Project 2 Design Doc - CS161

yiwen song (cs161-jp), Huadian Henry Zhang (cs161-sy)

1 Design Summary

The basic structure of our secure distributed storage system is as follows:

1. Userdata block to generate user-specific keys
2. File Metadata block to generate file-specific keys
3. Sharing block to generate keys for files shared with the user
4. Data block to actually store the files

At the beginning, we have the RSA keys for the user (K_U, K_U^{-1}). All non-MAC and non-Signature entries will be put into a JSON string. Any MACs and RSA signatures used will be inputted with this JSON string. Then the original JSON string and the MAC/Signature are put into a new JSON string and uploaded.

This design uses a PRNG. This PRNG is made with AES-CTR. Setting the IV and counter to be 0, we then input the seed as the key of this encryption algorithm. The resulting stream of output is used as keys for other things such as encryption or MAC.

1.1 Userdata

1.1.1 Naming

The Userdata section is named *username*`.userdata`.

1.1.2 Contents

The Userdata block contains the following:

1. Username
2. PRNG seed S_U
3. RSA Signature

1.1.3 Actions

The Userdata block is encrypted with asymmetric RSA using the user's public key. The signature is signed using the user's private key. After obtaining the Userdata block, the user will generate the following keys using the PRNG with seed S_U:

1. Encryption key for Metadata block K_E
2. MAC key for Metadata block K_A
3. Name-hash key for Metadata block K_N
4. Encryption key for Sharing block K_F
5. MAC key for Sharing block K_B

We should also check that the username matches and use verify on the signature to make sure that the data is not tampered with.

1.2 Metadata

1.2.1 Naming

Let $s = username.filename.\texttt{meta}$. Using K_N found in 1.1.3, we find $m = \text{SHA256-HMAC}_{K_N}(s)$. This file will be stored at m.

1.2.2 Contents

The Metadata block contains the following:

1. Filename
2. PRNG seed S_M
3. Number of Blocks
4. Sharing list
5. MAC

1.2.3 Actions

The Metadata block is encrypted with AES-CBC using K_E. The MAC of the Metadata block is calculated using K_A. (Keys are from 1.1.3) The number of blocks is calculated by taking the ceiling of the filesize divided by the block size, which is set to be 32 KB. From S_M, we should use the PRNG to generate the following keys:

1. Encryption key for Data block K_G
2. MAC key for Data block K_C

3. Name-hash key for Data block K_M

When the user gets the block from the server, the two checks that occur are that the MAC is valid and that the filename matches the filename that the user is attempting to access.

Whenever we add a user as a collaborator on a file, we generate a random seed and copy information from this Metadata block (more about this in 1.3) to the file corresponding to that seed. Then, we add the tuple $(username, randomname)$ to the sharing list.

To remove collaborators, we have to generate an entirely new Metadata block and copy all contents of a file to a new location. Then we remove the tuple from the Sharing list.

1.3 Sharing

1.3.1 Naming

The sharing block for a user is named *username*`.sharing`.

1.3.2 Contents

1. Username
2. Shared list
3. MAC

1.3.3 Actions

The Sharing block is encrypted with K_F and MAC with K_B (both from 1.1.3). The shared list contains tuples in the format $(username, filename, S_F)$. For sharing, we can generate 3 keys from the seed:

1. Key for encryption K_{SE}
2. Key for MAC K_{SA}
3. Key for naming K_{SN}

The Metadata block will be stored using the same scheme as the regular Metadata storage, and this block can be accessed as though the file belongs to the user that is shared with. When a user receives a sharing message, he should decrypt that message with his private RSA key and add the tuple to the Shared list in this block. Similarly, when a user shares any file, the share should send a message that is this tuple encrypted with the receiving user's public RSA key.

1.4 Data

1.4.1 Naming

Let $s = username\texttt{.}filename\texttt{.}n\texttt{.data}$ where n is the block number. Using K_M found in 1.2.3, we find $m = \text{SHA256-HMAC}_{K_M}(s)$. This file will be stored at m.

1.4.2 Contents

File name and block numbers are stored to check for block swapping by adversaries. 32KB of data is stored. Each block has a randomly generated salt (generated every time a block is made) to prevent adversaries from being able to compare two blocks with the same information and knowing what they are. Finally, there is a MAC to make sure that nothing is tampered with.

1.4.3 Actions

This block is encrypted with K_G and the MAC uses K_C, both from 1.2.3. The MAC should be checked before the data is used. When the data is received, the file name and block number should also be checked to make sure that they match what is expected. When all the Data blocks are downloaded and verified for integrity, simply piece the data in block order and return the string.