6.858 Lab 5

Ismael Gomez

Garron Charles

Reymundo Cano

Final Project: SecFS

Intro

For the final project we decided to go with the Serialized SUNDR project, and, in implementing this system, we had a few import design decision to make. These decisions were on how to handle user-group implementations, how VSLs are represented and signed, and how files are read-protected.

Groups

For this implementation detail we decided that it would be easier to write the block to the server as if the group's most recent user made the request (like we normally would), but with the added step of pointing the group's i-table entry for this block to the user's i-table entry for this block. This design choice allows us to resolve the group's last fetch and modification request to the user who made the request, this 'linked-list' style structure also allows us trace this updated block from the group's i-table entry to the block's i-node without too much added over-head.

VSL

For this implementation detail we decided to create a Class to represent our VSL and another Class to represent VSs. Our VSL has a key structure (basically a hashmap) that maps principals to corresponding VSs. Every time something is updated in our server we create a new VS instance and then update our version numbers, once we have populated the new VS we simply sign it and put it into our hash map. Signing is done by employing a symmetric hash scheme, we chose this detail for its simplicity.

Read-Protection

For this implementation detail we decided to create to cryptographically encrypt blocks. This basically ensures that we never have plain-text files on our server that a malicious user can just go ahead and read, this also ensures that only users with correct keys can see these files completely.

Extra-Feature

Not decided yet. Probably garbage collection or deleting and moving files.