

Cepheus: A New Encrypted File System with Group Sharing and Integrity Protection

Kevin Fu

Course VI-3A April 26, 1999

On-Campus Thesis Advisor: Ron Rivest
Bellcore Company Supervisor: S. Rajagopalan

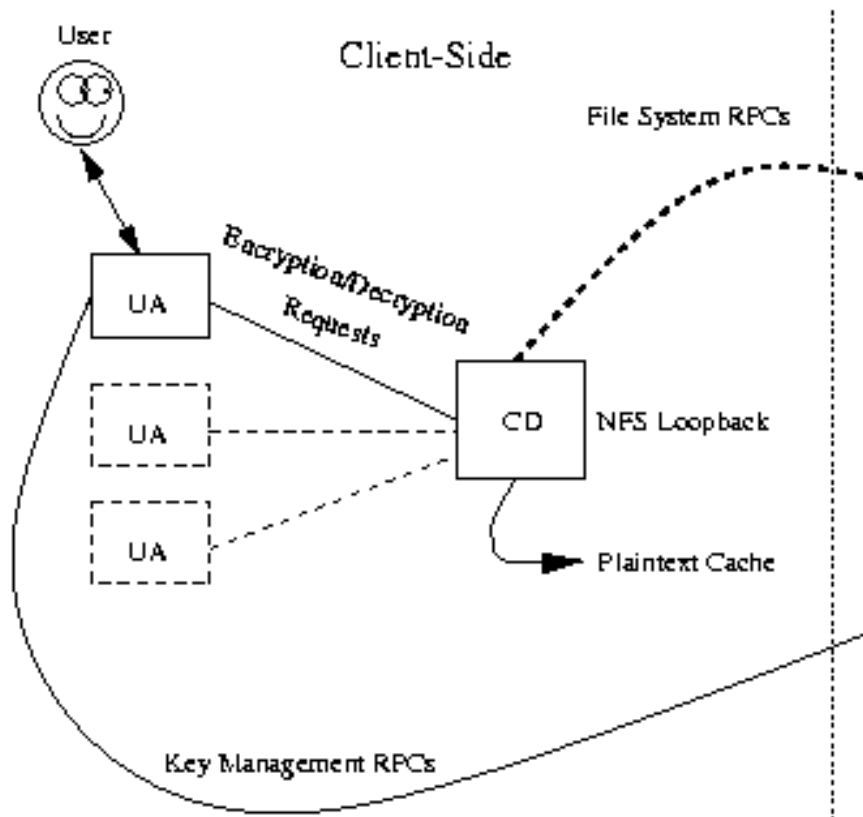
What is Cepheus?

- Confidentiality and integrity protection of data stored on a network file system
- Securely maintain UNIX semantics (file sharing, random access)
- NFS drop-in replacement

Key Problems for Secure Storage

- Problems:
 - Manual encryption cumbersome
 - Protection against malicious system administrators
- Solution: Encrypt stored data
- Side effects:
 - Loss of random access to data
 - No guarantee of integrity

