CS628A Assignment 1 (Part 2)
Design Review of Ankit Bharadwaj(150101), Rohit Gupta(150593)
Reviewers: Abhishek Yadav(160040), Naman Jain(160427)

Major fault in design is that one is not allowed to store anything on client side. Storing the symmetric key of files on client side is not allowed.

Dictionary attack is possible in a way one can brute-force the KeyStore to get all the Username-PublicKey pair and then locations of User structure becomes apparently visible (K_{PUB} stored as plain text), so user specific attacks are possible. This also leads to location leak of files (as rest everything at DataStore is file) which may then alter shareholders to deny access to files without owner not knowing if any harm done to file holders.

1. **User creation and authentication (properties 1 and 2)**

   [5 points]

   - The implementation is a bit slow as they are traversing the entire structure and checking if the public key matches with that corresponding to the username.
   - No apparent vulnerability.

2. **Integrity preservation in the simple secure client (property 3)**

   [5 points]

   - No apparent vulnerability apart from mentioned at the top

3. **Confidentiality in the simple secure client (property 4)**

   [5 points]

   - No apparent vulnerability apart from mentioned at the top

4. **AppendFile implementation and efficiency (property 5)**

   [3 points]

   - There is no visible vulnerability but design is quite inefficient.
   - Loading, decrypting, appending and then again encrypting the whole data would be quite slow.

5. **Sharing implementation (property 6)**

   [2 points]

   - The shareholder cannot trust the token since it does not contain anything to verify integrity. Decrypting may lead to any gibberish value which cannot be checked to be correct by receiver.
   - HMAC of token could be sent to check integrity at receiver end.
   - Believing it to be correct may lead to receiver adding a file pointing to any random location.

6. **Revocation implementation (property 7)**

   [3 points]

   - Attacker and malicious DataStore may collaborate and may add attacker to the shareholders without allowing owner to verify integrity and confidentiality. This is because location and symmetric key is still with the attacker.

7. **Clarity of the design document**

   [4 points]

   - Some implementation issues were not clear. As storing owner, shareholders, key encrypted data and hashed data in a same file separated by "\n". As encrypted data may containing "\n", the question of how the separator would be found was not clear.
   - User's private keys also seems to be stored at client side though not mentioned.
   - Design was not clear about which encryption techniques they would be using.