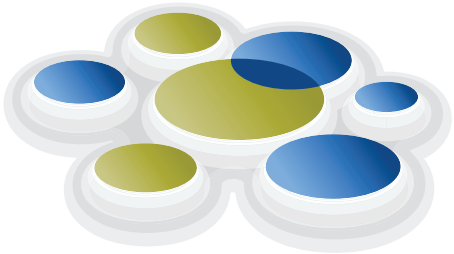

CohortFS

Some Considerations for End to End File Integrity and Privacy in NFS

Marcus Watts
mdw@cohortfs.com
Matt Benjamin
matt@cohortfs.com

November 5, 2013

CohortFS

environment

storage on nfs server managed by “somebody else” that we don’t completely trust

- encrypted so others can’t read data
- integrity determine that what we fetch is what we stored earlier
- compatible with pnfs operate on dispersed data and provide for parallism while doing so

CohortFS

goals

- encrypt file contents
- encrypt file names
- merkle tree for integrity
- file contents need to be managed “per-segment” for pnfs

CohortFS

key management

- lockbox, using acls as “hook” to share keys
- nonces on objects as needed to augment per-data security

CohortFS

consistency

need atomicity for

- file creation open plus any metadata should appear “all at once”
- write operations half-completed write operations will fail integrity!
- pnfs failure of client mid-operation should not result in “broken” file

CohortFS

misc

if file contents are encrypted, can use krb5i not krb5p.

CohortFS

existing mechanisms

- dix/dif
  - low level: 512 + 8 = 520 byte blocks
- named attributes
  - (but only as files not on segments)
  - (accessed thru mds, non-atomic access)

CohortFS

proposed

- layouts some level of atomicity? maybe?
- chuck lever's "data integritys"
  - only covers read + write, nothing for filenames