Intra-file Security for a Distributed File System

Zachary Peterson
Scott Banachowski
Ethan Miller
Scott Brandt

Storage Systems Research Center
University of California, Santa Cruz

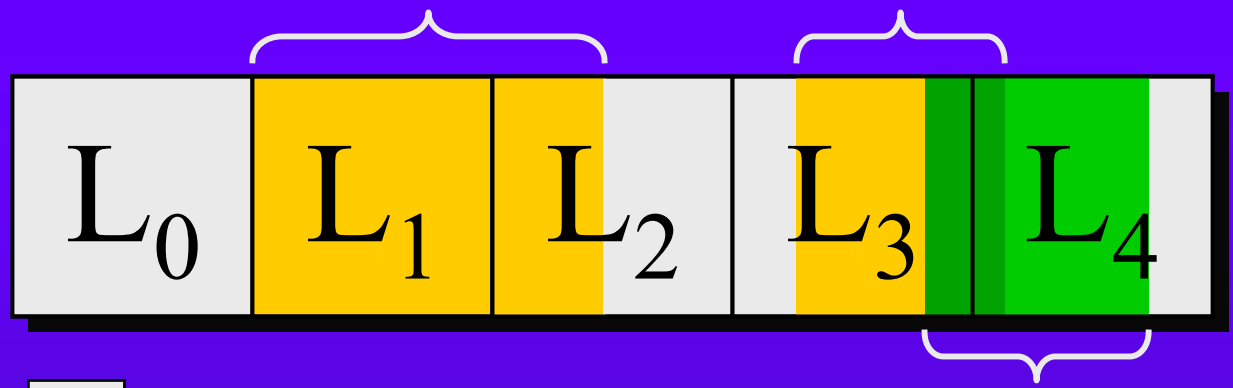
State of the Art in FS Security

- Today, cryptographic file systems take an "all-or-nothing" approach.
  - Files are encrypted on a per-file or per-directory basis, or not at all.
- This is sufficient only for files that must be accessed in their entirety for coherency.
  - Counter example: large shared scientific data, flat file database or a recipe with a secret ingredient.

Intra-file Security

- We present *intra-file security* (IFS), an end-to-end file system encryption technology.
  - Provides the ability to encrypt independent file extents.
  - Flexibility in encryption region size.
  - A single file may contain one or more isolated or overlapping secure regions.
  - Transparent to the user.
  - Supports strong encryption.

An Intra-file Security Example



Insecure Region

Secure Region 1

Secure Region 2

Secure Segment: A single encrypted portion of a file.

Secure Region: A set of secure segments encrypted with the same key.

Cipher Techniques

Because secure regions are not restricted to a block size, IFS is well-suited for stream ciphers, but block ciphers are still possible.

- Block Cipher
  - DES/AES or Blowfish
  - Combine noncontiguous secure segments into a temporary contiguous buffer.
  - Cipher block chaining with initialization vectors (IVs).
  - Employ cipher-text stealing to ensure correct secure segment size.
- Stream Cipher
  - RC4 or SEAL.
  - Data is encrypted in place.
  - Employ feedback chaining and IVs to hide data patterns.
  - No need for block copying or cipher-text stealing.

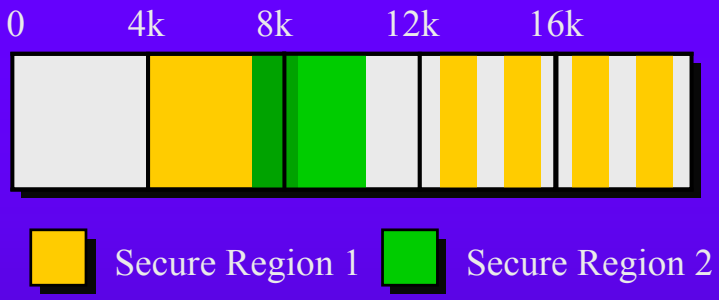
IFS Metadata: The S-Node

- We must keep information to locate secure regions in a file. We introduce the security node, or *s-node*.

An s-node contains:

- Start offset and length of a secure region.
  - Offset is relative to the last secure region.
- Count
  - Regions may be in a repeating pattern. Count provides a shorthand way to represent this.
- S-group
  - An identifier for the key used in encryption.
- Initialization vectors (optional)
  - IVs have the option of being generated on the fly.

