAC is extracted and Integrity check is done.
- Then the Data is decrypted using Appended IV and is returned as a stucture.

4 ShareRecord

- int Offset
- [ ][ ]byte Blockuid
- Offset will tell us about the number of DataBlocks for file
- Blockuid Array will be useful in accessing uid of random Block.

5 StoreFile

- It takes filename and data as function arguments.
- Then the function checks if the data is in multiple of BlockSize or not.
- For each file, a ShareRecord is maintained.
- Then for each Data Block, Random uid is generated using RandomBytes().
- Each block is encrypted with Key using CFBEncrypter and then HMAC is computed using Hashkey.
- ShareRecord contains uid for each block of the file and an offset is maintained. And it is encrypted in the same way as data block.

6 AppendFile

- First, check if file to be appended exists or not and then data to be stored is in multiple of BlockSize.

- Create Block then generates uid for each block. Data will be stored in block and will be encrypted and appended with the computed HMAC value.
- Sharing Record will be updated corresponding to the file.

7 LoadFile

- Check whether the file exists or not.
- The keys are extracted from the Filerecord using filename.
- The block is then downloaded using offset value, encrypted and integrity is checked.
- If the block is not corrupted, it is returned successfully.

8 ShareFile

- Check whether the file exists or not.
- Check if the file is corrupted or not.
- Transfer ShareRecord's uid to the receiver. and then encrypt it using Public Key of receiver and Signed using Private Key of sender.
- Encrypted Data + Signature will be appended to the encrypted Data.

9 ReceiveFile

- If the file already exists, and error is thrown.
- The Public Key of receiver is downloaded from KeyStore and the file is verified.
- The file is decrypted using Private key of Receiver and the filename is added to the FileRecord of Receiver.

10 RevokeFile

- A new ShareRecord will be created and data from each block will be copied to new blocks of different location
- Each newly created blocks will be encrypted, appended with computed hash value for checking integrity and then store in DataStore.
- Each newly created Block's uid will be stored in new ShareRecord.

- This new ShareRecord will be encrypted, appended with computed hash value and then store in DataStore.