Client-Side



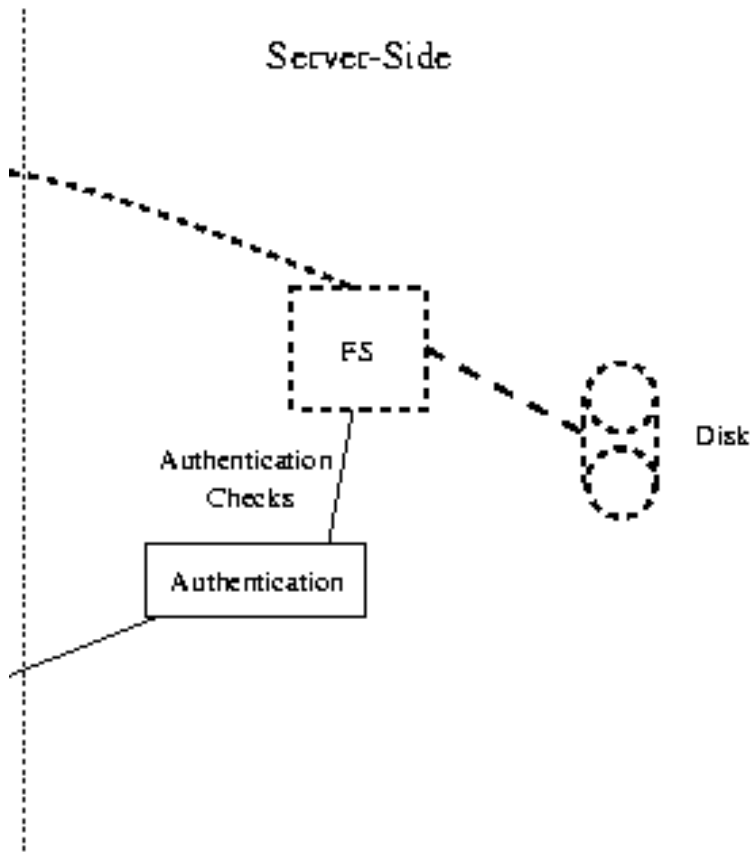
User Agent

- Encryption/decryption
- Integrity check

Client Daemon

- Cache per user agent
- Delayed-write-encryption policy for caching
- Delayed re-encryption for distributed re-encryption

