POTSHARDS: SECURING DATA FOR LONG-TERM ARCHIVES WITHOUT ENCRYPTION

Kevin Greenan, Mark W. Storer, Ethan L. Miller, Kaladhar Voruganti*

Storage System Research Center, University of California, Santa Cruz; *IBM Almaden Research Center

1. The Problem with Encryption in Long-term Storage
- Cryptographic keys are single point of failure as loss of a key is effectively the same thing as deletion and this weakness affects the file system even after encryption.
- Key management is an even more difficult over the long data lifetimes in a long-term archival storage system.
- Fixed encryption is only computationally secure. In 1999 a DES encrypted message was cracked in less than a day. What seemed infeasible in 1976 took 22 hours and 15 minutes in 1998. Advances in areas such as quantum cryptography make predicting the future very difficult.

2. A Solution for Security Through Authentication
- The POTSHARDS system addresses the problem of long-term encryption by moving security from encryption to authentication and utilizing secret splitting for secrecy.
- The POTSHARDS system preserves the data as submitted by the client. Thus, archiving data in the system does not preclude the client’s ability to apply their own compression and encryption.

3. The Need for Secure, Archival Storage
- Archival storage system address the special needs of long-term data storage.
- Redundancy, longevity and high throughput are core requirements.
- As more information is produced and distributed solely as digital data the need for long-term preservation increases. Additionally, legislation such as Sarbanes-Oxley requires the retention of audit data for seven years.

Client Layer
- POTSHARDS preserves the data as submitted by the client. Thus, archiving data in the system does not preclude the client’s ability to apply their own compression and encryption.

Placement Layer
- The Placement Layer maintains enough information about the archives to make intelligent placement decisions.

Redundancy
- The redundancy step provides the longevity of the user’s data and allows a user to retrieve their data in the event that a particular archive is unavailable.

Transformation Layer
- The transformation layer is responsible for dealing with the data operations that provide for secrecy, redundancy, longevity and integrity: it contains
  - Pre-processing function
  - Secret splitting function
  - Redundancy function
- The design of the transformation layer allows it to exist on multiple hosts so long as it can communicate to the placement layer.
- Security is provided through the use of secret splitting. This provides a number of advantages compared to the use of encryption:
  - Provably secure
  - Not dependent on a single key
  - Tunable parameters (e.g. in m of n schemes)

Archive Layer
- The Archive Layer handles the actual data storage chores. It ensures data integrity and safety, archiving storage level through the use of internal and external data auditing and access monitoring.

Each disk in the system contains a superblock and set number of shards for fixed sized shards.

Network Model
- Each layer consists of two server ports and two client ports. Requests travel down the stack and responses go back up the stack.

Read Requests:
1. Client sends read request down the stack to the archives.
2. Index information travel back up the stack to the user.

Write Requests:
1. Client sends write request down the stack to the archives.
2. The client’s data travels back up the stack to be reconstructed.

Clients maintain an index over their own shards. They submit objects to store within the system and receive index information in return. This index information can be used later to retrieve data data objects from the system.

Client
- Clients object to store in the system
- Output: Object + hash over the object data
- The hash over the object data used for integrity and reconstruction verification.

Transformation Layer
- Pre-processing
- Secret splitting
- Redundancy
- The hash over the object is used for reconstruction verification.

Redundancy
- Fragment + hash over fragment data
- Output: Fragment + hash over fragment data
- The redundancy step provides data secrecy as high up in the stack as possible. The hash over the object is used for reconstruction verification.

Placement Layer
- The Placement Layer maintains enough information about the archives to make intelligent placement decisions.

Constraint Checking
- The constraint checking function checks the availability and security for the user's data.

Object Layer
- Object hash
- Input: Object + hash over the object data
- Output: Set of shards + hash over fragment data
- The secret-splitting step provides data secrecy as high up in the stack as possible. The hash over the object is used for reconstruction verification.

Shard Group consisting of a header and data for fixed sized shards
- Each disk in the system contains a superblock and a set of shard groups.

Archive
- Each disk in an archive contains a superblock and set number of shards for fixed sized shards and parity information.
- Parity is calculated across archives and provides a level of external reliability. Additionally, each archive can utilize parity across disks to provide internal reliability.