CS628A Assignment 1 (Part 2)
Design Review of **Abhishek Yadav**, **Naman Jain**

Reviewers: Aniket Pandey (160113), Ashish Kumar (160160)

Property 1 and 2: User creation and authentication

5 Points No Apparent Vulnerabilty

Property 3: Integrity preservation in the simple secure client

2 Points
• There is no integrity check in case of *User* structures. A random manipulation can change any underlying field resulting in an incorrect authentication later on.
• Method of checking the integrity of *struct File* during *LoadFile()* is weak. Its possible that a tamper changes everything apart from stored "Public Key".
• **Key Value Swap Attack:** An adversary whose access to a file was earlier revoked, can still swap multiple blocks of file_data corresponding to a file, resulting in incorrect order for next *LoadFile()* invocation.

Property 4: Confidentiality in the simple secure client

4 Points
• The unencrypted *username-User* look-up table leaks the number of users registered and the location of User structs.
• The unencrypted *username filename-File* look-up table leaks the total number of files and their individual locations.

Property 5: AppendFile implementation and efficiency

5 Points: No apparent vulnerability. *AppendFile()* is adequately efficient.

Property 6: Sharing implementation

4 Points
• There is no encryption of the auth token to be shared through a medium susceptible to *Man in the Middle attack*. Hence, an adversary can learn about the location of shared file and lauch a personalized attack.

Property 7: Revocation implementation

5 Points: No apparent vulnerability.

Overall: Clarity of Design Document

4 Points
• The design incorrectly interprets the use of HMAC to generate keys and IVs, rather than verifying the integrity. Also, the arguments to functions are not specified properly, giving a vague idea about the implementation on multiple occasions.
• There is no explanation about the unused *File.hash_locations[]* field and the unnecessary map of Public keys in User struct, when only Private Key mapping was sufficient.