Server-Side



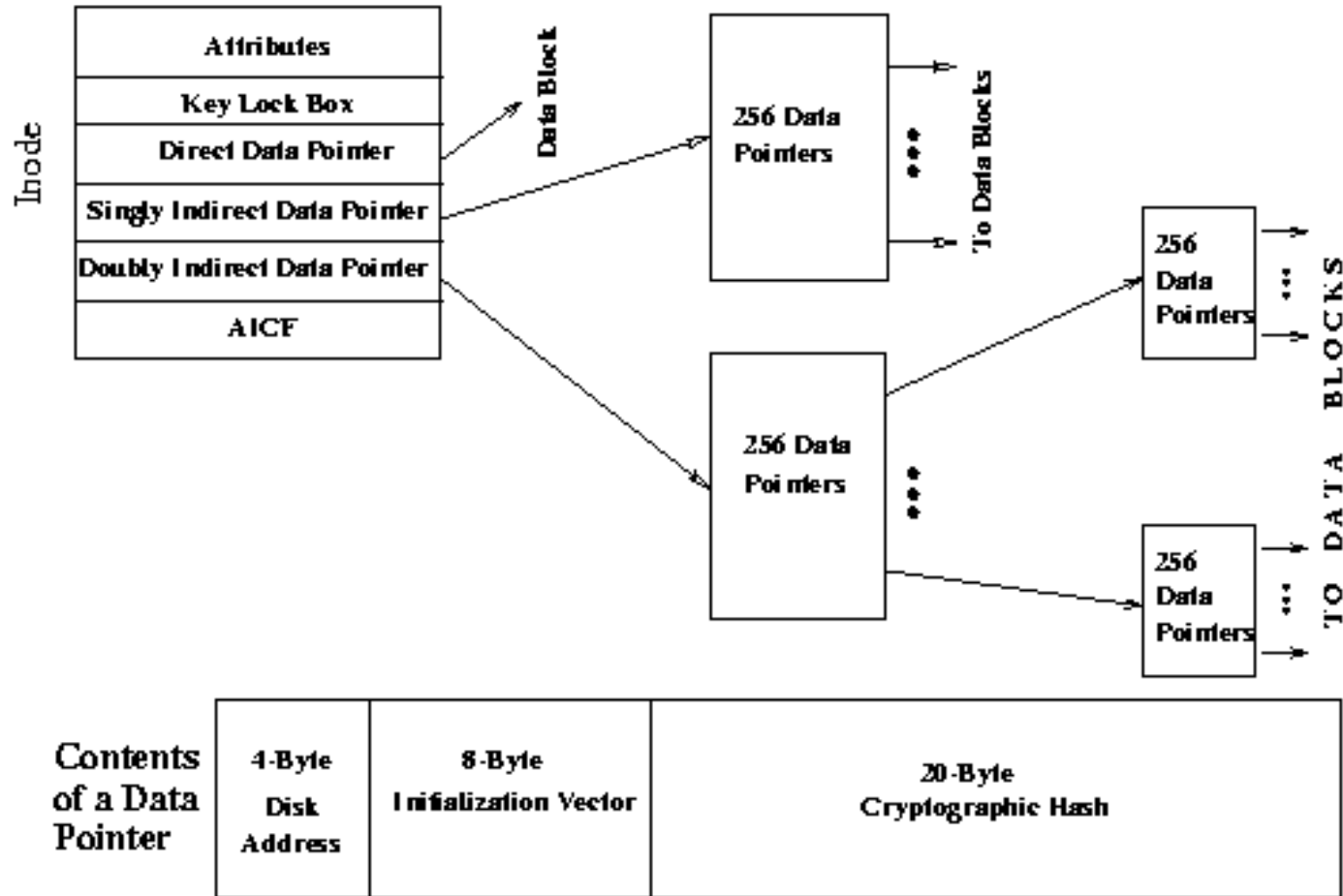
File Server

- Encrypted storage
- Hash tree structure beneath the inode for integrity

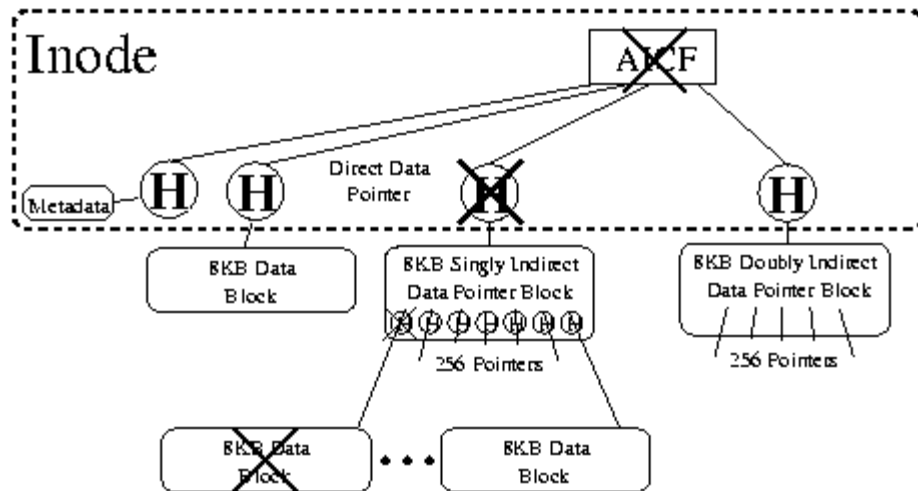
Authentication Server

- Key distribution
- Key recovery

File Structures



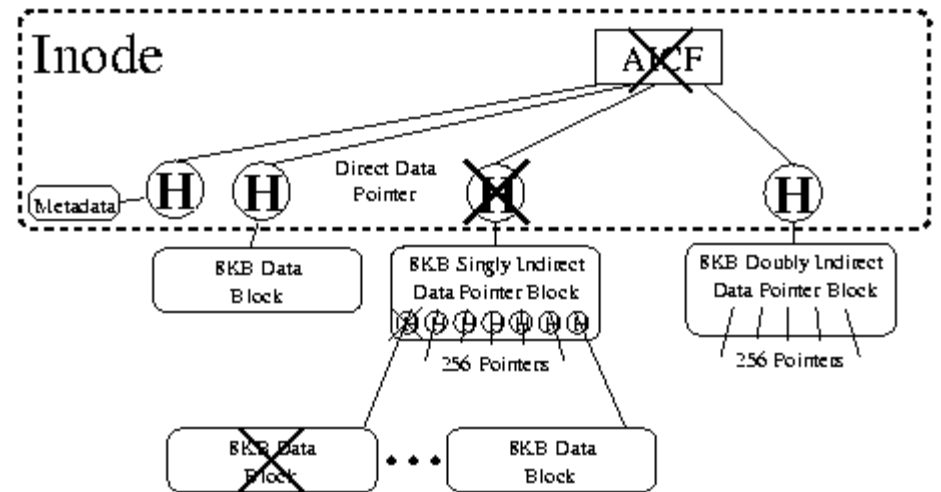
O(1) Sequential Read of a Block



- If block not cached, CD obtains ciphertext block from SD
- If block not decrypted, request UA to decrypt
- If hash path unauthenticated, compute hashes and AICF