An S-Node Example



Start	Length	Count	S-group
4096	5000	1	A
3000	3995	1	B
6144	128	1	A
256	128	3	A

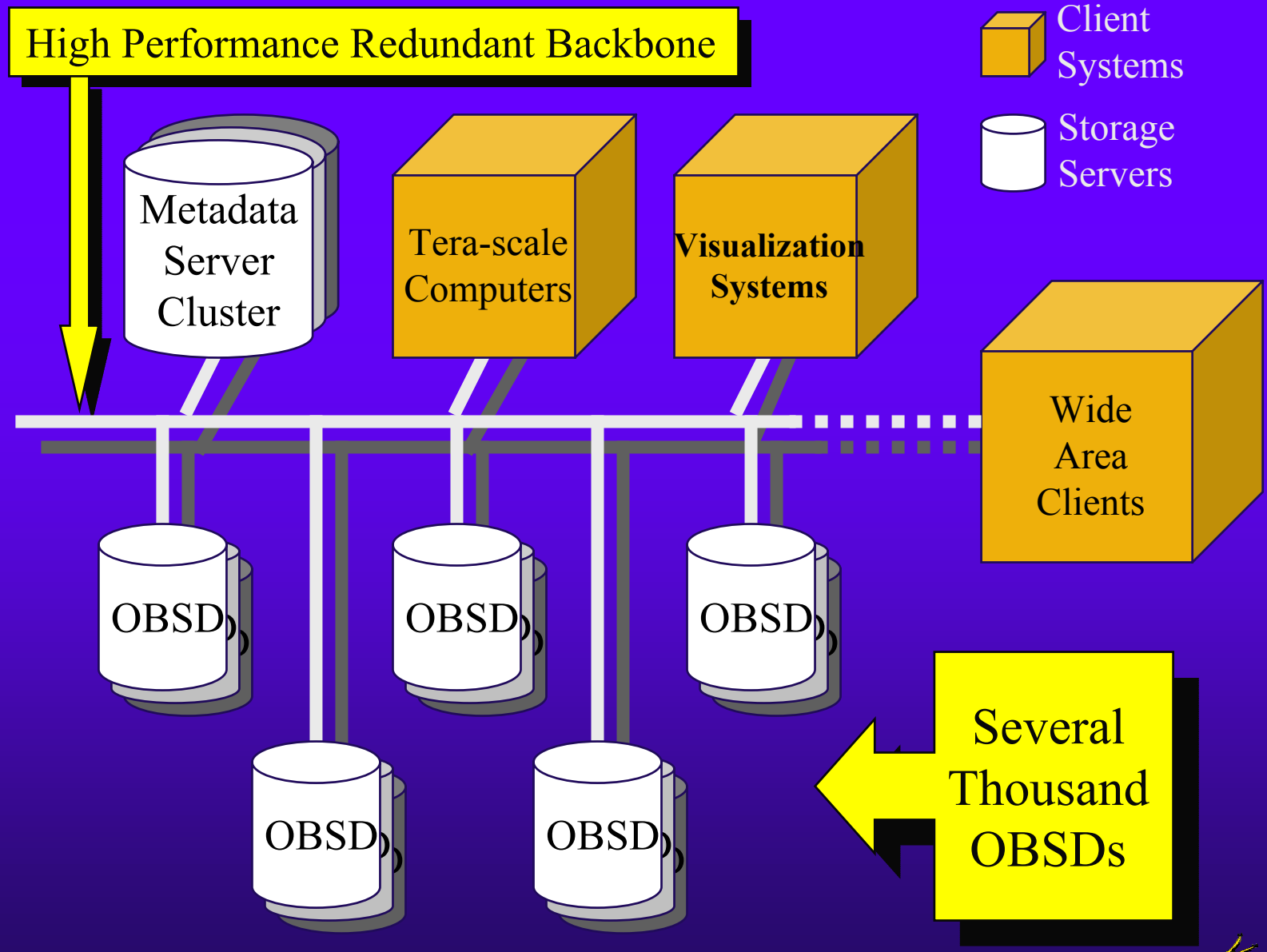
- S-nodes are stored similarly to i-nodes.
  - Groups of s-nodes can be combined, and stored together.
  - Stored on disk with other metadata.
- Future s-nodes might:
  - Utilize gamma compression.
  - Use more complex encryption patterns.
  - Include a bit for relative or absolute offsets.

