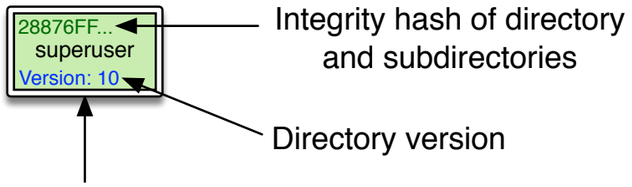
# Unix-like Access Permissions in Fully Decentralized File Systems

Johanna Amann, Thomas Fuhrmann — Technische Universität München, Germany

1

Each user and each group owns a dedicated directory tree. It contains all data belonging to the user/group. A hash-tree secures the directory tree. The top-level directory is signed. Thus the whole tree can be verified.




2

The visible root of the file system is contained within the dedicated directory structure of the file-system superuser.

3

Redirects glue the different user- and group directory structures together.

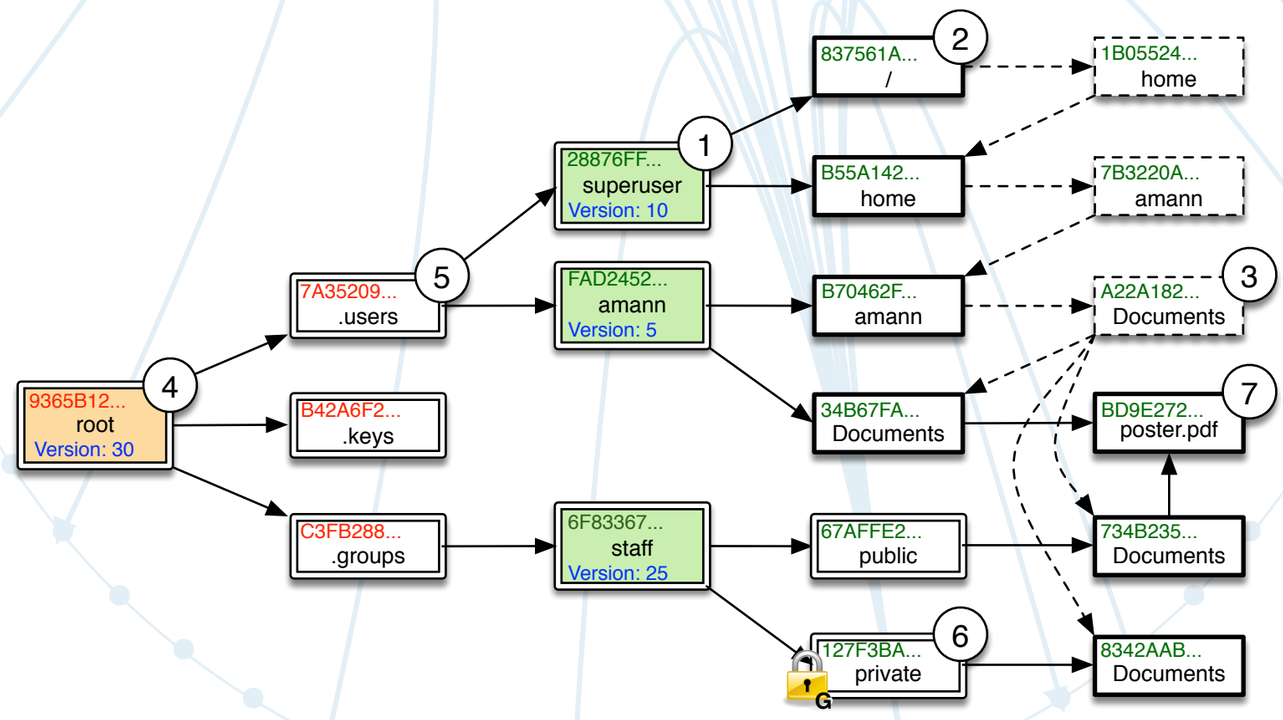
Files and directories are referenced multiple times in the hierarchy.

4

The superuser signs the root directory in the same way the user and group directories are signed.

The hash-tree of the root-directory only protects user and group directories. It does not include their contents.


Providing Confidentiality, Authenticity, and Access Permissions on Fully Decentralized Untrusted Storage




**Features:**

- Unix-like permissions on untrusted storage
- enforced only by cryptography
- fork consistency i.e. resistent against rollback attacks
- fast, only requires symmetric cryptography
- ACLs can be layered on top of this approach, albeit with considerable overhead
- works on block/chunk oriented storage

5

All users have access to the keys directory, which stores the data needed to verify user signatures. The superuser hash tree protects the signatures.

6

Groups are split into a public and a private part. The pointers to the private subdirectory are encrypted with the current group key.

Group keys are distributed to users using the subset difference algorithm. Keys have to be changed upon change in group membership.

7

A group- and world-readable file is present in the user- and public group directory structure.

Depending on the signed parameters in the user directory it may also be group- and world- writable.