Writes $O(\log n)$

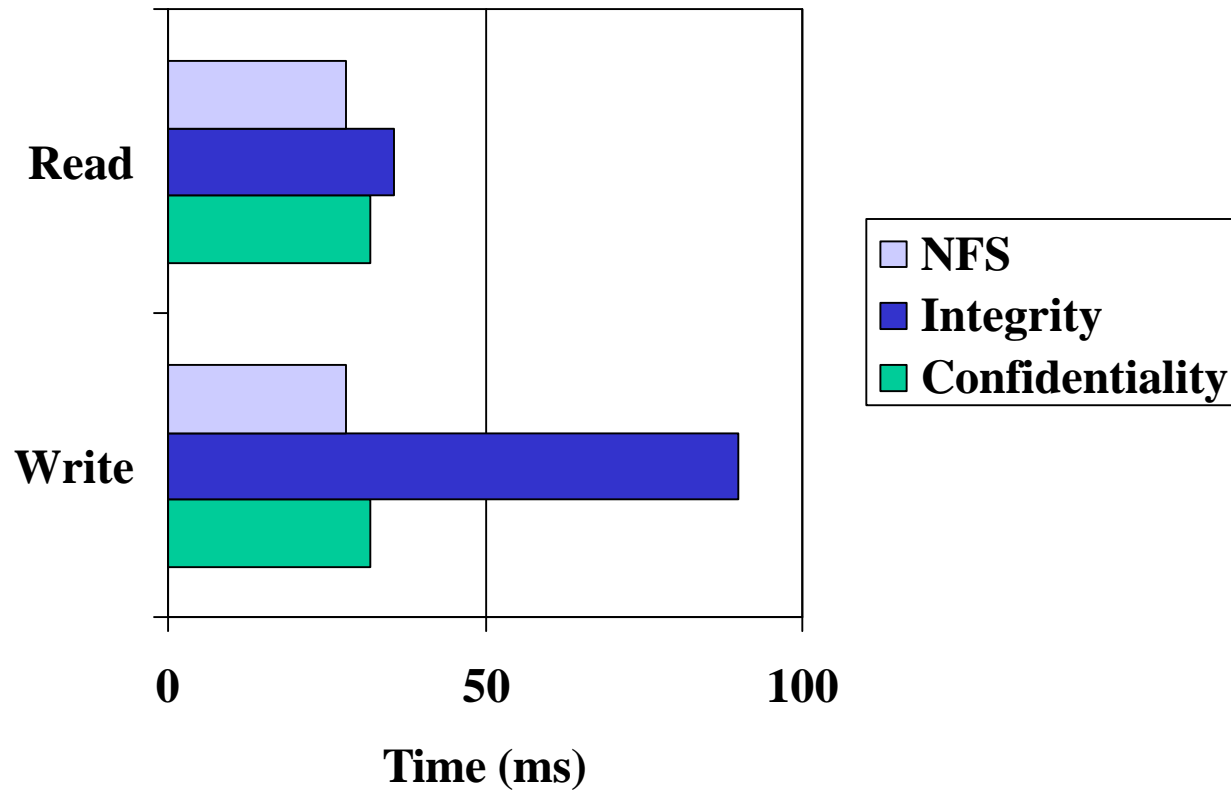
- CD writes plaintext block to cache, not SD
- When cache flushed:
 - Compute hash paths of dirty blocks.
 - Compute AICF
 - Write changed hash paths and AICF to SD
 - Encrypt, send to SD.



Integrity Failures

- When an integrity check fails, the client daemon refuses to serve the file (returns NFS_ERR_IO)
- User agent notified of integrity check failure
- Can attempt recovery of file via user agent

Performance Results



Conclusions

- Provides efficient random access to confidential, integrity-protected data
- Enables secure group sharing
- Uses a well-understood file system interface
- Surveys a wide range of cryptographic storage file systems

Anticipated Q/A