Integration of Intra-file Security

- IFS can be simply integrated into any type of file system.
- IFS is also very powerful and useful in a distributed environment.
  - Distributed environments benefit from good end-to-end encryption and often deal with large, shared data sets.
- With distribution comes complication.
  - Authentication
  - Key management and distribution
  - Metadata management

Object-based Storage Device

- Storage element in a storage area network.
- High performance:
  - Large bandwidth
  - Low latency
  - Massively parallel
- Decentralized data. Centralized metadata.
- Network of OBSDs seen by a user as a single device.
- High-level allocation (striping) managed by metadata server (MS). Low-level allocation delegated to OBSDs.


High Performance Redundant Backbone
Client Systems
Storage Servers
Metadata Server Cluster
Tera-scale Computers
Visualization Systems
Wide Area Clients
OBSD
OBSD
OBSD
OBSD
OBSD
Several Thousand OBSDs

Authentication

- Regardless of end-to-end encryption technology, some form of authentication system is required.
  - Many techniques already exist, and a new authentication scheme is not the focus of this work.
- Metadata Server (MS) performs all authentication, permissions and access control.
- MS then generates *tokens* to be presented to an OBSD for access data.
  - Tokens contain s-nodes.
- Tokens are similar to *capabilities* in NASD.

Key Management

- We require the existence of a key server (KS) that manages s-groups and keys.
  - KS tightly coupled with MS.
  - Possibly same machine.
- Creates a key for each s-group and for each file. Keeps them secret.
- CLIQUES protocol suite. [Steiner,ICDCS,98]

S-group Management

- Independent of UNIX groups.
- S-groups are created by:
  - Users dynamically
  - System administrator
- S-group specifications are lists of existing user and group names. Can be created on the fly.
- Translation function included in the interface that converts s-groups to a key identifier.

Interface- Reads

- IFS conforms to POSIX file semantics.
- Users each have a key ring containing keys for each s-group to which they belong.
- Read interface is unchanged and decryption is transparent:
  - If client has appropriate key, data is decrypted by client.
  - Otherwise, data is unreadable.

Interface- Writes

- Normal write calls are only allowed to unencrypted segments.
  - Writes that span secure regions are copy-on-written by OBSD to protect the integrity of the encrypted data.
  - Unauthorized changes are discarded.
  - During OBSD inactivity, writes may be merged into original file.
- Secure writes use an explicit system call.
  - Generates s-node information.

Performance & Applications

- IFS eliminates the need to fragment data into multiple files.
  - Ensures high-performance sequential and random access.
- Single-file semantics important for
  - Flat-file databases
  - Extremely large files.
  - Simplifying data management.
- Could be used in combination with the Low-Bandwidth File System (LBFS) [Muthitacharoen, SOSP, 01] used for transferring partial files to slow clients.

Related Work

- Secure File Systems
  - CFS [Blaze, ACM, 93]
  - Cryptfs [Zadok, Tech Report, 98]
  - Self Securing Storage [Strunk, OSDI, 00]
- Architectures for networked attached disks.
  - Network-Attached Secure Disks (NASD) [Gibson, ASPLOS, 98]
  - Secure Network-Attached Disks (SNAD) [Miller, FAST, 02]
  - SeCure Authentication for Remotely Encrypted Devices (SCARED) [Reed, IEEE Micro, 00]

Related Work

- Encryption Technology
  - Feedback chaining
  - Initialization Vectors [Schneier, Book, 96]
  - Cipher-text stealing [Daeman, PhD Thesis, 95]
- Authentication
  - Kerberos [Neumann, USENIX, 88]
  - Cryptographic hashes [Miller, FAST, 02][Reed, IEEE Micro, 00]
- Key Management
  - CLIQUES [Steiner, ICDCS, 98]

Conclusions

- IFS increases the granularity and ease with which users may encrypt data with little change to the file system interface.
  - Especially useful in an high-performance distributed environment.
- Able to separate secure keys from insecure metadata.
- Predicted performance to be similar to other cryptographic file systems.
- An IFS implementation is in progress, as well as other exciting work on OBSDs.

Thank you. Questions?

zachary@cse.ucsc.edu

http://csl.cse.ucsc.edu