Security in Storage and File Systems

Rajeev Thakur
Argonne National Laboratory

Nice Survey Papers

- "Securing Data in Storage: A Review of Current Research"

  Paul Stanton, UIUC

  http://arxiv.org/ftp/cs/papers/0409/0409034.pdf

  - A good overview of various projects in the field

- "A Framework for Evaluating Storage System Security"

  Erik Riedel, Mahesh Kallahalla, Ram Swaminathan, HP Labs

  www.hpl.hp.com/SSP/papers/FAST2002-**security**.pdf

Goals of Storage Security

- Confidentiality of data
  - Only authorized users have access to data
  - May require encryption of data if network or storage are untrusted
- Integrity of data
  - No data is subject to unauthorized modification
  - Maintaining data consistency in case of accidental or malicious attacks
  - Data modification *prevention* and data modification *detection*
- Availability
  - Data is available to an authorized user within an acceptable time period
  - An adversary cannot prevent authorized access to data via a DoS attack
- Performance
  - All this without substantially degrading performance!

Different Approaches

- No encryption of data
  - Storage servers and network are trusted
  - Only user authentication and authorization are performed
- Encrypt on wire
  - Only storage servers are trusted
  - Data is stored unencrypted, but is encrypted during communication between client and server
- Encrypt on disk
  - Even storage servers are untrusted
  - Data is stored encrypted on disk

Incomplete List of Projects

- No Encryption
  - SFS (Secure File System)
  - LWFS (Lightweight File System)
- Encrypt on wire
  - NASD (Network Attached Secure Disks)
- Encrypt on disk
  - CFS (Cryptographic File System)
  - SFS-RO (Secure File System – Read Only)
  - SNAD (Secure Network Attached Disks)
  - Plutus
  - SiRiUS (Securing Remote Untrusted Storage)

CFS (Cryptographic File System)

- One of the early attempts at security for storage
  - Matt Blaze, Bell Labs, 1993
- www.crypto.com/papers/cfs.pdf
- Standard Unix file system interface to encrypted files, accessible under `/crypt` mount point
  - It's a *local* file system
- Uses symmetric keys
- User specifies the key for a directory by using `cattach` command
- From then on, the user "sees" the directory as a normal directory
- To share a file, the user must hand the key to the other user

SFS (Secure File System)

- David Mazieres, Frans Kaashoek, et al, MIT
- http://pdos.csail.mit.edu/papers/sfs:sosp99.pdf
- Separates key management from file system security
- Servers and clients perform mutual authentication
- Server's public key is embedded in the path name of a file
- Data is stored unencrypted (server is trusted)

SFS-RO (Secure File System – Read Only)

- Kevin Fu, Frans Kaashoek, David Mazieres, MIT
- www.scs.cs.nyu.edu/~dm/sfsro-tocs.pdf
- SFS extended to support storage and retrieval of *encrypted* read-only data
- Useful for content distribution over the Internet by using insecure mirrors as storage servers
- A publisher creates a digitally signed database out of a file system's contents
- The database is replicated on untrusted mirror sites
- Decryption is performed on the client, hence scalable

LWFS – Lightweight File System

- Ron Oldfield, Lee Ward, Barney Maccabe, Sandia-UNM
- Assumes trusted network and storage servers, hence no encryption
- Focuses on authentication and authorization without significant performance impact
- Ron can tell you more about it ☺

NASD (Network Attached Secure Disks)

- Garth Gibson et al, CMU
- www.pdl.cmu.edu/PDL-FTP/NASD/asplos98.pdf
- Data is not stored encrypted on disk, but all communication is encrypted
- User contacts server *once* to obtain a capability key
- With this capability key, user can access the appropriate disks directly without further server interaction
- Relieves server bottleneck
- Disks must be “intelligent” enough to process the capability key and service file access requests

PASIS – Survivable Storage

- Jay Wylie, Greg Ganger, et al, CMU
- www.pdl.cmu.edu/PDL-FTP/Storage/PASIS.pdf
- Can withstand some number of compromised servers
- Uses a *threshold scheme* (or *secret-sharing scheme*), not cryptography
- A *p-m-n* threshold scheme:
  - divides data into *n* parts, such that
  - any *m* of the parts can reconstruct the original data
  - fewer than *p* parts reveal no information about the original
- Can be combined with cryptography for higher security

S4 – Self Securing Storage

- Greg Ganger, Garth Goodson, et al, CMU
- www.pdl.cmu.edu/PDL-FTP/Storage/s4.pdf
- Focuses on maintaining data integrity rather than confidentiality
  - No fancy authentication or encryption
- Disks don't trust even the host operating system because a clever intruder could have broken in and remained undetected
- Disks internally audit all requests and keep old versions of data and metadata for a window of time
- Enables intrusion diagnosis and recovery

SNAD (Secure Network-Attached Disks)

- Ethan Miller et al, UC Santa Cruz
- www.soe.ucsc.edu/~elm/Papers/fast02.pdf
- Combines the functionality of CFS and SFS in some sense
- All data is stored and transferred encrypted; client does the decryption
- A file consists of variable-sized data objects, each individually encrypted with a symmetric key
- Users authenticated via certificate objects

Plutus

- Mahesh Kallahalla et al, HP Labs
- www.hpl.hp.com/research/ssp/papers/FAST2003-plutus.pdf
- Provides scalable key management while enabling users to directly control authorizing access to their files
- All data is stored encrypted; all encryption/decryption is done on the client
- Key management and distribution handled by the client
- Groups files (not users) into *filegroups*
- Keys are shared among files in a filegroup, thus reducing the total number of keys

SiRiUS – Securing Remote Untrusted Storage

- Dan Boneh et al, Stanford
- www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/9.pdf
- A secure file system intended to be layered over an insecure network and P2P file systems such as NFS, CIFS, OceanStore, and Yahoo! Briefcase
- Appears to a user as a local file system with usual hierarchical view of files and directories
- Files are stored on the server in two parts:
  - a file containing data, which is encrypted
  - a metadata file containing access control information

Other Projects

- Microsoft Farsite
- Berkeley OceanStore
- Cryptfs/NCryptfs (Stackable encryption file system), Stony Brook
- TCFS (Transparent Cryptographic File System), Italy
- Cepheus (file system on top of MIT's SFS)
- BestCrypt (commercial product. Loopback device driver supporting many ciphers.)
- Windows EFS (Encryption File System)
- StegFS (file system for Linux that uses steganography and encryption)

David’s Questions

Does Time Matter?

- *Does longer term storage require higher security?*
- A longer time period may draw your attention to the problem, but, in reality, the problem was always there
- I would say the sensitivity or importance of data should determine the security requirement, not the time
- Once the right security level is determined, it should be applied immediately
- Data shouldn’t be left at a lower security level even for a short period

Leveraging Commercial Security Advances in HPC

- I would say the core technology is the same in both environments, although the applications may be different
- Some products may be usable in both environments
- What matters a lot is
  - How well the technology is implemented and deployed
  - Whether there are any bugs in the software
  - Whether there are any holes in the end-to-end framework (weak links in the chain)